Detecting common secondary structure elements in RNA sequences

by

Sohrab P. Shah

B.Sc. (Hons) Biology, Queens University at Kingston, 1996

B.Sc. Computer Science, University of British Columbia, 2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

The University of British Columbia

April 2005

Abstract

As evidence for the important and diverse roles of RNA molecules in our cellular machinery continues to grow, there is an increasing interest in developing computational methods to analyse RNA sequences. Sets of evolutionarily related RNA sequences contain signals at both the sequence and secondary structure levels that can be exploited to detect motifs common to all or a portion of those sequences. Motifs conserved in evolution are believed to be functionally important and therefore detection of such motifs could yield novel functional RNA sequences.

We developed an algorithm called DISCO to detect conserved motifs in a set of unaligned RNA sequences. Our algorithm uses a powerful probabilistic formalism called covariance models (CM) to model motifs. We introduce a novel approach to initialise a CM using pairwise and multiple sequence alignment. The CM is then iteratively refined using expectation maximisation. Our initialisation method can operate on sequence signals alone using only a portion of the input sequences to initialise a CM to recover the remaining motif instances.

We tested our algorithm on 26 data sets derived from Rfam seed alignments of microRNA (miRNA) precursors and conserved elements in the untranslated regions of mRNAs (UTR elements). By three measures of specificity and positive predictive value, our algorithm performed well on the miRNA data sets and showed a bi-modal distribution for the UTR element data sets where the motif was completely missed, or very accurately predicted. In a comparison test with a competing algorithm, DISCO outperformed RNAProfile in measures of sensitivity and positive predictive value, although the running time of RNAProfile was considerably faster. The accuracy of our algorithm was unaffected by average percent pairwise sequence identity, overall length or number of sequences in the input data, indicating that DISCO could be run with similar accuracy on diverse data sets. The running time of DISCO is $O(W^3 + L^2W^2 + L^3)$ where W is the width of the motif and L is the length of the longest sequence in the input data. This is an improvement on SLASH, the only other RNA motif finding algorithm in the literature that uses CMs.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acknowledgements

I would like to thank my advisor Anne Condon for her guidance, patience and dedication to teaching. Thanks also to Wyeth Wasserman for invaluable comments on this work and to Francis Ouellette for providing me with professional experiences that gave me a thorough immersion into the field of bioinformatics. Thanks to my colleagues in the Beta-lab at UBC Computer Science for stimulating discussions about their work, my work and bioinformatics in general. Most of all, thank you to my family: Nikiah, Zubin and Zahra for their support and patience on rainy Sundays and to my parents for always being there when I needed them.

SOHRAB P. SHAH

The University of British Columbia
April 2005

To my son Zubin and my daughter Zahra.

Chapter 1

Introduction

One of the most surprising facts revealed by the Human Genome Project is that our complement of DNA contains much fewer protein coding genes than originally thought [10, 69]. The most recent estimates put the number of protein coding genes at between 20,000 and 25,000 [11]. In comparing other eukaryotic organisms' genomes such as the worm (13,000), or the fly (18,000), it is clear that the number of genes in an organism is not a good measure of biological complexity. If not the number of protein coding genes, then what can explain the complexity of our species in comparison to others?

A growing body of literature is pointing to RNA genes as one possible explanation. RNA has been largely under-studied in comparison to proteins, but work done in the last decade is revealing promising insights into the importance of RNA. One class of RNAs called non-coding RNAs (ncRNAs) are transcribed from DNA, but do not get translated into proteins - ncRNAs are the functional molecules themselves. While some ncRNAs such as transfer-RNAs and ribosomal RNAs have been well characterised in the literature for decades, it is becoming increasingly apparent that a diverse array of ncRNAs that have important roles in the normal function of cellular processes remain to be discovered. In addition to ncRNAs there are RNA elements embedded in other RNA molecules that also influence biochemical

processes in the cell. RNA elements are relatively small RNAs that are part of a larger RNA molecule. These elements are often responsible for fine-grained control of translation of mRNA transcripts[1].

Numerous examples of ncRNAs and RNA elements are included in repositories such as Rfam (http://www.sanger.ac.uk/Software/Rfam/) [20, 21]. As evidenced by the growth of Rfam (approximately 400 families of ncRNAs and counting), new RNA molecules are being discovered at a rapid rate. These new discoveries of RNAs with diverse and critical functions give us strong evidence that RNA molecules play a more significant role in cellular process than previously thought.

1.1 Challenging the central dogma of biology

In a recent review article, Mattick [50] describes RNA as the "architects of eukaryotic complexity". He suggests that the central dogma of molecular biology that governs the "gene → RNA transcript → protein" process is somewhat incomplete. Mattick paints a picture of a system of RNA molecules that are an intricate part of the biochemical processes in the cell and exert control over transcription, translation and thereby gene expression. In another review article, Eddy describes experimentally determined families of RNA genes that have a wide array of functions [14]. He describes a rapidly evolving, diverse array of RNA molecules that are involved in gene regulation, RNA processing, catalysis of sub-unit formation in critical cellular machinery. Each of these reviews postulate that we are gaining new insights about the RNA world. More discoveries of RNA molecules will further our understanding of how they contribute to biochemical pathways and cellular processes in all forms of life. Furthermore, it is now clear that a complete understanding of cellular processes must include RNA molecules as key participants.

[1] An mRNA is an intermediate biochemical form of a protein coding gene that is created through the process of transcription from DNA. The mRNA is then translated (in whole or in part) into a protein through the process of translation.

1.2 Terminology

RNA (ribonucleic acid) molecules are polymers composed of four different nucleotides - Adenosine (A), Cytosine (C), Guanine (G) and Uracil (U). An RNA sequence has a directionality due to the biochemically distinct 5' and 3' ends of the nucleotides[2]. The order of the nucleotides is called the sequence of the molecule. RNA molecules also exhibit folding patterns where nucleotides in the sequence can base-pair with other nucleotides in the sequence and form hydrogen bonds. This acts to thermodynamically stabilise the molecule in the cell. The resultant set of pairs of nucleotides defines the secondary structure of the molecule. The secondary structure is closely linked to the function of the molecule and is conserved in evolution, despite potential sequence divergence. The concept of evolutionary conservation of secondary structure is highlighted throughout this thesis and will be discussed in depth in later sections.

