Genome-wide discovery of miRNAs using ensembles of machine learning algorithms and logistic regression

Benjamin Ulfenborg*, Karin Klinga-Levan and Björn Olsson

Systems Biology Research Centre, School of Bioscience, University of Skövde, Skövde, Sweden
Email: benjamin.ulfenborg@his.se
Email: karin.klinga.levan@his.se
Email: bjorn.olsson@his.se
*Corresponding author

Abstract: In silico prediction of novel miRNAs from genomic sequences remains a challenging problem. This study presents a genome-wide miRNA discovery software package called GenoScan and evaluates two hairpin classification methods. These methods, one ensemble-based and one using logistic regression were benchmarked along with 15 published methods. In addition, the sequence-folding step is addressed by investigating the impact of secondary structure prediction methods and the choice of input sequence length on prediction performance. Both the accuracy of secondary structure predictions and the miRNA prediction are evaluated. In the benchmark of hairpin classification methods, the regression model achieved highest classification accuracy. Of the structure prediction methods evaluated, ContextFold achieved the highest agreement between predicted and experimentally determined structures. However, both the choice of secondary structure prediction method and input sequence length had limited impact on hairpin classification performance.

Keywords: miRNA prediction; miRNA discovery; RNA structure prediction; GenoScan; ensemble classifier; regression model; machine learning.


Biographical notes: Benjamin Ulfenborg is a PhD student in Bioinformatics in the Department of Bioscience, University of Skövde, Sweden. He received his MSc degree in Molecular Biology from University of Skövde in 2008. One of his main research interests is the development of machine learning algorithms and other data mining techniques, and their application to biomedical datasets with a focus on cancer and microRNA.

Karin Klinga-Levan is Professor of Biomedicine in the Department of Bioscience, University of Skövde, Sweden. She received her PhD degree in Genetics from Lund University in 1988. Her current research interests are in the areas of tumour genetics and bull fertility with special focuses on the influence of miRNA expression on tumour development and the influence of sperm borne miRNA expression on foetus development.
1 Introduction

It is becoming increasingly clear that non-coding RNAs (ncRNAs) play key roles in the regulation of gene expression. The discovery of several new classes of ncRNAs transformed our understanding of transcriptomics and challenged the traditional view of proteins as the main effector molecules in cells (Mattick et al., 2010). One class of ncRNAs that has received a lot of attention is microRNAs (miRNAs), which act as post-transcriptional regulators of gene expression. They have been shown to regulate many different cellular processes (He and Hannon, 2004; Mattick and Makunin, 2005; Bartel, 2009) and to be dysregulated in different diseases, such as cancer (Espinosa and Slack, 2006; Sassen et al., 2008; Munker and Calin, 2011). The biogenesis of miRNAs starts with transcription of primary miRNAs (pri-miRNAs), which are cleaved by the enzyme Drosha into precursor miRNAs (pre-miRNAs) that fold into characteristic hairpin shapes. The pre-miRNA is exported to the cytoplasm and cleaved by Dicer to release the single-stranded mature miRNA. Finally, the mature miRNA is incorporated into the RNA-induced silencing complex, which mediates the regulatory action by inhibiting translation, or inducing degradation, of specific target mRNAs (He and Hannon, 2004).

A large number of computational methods have been developed to facilitate the identification of novel miRNA-coding genes in sequence data. The process of in silico miRNA prediction generally consists of three steps. First, sequences in which to search for miRNAs are selected. This could be an entire genome, or regions from a genome that are homologous to known miRNAs, conserved between species, or transcribed (Lim et al., 2003; Huang et al., 2007; Friedländer et al., 2008). Restricting the search reduces the false discovery rate, which is a prevailing problem in miRNA prediction since miRNA hairpins are short (~70 nt.) and since the genome contains a vast number of sequences that, when transcribed, can fold into hairpin structures (Bentwich, 2005). A disadvantage of restricting the search is that it will discover only miRNAs that are similar to known ones or expressed in a particular experiment. The second step in miRNA prediction is to fold the sequences of interest using secondary structure prediction software such as RNAfold (Lorenz et al., 2011). Third, sequences with hairpin structures are evaluated to determine if they are real miRNAs or not.

Computational discovery of miRNAs was pioneered with MIRscan (Lim et al., 2003) and miRseeker (Lai et al., 2003), which sought to identify novel miRNAs in C. elegans and D. melanogaster, respectively. These and other early methods were filter-based, i.e. they defined a set of sequence and structure features for the miRNA hairpins and
thresholds for feature values. Commonly used features include Minimum Free Energy (MFE) of the secondary structure, length of the hairpin and base pair propensity. Examples of early machine learning classifiers for miRNA discovery are Triplet-SVM (Xue et al., 2005) and mir-abela (Sewer et al., 2005), which are based on Support Vector Machines (SVMs). Subsequently, miRNA prediction algorithms relying on many different classifiers have been published, including Hidden Markov Models (HMMs) (Terai et al., 2007; Agarwal et al., 2010), random forests (Jiang et al., 2007; Gudyś et al., 2013), artificial neural networks (Rahman et al., 2012) and decision trees (Tyagi et al., 2008). The emergence of next-generation sequencing (NGS) technologies led to development of prediction methods using read mapping in genomes. The earliest of these was miRDeep (Friedländer et al., 2008), followed by miRanalyzer (Hackenberg et al., 2009), MIRENA (Mathelier and Carbone, 2010) and others.

We have developed a genome-scale miRNA discovery software package named GenoScan, which allows scanning of an entire genome or a set of genomic regions. When developing GenoScan two methods for hairpin classification were evaluated. The first is based on the results of a benchmark study of previous hairpin classification methods to evaluate their suitability for genome-wide scans. A selection of the suitable algorithms was then integrated into an ensemble classifier for improved performance. The second method is a logistic regression model trained on the datasets used during benchmarking. Additionally, we investigated two ways of improving the confidence of miRNA discovery from genomic sequences and implemented them in GenoScan. First, performance of secondary structure prediction algorithms was investigated by evaluating their dependence on sequence length and their agreement with experimentally determined structures. Second, the datasets used to train GenoScan were reconstructed by mapping mature miRNAs to predicted transcript structure. These new datasets were used to retrain the GenoScan regression model and the impact on performance was evaluated.

