Structural bioinformatics

Comparison of \textit{P-RnaPredict} and \textit{mfold}—algorithms for RNA secondary structure prediction

Kay C. Wiese* and Andrew Hendriks

School of Computing Science and InfoNet Media Centre, Simon Fraser University, 15th Floor, Central City Tower, 13450 102nd Avenue, Surrey, BC, Canada V3T 5X3

Received on November 4, 2005; revised on February 2, 2006; accepted on February 3, 2006
Advance Access publication February 10, 2006
Associate Editor: Anna Tramontano

ABSTRACT
Motivation: Ribonucleic acid is vital in numerous stages of protein synthesis; it also possesses important functional and structural roles within the cell. The function of an RNA molecule within a particular organic system is principally determined by its structure. The current physical methods available for structure determination are time-consuming and expensive. Hence, computational methods for structure prediction are sought after. The energies involved by the formation of secondary structure elements are significantly greater than those of tertiary elements. Therefore, RNA structure prediction focuses on secondary structure.

Results: We present \textit{P-RnaPredict}, a parallel evolutionary algorithm for RNA secondary structure prediction. The speedup provided by parallelization is investigated with five sequences, and a dramatic improvement in speedup is demonstrated, especially with longer sequences. An evaluation of the performance of \textit{P-RnaPredict} in terms of prediction accuracy is made through comparison with 10 individual known structures from 3 RNA classes (5S rRNA, Group I intron 16S rRNA and 16S rRNA) and the \textit{mfold} dynamic programming algorithm. \textit{P-RnaPredict} is able to predict structures with higher true positive base pair counts and lower false positives than \textit{mfold} on certain sequences.

Availability: \textit{P-RnaPredict} is available for non-commercial usage. Interested parties should contact Kay C. Wiese (wiese@cs.sfu.ca).

Contact: wiese@cs.sfu.ca

1 INTRODUCTION

The shape of a given RNA molecule essentially governs its functionality. RNA has key structural and functional roles in such processes as protein synthesis. Various algorithms employed in the prediction of RNA structure include comparative methods (Gardner and Giegerich, 2004), kinetic folding (Flamm et al., 1999), evolutionary algorithms (EAs) and dynamic programming (DP).

Comparative methods work by analyzing multiple sequences to determine points of base covariance within the sequences; these points indicate probable base pair positions. Kinetic folding attempts to simulate the folding dynamics of RNA secondary structure through the gradual modification of base pair positions.

Structure prediction can be expressed as an energy minimization problem through the use of thermodynamic models; this is how structure prediction is performed in DP and EAs. Gultyaev et al. (1995, 1998) have established that an EA can predict RNA secondary structure with a high degree of accuracy. Also, the addition and deletion of stems by the EA approximates the RNA folding pathway followed during the folding process.

Although it has been demonstrated that DP can accurately predict a minimum energy structure within a given thermodynamic model, the native fold is often in a suboptimal energy state (van Batenburg et al., 1995) which varies greatly from the predicted one. A case may be made that the natural folding process of RNA and the simulated folding of RNA using an EA, which includes intermediate folds, have much in common.

A massively parallel EA for RNA folding was introduced by Shapiro and Navetta (1994) and implemented on a specialized MasPar platform; an improved mutation operator for this algorithm (Shapiro and Wu, 1996) was added later.

Other related research on EAs and RNA structure includes Currey and Shapiro (1997), Chen and Dill (2000), Titov et al. (2002) and Fogel et al. (2002). Currey and Shapiro (1997) discuss the enhancements of secondary structure prediction of the poliovirus 5' non-coding region by a genetic algorithm. Chen and Dill (2000) provide an interesting study on RNA folding energy landscapes that demonstrates that the folding of RNA secondary structures may involve complex intermediate states and rugged energy landscapes. In Titov et al. (2002) a fast genetic algorithm for RNA secondary structure prediction and subsequent structure analysis is introduced. Their GARNA algorithm was used for fast calculations of RNA secondary structure and GARNA results are used to interpret how secondary RNA structure influences translation initiation and expression regulation. In Fogel et al. (2002) a method based on evolutionary computation to search possible RNA secondary structures for common elements across multiple sequences without requiring pre-alignment of the sequences is described.

Coarse-grained distributed EAs (Cantú-Paz, 2000) offer significant advantages beyond the speedup gained through parallelization. These include the prevention of premature convergence by maintaining diversity, an increase of the selection pressure within the entire population and also reduction of the time to
convergence. This type of EA differs from a massively parallel EA such as the one implemented by Shapiro et al. (2001) in how the population is distributed. A distributed EA maintains its individuals in two or more subpopulations which occasionally exchange individuals through ‘migration.’ Each subpopulation is allocated its own processor. By contrast, massively parallel EAs typically distribute their population across a two-dimensional grid with one individual per grid point, preferably allocating one processor per individual.

P-RnaPredict is a fully parallel implementation of a coarse-grained distributed EA for RNA secondary structure prediction. It is based on RnaPredict, a serial EA for the same purpose developed by Wiese and Glen (2003) who have proposed to use permutations to encode RNA secondary structures. In this paper, the authors focused on demonstrating the superiority of permutation-based over binary-based representation and studying the effects of various crossover operators and selection strategies such as keep-best reproduction (KBR) (Wiese and Goodwin, 2001). RnaPredict was later adapted in Descheˆnes and Wiese (2004) to include two stacking-energy-based thermodynamic models; testing with three RNA sequences demonstrated the improved prediction accuracy with the new thermodynamic models.