1.3 Examples of ncRNAs and RNA elements

Recent experimental evidence has uncovered several examples of ncRNAs and RNA elements that influence biochemical activities in the cell. These include microRNAs (miRNAs), which are small (approximately 22 nucleotide) molecules that bind to mRNAs and influence protein expression [38, 40]. In addition, RNA elements in the untranslated regions (UTRs) of mRNA transcripts of certain genes are essential for regulation of the translational machinery [9, 23, 25, 39]. There are also numerous examples of human disease genes whose causative agents are gain of function mutations related to changing RNA secondary structure of their mRNA [4, 31], and elements in human viral pathogens that are being investigated as possible drug tar-

[2]The 5th and 3rd carbon atoms in the sugars of all the non-terminal nucleotides of an RNA molecule are covalently bonded to a phosphate group. The end nucleotides of an RNA molecule either have their 5th carbon unbonded to a phosphate (5' end), or the 3rd carbon unbonded to a phosphate group (3' end).

gets [44, 67]. The remainder of this thesis, however, will focus on miRNAs and UTR elements and the following section will give detailed explanations and examples of these RNAs in particular.

1.3.1 UTR elements

Generally speaking, UTR elements are small RNA structures that occur in the 5' or 3' UTRs of mRNA transcripts. Usually, these elements are target binding sites for proteins that once bound, influence the process of translation of the mRNA into protein. We outline several examples of UTR elements below.

Iron response element

The iron response element (IRE) is an RNA element of approximately 30 nucleotides found in the 5' UTR of ferritin genes and the 3' UTR of the transferrin receptor genes. The IRE forms a hairpin secondary structure which is a binding target for two forms of the iron regulation protein (IRP) [9, 24]. In low concentrations of iron, IRP binds to the IRE which prevents the binding of the 43S ribosomal subunit complex, thereby inhibiting the translation of the ferritin protein. In high concentrations of iron, the IRP does not bind to the IRE and translation of ferritin proceeds so that iron can be sequestered by the ferritin protein [25]. In contrast, the transferrin receptor IREs bind IRP in the presence of iron. This has the effect of stabilising the mRNA of the receptor against degradation and enables translation [25]. These two essential proteins in iron metabolism, ferritin and transferrin receptor, are under translational control that is mediated in part by a UTR element.

Selenocysteine insertion sequence (SECIS)

The selenocysteine insertion sequence (SECIS) is an RNA element found in the 3' UTR of animal selenoprotein mRNAs [39]. The selenocysteine codon[3] (UGA) also

[3]A codon is a group of three successive nucleotides in an mRNA that code for a specific amino acid. There are $4^3 = 64$ possible codons, and each codon encodes one of 20 possible

codes for a stop codon. The selenocysteine insertion sequence is a hairpin structure required by the translational machinery to distinguish the selenocysteine codon from a stop codon [39].

Internal ribosomal entry sites

Another class of UTR elements are the internal ribosomal entry sites (IRES) present in some eukaryotic and viral mRNAs. These RNA structure elements permit initiation of translation under physiological circumstances such as mitosis, apoptosis, hypoxia and some viral infections [23]. Several genes are known to contain IRES whose structures are conserved, with significant sequence variability [23].

1.3.2 microRNAs

MicroRNAs are a class of ncRNAs that are known to influence the function of the post-transcriptional machinery in eukaryotes [33, 59]. An initial genome wide survey of human miRNAs estimates that miRNAs constitute 1-2% of all eukaryotic genes and that they exert regulatory influence on the production of 10% of all human protein products [33].

MiRNAs are first transcribed as miRNA precursors that form stem-loop secondary structures [37]. The mature miRNA sequence (21-25 nucleotides) is excised from the hairpin by an enzyme called Dicer [37]. The mature miRNA binds to its mRNA target by complementary base pairing [33]. This binding process induces cleavage of the mRNA or represses translation by unknown mechanisms [33].

First discovered in the nematode worm (*Caenorhabditis elegans*), the *lin-4* and *let-7* miRNAs were shown to be involved in temporal control of development events [38, 40, 43]. These miRNAs exhibited expression patterns in larval development and homologues were found in animals with bilateral symmetry [43].

---

amino acids. The translation machinery moves along the mRNA codon by codon and adds the appropriate amino acid to the growing protein.

The studies outlined in [38, 40, 43] represented the first high-thoughput discoveries of miRNAs. Subsequent papers have hinted at the extent to which miRNAs are active in eukaryotic organsims, especially higher order plants and animals. John *et al.* [33] reported more than 2000 mRNAs with miRNA target sites conserved across mammals and Sempere *et al.* [59] reported tissue-specific expression patterns for 119 previously unreported miRNAs including 19 brain-specific miRNAs implicated in human neuronal development. Pfeffer *et al.* [56] found several miRNAs expressed in Eppstein-Barr virus (EBV) that exploit silencing of translation as a mechanism to regulate host genes. This work implies that other viruses may also encode miRNAs and it further demonstrates the diverse functionality of miRNAs.

1.4 Discovering new ncRNAs and RNA elements

The examples outlined above illustrate the importance of ncRNAs and RNA elements in biological functions. It is compelling to consider the possibility that numerous ncRNAs and RNA elements with diverse functional roles remain to be discovered. Developing computational methods to detect previously unknown ncRNAs and elements will contribute to this discovery process. Accurate tools will help identify putative targets for biochemical assays.