2 Methods

2.1 Datasets for benchmarking hairpin classification algorithms

Three datasets were generated for benchmarking of published hairpin classification methods: a positive, negative and genomic dataset. The positive dataset was extracted from the 1600 H. sapiens pre-miRNA hairpin sequences in miRBase ver. 19 (Kozomara and Griffiths-Jones, 2011). All pairs of sequences were aligned with BLAST (Altschul et al., 1990) and one sequence removed for every pair with \( \geq 95\% \) identity. Every hairpin fulfilling all of the following criteria, according to miRBase annotation, was removed: (a) not supported by cloning; (b) supported by fewer than three experiments and (c) supported by fewer than ten NGS reads. This removes degradation products from NGS experiments that have been wrongly annotated as miRNAs. The remaining sequences were folded by RNAfold (Lorenz et al., 2011) and hairpins containing more than one terminal loop were removed. The remaining 1200 hairpins constitute the positive dataset.

The negative dataset included permutations of known human pre-miRNAs, hairpins extracted from CDS loci and hairpins extracted from ncRNA loci (excluding miRNAs). Permutations were generated by the uShuffle tool (Jiang et al., 2008), using \( k = 2 \), which preserves the dinucleotide frequencies of the original sequence. CDS and ncRNA
Genome-wide discovery of miRNAs

sequences based on NCBI Refseq annotation were extracted from the *H. sapiens* genome build hs_ref_GRCh37.p10 (hg19) (Pruitt et al., 2012) and folded with RNAfold followed by hairpin extraction. All pairs in the negative dataset were aligned with BLAST and one sequence removed from every pair with identity ≥95%. Hairpins with more than one terminal loop and those with a sequence identity ≥95% to known human pre-miRNAs were also removed. The remaining hairpins were merged into a single negative dataset of 1361 hairpins, where 1/3 comes from permutations, 1/3 from CDSs and 1/3 from ncRNAs.

The genomic dataset was generated by moving a 200 nt. window with a step size of 20 nt. across the genome (excluding CDS regions), folding the sequences with RNAfold and extracting hairpins. All hairpins were pairwise aligned with BLAST and one sequence removed for every pair with identity ≥95%. Hairpins containing more than one terminal loop were also removed. From the remaining hairpins, 10,000 sequences were randomly selected for the genomic dataset.

The following hairpin extraction algorithm was used when generating the negative and genomic datasets. For each hairpin loop in a sequence, the algorithm extends the stem of the hairpin until the length is 120 nt. or a hairpin junction is seen. Hairpins were retained if they fulfilled all of the following criteria: (1) sequence length (min: 40); (2) GC-content (min: 0.2, max: 0.8); (3) triplet-repeat score (max: 5); (4) stem length (min: 20); (5) loop size (max: 20); (6) base pair propensity (min: 0.5); (7) unpaired stretch propensity (max: 0.35); (8) normalised unpaired stretch length (max: 0.35). The triplet-repeat score (3) is calculated as the total number of base triplets divided by the number of unique triplets. Base pair propensity (6) is the number of base pairs in the hairpin stem divided by the stem length. Unpaired stretch propensity (7) is the number of unpaired base stretches in the stem divided by stem length. An unpaired base stretch is a symmetric/asymmetric loop or bulge in the hairpin stem. The normalised unpaired stretch length of an unpaired stretch is calculated as the number of bases in the unpaired stretch divided by stem length.

2.2 Hairpin classification benchmark measures

Benchmarking of the hairpin classification methods was based on Sensitivity (SN), specificity (SP), Matthew’s Correlation Coefficient (MCC), False Discovery Rate (FDR), Genome Prediction Rate (GPR) and Computational Time (CT). Sensitivity and specificity are defined as the fraction of miRNA hairpins classified as miRNAs and the fraction of non-miRNA hairpins classified as non-miRNAs, respectively. Matthew’s correlation coefficient ranges from –1 to 1, where –1 indicates complete disagreement between assigned class and real class and 1 indicates perfect agreement. False discovery rate is the fraction of hairpins classified as miRNAs that are non-miRNA. Genome prediction rate is the fraction of hairpins in a genome (*N*_genome) that are classified as miRNAs (*N*_classified). Computational time *CT* is the average number of seconds required to classify a single hairpin, where *T* is the number of CPU seconds taken to classify all hairpins in a dataset and *D*_size is the number of sequences of the dataset. The definitions of all measures are given in equations (1)–(6).

\[
SN = \frac{TP}{TP + FN}
\]
\[ SP = \frac{TN}{TN + FP} \]  \hspace{1cm} (2) \\
\[ MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \]  \hspace{1cm} (3) \\
\[ FDR = \frac{FP}{FP + TP} \]  \hspace{1cm} (4) \\
\[ GPR = \frac{N_{\text{classified}}}{N_{\text{genome}}} \]  \hspace{1cm} (5) \\
\[ CT = \frac{T}{D_{\text{size}}} \]  \hspace{1cm} (6)

### 2.3 Hairpin classification methods

The literature was surveyed for hairpin classification methods that: (a) are able to classify hairpins as miRNAs or non-miRNAs; (b) can take nucleotide sequences as input and (c) are available as a stand-alone executable implementation, or a web-server with batch functionality. The survey covered 57 methods, many of which lacked executable implementation. Of the 15 methods that fulfilled all criteria, four are based on SVMs, four on HMMs, two on algorithm-specific hairpin criteria, two on ensemble methods and three on decision trees. The most straightforward methods are those that classify hairpins as miRNAs if they pass predefined thresholds for different hairpin sequence and structure features. MIRcheck (Jones-Rhoades and Bartel, 2004) and MIReNA (Mathelier and Carbone, 2010) are the two criteria-based methods included in the study, where MIRcheck is designed to predict miRNAs in plant genomes, while MIReNA predicts miRNAs in any species using genome sequences or NGS reads as input. For these methods the classification process of hairpins is transparent, which is rarely the case for machine learning methods. Triplet-SVM (Xue et al., 2005), microPred (Batuwita and Palade, 2009), MiRPara (Wu et al., 2011), Mirident (Liu et al., 2012) all rely on SVMs, but differ with respect to the hairpin features included. Triplet-SVM is a relatively simple model that relies on that the propensity of secondary structure triplets within the hairpin structure being different between miRNAs and pseudo hairpins. In contrast, microPred is based on 48 hairpin features, MiRPara on 42 and Mirident on 1300 features. The training of Mirident was preceded by a feature space exploration where a large number of sequence-structure motifs were identified and ranked. The algorithm was subsequently trained with the 1300 highest ranked motifs.