The work presented here builds on Hendriks et al. (2004), in which a serial simulation of a distributed version of RnaPredict was implemented that predicted the secondary structure of RNA molecules from input RNA sequences. In this research, the distributed EA was demonstrated to perform comparably to the original serial implementation of RnaPredict on three known structures. The effects of two new stacking energy thermodynamic models were also tested with the serial simulation.

Most recently, the serial simulation was adapted into a fully parallel implementation, P-RnaPredict (Wiese et al., 2005). Here, the effects of pseudorandom number generators on the EA were investigated, along with its performance on five known structures.

The first application of DP to RNA secondary structure prediction was developed by Nussinov et al. (1978) and functioned by maximizing the number of base pairs in a predicted structure. In this case it was assumed that the formation of a given base pair was independent of all other base pairs. The main problems with this first DP algorithm were as follows. First, there were no constraints on hairpin loop length, whereas actual RNA requires a minimum of ~3 nt. Second, there were no size constraints on bulges and internal loops; this resulted in helices with shorter lengths than what would occur in actual RNA. Later, Zuker and co-workers (Zuker and Stiegler, 1981; Zuker, 1989, 1994; Zuker et al., 1999; Zuker, 2000, 2003) developed an alternate approach, called mfold, which employed thermodynamic models to minimize the free energy of the predicted structure with increased prediction accuracy.

This paper will present a comparison of the prediction accuracy of P-RnaPredict with that of the mfold DPA (Zuker, 2003). The objectives of this paper are as follows:

- To present the speedup achieved through parallelization
- To compare the accuracy of structures predicted by P-RnaPredict against the accuracy of structures predicted by the mfold DPA.

First, we describe the algorithm design in Section 2. Next, Section 3 presents the experiment design and their results. Finally, Section 4 comments on the results, summarizes the paper and offers a brief summary of future work.

### 2 APPROACH

#### 2.1 Evolutionary algorithms

Evolutionary computation (Bäck, 1996) refers to methods by which evolution is simulated on a computing platform. The algorithms reliant on these methods are known as evolutionary algorithms (EAs), and include evolutionary programming (Fogel et al., 1966), evolution strategies (Rechenberg, 1973; Schwefel, 1977), genetic algorithms (Holland, 1975) and genetic programming (Koza, 1992). EAs are widely applicable search and optimization techniques; P-RnaPredict is a type of EA. In general, EAs have five basic components: a genetic representation of solutions to the problem, a way to create an initial population of candidate solutions, an evaluation function rating solutions in terms of their fitness, genetic operators that alter the genetic composition of offspring during reproduction and values for various parameters.

EAs maintain a population of individuals in which each individual represents a potential solution to the given problem. In each cycle or ‘generation’ each individual is evaluated to determine its relative fitness within the population; individuals representing the population’s best solutions to the problem are retained via a selection operator. Mutation and crossover may also be applied to the population. Mutation creates new individuals by randomly altering individuals, while crossover combines elements from two separate individuals to make new individuals. The generation is then complete, and the cycle repeats itself with the new population. After a varying number of generations, the algorithm will converge to the best individual, which represents a suboptimal or optimal solution to the given problem. EAs execute an essentially blind search in complex fitness landscapes; selection operators should direct the search toward the most favorable areas of the landscape. EAs must strike a balance between exploitation and exploration to be successful (Gen and Cheng, 2000).

In P-RnaPredict, RNA secondary structure prediction proceeds with an ab initio method, where only the linear sequence of nucleotides making up a given RNA molecule is known. These nucleotides, or bases, are adenine, guanine, cytosine and uracil (A, G, C and U). In our model, these bases bond to what are known as the canonical base pairs: GC, AU and GU, and their mirrors. RNA secondary structure ultimately results from these base pairings. Determining every possible canonical base pair which could form in an RNA nucleotide sequence is rapid and straightforward. However, it is much more difficult to predict the specific base pairings that result in the native RNA structure.

Higher order RNA secondary structure elements include hairpin loops, internal loops, bulges, multi-branched loops and external bases. Stacked base pairs form helices; these add stability to the secondary structure.

The first stage of P-RnaPredict’s algorithm is to build a set \( H \) of all the potential helices which could form in a given RNA sequence under our model. First, all canonical base pairs for a given RNA sequence are iterated through, and the algorithm attempts to build a helix by stacking additional base pairs on top of existing ones. Consider an RNA sequence \( r \) of length \( n \), and a canonical base pair \((r_i, r_j)\), where \( 0 \leq i < j < n \). A base pair that does not meet the condition of \( |j - i| > 7 \) is discarded, as it could not be the base of a valid helix under our model. P-RnaPredict checks \((r_{i+1}, r_{j-1})\) to determine if it is a canonical base pair. If so, this base pair is added to the potential helix \( H \), and the next potential base pair \((r_{i+k}, r_{j-\lambda})\) is checked. This proceeds until the first non-canonical base pair is encountered, or until we reach \((r_{i+k}, r_{j-\lambda})\).
Thus, representation. In the EA encodes a potential RNA structure via a permutation-based permutation is decoded from left to right. The helix specified by each integer number of generated helices. To produce chemically feasible structures, each models and are explained in Section 2.2.