As previously mentioned, ncRNAs and RNA elements exhibit sequence and secondary structure conservation through evolution. This implies that regions of DNA or RNA sequences that have functional roles will exhibit shared sequence and secondary structure patterns. Computational tools can exploit this phenomenon by searching sequences from different organisms in order to detect shared sequence and secondary structure patterns, and thereby detect putative functional regions in RNA sequences. The work described in this thesis investigates a new method for computational discovery of shared sequences and secondary structures in a set of RNA sequences. We test this method for its ability to detect UTR elements and miRNAs in publicly available RNA data sets.

1.5 Thesis outline

The remainder of thesis describes computational tools for processing RNA sequences (Chapter 2), formalises the computational problem to detect conserved secondary structure motifs in unaligned RNA sequences (Chapter 3), describes a new approach to address this problem (Chapter 3), describes experiments to test the performance of the algorithm (Chapter 4), presents the results of those experiments (Chapter 5) and concludes with a discussion of those results (Chapter 6) and future directions for this work.

Chapter 2

Related Work

To address the need for computational tools to analyse RNA sequences, numerous methods have been developed. The following sections discuss the different classes of computational problems related to RNA sequence and structure pattern discovery, review current methods and discuss some of their strengths and limitations. This chapter is meant to be a broad survey of the literature related to RNA sequence analysis and serves as necessary background material for the formal description of the computational problem addressed by our algorithm described in Chapter 3.

2.1 Considering sequence and secondary structure in RNA sequence analysis

DNA and RNA sequences are under different evolutionary selection pressures. When analysing a set of DNA sequences, the analysis of the sequence itself is usually sufficient to detect patterns. RNA, however is subject to evolutionary constraints at the secondary structure level. This results in the phenomenon that related RNA genes share common secondary structure, but may have little sequence similarity. This renders most DNA and protein sequence motif finding and alignment algorithms potentially unsuitable for RNA sequence analysis. There are several categories of

algorithms that take both sequence information and secondary structure information of a set of RNA sequences into account. These categories are defined by the computational problem they address. We outline five categories of computational problems and published algorithms designed to solve these problems. All categories are included as they have some relevance to the novel algorithm introduced in Chapter 3.

2.2 Techniques for RNA sequence analysis

We begin with a description of several techniques which are used for RNA sequence analysis, RNA sequence and structure modeling and aligning RNA sequences and secondary structures.

2.2.1 Dynamic programming with thermodynamic energy models

Most of the work we will describe deals with the prediction of secondary structure using multiple sequences as input. Often, it is necessary to predict the secondary structure of a single sequence. This problem has been solved by Zuker and Steigler [71] who used a dynamic programming algorithm based on a thermodynamic energy model scoring function. This approach is implemented in the mFold [71] program, the RNAFold program [26] and PairFold [1].

2.2.2 Probabilistic RNA modeling using covariance models

A major contribution to RNA structure modeling is the concept of covariance models (CM) due to Eddy and Durbin [16]. The CM framework allows for RNA sequences and structures to be modeled probabilistically. The framework uses stochastic context free grammars (SCFG) which can be constructed from a multiple alignment of related RNAs and a consensus secondary structure. Analogous to hidden markov models (HMMs) for sequence alignment, SCFGs provide a powerful formalism for

aligning sequences to a model of an RNA family. In fact, a CM is a generalisation of an HMM-profile for modeling sequence motifs or protein families [16].

Using the CYK/INSIDE scanning algorithm (see [15]), one can generate an optimal alignment (using dynamic programming) of a sequence to a SCFG that can include insertions and deletions [13]. Using this algorithm, instances of the RNA family represented by the model can be detected in a genome or a database of sequences of interest. To understand this alignment step, we need to first define SCFGs and CMs and describe how a CM is built from the input. We will describe how CMs are constructed and the INSIDE alignment algorithm in detail in the following sections.

Formal descriptions of SCFGs and CMs

A CM is a specialised SCFG designed to model a multiple RNA sequence alignment and consensus secondary structure with position-specific scores [13, 15, 16]. An SCFG is made up of a set of M non-terminal states, K different terminal symbols over an alphabet (A,C,G,U for RNA), and a set of production (or emission) rules of the form $V \rightarrow \gamma$, where V is a non-terminal state and γ is a string over the set of (non-terminal and terminal) symbols that includes the empty string ϵ. Each production rule is associated with a normalized probability (summing to 1) for any given non-terminal V. A CM is a specific type of SCFG adapted to model RNAs at the sequence and secondary structure levels. A CM has seven types of states and production rules (see Table 1 in [15]). The production probability of a given state v is the product of an emission probability e_v and a transition probability t_v. CMs are made up of seven types of states (P, L, R, B, D, S, E - see Table 1 in [15]). Each state has its own emission and transition probabilities which can be derived from the input alignment and consensus secondary structure (see below). The types of states correspond to alignment 'events'. For example, consensus base pairs are modeled with a P state, consensus single stranded residues by L and R

states, deletions relative to the consensus by D states. The branching topology of the RNA secondary structure is modeled by begin (B), start (S) and end (E) states. This topology creates an ordered tree. The tree structure makes the alignment algorithm tractable, however it creates a limitation of CMs in that they cannot model pseudoknots or base triples. Modeling pseudoknots would require a more complicated data structure that supported cycles. This is beyond the framework of SCFGs.

The next section will describe how to create and set parameters corresponding to the states and their emissions and transitions.

Steps in constructing a CM

Creating a CM from a multiple alignment and a consensus structure involves three steps. First, a guide tree (explained below) is created from the consensus structure. Next, an empty CM (with no emission probabilities or transition probabilities) is created from the guide tree. Finally, the CM transition probabilities and emission probabilities are calculated from the sequences in the alignment. We will now discuss each step in detail.