The four included methods based on HMMs are ProMir (Nam et al., 2006), HHMMiR (Kadri et al., 2009), CSHMM (Agarwal et al., 2010) and FOMmir (Shen et al., 2012). Although using different variants of the Markov modelling approach, they all have in common that they calculate the probability of a hairpin being a miRNA by traversing a model containing different states representing possible sequence and structure features of a hairpin. The algorithm will visit states in the model according to observed features in the analysed hairpin and multiply the probabilities associated with each visited state and transition probability. Finally, the algorithm classifies the hairpin by the overall probability assigned to it by the model.
Of the three tree-based methods included, MiPred (Jiang et al., 2007) and HuntMi (Gudyś et al., 2013) are based on random forests and CID-miRNA (Tyagi et al., 2008) on decision trees. The hairpin classifier used by CID-miRNA is using a stochastic context-free grammar and is implemented with J48 (Quinlan, 1993). The algorithm classifies hairpins by traversing the tree using decision thresholds for selected features. Random forests extend the decision tree approach by creating an ensemble of trees and giving each tree a vote. Hairpins are then classified according to which class receives the most votes. MiPred relies on MFE, triplet structure features and a $p$-value for how significantly different the MFE is from that of random sequences, while HuntMi relies on the features used by microPred and seven additional features. Finally, miR-BAG (Jha et al., 2012) and HeteroMirPred (Lertampaiporn et al., 2012) are ensemble-based. While miR-BAG combines SVMs, naïve Bayes and best-first tree, HeteroMirPred uses SVMs, k-nearest neighbours and random forests. In the benchmark, all algorithms were run with default parameters, except MiRPara for which the ‘animal 15’ model was used. For miR-BAG, organism was set to *H. sapiens*.

For the ensemble classifier in GenoScan, algorithms were selected based on the results from benchmarking and divided into primary and secondary algorithms. Primary algorithms are suitable for generating an initial set of predictions by achieving high sensitivity and low computational time. Secondary algorithms are suitable for increasing the confidence in the predictions and to remove false positives, due to their high specificity and low false discovery rates. The ensemble classifier was evaluated on the positive, negative and genomic datasets.

### 2.4 Logistic regression modelling

A novel hairpin classification algorithm based on a logistic regression model was implemented in R using the *glm* function. The initial model contained the following covariates: (1) hairpin stem length; (2) terminal loop size; (3) hairpin minimum free energy divided by stem length, proportion of (4) symmetric loops; (5) asymmetric loops; (6) gaps; (7) bulges; (8) wobbles; (9) GC nucleotides and (10–17) the eight possible secondary structure triplets, i.e. combinations of ‘(‘ and ‘.’, in the stem. During model simplification by iteratively removing the least significant covariate until only significant ones remained ($p \leq 0.05$), covariates 4 and 5 were removed: as well as the secondary structure triplets ‘(‘ and ‘.’. Thus the final model contained 13 covariates. A chi-square test of the final model against the null model showed statistical significance ($\chi^2 = 2044, p < 0.0001, \text{df} = 13$) indicating that the covariates distinguish between miRNAs and pseudo hairpins. The performance of the final model was evaluated by leave-one-out cross validation on the positive and negative datasets. In contrast to the criteria-based and machine learning methods included in the study, the regression model calculates a weighted sum of all hairpin features in the model. This allows the algorithm to determine the probability that a hairpin is a miRNA and to classify the hairpin by comparing this probability to a predefined threshold.

### 2.5 Secondary structure prediction methodology

In order to encode a pre-miRNA, a transcript must be able to locally fold into a hairpin secondary structure. Therefore, before running a miRNA prediction algorithm, it is
necessary to determine if the sequences of interest can adopt a hairpin shape. This is accomplished using secondary structure prediction software, such as RNAfold (Lorenz et al., 2011) that determines the MFE structure of sequences. When GenoScan was developed, an arbitrary window size of 200 and step size of 20 nucleotides were defined for genome scanning. Similar methods are frequently used in the literature, relying on the assumption that pre-miRNAs are shorter than 200 nucleotides for organisms of interest, which is in agreement with miRNA biogenesis experiments (Lee et al., 2002). However, secondary structure prediction accuracy may be dependent on sequence length, because longer sequences take additional nucleotides into account that can form base pairs in the hairpin structure. It can thus be questioned if a window size of 200 nucleotides is adequate for structure prediction. After all, intergenic pre-miRNAs are found on transcripts with lengths ranging from a few hundred to several thousand nucleotides (Saini et al., 2007). Furthermore, the MFE structure is not guaranteed to be the true biological structure of the transcript (Puton et al., 2013). These concerns motivate closer inspection of the secondary structure prediction step for in silico prediction of pre-miRNAs.

What is of interest is the true biological structure of the region in the transcript where the putative pre-miRNA is located. In pursuit of this structure, three questions arise: (a) How close is the predicted secondary structure to the true biological structure? (b) To what extent is the predicted structure dependent on sequence length? (c) If the predicted structure varies considerably as sequence length is varied, which predicted structure most accurately approximates the true biological structure? The first question can be addressed by benchmarking secondary structure prediction software against experimentally determined structures. The second and third questions should be answered by predicting the structure of sequences using different flanking region lengths and comparing the predicted structures to each other and to experimentally determined structures.