As mentioned, RNA folds into a structure with near minimal free energy. Thus, P-RnaPredict attempts to find the combination of helices that result in a chemically feasible structure with the lowest free energy possible. A given RNA secondary structure is ultimately encoded as a permutation. Each helix in H is numbered by an integer ranging from 0 to n − 1, n being the total number of generated helices. To produce chemically feasible structures, each permutation is decoded from left to right. The helix specified by each integer is checked for conflicts with helices to its left. If no conflicts are found, the helix is retained; otherwise it is discarded. In summary, each individual in the EA encodes a potential RNA structure via a permutation-based representation.

Unlike RnaPredict (Wiese and Glen, 2003) which maintains a single panmictic population, P-RnaPredict separates its population into segregated subpopulations which are also known as demes. The isolated demes behave essentially like miniature serial EAs and exchange individuals with each other on an intermittent basis through migration. This is known as a coarse-grained distributed model and refers to a high ratio of time spent in computation versus the time spent in communication. Coarse-grained distributed EAs are referred to as ‘pleasingly parallel’, as each deme may be assigned to a separate processor.

P-RnaPredict should improve performance by preserving diversity in the population through multiple demes, while increasing the selection pressure through periodic migration (Hendriks et al., 2003). A detailed description of the design of the P-RnaPredict algorithm can be found in Hendriks (2005).

2.2 Thermodynamic models

P-RnaPredict uses two thermodynamic models: individual nearest neighbor (INN) (Serra and Turner, 1995) and individual nearest neighbor-hydrogen bond (INN-HB) (Xia et al., 1998). The essential idea behind the INN and INN-HB stacking-energy models employed in P-RnaPredict is that the stabilizing contribution each base pair makes to its helix depends on that base pairs’ nearest neighbors. For example, the free energy contribution of a GC base pair would vary depending on whether the adjacent base pair in the helix is either an AU base pair, or its mirror a UA base pair.

2.2.1 Individual nearest-neighbor model There are two distinct components to computing the free energy of a helix using INN. The first is initiation, or the formation of the first base pair. Initiation brings the two strands together and entails hydrogen bonding. The second component is propagation, or the continued formation of subsequent base pairs. Propagation involves nearest-neighbor or stacking interactions as well as hydrogen bonding. The nearest-neighbor thermodynamic parameters used in the INN model were initially measured at 25°C (Borer et al., 1974), but were later re-measured and extended at 37°C (Freier et al., 1986; He et al., 1991; Sugimoto et al., 1986; Wu et al., 1995). A thorough review of the INN model complete with thermodynamic parameters can be found in Serra and Turner (1995).

2.2.2 Individual nearest-neighbor hydrogen bond model Later experimentation determined that duplexes with identical nearest neighbors but varying terminal ends also differed in their stabilities. Specifically, a duplex with one additional terminal GC pair and one less terminal AU pair is always more stable (Xia et al., 1998). This new thermodynamic detail is added to the INN model, resulting in the individual nearest-neighbor hydrogen bond model. Although the INN-HB model only specifies a penalty for terminal AU pairs, we give terminal GU pairs the same penalty as suggested by Mathews et al. (1999).

2.3 Dynamic programming

DP is a technique typically applied to optimization problems which possess two properties. The first is optimal substructure, where the optimal solution to the problem has optimal solutions to its subproblems. The second property is overlapping subproblems, in that a recursive algorithm for the problem solves the same subproblems repeatedly. Thus, the prediction problem can be divided into subproblems, the subproblems solved and their solution tabulated, and the final structure computed from the tabulated subproblem solutions using DP. As with our EA approach, the DP approach to RNA secondary structure prediction proceeds with an ab initio method, where only the primary structure of the RNA molecule in question is known.

3 METHODS AND RESULTS

All P-RnaPredict experiments were conducted on a 128 node Beowulf cluster. Each node’s CPU is a 3 GHz Intel Pentium 4, and the nodes are interconnected with a Gigabit Ethernet network. P-RnaPredict itself was implemented in 18 000 lines of C++ code; the Message Passing Interface (MPI) parallel standard was employed for communication. The pseudorandom number generator implemented in P-RnaPredict is Dynamic Creation, a parallelized Mersenne Twister (Matsumoto and Nishimura, 1998).

A total of 10 sequences were employed for testing P-RnaPredict; one RNA sequence was taken from each of the following genomes: Saccharomyces cerevisiae, Halocarcinula marismortui, Aureoumbra lagunensis, Hildenbrandia rubra, Acanthamoeba griffini, Caenorhabditis elegans, Drosophila virilis, Xenopus laevis, Homo sapiens and Sulfolobus acidocaldarius. All sequences were taken from the Comparative RNA web site (Cannone et al., 2002), and their details are summarized in Table 1. For brevity, throughout this manuscript we refer to these specific RNA sequences by the name of the organism that they were taken from.