The guide tree

The guide tree is made up of eight types of nodes (node type shown in parentheses): MATP (P) which models a base pair, MATL (L) and MATR (R) which model an unpaired base, BIF (B) which models a branch in the structure, ROOT (S) which is the begin node, BEGL (S) and BEGR (S) which are the begin nodes for adjacent branches in the tree, and END (E) which is the end node for a branch. The guide tree is created in the following way from the input: first, a consensus structure is derived from the input (base pairings are assigned and columns with a high proportion of insertions are ignored). Once the consensus structure is determined, the guide tree is created by first creating a root node, then traversing the structure,

assigning unpaired bases to MATL or MATR nodes and paired bases to MATP nodes. Branching structures are specified with BIF nodes and then starting each branch with either a BEGL or BEGR node. All branches end in END nodes. By convention, MATL nodes are always used before MATR nodes in the case of unpaired bases. (See [15] for more details on this process). The result of the guide tree creation is a binary tree made up of nodes in the set {MATP, MATL, MATR, BIF, ROOT, BEGL, BEGR, END}.

Guide tree to covariance model

Once the guide tree is created, it is 'expanded' into a covariance model. Each node type in the guide tree can be in a particular state, where the state is in the set of possible states for that node type. Consider a given node to be a discrete random variable. The set of states for a node are just the possible values of the random variable (eg six sides of a dice). The possible states for each node type are listed in Table 3 of [15]. There are two types of states for each node type: *split set* states and *insert set* states. For example, MATP nodes have 6 possible states: MP, ML, MR, D *split set* states and IL, IR *insert set* states. Each state has a set of transitions to their child states. The set of transitions depends on whether the state is a *split set* state or *insert set* state. *Split set* states transition to every *insert set* state in the same node and to every *split set* state in the next node. *Insert set* states self transition and also transition to every *split set* state in the next node. For IL states, there is a special case in that there is a transition to the IR state in the same node. The B state makes an obligate transition to the S states of the child BEGL and BEGR nodes.

So in summary, a guide tree is a set of nodes connected by the consensus secondary structure. The CM is a set of states determined by the nodes in the guide tree that can connect to other states within the same node, or other states in the child node. The structure of the CM is therefore made of 1) an array of states

in the set {MP,ML,MR,D,IL,IR,B,S,E} whose ordering is constrained by the guide tree from which it was derived and 2) a set of directed 'edges' or transitions between states. The edges can connect a state to itself, or can connect a state to a state in an adjacent downstream node in the guide tree.

With the 'empty' CM now in place, the transition probabilities and emission probabilities can now be calculated from the sequences in the input multiple alignment. This is the parameterisation step, analagous to determining the state transition probabilities in an HMM. This step is also known as the training step in the broader machine learning literature.

Parameterizing a CM

Each sequence in the multiple alignment can be represented unambiguously by a unique parse tree over the topology of the 'empty' CM. Once each parse tree has been determined, the transition and emission probabilities can be computed. The transition probabilities are the observed counts of transitions from state $v \rightarrow \gamma$ where v is the parent state and γ is the child state. These counts are normalised over the child states and a Dirichlet prior is used to smooth the probabilities - this results in no 0 probabilities. The emission probabilities are similiarly computed by counting the observed unpaired nucleotides for MATR and MATL states and the observed pairs for MATP states. Again the counts are normalised and a Dirichlet prior is used. A paramaterization therefore results in a M-by-M transition matrix where M is the number of types of states in the model and $t_v(\gamma)$ are the entries in the matrix, a 1-by-4 matrix for the unpaired emissions and a 1-by-16 matrix for the paired emissions. Note that in the special case of B states, the transition probabilities are 1 to the S states of the branches.

From CMs to sequence alignments

As stated previously, the most practical use of a CM is to align a sequence to it. Similar to sequence alignment problems, we wish to compute the optimal alignment of the sequence to the RNA profile represented by the CM. It is possible to calculate the log-probability of the most likely CM parse tree using the CYK/INSIDE algorithm (see [15]). One can also calculate the probability of all possible parse trees using a 'sum' instead of 'max'. This results in the likelihood of the data given the model, or $P(x|\theta)$ where x is the sequence and θ are the parameters of the CM. We also would like the unambiguous alignment of each nucleotide in the sequence to each nucleotide in the structure that is modeled by the CM (allowing for gaps). This can be derived from the optimal parse tree which is computed using the INSIDE$^\tau$ algorithm described in [15]. This algorithm includes a traceback procedure to recover the optimal parse tree from the matrices computed in INSIDE.

The INSIDE algorithm

The INSIDE algorithm is analogous to the Forwards algorithm for HMMs and is described in [15]. It computes a 3-dimensional matrix $\alpha_v(i,j)$ where v is a state in the CM and $i..j$ is a subsequence for which the parse tree is being calculated. Each entry in the matrix is the maximum likelihood of the sub-parse-tree rooted at state v that generates the subsequence $i..j$. This algorithm works from the inside of a structure and proceeds outwards. In other words, it starts at the empty string and null trees of end states and proceeds outward, generating longer subsequences and larger parse-trees. It begins at the highest numbered state and proceeds to the lowest numbered state. For each state, the resultant matrix is upper-triangular containing in each cell the maximum likelihood over the possible child states of v of transitioning from the current state to the next state given i and j and the current state v. Because it works backwards through the states, the matrices for the child states of v are already filled in. As an aside, the states in the CM are enumerated

such that the indices of child states are always greater than their parent states. At the end of the run, the likelihood of the optimal parse tree is in the cell $\alpha_v(1, L)$ where L is the length of the sequence. The score of the alignment is a log-odds ratio, which is the difference of the log-likelihood of the alignment and the log likelihood that the sequence was generated at random with independently and identically distributed nucleotide emission probabilities [16]. **The log-odds score will prove to be important when we introduce our approach in Chapter 3.**