2.6 Structure prediction datasets

To evaluate the performance of secondary structure prediction software in terms of agreement between predicted and true structure, two datasets were constructed based on known RNA secondary structures in the RNA STRAND database v2.0 (Andronescu et al., 2008). All RNAs containing only canonical bases (AUGC) were downloaded (3235 RNAs). All pseudoknots were replaced with a dot in the dot-bracket notation (consisting of ‘(’ and ‘)’ for paired bases and ‘.’ for unpaired), since the programs evaluated here do not predict pseudoknots and their presence would interfere with their evaluation. Removing all RNAs that contain pseudoknots would take away half of the RNAs in the dataset, while simply ignoring the pseudoknots allows all RNAs to be retained. This does not affect the remaining secondary structure of the RNAs. Sequences with more than 60% unpaired bases were removed (3038 RNAs remaining), since these are unsuitable for evaluating structure predictions where the assumption is that the sequences take on a low-energy structure.

The first dataset, referred to as the short RNA dataset, was generated by extracting RNAs with 50–200 nucleotides from the 3038 RNAs, resulting in 648 RNAs. Using BLAST to align all RNA pairs, one sequence was removed for every pair with ≥95% identity, after which 379 RNAs remained. Similarly, the long RNA dataset was generated
by extracting RNAs with $\geq 1000$ nucleotides. The resulting 433 sequences were filtered to remove duplicates using the same criteria as before, after which 106 RNAs remained. The final datasets contained the following RNA types: ribonuclease P, ribosomal, signal recognition particle, synthetic, group II intron, ribozyme, tRNA, snRNA, Y RNA, telomerase RNA and cis-regulatory elements. The reason that no pre-miRNA structures could be included in the small RNA dataset is that their structures are not available. However, since the short RNAs included also form hairpin structures they can be used to evaluate the agreement between predicted and experimental structures.

### 2.7 Secondary structure in silico experiments

It is beyond the scope of this study to evaluate all secondary structure prediction software found in the literature. Comprehensive and up-to-date benchmarks can be found at the CompaRNA website (Puton et al., 2013). The purpose here is to investigate more closely the performance of some structure prediction software either used by pre-miRNA discovery tools or performing well in the CompaRNA benchmark. The software tested here include RNAfold, UNAfold (Markham and Zuker, 2008), Sfold (Ding et al., 2004) and ContextFold (Zakov et al., 2011). Three in silico experiments were carried out:

- **Whole structure prediction of short RNAs:** The four structure prediction programs were used to fold sequences in the short RNA dataset and the results compared to the experimentally determined structure of the RNAs. All programs were run with default parameters.

- **Whole structure prediction of long RNAs:** The same procedure as above, but using the long RNA dataset.

- **Window structure prediction test:** Subsequences (windows) $w = \{w_1, \ldots, w_{17}\}$ with size $s = \{200, 250, 300, \ldots, 1000\}$ were extracted from the sequences in the long RNA dataset. These windows were centred on the middle of the sequences such that $w_i$ contained $w_{i-1}$ with 25-base flanks on each side. All windows were folded with RNAfold and ContextFold. UNAfold and Sfold were not included in this test because they showed very similar performance to RNAfold in terms of prediction accuracy. Three comparisons were carried out with the resulting predictions, where only the middle 200 bases in each window were considered. To determine the effect of window size on prediction accuracy, the predicted window structure was compared to the experimentally verified structure for the window. The predicted structure of each window $w_i$ was then compared to the structure of the previous window $w_{i-1}$. The purpose of this is to investigate if and to what extent, the secondary structure prediction of the middle 200 bases changes as larger flanks are added. Finally, all windows were compared to $w_1$ to investigate the relationship between their structures and the initial structure.

In all structure prediction experiments performance was measured by sensitivity (equation (1)), specificity and Matthew’s correlation coefficient, where the definitions of SP and MCC differed slightly from equations (2) and (3).

$$SP = \frac{TN}{TN + FP - \xi}$$  \hspace{1cm} (7)
To count the number of True Positives (TP), False Negatives (FN), True Negatives (TN) and False Positives (FP) the convention used by Gardner and Giegerich (2004) and CompaRNA (Puton et al., 2013) was applied. TP denotes predicted base pairs that also exist in the true structure; FN denotes true base pairs not in the predicted structure; TN denotes base pairs not in the true structure that are not predicted; FP denotes base pairs that are predicted but do not exist in the true structure. As in the study by Gardner and Giegerich (2004), false positives that are labelled ‘compatible’ are subtracted from FP (denoted by $\xi$ in equations (7) and (8)) since they are considered neutral for prediction accuracy. TN was assumed to be equal to the number of possible (but absent) base pairs in the true structure that are not predicted. Possible base pairs are all A-U, G-C and G-U pairs between unpaired bases at least one base apart, except when the base pair leads to a pseudoknot.

2.8 Hairpin classification dataset reconstruction

As is common practice in the miRNA prediction literature, the GenoScan regression model was initially trained and evaluated on pre-miRNAs from miRBase. However, since the length of pre-miRNA hairpins in miRBase is arbitrary and their structure is predicted, the adequacy of this procedure is questionable. When GenoScan is used in practice, the regression model will classify hairpins extracted from the genome of interest. These hairpins are generated by GenoScan’s internal extraction procedure and may differ from hairpins extracted by other methods. Thus, the genomic hairpins classified by the regression model may be very different from the precursors in miRBase and results obtained when evaluating the model on miRBase pre-miRNAs may not reflect the real performance of the model. For this reason, the datasets used to train GenoScan were reconstructed and used to retrain the regression model.