The parameters which remained fixed throughout all experiments presented here are as follows: 700 generations, a crossover probability (Pc) of 0.7, a mutation probability (Pm) of 0.8, standard roulette wheel selection (STDS) with 1-Elitism, a migration interval of 20 generations, a migration rate of 10%, a fully connected (island) topology and a migration policy of best-replace-worst. These fixed parameters were determined experimentally to be among the best performing runs in the serial simulation of the distributed EA (Hendriks et al., 2003; Wiese et al., 2003).

3.1 Parallel speedup

A series of experiments were conducted to investigate the speedup and efficiency of P-RnaPredict. Speedup, or the speedup factor S(t), is defined as the ratio of the serial time ts to the parallel
time \( t_p(n) \), where \( n \) is the number of processors:

\[
S(n) = \frac{t_s(n)}{t_p(n)}.
\]

The system efficiency \( E(n) \), where \( n \) is the number of processors, is defined by the following equation:

\[
E(n) = \left( \frac{t_s(n)}{t_p(n) \cdot n} \right) \cdot 100\% = \frac{S(n)}{n} \cdot 100\%.
\]

Figure 1 is taken from an experiment (Hendriks, 2005) on the \textit{H. rubra} sequence from Table 1. The objective was to determine the effects of subdividing a population of constant size among an increasing number of processors. Thus, the number of demes was varied while holding the total population size constant at 700. It can be seen that while \( S(n) \) increases as the number of demes increases, \( S(n) \) does not double when the number of processors is doubled. This reduction in \( E(n) \) is due to rising communications overhead.

A second series of experiments were conducted to investigate the speedup of \textit{P-RnaPredict}. Both serial and parallelized versions of \textit{P-RnaPredict} were tested with the following five RNA sequences from Table 1: \textit{C. elegans}, \textit{A. griffini}, \textit{H. rubra}, \textit{H. marismortui} and \textit{S. cerevisiae}.

The parameters that were varied for the speedup experiments were as follows. A cycle crossover (CX) (Oliver \textit{et al.}, 1987) and the INN-HB thermodynamic model were employed. The deme size and deme count were varied between two sets of values: (50, 14) and (70, 10). These were determined experimentally to be among the best performing runs in the serial simulation of the distributed EA (Hendriks \textit{et al.}, 2003; Wiese \textit{et al.}, 2003). Each experiment was performed with 30 randomly seeded runs. With 2 parameter sets and 5 sequences with 30 runs each, a total of 300 runs were performed. Note that in the parallel implementation, each deme is assigned to its own processor.

Table 2 compiles the results of the speedup test runs. Deme count is the total number of demes in the experiment, while deme size is the total number of individuals per deme in the experiment; the results in Table 2 are split between the deme counts of 10 and 14 for clarity and sorted by sequence length. A clear improvement can be seen in \( S(n) \) and \( E(n) \) as the sequence lengths increase. Note that the differing deme counts result in differing relationships with \( E(n) \). For example, the \textit{C. elegans} entry with 10 demes and an \( S(n) \) of 6.3 results in an \( E(n) \) of 63\%, while the corresponding 14 deme entry has an \( S(n) \) of 7.6 but results in an \( E(n) \) of 54\%.

The \textit{H. rubra} sequence runtimes are notable. Although \textit{H. rubra} is shorter than \textit{A. griffini}, the \textit{H. rubra} sequence generates more potential helices under \textit{P-RnaPredict}'s model, and thus has a longer representation. Hence, the representation length rather than the sequence length ultimately influences the runtime. Also, note that the speedups in Figure 1 and Table 2 were derived with different experiment settings and cannot be compared directly.

It is also possible to see that there is not a linear relationship between sequence length and runtime. For shorter sequences, \( S(n) \) and \( E(n) \) appear low compared with the number of processors available (10 or 14). This occurs because the time required to process one sequence runtimes are notable. Although \textit{H. rubra} is a shorter sequence than \textit{A. griffini}, the \textit{H. rubra} sequence generates more potential helices under \textit{P-RnaPredict}'s model, and thus has a longer representation. Hence, the representation length rather than the sequence length ultimately influences the runtime. Also, note that the speedups in Figure 1 and Table 2 were derived with different experiment settings and cannot be compared directly.
the time consumed through communication-related activities such as migration. As the sequences increase in length, there is a corresponding increase in the representation length. Thus, the amount of time spent in computation in an individual deme considerably exceeds the time spent in communication, and $S(n)$ and $E(n)$ improve dramatically. Higher values for $S(n)$ and $E(n)$ for longer sequences demonstrate that parallelization is worthwhile in order to drastically reduce execution times for the experiments.

### 3.2 Comparison to mfold

What follows is a quantitative measure of P-RnaPredict’s performance through a comparison between a set of 10 known structures, the structures predicted by P-RnaPredict, and structures predicted by mfold. The known structures are determined via comparative methods, and are taken from the Comparative RNA Web Site (Cannone et al., 2002); their details are in Table 1. The primary criteria for comparison are true-positive matching base pairs and false positives. Also, sensitivity, specificity and F-measures are discussed. The parameters which varied for these experiments were chosen based on prior research (Hendriks et al., 2003; Wiese et al., 2003; Deschênes and Wiese, 2004) and were set as follows: The INN and INN-HB thermodynamic models, and the OX2 (Starkweather et al., 1991) and CX (Oliver et al., 1987) crossovers were selected. A notable exception was H.marismortui, for which P-RnaPredict produced the best structures with the Edge Recombination Crossover (ERC) operator (Whitley et al., 1991). Three separate sets of deme sizes and deme counts were employed: (50, 14), (70, 10) and (100, 10), respectively. The parameter set for each experiment was repeated with 30 random seeds and the results averaged. With 12 parameter sets and 10 sequences with 30 runs each, a total of 3600 P-RnaPredict runs were performed. The mfold results presented here were generated using mfold web server version 3.1 with default settings. Tables 3–9 present the results of the comparison between mfold and P-RnaPredict. Superior results (high true positive base pair predictions/low false base pair predictions) are presented in bold font.