Using EM to iteratively refine a CM

Eddy and Durbin [16] introduce an EM method for learning the most likely CM from a set of unaligned RNA sequences. Consider the case where a multiple alignment is given. This approach uses covarying positions in the multiple alignment to determine the RNA consensus structure. Their algorithm uses the Nussinov-Jacobsen/Zuker folding algorithm [71], but instead of maximizing the stacking energies, they maximize the mutual information content, which is calculated on compensatory mutations in the sequence alignment that preserve the secondary structure. Once the consensus structure backbone is in place, it is converted into a CM and then the parameters are estimated in the usual way. Alignments are re-estimated by then aligning the sequences to the model. Then the consensus structure and the new CM are re-estimated given the new alignment. This procedure iterates until the model parameters do not change significantly between iterations. A significant problem is how to achieve a multiple alignment to start with. **Creating an initial multiple alignment for a CM and EM as a method for iterative refinement are discussed extensively when we introduce our algorithm in Chapter 3.**

2.2.3 Simultaneous alignment and consensus structure prediction of unaligned RNA sequences

Aligning two sequences allows us to infer the positions in the alignment that are conserved and the positions that are not. We can assume that conservation gives us information at a nucleotide level as to which positions are evolutionarily constrained. Furthermore, once an alignment is attained, a common secondary structure can be inferred by examining positions in the alignment where compensatory mutations have occurred to preserve a secondary structure. Algorithms have tried to exploit these characteristics of RNA sequences to simultaneously align and predict the secondary structure of two or more sequences. Such algorithms operate under the assumption that sequence conservation and compensatory mutations which preserve secondary structure imply evolutionary constraint and furthermore imply biologically functional importance. These algorithms attempt to represent a set of RNA sequences with a single consensus secondary structure and produce a multiple alignment of the sequences in the process. Examples of algorithms that compute a simultaneous alignment and consensus structure prediction of unaligned RNA sequences are given in Section 2.7.1 and Section 2.7.2.

2.3 Consensus structure prediction from aligned RNA sequences

The algorithms described in this section address the prediction of a consensus structure for a set of aligned RNA sequences. They make use of sets of evolutionarily related sequences in order to infer a conserved secondary structure. In particular, these methods capitalise on the occurrence of compensatory mutations in the columns of the alignment. These compensatory mutations indicate the preservation of secondary structure even in the context of a mutated sequence. These algorithms avoid the high computational cost of simultaneously aligning and predicting sec-

ondary structure as in [18] and are therefore applicable to longer RNAs such as 16S or 23S ribosomal RNAs [28] provided a quality multiple alignment is available.

2.3.1 Alifold

The Alifold dynamic programming algorithm [28] incorporates both thermodynamic stability and compensatory mutations to generate a consensus secondary structure of a multiple RNA sequence alignment. The classical energy model for folding RNA sequences used in [71] is modified to include a covariance measure of pairwise columns in the multiple alignment. A simplified view of this model is that a compensatory mutation that preserves a base pair is comparable to the energy gained by extending a helix by one base pair.

The algorithm has an $O(L^3)$ running time where L is the length of the alignment. In contrast to other methods such as [35], the folding algorithm is only run once, thereby reducing computational effort. **The Alifold algorithm plays an important role in our approach as described in Chapter 3.**

2.3.2 Pfold

Pfold [35, 36] estimates the maximum a priori secondary structure from an alignment of RNA sequences assumed to have identical secondary structure. The key idea is that Pfold uses phylogenetic information combined with SCFGs. The algorithm produces the most likely structure based on an evolutionary model that takes into account both nucleotide substitution rates and base-pair substitution rates. These substitution rates are based on large sets of published alignments of tRNAs and large sub-unit ribosomal RNAs. The grammar used by Pfold was shown to be the most effective grammar in a recent evaluation [12].

The key idea of PFold is that the algorithm derives a phylogenetic tree from the alignment using a maximum likelihood method over the possible trees calculated from the substitution matrices. The tree is then used together with the alignment

and the grammar to infer the most likely structure. The authors demonstrate that using phylogenetic information confers a significant performance advantage in predicting the true structure. In recent work [36] the authors present an optimisation of the tree-estimation step which significantly improves performance.

The major advantage of this method is the incorporation of an evolutionary model into a Bayesian approach to inferring secondary structure. This allows the output to have a posterior probability associated with it - a quantitative probabilistic measure that is not available with most other approaches. Moreover, by using the SCFG formalism, one can easily compute the likelihood of each position in the structure, allowing users to assess which positions in the structure are more likely than others [35]. The major drawbacks are that PFold cannot handle pseudoknots (due to the use of SCFGs (see Section 2.2.2)), its iterative use of the relatively expensive $O(L^3)$ INSIDE algorithm for phylogenetic tree and secondary structure estimation and the need for a good input alignment to achieve satisfactory results. We will discuss the implications of iterations of the INSIDE algorithm in the context of optimising our own work in Chapter 6. Also, we will explore the idea of using phylogenetic information and evolutionary models further in Chapter 6.

2.3.3 Limitations of consensus structure prediction

The major limitation of both Alifold and Pfold is the requirement of a user-supplied alignment as input. Pfold has the added assumption that each sequence in the alignment share the identical secondary structure. This introduces a cyclical problem where a good quality multiple alignment can be achieved if a consensus structure is known, but consensus structure determination requires a good quality multiple alignment. We will address this problem in our work and present a solution based on pairwise sequence alignment with secondary structure filters.

2.4 ab-initio RNA gene detection algorithms

Ab-initio prediction of RNA genes has proved to be a significantly more difficult challenge than predicting genes that encode proteins. This difficulty is due to a lack of obvious statistical signals emitted by RNA genes [57] in genomic sequence. Whereas protein-coding sequences emit signals based on codon bias and specific signals called splice site consensus sequences in eukaryotic genomes [8], ncRNAs lack such discriminating features. However, two notable efforts have shown progress in ab-initio RNA gene detection.