All known human mature miRNA loci from miRBase ver. 19 were downloaded and used to generate four new positive datasets. The 2233 mature miRNAs were filtered using the same experimental support criteria as before. From the remaining 1522 mature miRNAs 200-nucleotide (set I) and 500-nucleotide (set II) windows centred on the mature miRNA loci were extracted from the human genome. If both the 5p and 3p versions of a mature miRNA existed, only the 5p was used. All pairs of the 1108 extracted sequences were aligned with BLAST and one sequence from every pair with $\geq 95\%$ identity was removed. Set I and II were treated separately, resulting in 987 and 942 sequences, respectively. The remaining sequences in each set were folded with RNAfold and ContextFold and hairpins containing the mature miRNAs extracted from the folded sequences. Hairpins containing more than one terminal loop were removed, leaving 886 and 788 sequences (set I and II) for RNAfold and 857 and 763 sequences (set I and II) for ContextFold. Finally, the intersection of these four sets was taken, resulting in 661 sequences. These hairpins were placed in four new positive datasets referred to as the RNAfold positive 200 and 500 and ContextFold positive 200 and 500 datasets. The negative and genomic datasets were reconstructed by the same procedure. Thus, four negative and four genomic datasets were constructed. The negative dataset was generated.
from permutations of the positive dataset and hairpins extracted from CDS and ncRNA loci (excluding miRNAs) in the human genome. As a final step, sequences in the negative dataset were aligned to the positive dataset with BLAST. Negative sequences with ≥95% identity to positive sequences were removed. The intersect step was not possible to carry out for the negative dataset because too few sequences would be retained. Thus the final negative datasets contained 722, 707, 727 and 698 sequences for RNAfold 200/500 and ContextFold 200/500, respectively. The genomic dataset was based on randomly extracted hairpins from the human genome and contained 10,000 sequences. The steps for hairpin dataset reconstruction are shown in Figure 1. Finally, the datasets were grouped into sets of three, such that each set contained a positive, negative and genomic dataset. The sets are referred to as RNAfold 200, RNAfold 500, ContextFold 200 and ContextFold 500.

Figure 1 Flowcharts of hairpin dataset construction. A: positive dataset; B: negative dataset; C: genomic dataset

2.9 GenoScan update and performance evaluation

Based on the results of the secondary structure prediction experiments, command line arguments were added in the GenoScan software to allow users to choose between RNAfold and ContextFold for secondary structure prediction and to set the window size for genome scanning. In addition, the regression model was retrained using the RNAfold 200 dataset.
The regression model classification performance was evaluated by leave-one-out cross validation on the four sets of datasets. The positive and negative datasets were used to estimate SN, SP, MCC and FDR, while the genomic dataset was used to estimate GPR. For comparison, the original regression model of GenoScan was used to classify the reconstructed datasets.

3 Results and discussion

3.1 Hairpin classification benchmarking results

Benchmarking was carried out in Ubuntu 13 and Fedora 18 using an Intel Core i5 2300 processor (2.8 GHz, four cores) and 12 GB DDR3 memory. Most methods show a performance ranging from 0.5 to 0.7 MCC, with a mean of 0.48 (Table 1). This indicates fairly good classification accuracy, although with some room for improvement. MIRENA achieved the highest MCC (0.68), followed by CID-miRNA (0.66) and FOMmir (0.65). HuntMi achieved the highest SN of 100%, but low SP (50%). The three other methods with high SN, MiPred (96%), Mirident (90%) and microPred (90%), all had low SP (45%, 70% and 47%, respectively). ProMir had the highest SP of 98%, but the second lowest SN. There were generally large gaps between SN and SP, with an average of 29 percentage units.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SN (%)</th>
<th>SP (%)</th>
<th>MCC</th>
<th>FDR (%)</th>
<th>GPR (%)</th>
<th>CT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIRcheck</td>
<td>5.6</td>
<td>96</td>
<td>0.05</td>
<td>41</td>
<td>2</td>
<td>0.002</td>
</tr>
<tr>
<td>Triplet-SVM</td>
<td>67</td>
<td>92</td>
<td>0.62</td>
<td>12</td>
<td>6</td>
<td>0.003</td>
</tr>
<tr>
<td>ProMir</td>
<td>54</td>
<td>98</td>
<td>0.59</td>
<td>4</td>
<td>2</td>
<td>0.004</td>
</tr>
<tr>
<td>MiPred</td>
<td>96</td>
<td>45</td>
<td>0.47</td>
<td>39</td>
<td>38</td>
<td>6.067</td>
</tr>
<tr>
<td>CID-miRNA</td>
<td>69</td>
<td>94</td>
<td>0.66</td>
<td>9</td>
<td>6</td>
<td>6.425</td>
</tr>
<tr>
<td>HHMMiR</td>
<td>60</td>
<td>89</td>
<td>0.51</td>
<td>18</td>
<td>17</td>
<td>0.054</td>
</tr>
<tr>
<td>microPred</td>
<td>90</td>
<td>47</td>
<td>0.41</td>
<td>40</td>
<td>48</td>
<td>11.310</td>
</tr>
<tr>
<td>CSHMM</td>
<td>86</td>
<td>74</td>
<td>0.60</td>
<td>26</td>
<td>31</td>
<td>1.986</td>
</tr>
<tr>
<td>MIRENA</td>
<td>76</td>
<td>90</td>
<td>0.68</td>
<td>12</td>
<td>5</td>
<td>0.002</td>
</tr>
<tr>
<td>MiRPara</td>
<td>66</td>
<td>83</td>
<td>0.51</td>
<td>22</td>
<td>8</td>
<td>0.248</td>
</tr>
<tr>
<td>Mirident</td>
<td>90</td>
<td>70</td>
<td>0.61</td>
<td>27</td>
<td>34</td>
<td>0.269</td>
</tr>
<tr>
<td>miR-BAG</td>
<td>81</td>
<td>47</td>
<td>0.25</td>
<td>43</td>
<td>45</td>
<td>171.200</td>
</tr>
<tr>
<td>FOMmir</td>
<td>88</td>
<td>77</td>
<td>0.65</td>
<td>23</td>
<td>25</td>
<td>–</td>
</tr>
<tr>
<td>HeteroMirPred</td>
<td>83</td>
<td>77</td>
<td>0.60</td>
<td>24</td>
<td>35</td>
<td>2.430</td>
</tr>
<tr>
<td>HuntMi</td>
<td>100</td>
<td>50</td>
<td>0.56</td>
<td>36</td>
<td>35</td>
<td>11.890</td>
</tr>
</tbody>
</table>

Dramatic differences could be observed with respect to computational time of the stand-alone implementations. MIRcheck, Triplet-SVM, ProMir and MIRENA required only milliseconds per hairpin, while MiPred and CID-miRNA took 6s and microPred and HuntMi more than 11s. The slowest method was miR-BAG, with 171s per hairpin. This has enormous implications for scanning a whole genome sequence.
A potential source of bias in benchmarking is to include some items in the test set that were used in the training set, leading to overestimation of performance. This was difficult to avoid in this study, since most methods cannot be retrained on new datasets. Two notable exceptions are CSHMM and HuntMi. In addition, the datasets used by most methods are not available, making it difficult to deduce which hairpins have been used as positive and negative training sets. Since the known human miRNAs are relatively few, excluding all miRNAs used by all methods would probably result in exclusion of most known miRNAs.