Table 3 presents the results of comparing the prediction accuracy of P-RnaPredict (minimum ΔG structure) and mfold (minimum ΔG structure) on 10 RNA sequences from different organisms and differing lengths. The sequences are ordered by increasing length. The first multicolumn compares the absolute number of correct base pairs predicted by mfold and by P-RnaPredict, while the second multicolumn compares the percentages of true positive base pairs predicted by the respective algorithms. In general, we see that both algorithms have a good prediction accuracy for shorter sequences while for longer sequences the prediction accuracy is generally lower. For the S.cerevisiae sequence, both mfold and P-RnaPredict predict an identical amount of correct base pairs. In 3 out of 10 cases P-RnaPredict predicts a structure with more true positive base pairs than mfold. P-RnaPredict performs relatively poorly on the H.marismortui sequence.

Table 4 compares the amount of false positive base pairs predicted by mfold with the false positive predictions of P-RnaPredict (EA). We note that with shorter RNA sequences,

| Organism          | Seq. length (nt) | Serial time (s) | Parallel time (s) | S(n) | E(n) (%)
|-------------------|------------------|-----------------|-------------------|------|--------
| C.elegans         | 697              | 1264.4          | 202.4             | 6.3  | 63     |
| A.griffini        | 556              | 754.0           | 127.8             | 5.9  | 59     |
| H.rubra           | 543              | 871.9           | 145.0             | 6.0  | 60     |
| H.marismortui     | 122              | 32.1            | 11.5              | 2.8  | 28     |
| S.cerevisiae      | 118              | 27.5            | 10.9              | 2.5  | 25     |
| Proc. count: 14   |                  |                 |                   |      |        |
| C.elegans         | 697              | 1255.8          | 165.0             | 7.6  | 54     |
| A.griffini        | 556              | 759.3           | 106.8             | 7.1  | 51     |
| H.rubra           | 543              | 870.9           | 203.1             | 4.3  | 31     |
| H.marismortui     | 122              | 31.6            | 11.3              | 2.8  | 20     |
| S.cerevisiae      | 118              | 27.2            | 10.9              | 2.5  | 18     |

Table 3. Minimum ΔG mfold versus minimum ΔG P-RnaPredict (EA): true positive base pairs

<table>
<thead>
<tr>
<th>Sequence</th>
<th>True positive base pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mfold</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>S.cerevisiae</td>
<td>33</td>
</tr>
<tr>
<td>H.marismortui</td>
<td>29</td>
</tr>
<tr>
<td>A.lagunensis</td>
<td>60</td>
</tr>
<tr>
<td>H.rubra</td>
<td>49</td>
</tr>
<tr>
<td>A.griffini</td>
<td>67</td>
</tr>
<tr>
<td>C.elegans</td>
<td>40</td>
</tr>
<tr>
<td>D.viridis</td>
<td>37</td>
</tr>
<tr>
<td>X.laevis</td>
<td>92</td>
</tr>
<tr>
<td>H.sapiens</td>
<td>95</td>
</tr>
<tr>
<td>S.acidocaldarius</td>
<td>261</td>
</tr>
</tbody>
</table>

Table 4. Minimum ΔG mfold versus minimum ΔG P-RnaPredict (EA): false positive base pairs (overprediction)

<table>
<thead>
<tr>
<th>Sequence</th>
<th>False positives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(mfold)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>S.cerevisiae</td>
<td>8</td>
</tr>
<tr>
<td>H.marismortui</td>
<td>5</td>
</tr>
<tr>
<td>A.lagunensis</td>
<td>68</td>
</tr>
<tr>
<td>H.rubra</td>
<td>127</td>
</tr>
<tr>
<td>A.griffini</td>
<td>105</td>
</tr>
<tr>
<td>C.elegans</td>
<td>177</td>
</tr>
<tr>
<td>D.viridis</td>
<td>199</td>
</tr>
<tr>
<td>X.laevis</td>
<td>157</td>
</tr>
<tr>
<td>H.sapiens</td>
<td>163</td>
</tr>
<tr>
<td>S.acidocaldarius</td>
<td>233</td>
</tr>
</tbody>
</table>

Table 4. Minimum ΔG mfold versus minimum ΔG P-RnaPredict (EA): false positive base pairs (overprediction)

'Organism' is the name of the organism from which the RNA sequence is taken. 'Seq. Length (nt)' is the length of the specific sequence in nucleotides (nt). 'Serial time (s)' is the total runtime of the serial simulation of the distributed EA in seconds, averaged over 30 randomly seeded runs. 'Parallel time (s)' is the total runtime of the fully parallel distributed EA in seconds, averaged over 30 randomly seeded runs.