2.4.1 QRNA

The QRNA algorithm [58] takes two related DNA sequences as input and classifies the individual nucleotides in the sequences as protein coding, ncRNA or 'other'. This method exploits patterns of compensatory mutations observed in ncRNAs and patterns of synonymous codon substitutions in protein coding sequence. The authors implement a pair-SCFG model for ncRNAs and a pair-HMM for protein-coding and 'other'. The algorithm takes in a pair-wise alignment of two sequences and calculate a log-odds score for the ncRNA model compared to the two HMMs. This method was used to classify regions in the *E. coli* genome not known to code for proteins. Four other bacterial species were used as comparative sequence in the input. The authors speculate that this method could be used as a screening tool to detect ncRNAs in genomic sequence. The major drawbacks of this method are the $O(L^3)$ running time for the pair-SCFG scanning algorithm which makes the algorithm unusable for long sequences, and the limitation to two aligned sequences as input.

2.4.2 RNAz

Washietl *et al.* [68] used comparative sequence analysis and thermodynamic stability to classify multiple sequence alignments as ncRNA sequence or non-ncRNA sequence. The algorithm takes as input a multiple sequence alignment of two or

more sequences. Using Alifold, the algorithm first calculates the average minimum free energy (MFE) of the consensus secondary structure of the input multiple alignment. The MFE measure is normalized using standard regression techniques into a z score. A structure conservation index (SCI) is calculated that measures the degree of conservation of secondary structure of the sequences in the input multiple alignment. Finally using both the z score and the SCI, the alignment is classified with a support vector machine as ncRNA or not. The authors report better results and substantially improved running times when compared to QRNA [58]. The method is a significant step forward in ab-initio RNA gene detection. Notably, previous work has shown that MFE of RNA sequences does not emit strong enough statistical signals to be used in a RNA detection algorithm [57]. This work shows that thermodynamic stability can indeed be useful for prediction of ncRNAs. We will discuss this notion further in Chapter 6.

2.5 Searching for RNAs from predefined models

2.5.1 RNAMotif

RNAMotif [47] uses deterministic descriptors to model RNA secondary structure motifs. The descriptors describe rules that instances of the motif must follow in order to be retrieved in a search of a sequence database. RNAMotif, therefore provides a RNA secondary structure definition language. The language allows RNA structures to be strictly or loosely defined depending on the amount of specific knowledge (sequence or base pairing constraints) the user has about the RNA structure they are trying to model. The RNAMotif description language can describe complex RNA structures at both the secondary and tertiary levels. The major drawback of this type of modeling is that if the constraints are too specific, the searching algorithm is bound to miss instances of the motif that are subtle variants. Furthermore, if the constraints are too general, many false positives will be returned. This latter issue

is addressed by Fogel *et al.* [17] who use evolutionary computation to filter large 'hit lists' returned from the RNAMotif search algorithm.

2.5.2 Infernal and Rfam

Infernal [15] is follow up work to probabilistic modeling with CMs [13, 16]. This work describes a memory-efficient improvement to the original CM INSIDE alignment algorithm that reduces the memory requirement to $O(L^2 log L)$ from $O(L^3)$ where L is the length of the sequence. This improvement enabled the development of the Rfam database [20] (http://www.sanger.ac.uk/Software/Rfam/), by making alignments of CMs to longer sequences tractable with respect to memory requirements. Rfam now has sequence and secondary structure multiple alignments and CMs of nearly 400 RNA families whose activity has been experimentally verified and published in the literature. This enables searching any new genomic sequence for homologues of the RNA families included in Rfam. The major limitation is that the alignment algorithm remains $O(L^3)$ in time complexity and so searching large genomes remains a computational challenge. This is addressed to an extent by using heuristics to filter the sequence being searched using sequence alignment first [20] or using more robust techniques that filter out low-scoring sequences in $O(L^2)$ time and only run INSIDE on the remaining sequences [70].

Another important use of Rfam is in testing motif detection algorithms and RNA searching tools. The database contains two types of structural multiple alignments, seed and full, that represent families of ncRNAs and UTR elements. The seed alignments are hand curated multiple alignments with accompanying annotated consensus structure information. The seed sequences are all derived from GenBank/EMBL sequences so obtaining the sequence that flanks the seed sequence is trivial. The seed sequences themselves are experimentally validated sequences published in the literature. The quality of alignments and ease of use make this data attractive to use as test data for new algorithms. Indeed, several studies have

recently used Rfam alignments as test data [66, 68, 70]. This indicates that Rfam data is gaining acceptance as a quality source for RNA sequence and structure alignments. We also chose to use Rfam data as a source of high quality alignments to evaluate our own algorithm (see Section 4.2 for more details).

2.5.3 Rsearch

The Rsearch algorithm [34] is built on top the Infernal [15] system. The authors present a local alignment search tool that allows a user to search a database of sequences for homologues of a single RNA sequence with known secondary structure. The key contribution of this work is the development of empirically derived substitution matrices (called RIBOSUM matrices) that are specific to RNA sequences and secondary structures. These matrices were developed under the BLOSUM model of evolutionary divergence and contain substitution rates for unpaired nucleotides and base-paired nucleotides. The matrices are used to parameterise a 'single-sequence' CM. Instead of deriving the emission probabilities from a multiple sequence alignment as described in Section 2.2.2, the emission probabilities are set from a RIBOSUM matrix. Transition probabilities are derived using a standard affine gap formulation (see [34] for further details). By using RIBOSUM matrices, a CM can be constructed from a single sequence with known secondary structure. Another key contribution of this work is a local alignment variant of the CM alignment algorithm presented in [13, 16]. The authors report higher sensitivity and specificity than standard local sequence alignment search tools that do not consider secondary structure.

2.6 Specific RNA gene detectors

The work described thus far addresses general approaches for RNA sequence analysis. Specific RNA families emit characteristics that can be exploited for better predictions. For example, the snoscan algorithm [46] searches for 2'-O-ribose methy-

lation guide snoRNA genes in a sequence database based on specific sequence and secondary structure patterns derived from experimentally determined snoRNA sequences. Similarly, trnascan-SE algorithm [45] scans for tRNA genes. These algorithms are by design highly specific to particular gene families. Considering the diversity of ncRNAs believed to exist in nature [14, 50], it is impractical to conceive of designing an algorithm for each type of ncRNA. Tools such as CMs offer generality and hence are advantageous in this regard when building algorithms designed to discover new ncRNAs or RNA elements. However, it would be an improvement if specific attributes of an RNA family could be incorporated into a general model. We explore this idea in our algorithm by fine-tuning parameters to detect specific types of RNA sequences.