3.2 The GenoScan hairpin classification methods

The GenoScan classification methods were designed with four goals in mind: high performance (SN and SP), reasonable GPR, low FDR and reasonable computational time. GenoScan takes genomic sequences as input, followed by optional filtering based on supplied sequence annotation. Sequences that pass the filter are segmented into 200 nt. windows with 20 nt. overlaps, folded with RNAfold and passed to the Classify hairpins step. Two approaches to hairpin classification were evaluated. The first is an ensemble classifier (Figure 2) that was created by integrating published hairpin classification algorithms to provide accurate and confident hairpin classification. Initial miRNA candidates are generated by Mirident (primary algorithm), which combines high SN with low computational time. The candidates from Mirident are passed to HHMMiR, MiReNA and Triplet-SVM (secondary algorithms), which have high SP and low FDR. Hairpins passing HHMMiR, MiReNA or Triplet-SVM are merged into a single set and filtered based on the confidence parameter \( C = \{1, 2, 3\} \), determining how many secondary algorithms a hairpin must pass in order to qualify as a miRNA candidate. The \( C \) parameter allows adjustment of the balance between SN and SP and to control the number of candidates reported.

**Figure 2** Outline of the ensemble classifier method
The second method is a logistic regression model that was trained on the positive and negative datasets. The model calculates the probability that a given hairpin is a miRNA and reports it as a miRNA candidate if the probability is equal to or greater than the threshold $P$. The purpose of evaluating this method is to investigate the usefulness of logistic regression for miRNA discovery and to see if single algorithms can achieve classification performance comparable to the ensemble approach.

**Figure 3** Performance of benchmarked algorithms compared with the GenoScan ensemble (GS-E) and regression (GS-R) classifiers, with $C = 1$ for GS-E and $p \geq 0.5$ for GS-R

Notes: Upper panel: sensitivity vs. specificity. Algorithms included in the ensemble classifier are represented by filled grey markers. Lower panel: sensitivity vs. $100 – \text{GRP} (%)$. 
3.3 GenoScan performance evaluation

The ensemble classifier was evaluated by classifying the hairpins in the positive, negative and genomic datasets with $C = 1, 2$ and $3$ (Table 2). Increasing $C$ results in better SP, FDR and GPR, but reduces SN. In comparison to the individual methods, the ensemble achieves better balance between SN and SP (Figure 3) for $C = 1$, although the GPR is relatively high (17%). For $C = 3$ the GPR drops to 0.5%, thus surpassing the best GPR for the individual classification methods, which ranged from 2% for ProMir and MIRcheck to 48% for microPred. The cost is a dramatic reduction of SN to 51%, however with an SP of 99%. For individual methods with low GPR (2–8%) the sensitivity ranges from 5.6% to 76%. Individual methods with higher sensitivity have considerably higher GPR. When scanning whole genomes, the number of hairpins to be classified potentially number in the millions and this demands a method combining very low FDR and GPR with reasonable computation time. The lower GPR of the ensemble classifier makes it more suitable for whole-genome miRNA discovery than the individual methods. For the human genome, the hairpin extraction algorithm described in the methods section identifies ~27 million single terminal loop hairpins, of which ~19 million are extracted.

Table 2  Classification performance of the ensemble classifier

<table>
<thead>
<tr>
<th>Confidence</th>
<th>Sn (%)</th>
<th>Sp (%)</th>
<th>MCC</th>
<th>FDR (%)</th>
<th>GPR (%)</th>
<th>CT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>82</td>
<td>84</td>
<td>0.66</td>
<td>18</td>
<td>17</td>
<td>0.31</td>
</tr>
<tr>
<td>2</td>
<td>68</td>
<td>94</td>
<td>0.65</td>
<td>9</td>
<td>7</td>
<td>0.31</td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>99</td>
<td>0.58</td>
<td>2</td>
<td>0.5</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Figure 4  Hairpin probabilities from the regression model

The logistic regression model was evaluated with different values for the probability threshold $P = 0.1, 0.2, …, 0.9$, where hairpins with a probability of $P$ or higher are classified as miRNAs. The probabilities assigned to the left-out positive and negative hairpins during cross validation are shown in Figure 4. The SN and SP of the model peaks at $p \geq 0.5$ and 0.6, with MCC = 0.75 (Figure 5). This is higher than the MCC of all the benchmarked methods, where MIReNA had the highest (0.68) and also higher than
the MCC of the ensemble classifier (0.66 for $C = 1$). At $p \geq 0.6$ the regression model has FDR = 9% and GPR = 5%, compared to 12% and 5% for MIReNA. SN and SP of MIReNA was 76 and 90%, respectively, while the regression model achieved 81% and 93%. At $p \geq 0.9$ FDR and GPR are considerably lower (2.5% and 0.6%, respectively) and surpass the individual classification methods, at the cost of a dramatic loss of sensitivity.

Figure 5  Performance of the regression model classifier at $P = \{0.1, \ldots, 0.9\}$

Notes: All performance measures are shown in percentages, except MCC, which is shown as the original MCC value multiplied by 100. (The peak MCC value at $p = 0.5$ and 0.6 is 0.75.)

In summary, both methods in GenoScan give highly accurate classification results with a good balance between sensitivity and specificity. The results also show that a regression model with only 13 covariates achieves higher classification accuracy than previously published algorithms.