'Sequence' is the specific sequence from which the RNA sequence is taken.

'EA' performs relatively poorly on the H.marismortui sequence.

'Table 4' compares the amount of false positive base pairs predicted by mfold with the false positive predictions of P-RnaPredict (EA). We note that with shorter RNA sequences,
there are relatively few false positives and the number of over-
predictions increases quite quickly with increasing sequence length.
*X.laevis* is somewhat of an exception to this trend. It is interesting to
note that for *S.cerevisiae* both *mfold* and *P-RnaPredict* predicted
33 identical true positive base pairs. However, *mfold* predicted two
more false positives, therefore the predicted structures are different.
Overall, for this test set, *P-RnaPredict* had fewer false positives than
*mfold* for 3 out of 10 RNA structures. Three trends are visible within
this table. First, all runs involving short RNA sequences such as
*C.elegans* converged to identical results regardless of para-
meter settings. Second, the OX2 crossover operator either produced
the best result or tied for best result for all but 1 of the 10 sequences.
Lastly, runs with deme sizes of 100 also performed best or tied for
best result for all but 1 sequence.

One advantage of EAs is that they maintain a population of
candidate solutions. We have searched for structures with a high
true positive base pair count in the final population of *P-RnaPredict*
regardless of free energy. The results of a comparison between these
structures and *mfold* are presented in Table 5. For the longest
sequence (*S.acidocaldarius*), the structure predicted by *mfold* has
a substantially higher number of true positive base pairs than
*P-RnaPredict*. In the case of *S.cerevisiae*, there is again a tie. For
*H.marismortui* and *H.sapiens* the true positive base pair count is
very similar between the two algorithms (within 2 and 3 bp,
respectively). For the remaining six RNA sequences the structures
predicted by *P-RnaPredict* have a substantially higher true positive
base pair count than the corresponding structures predicted by
*mfold*.

A comparison of the number of false positive base pairs is present-
ed in Table 6. For *S.acidocaldarius* *mfold* has substantially fewer
false predictions. For *X.laevis* both algorithms perform almost
equally with *mfold* having 157 overpredictions compared with
158 for *P-RnaPredict*. In the remaining 8 out of 10 predicted
RNA structures, *P-RnaPredict* has substantially fewer false posi-
tives than *mfold*. Reviewing trends in the parameter settings, we see
that again runs with short sequences tend to converge to identical
structures regardless of parameter settings. Another point is that
larger deme sizes again have a significant effect on the results; the
smallest deme size of 50 does not produce the competitive results
for any but the shortest sequences.

### Table 5. Minimum ΔG *mfold* versus best base pairs structure *P-RnaPredict* (EA): true positive base pairs

<table>
<thead>
<tr>
<th>Sequence</th>
<th>True positive base pairs</th>
<th>True positive %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>mfold</em></td>
<td><em>P-RnaPredict</em></td>
</tr>
<tr>
<td><em>S.cerevisiae</em></td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td><em>H.marismortui</em></td>
<td>29</td>
<td>27</td>
</tr>
<tr>
<td><em>A.lagunensis</em></td>
<td>60</td>
<td>73</td>
</tr>
<tr>
<td><em>H.rubra</em></td>
<td>49</td>
<td>71</td>
</tr>
<tr>
<td><em>A.griffini</em></td>
<td>67</td>
<td>79</td>
</tr>
<tr>
<td><em>C.elegans</em></td>
<td>40</td>
<td>56</td>
</tr>
<tr>
<td><em>D.viridis</em></td>
<td>37</td>
<td>66</td>
</tr>
<tr>
<td><em>X.laevis</em></td>
<td>92</td>
<td>100</td>
</tr>
<tr>
<td><em>H.sapiens</em></td>
<td>95</td>
<td>92</td>
</tr>
<tr>
<td><em>S.acidocaldarius</em></td>
<td>261</td>
<td>159</td>
</tr>
</tbody>
</table>

The best results for each sequence are in bold.

### Table 6. Minimum ΔG *mfold* versus best base pairs structure *P-RnaPredict* (EA): false positive base pairs (overprediction)

<table>
<thead>
<tr>
<th>Sequence</th>
<th>False positives</th>
<th>False positives (EA)</th>
<th>Crossover-deme size-deme count-model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(mfold)</td>
<td>(EA)</td>
<td></td>
</tr>
<tr>
<td><em>S.cerevisiae</em></td>
<td>8</td>
<td>6</td>
<td>ALL-ALL-ALL-INNHB</td>
</tr>
<tr>
<td><em>H.marismortui</em></td>
<td>5</td>
<td>3</td>
<td>ERC-ALL-ALL-INNHB</td>
</tr>
<tr>
<td><em>A.lagunensis</em></td>
<td>68</td>
<td>59</td>
<td>CX-70-10-INNHB</td>
</tr>
<tr>
<td><em>H.rubra</em></td>
<td>127</td>
<td>88</td>
<td>OX2-100-10-INNHB</td>
</tr>
<tr>
<td><em>A.griffini</em></td>
<td>105</td>
<td>78</td>
<td>CX-70-10-INNHB</td>
</tr>
<tr>
<td><em>C.elegans</em></td>
<td>177</td>
<td>146</td>
<td>OX2-70-10-INN</td>
</tr>
<tr>
<td><em>D.viridis</em></td>
<td>199</td>
<td>168</td>
<td>CX-100-10-INN</td>
</tr>
<tr>
<td><em>X.laevis</em></td>
<td>157</td>
<td>158</td>
<td>OX2-100-10-INN</td>
</tr>
<tr>
<td><em>H.sapiens</em></td>
<td>163</td>
<td>135</td>
<td>OX2-70-10-INNHB</td>
</tr>
<tr>
<td><em>S.acidocaldarius</em></td>
<td>233</td>
<td>288</td>
<td>OX2-100-10-INNHB</td>
</tr>
</tbody>
</table>

The best results for each sequence are in bold.