2.7 RNA motif discovery in unaligned sequences

Often only parts of related RNA sequences have functional elements or motifs that are conserved. Until now, we have only considered prediction of secondary structure and alignments where a given set of sequences contain a shared structure that spans the entire length of the sequences. As previously discussed in Chapter 1, there are numerous examples of RNA elements that represent only a portion of each sequence in a set of related sequences. The UTR elements such as IRE and SECIS are two such examples. In addition, several algorithms presented in previous sections of this chapter have required a multiple sequence alignment as input. In this section we present algorithms that detect conserved motifs in a set of unaligned sequences that are expected to have only a portion of the sequence contain the motif. These algorithms receive the most treatment in this chapter as they address the same computational problem as our approach.

2.7.1 FOLDALIGN

One approach to this problem is outlined in Gorodkin *et al.* [18]. The authors describe an algorithm, FOLDALIGN, that uses a greedy strategy to create a multiple alignment of RNA sequences that have a common structure in a portion of each sequence. A 4-D dynamic programming alignment algorithm is used to perform pairwise alignments. The scoring system takes into account both sequence and structure similarity of gapped pairs of residues in the two sequences. The alignment algorithm can be thought of as a mixture of the Smith-Waterman [62] and Nussinov-Jacobsen algorithms [52]. A scoring matrix S is developed that contains substitution scores for aligning a_{ij} to b_{kl} where a and b are input sequences with the nucleotide at position i in a, paired to the nucleotide at position j in a, and the nucleotide at position k in b, paired to the nucleotide at position l in b. S is a 25 by 25 matrix that includes values for substituting all combinations of paired substitutions of ACGU-, where '-' represents a gap that has been inserted in a sequence for the purpose of alignment. The 4-D dynamic programming matrix $D_{ij,kl}$ contains the best score of aligning $a_i...a_j$ with $b_k...b_l$.

The greedy strategy is used to build multiple alignments using the results of all the possible pair-wise alignments in the input sequences. First, all sequences are compared to each other, then all pairwise alignments are compared to all the sequences such that no one sequence is included more than once in each comparison. Next the 'triplet' alignments can be compared to each sequence such that each sequence is only included once in the result. This may continue until all sequences have been compared. This has complexity $O(N^N L^4)$ where N is the number of sequences and L is the length of the longest sequence. To optimize the greedy algorithm, the search space is heavily pruned by eliminating redundant and low-scoring alignments. Only a fixed number of highest scoring alignments are kept at any one iteration of the process. This reduces the complexity to $O(N^4 L^4)$.

This algorithm introduces an optimal pairwise local alignment procedure

and scoring matrix that takes into account both sequence and secondary structure. While the approach is mathematically rigorous, the time complexity is prohibitive for longer sequences.

2.7.2 SLASH

Gorodkin *et al.* [19] combine CMs with FOLDALIGN. This is done to improve on the time complexity of FOLDALIGN. The algorithm uses FOLDALIGN to generate a 'seed' alignment and consensus secondary structure to initialise a CM. The key point is that this step only uses a subset of the input sequences. From this seed alignment and consensus secondary structure, a CM is constructed (see Section 2.2.2) and the remaining sequences are aligned using the CM INSIDE alignment algorithm described in Section 2.2.2. The notion of a CM being composed of only a portion of the input sequences and then used to 'recover' the motifs in the remaining sequences is important as we incorporate a similar technique in our algorithm. Despite the improvement, this algorithm still has complexity $O(N^4L^4)$ where N is the number of sequences used for the 'seed' alignment and L is the length of the longest of those sequences. This remains usable only on small sets of short sequences. We introduce a method of 'seeding' a CM that has a much lower time complexity in Chapter 3.

2.7.3 RNAProfile

The RNAProfile algorithm [53] outputs the most conserved regions of a set of unaligned RNA sequences according to a similarity measure that accounts for both sequence and secondary structure. The algorithm proceeds by first selecting a set of candidate subsequences from the input sequences that will be further analysed in a later step. This initial step exhaustively searches all possible subsequences (referred to as regions) in the input for candidate regions that contain a given number of stems in the secondary structure (derived by dynamic programming with a thermodynamic energy model). The authors estimate there to be $O(L)$ candidate

regions where L is the length of the sequence [53]. Once initial candidate regions are selected, they are then evaluated for their similarity to each other. The algorithm proceeds by aligning each candidate region in sequence $S1$ to every other candidate region in sequence $S2$ using a Needleman-Wunsch variant that takes into account both sequence and secondary structure in the scoring function. We will explore a similar method in our approach outlined in Chapter 3. The resulting alignments are converted to position specific profiles that represent the frequency of each nucleotide (and gap) at each position in the alignment. These profiles are ordered based on the alignment scores and only a fixed number of the best-scoring profiles are kept. After this step, the candidate regions from sequence $S3$ are aligned to each profile and a new set of high scoring profiles is stored. This procedure continues until the candidate regions from all of the sequences have been aligned. The output is the highest scoring profiles after all sequences have been processed.

The RNAProfile method has several distinguishing features. The major advantage is the lack of required prior information. The only required input parameter is the number of stems expected to occur in the motif. In addition, the use of profiles enables the algorithm to more readily detect instances of the motif that may have diverged considerably. The output has a quantitative measure of the 'fitness' of the region predicted to be an instance of the motif. The algorithm relies on predicting the secondary structure through dynamic programming and a thermodynamic energy model. While this is expected to work well for motifs < 100 nucleotides in length, the accuracy of RNAProfile is tied to the accuracy of the folding routines. This is also an issue in our approach and we will explore this further in Chapters 3 and 6. Another drawback is that RNAProfile does not report aligned instances of the motifs it predicts.