3.4 Whole structure prediction tests

To evaluate performance in terms of agreement between predicted and experimentally determined structures, structure prediction was carried out on whole RNA sequences in the short RNA and long RNA datasets, followed by comparing the entire predicted and experimental structures. RNAfold, UNAfold and Sfold achieved similar performance with median MCC values of 0.47–0.48 and 0.15–0.20 for the short RNA and long RNA datasets, respectively (Figure 6). ContextFold clearly outperforms the other methods, achieving an MCC of 0.80 and 0.56, respectively. The similar performance obtained for the energy-based methods (RNAfold, UNAfold and Sfold) and the higher performance of ContextFold is consistent with previous research (Gardner and Giegerich, 2004; Puton et al., 2013). For both datasets, all methods obtained a higher median specificity than sensitivity (Table 3), indicating that it is more difficult to predict the true pairs than to not predict the false ones. The difference between SN and SP is 5–8 percentage units for short RNAs and 29–40 for long RNAs.
Figure 6  Agreement between predicted and experimentally determined structures

Notes: MCC values were calculated from comparing the entire predicted structure of RNAs with the entire experimental structure. Panel A shows results of the short RNA dataset, while panel B shows results for the long RNA dataset.

Table 3  Performance measures for secondary structure prediction

<table>
<thead>
<tr>
<th></th>
<th>Short RNA dataset</th>
<th></th>
<th>Long RNA dataset</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SN</td>
<td>SP</td>
<td>MCC</td>
<td>SN</td>
</tr>
<tr>
<td>RNAfold</td>
<td>73</td>
<td>78</td>
<td>0.48</td>
<td>38</td>
</tr>
<tr>
<td>UNAfold</td>
<td>73</td>
<td>79</td>
<td>0.47</td>
<td>38</td>
</tr>
<tr>
<td>Sfold</td>
<td>73</td>
<td>79</td>
<td>0.48</td>
<td>38</td>
</tr>
<tr>
<td>ContextFold</td>
<td>87</td>
<td>95</td>
<td>0.80</td>
<td>64</td>
</tr>
</tbody>
</table>

Notes: Values are medians for all sequences in each dataset (short or long RNAs). SN: sensitivity, SP: specificity, MCC: Matthew’s correlation coefficient.

Performance was considerably better for the short RNA dataset, reflecting the fact that the structure of longer RNAs is more difficult to predict. The much larger gap between SN and SP for long RNAs is explained by the fact that the number of possible structures increases exponentially with sequence length.

One explanation for the superior performance of ContextFold is that it relies on a machine-learning model, instead of the energy-based models employed by RNAfold, UNAfold and Sfold. By using parameters estimated from experimental structures, the machine learning model escapes the limitations of relying on empirically estimated parameters in the energy model. Thereby, ContextFold may predict structures closer to the experimental ones, rather than the one with lowest energy. These results indicate that ContextFold gives more reliable predictions for both short and long RNAs and may be more suitable than RNAfold for pre-miRNA prediction.

3.5 Window structure prediction tests

The window structure prediction tests were conducted by extracting 17 subsequences of different size from the RNAs in the long RNA dataset. The windows were centred on the
RNAs such that $w_i$ contains $w_{i-1}$ and 25 base flanks on both sides. These windows were then folded with RNAfold and ContextFold to evaluate the effect of window size on accuracy and the structure obtained for the middle bases of the window. Hence, all comparisons based on the folded window sequences only consider the middle 200 bases. The predicted structures of the windows were compared to the corresponding experimentally derived structures (Figure 7). While specificity remains fairly constant for both methods as the window size increases, a small increase in sensitivity of RNAfold is observed from 200 to 400 bases, followed by a decline from 600 to 1000 bases. For ContextFold, sensitivity increases from 200 to 400 bases, after which it remains around 60–70%. ContextFold outperforms RNAfold by at least 20 percentage units both in terms of median sensitivity and specificity. In other words, structure predictions by ContextFold more closely approximate the true experimental RNA structures compared to RNAfold.

One explanation for the increased sensitivity of ContextFold for larger windows is that it allows the structure prediction algorithm to take additional bases into consideration. Those bases may form base pairs with the middle 200 nucleotides in the experimental structure, which can only be predicted if the flanking bases are included in the window. RNAfold also demonstrates a slight performance increase for larger windows of 400 to 600 bases, but the improvement is marginal and degrades for larger windows. One possible reason for this is that the predicted MFE structure is not a good approximation for the structure of longer RNAs. Longer RNAs have many more possible conformations than short RNAs and the MFE structure is only one possible conformation. This problem is consistent with the results in Figure 6, which clearly demonstrate a performance drop for methods based on the free energy model. Taking both the results from whole and window structure prediction into consideration, ContextFold appears more suitable for secondary structure prediction applied to \textit{in silico} miRNA discovery.

Figure 7  Effect of window size on structure prediction accuracy for the middle 200 bases of windows

A

B

Notes: Circles represent sensitivity and triangles represent specificity of structure prediction for different window sizes. Panel A shows results for RNAfold and panel B shows results for ContextFold.
In addition to comparing the predicted structure of windows to the corresponding experimental structure, the window structures were also compared to each other (for each RNA individually). The purpose of this was to investigate the effect of window size on the structure prediction itself, to see if and to what extent, it changes as larger flanks are added around the middle 200 bases. As when comparing the predicted structure to the experimental one, only the middle 200 bases were considered when comparing the windows to each other. First, the structure of each window \( w_i, i = 2, \ldots, 17 \), were compared to the structure of \( w_{i-1} \). This shows the extent of change in structure prediction for each 25 base flanks added. Second, the structure of each window \( w_i, i = 2, \ldots, 17 \), were compared to the structure of \( w_1 \). For comparisons between \( w_i \) and \( w_{i-1} \), both RNAfold and ContextFold show increased agreement for each pairwise window comparison, where the median MCC approaches 1 for windows \( \geq 600 \) bases (Figure 8). Some difference can be observed between the methods, where RNAfold shows a sharp increase in MCC from 500 to 550 bases, while ContextFold shows a smooth increase in MCC from 200 to 600 bases. The sharp increase in MCC by RNAfold indicates that the free energy structure landscape changes dramatically and that a very different structure becomes the new MFE structure. For comparisons between \( w_i \) and \( w_1 \) a slight decrease in agreement is observed from 200 to 400 bases, with more variation in MCC values for RNAfold compared to ContextFold.