While DP typically aims at producing a single optimum result
based on an objective function (here: minimum ΔG), EAs produce a
population of suboptimal solutions, which may contain the global
optimum. Tables 3 and 5 have demonstrated that structures from
*P-RnaPredict* with a higher ΔG can have a higher true positive
base pair count than the lowest ΔG structure found. Many of these
structures have compared favorably with the minimum ΔG
structures found by *mfold* (Tables 5 and 6). However, *mfold* also
offers an option to produce suboptimal RNA foldings. We have used
the *mfold* default setting of 5% suboptimality and searched the
produced suboptimal foldings for structures with high base pair
overlap compared with the known structures (Tables 7 and 8).

Table 7 compares the best base pair structure found by *mfold* with
that of *P-RnaPredict* in terms of true positive base pairs. The
performance of both approaches is identical for *S.cerevisiae*. For
*A.lagunensis*, *H.marismortui* and *H.sapiens* the performance is
within 1, 2 and 3 bp, respectively. For *C.elegans* the structure
predicted by *P-RnaPredict* is substantially better than the *mfold*
structure. For the remaining five sequences, the *mfold* structures
were substantially better.

The data on false positives (Table 8) indicate that for
*A.lagunensis* both approaches predict an equal number of false
positives. For the remaining sequences, *mfold* has fewer false
positives on three structures while *P-RnaPredict* has fewer false
positives on six structures. It is interesting to note that while the
overall performance of *mfold* suboptimal structures in terms of true
positive base pairs was higher than the one of *P-RnaPredict*, the
performance of *mfold* is worse when it comes to false positive
predictions on a per sequence basis. This should be seen as a
tradeoff.

A more detailed analysis of our results of comparing the best
structure from *P-RnaPredict* and the best structure from *mfold* is
given in Table 9. Before discussing the results in Table 9 let us
define some of the measures used to assess the performance of both
algorithms. Let TP be the number of true positive base pairs
predicted, FP be the number of false positive base pairs predicted
and FN be the number of false negative base pairs predicted.
A commonly used measure of performance for a predictor is the
sensitivity, which is computed as TP/(TP + FN) and is the
proportion of correctly predicted base pairs versus the base pairs
found in the native structure. In general, increasing the number of
Another interesting observation is that the number of false positives is not desirable. To establish the effectiveness of a prediction method, another indicator, called specificity is often used. In Baldi et al. (2000) specificity is defined as TP/(TP + FP) and is also known as positive predictive value. It should be noted that this definition is different from the definition of specificity employed in medical statistics. Here, specificity is the proportion of all predicted true positive base pairs compared with all base pairs predicted. A predictor A that has the same TP count but has a higher FP count than a predictor B will have a lower specificity than B and is considered to have a lower performance. The F-measure is a metric that combines both specificity and sensitivity into a single measure: $F = 2 \times \frac{TP}{TP + FP} \times \frac{specificity + sensitivity}{specificity + sensitivity}$ and can be used as a single performance measure for a predictor. In Table 9, the first column lists the organism the RNA sequences were taken from, the second column lists the number of base pairs in the known structure, and the third double column lists the total number of base pairs predicted. The next three double columns list the true positive base pairs (TP), the false positive base pairs (FP) and the false negative base pairs (FN) predicted by P-RnaPredict and mfold. The sensitivity and specificity of both P-RnaPredict and mfold are given in double columns 7 and 8, respectively. The F-measure values for both P-RnaPredict and mfold are given in double column 9.

There is a noticeable performance drop of P-RnaPredict on S.acidocaldarius 16S rRNA over all other sequences. While the known S.acidocaldarius structure has 22 non-canonical base pairs out of a total of 468 bp, this alone does not explain the poor performance of P-RnaPredict on this particular sequence. Non-canonical base pairs are neither modeled by P-RnaPredict nor by mfold. We speculate that the increased sequence length might be a factor, but not the determining one. Also, compositional differences in the sequence could be a factor, but this is not clear yet. Another interesting observation is that S.acidocaldarius is an extremophile (acidophile) which lives in a hot (85°C) acidic (sulphurous) environment. This could also have an impact on its 16S rRNA structure and certainly thermodynamic models derived at 37°C may not be as suitable in such cases. Future work with multiple long 16S rRNA sequences from organisms that live in regular environments will hopefully provide further insights. Owing to this behavior, we consider S.acidocaldarius an outlier. In the discussion that follows we present performance measures for both the complete test set as well as the test set without the outlier.