**Figure 8** Agreement between structure predictions for windows of different size

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Graph A" /></td>
<td><img src="image2.png" alt="Graph B" /></td>
</tr>
</tbody>
</table>

**Notes:** Circles represent comparisons between window \( w_i \) and \( w_{i-1} \) and triangles represent comparisons between window \( w_i \) and window \( w_1 \). Comparisons between windows consider only the middle 200 bases. Panel A shows results for RNAfold and panel B shows results for ContextFold. MCC values are medians for all RNAs in the long RNA dataset.

### 3.6 Regression model performance for reconstructed datasets

To investigate the effect on classification performance by changing secondary structure prediction method and window size, the datasets that were initially used to train GenoScan were reconstructed. The new datasets were generated with RNAfold and ContextFold and a window size of 200 and 500 bases. Thus, four sets of datasets were
created and used to train and evaluate the performance of a new regression model (model B) by leave-one-out cross validation. In addition, the original regression model (model A) was used to classify the hairpins of the new datasets with \( p \geq 0.5 \) and results compared with the re-trained model (Table 4). A modest performance improvement can be observed, showing that the new datasets were easier to classify than the original dataset. For RNAfold 200 and 500, model A achieved higher SN and lower SP compared to the original data, indicating higher tendency to classify hairpins in the new dataset as miRNAs. The reverse is true for the ContextFold 200 and 500 datasets. Here lower SN and higher SP is observed, as well as lower GPR. The fact that these hairpins were folded with another secondary structure prediction method probably explains the lower tendency to classify hairpins as miRNAs.

The performance for model B on the reconstructed datasets is considerably higher than the performance of model A on the original data. This can partially be explained by the new datasets containing hairpins that are easier to classify. Another possible contributing factor is that the new positive datasets contain hairpins extracted from the human genome rather than miRBase precursors. By predicting the structure of the precursor while taking parts of the surrounding transcript into consideration, structural features may be obtained that contribute to discriminate between miRNA precursor hairpins and pseudo hairpins. Interestingly, although ContextFold has demonstrated superior performance in terms of agreement between predicted and true structure, the performance of model B is slightly higher for the RNAfold 200 dataset. Thus, in terms of classification performance the results indicate that changing secondary structure prediction method and window size has only minimal effect. By increasing the probability threshold to 0.9, performance in terms of MCC drops from 0.86 to 0.78. Although SN decreases from 92% to 76%, SP increases from 94 to 99 and a lower FDR and GPR are observed (1.2 and 1.5%, respectively). Given the number of hairpins in the human genome, minimising the GPR is very important for miRNA prediction methods that do not rely on conservation or expression of pre-miRNA candidates. Thus it may be acceptable to sacrifice SN for lower GPR in this case.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>SN</th>
<th>SP</th>
<th>MCC</th>
<th>FDR</th>
<th>GPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Original</td>
<td>85</td>
<td>90</td>
<td>0.75</td>
<td>12</td>
<td>7.7</td>
</tr>
<tr>
<td>A</td>
<td>RNAfold 200</td>
<td>92</td>
<td>87</td>
<td>0.79</td>
<td>13</td>
<td>5.4</td>
</tr>
<tr>
<td>A</td>
<td>RNAfold 500</td>
<td>91</td>
<td>87</td>
<td>0.77</td>
<td>14</td>
<td>5.4</td>
</tr>
<tr>
<td>A</td>
<td>ContextFold 200</td>
<td>79</td>
<td>97</td>
<td>0.78</td>
<td>4.2</td>
<td>1.8</td>
</tr>
<tr>
<td>A</td>
<td>ContextFold 500</td>
<td>80</td>
<td>96</td>
<td>0.77</td>
<td>4.9</td>
<td>2.0</td>
</tr>
<tr>
<td>B</td>
<td>RNAfold 200</td>
<td>92</td>
<td>94</td>
<td>0.86</td>
<td>6.9</td>
<td>4.2</td>
</tr>
<tr>
<td>B</td>
<td>RNAfold 500</td>
<td>91</td>
<td>92</td>
<td>0.83</td>
<td>8.7</td>
<td>4.5</td>
</tr>
<tr>
<td>B</td>
<td>ContextFold 200</td>
<td>91</td>
<td>93</td>
<td>0.84</td>
<td>7.6</td>
<td>6.0</td>
</tr>
<tr>
<td>B</td>
<td>ContextFold 500</td>
<td>90</td>
<td>93</td>
<td>0.84</td>
<td>7.4</td>
<td>5.7</td>
</tr>
</tbody>
</table>

Notes: Performance measures for the regression model were obtained with \( p \geq 0.5 \). Model A refers to the GenoScan model trained on the original dataset. Model B indicates the model trained on the reconstructed datasets.
4 Conclusions

This study presented GenoScan, a software package for genome-wide miRNA discovery. In the benchmarking of GenoScan’s ensemble-based and regression-based classifiers, along with published hairpin classification methods, the regression model achieved highest classification accuracy. GenoScan does not rely on conservation or transcriptome data for generating candidate miRNAs and supports user-defined filters to exclude or target specific regions. This makes GenoScan a flexible tool for genome-scale miRNA discovery. The performance of the RNA secondary structure prediction methods and their impact on pre-miRNA discovery were also investigated. Of the four evaluated methods, ContextFold had the highest agreement between predicted and experimentally determined structures. However, the choice of structure prediction method had only minimal impact on hairpin classification.

Finally, evaluation of the GenoScan regression model on different datasets showed that higher performance is achieved when pre-miRNAs extracted from genomic sequences are used compared to miRBase precursors. To further improve genome-wide discovery of miRNAs, future improvements of GenoScan will be aimed at lowering the genome prediction rate and implementing support for miRNA candidate validation by incorporating data such as expressed sequence tags, transcription start sites and NGS reads.

5 Availability

GenoScan is available as downloadable software from the Comprehensive Perl Archive Network (CPAN) in the form of the Software::GenoScan module. It was developed and tested in the Linux Ubuntu environment.

References


B. Ulfenborg, K. Klinga-Levan and B. Olsson


Genome-wide discovery of miRNAs