Several interesting observations can be made: (1) The average number of predicted base pairs of P-RnaPredict is much lower than the average for mfold: 188.5 bp versus 201.7 bp for all 10 sequences and 159.8 bp versus 169.0 bp for the 9 sequences when S.acidocaldarius is removed from the test set. (2) The average number of base pairs predicted by P-RnaPredict is very close to the average number of base pairs in the known structure: 188.5 bp versus 186.4 bp for the complete test set and 159.8 bp versus 155.1 bp for the test set without the outlier. (3) Comparing the sensitivity on the test set of 10 sequences, mfold has an average sensitivity of 55.9% and P-RnaPredict has an average sensitivity of 50.3%. When S.acidocaldarius is not considered, the average sensitivity of mfold remains almost unchanged at 55.6% while the sensitivity of P-RnaPredict increases to 52.1%. (4) In terms of specificity, mfold compares with 51.4% to P-RnaPredict at 49.6% on the 10 sequences. Both methods have a specificity of 51.1% if the longest sequence is not considered in the test set. (5) In terms of F-measure, P-RnaPredict outperformed mfold on 3 sequences and had a very similar performance on 2 other sequences (H.marismortui and A.lagunensis). For the remaining sequences, mfold outperforms P-RnaPredict. (6) For 6 out of 10 sequences P-RnaPredict had fewer false positives than mfold and tied in 1 case. The average amount of FPs is 112.9 bp for P-RnaPredict and 110.2 bp for mfold. If the outlier is not considered, the number of FPs drops to 93.4 bp for P-RnaPredict and 97.4 bp for mfold. Overall, P-RnaPredict produces more structures with fewer false positives.

### 4 DISCUSSION AND CONCLUSION

We have presented results obtained from a fully parallel version of P-RnaPredict, a parallel evolutionary algorithm for RNA secondary structure prediction which was implemented on a 128 node Beowulf cluster. We have demonstrated that noteworthy speedups were achieved allowing for a reduced time to predict structures, particularly for longer sequences.
In terms of overall performance, larger deme sizes had a significant impact on the results. In the majority of cases, runs with a deme size of 100 and deme count of 10 either produced the best results overall or tied for best result. This suggests that P-RnaPredict can employ a larger overall population size while still retaining the benefits of parallel speedup. Also, the OX2 crossover operator was found to perform best in the majority of cases.

We have compared the quality of the predicted structures of P-RnaPredict with 10 known structures and also to the accuracy of the respective predicted structures from mfold. One major advantage of mfold is that it employs a theoretically tractable DP algorithm which can find the minimum \(DG\) structure within its thermodynamic model in polynomial time. It is not possible to find a structure with a lower \(DG\). However, owing to limitations in current thermodynamic models the minimum \(DG\) structure is usually not the native fold and may differ significantly from it. This problem can be addressed by considering suboptimal structures produced by either P-RnaPredict or mfold.

Comparisons performed with mfold determined that P-RnaPredict can predict structures with a higher true positive base pair count and a lower false positive base pair count than the minimum \(DG\) structure computed by mfold for the majority of sequences tested. However, when suboptimal structures from mfold are considered, the ability of mfold to predict more true positive base pairs was higher. This came at the expense of more false positive base pairs on a per sequence basis and is considered to be a tradeoff between the two approaches.

Overall, the prediction accuracy of both methods is good for shorter sequences. Prediction accuracy decreases as sequence lengths increase, very likely owing to the limitations of the thermodynamic models.

An interesting point of observation is that clearly it seems current thermodynamic models are not sufficient in predicting structures that are very close to the native fold except for rather short sequences. This may have two reasons: first, they are not complete and not noise free. Problems exist with modeling global interactions and the data modeled come from wet lab experiments which are inherently noisy. Second, there is speculation that the native fold is not necessarily in a global minimum free energy state.

It is interesting to note that mfold employs a much more sophisticated thermodynamic model than P-RnaPredict, giving it a slight advantage. It can be speculated that with a more advanced thermodynamic model the prediction accuracy of P-RnaPredict will increase further.

Another interesting observation is that non-canonical base pairs such as CU or GA are not captured with the current thermodynamic models employed in P-RnaPredict and mfold. However, such non-canonical base pairs are found in native RNA structures.

In future work, improving the performance of P-RnaPredict will involve four important subproblems. These include modifying the helix generation algorithm to produce partially complete helices, employing more sophisticated thermodynamic models in the fitness function, modeling non-canonical base pairs and explicitly modeling pseudoknots to further improve the prediction accuracy of P-RnaPredict.

### ACKNOWLEDGEMENTS

The first author would like to acknowledge the support of the Natural Sciences and Engineering Research Council (NSERC) for this research under research grant number RG-PIN 238298 and equipment grant number EQPEQ 240868. The second author would like to acknowledge support from Simon Fraser University, the Advanced Systems Institute of British Columbia and an NSERC postgraduate scholarship (PGS-A). Both authors would also like to acknowledge the support of the InfoNet Media Centre funded by the Canada Foundation for Innovation (CFI) under grant number CFI-3648.

The authors are grateful for the invaluable comments from the
anonymous reviewers, which have helped in strengthening this manuscript. Funding to pay the Open Access publication charges for this article was provided by NSERC RG-PIN 238298.

Conflict of Interest: none declared.

REFERENCES