2.7.4 ComRNA

Ji *et al* [32] introduce a method for detecting conserved RNA structure motifs using graph theoretical methods. Their method has three major steps: 1) find all possible stable stems in a sequence, 2) find all potential conserved stems shared by subsets of sequences and 3) assemble compatible sets of conserved structures to construct consensus secondary structure profiles.

In step 1), the stable stems are identified through a branch and bound procedure combined with stacking energy parameters to evaluate biochemical stability.

In step 2), comparing stems across sequences is done by aligning each pair of sequences globally using the Needleman-Wunsch [51] algorithm to identify highly conserved regions. Conserved regions are defined as having at least 80% sequence identity over 10 or more nucleotides. The conserved regions are used as anchor regions for the stem comparisons. Similarities between stems are measured using five features: helix length, helix sequence, loop sequence, stem stability, and relative positions of the stem start and end coordinates in the whole sequence. These features are used to compute a weighted sum divided by the sum of the relative stability of the two stems in their respective sequences. The compatible set of stems are those that are found in a minimum k out of the N sequences.

In step 3), the population of stems in the entire input data are partitioned into N-partite graph with stems as nodes. Edges are only allowed between nodes in different partitions and are weighted according to how similar they are from step 2). N is the number os sequences in the input and stems that originate from the same sequence are placed in the same partition. The problem is to find maximal cliques (cliques that are not fully contained in larger cliques) of at least size k in the N-partite graph. The maximal cliques will contain stems that are shared by at least k sequences. Maximal clique finding is an NP-hard problem, so an approximation that uses a depth-first enumeration approach is used. The output of the algorithm is the stems found in the maximal cliques containing at least k stems.

This approach has several advantages: it can predict pseudoknotted structures, it allows for prediction of motifs not shared by all sequences and it reports a given number of best scoring motifs. A strong attribute of this paper is that the authors did a thorough comparison of their approach with other published methods and present a quantitative measure for evaluation of performance. We used a similar measure to assess the results of our experiments (see Chapter 4). The major disadvantage of their approach is that the maximal-clique finding step has an exponential run-time, although in practice, the authors report acceptable run-times for data sets that are of comparable size to our test data sets (see Chapter 4).

2.7.5 GPRM

The genetic programming for RNA motifs (GPRM) algorithm [30] uses a different technique to address the RNA motif finding problem. The GPRM algorithm focuses on finding base-paired segments and non base-paired segments that compose the secondary structure of the motif. The user specifies the maximum number of segments and the length range(s) of the segments. Under the genetic programming terminology, an individual in a population is a motif. GPRM randomly generates an initial population of motifs that follow the user's specifications. The fitness of the individuals in the population are measured as a function of the number of sequences in the inputted training set expected to contain the motif (sensitivity) and the number of sequences predicted to contain the motif that actually do contain the motif (positive predictive value). The algorithm randomly selects two motifs and the motif with the higher fitness gets selected for the genetic operation step. This step takes a motif and transforms it using specific rules (see [30] for more details). The new individual is added to the population and the process iterates until there are no more fit individuals in the population than the most fit individual that has already been processed, or a maximum number of iterations is reached. The running time of the algorithm is approximately $O(L^3N)$ where N is the total number

of sequences and L is the length of the longest sequence.

2.7.6 CARNAC

The CARNAC algorithm [66] predicts conserved secondary structure elements in a family of related ncRNAs. The algorithm combines energy minimization, phylogenetic comparison and sequence conservation in a three step approach. First, all predicted stems (by dynamic programming with a thermodynamic energy model) below a free-energy threshold are selected. Next all pairs of sequences are folded using a pairwise folding algorithm outlined in [55] to give an optimal consensus secondary structure for the pair. This step is done by choosing pairs of stems from the two sequences that are in locally conserved regions, are identical in structure and have at least one compensatory mutation. For N sequences, this results in $N-1$ structures for each sequence. Next, the predicted stems from the pairwise folds are filtered using graph-theoretic techniques. Using a graph structure with predicted stems from the first step as nodes, edges are drawn between two stems if they are from two different sequences and if the stem also appears in the pairwise fold of those two sequences. In addition all identical stems (not included in the previous step) are included at this time. Connected components in the graph are then scored based on the number of nodes, the number of stems in each sequence, the total number of edges and the number of edges between identical stems. The highest scoring connected components have 1 stem per sequence and are fully connected. Finally, the secondary structure for each sequence is determined by incorporating stems from the connected components greedily by first choosing stems from the highest scoring connected components. Overlapping stems are not permitted. The authors report good results for RNase P, ciliate telomerase RNA and enterovirus UTRs although the assessment of the results is largely qualitative. An advantage of this method is its lower computational complexity ($O(N^3)$) compared to FOLDALIGN ($O(N^4L^4)$) where N is the number of sequences and L is the length of the longest sequence.

The limitations are the lack of quantitative or probabilistic output and the lack of aligned conserved motifs. In other words, one could not build a CM for searching other sequence databases from the output of CARNAC.

2.7.7 Alidot

The Alidot algorithm [27] begins by performing a multiple sequence alignment using CLUSTAL-W. The secondary structures are predicted independently of the sequence alignment using the free-energy model implemented in the Vienna package [26]. The sequence alignment is used to introduce gaps into the structures, and they are in turn aligned. These aligned structures are then plotted using 'mountain plots' which are representations of secondary structures. The mountain plots of the aligned structures are used to produce a 'consensus mountain'. All predicted base pairs in the consensus mountain are then removed and sorted according to the following criteria: i) no nucleotide pairs more than once (no base triples), ii) no base pairs cross i.e. there may not exist two base pairs (i, j) and (k, l) such that $i < k < j < l$. Base pairs are ranked according to their 'credibility'. Credibility of a base pair (i, j) is determined using a number of criteria. First, the residues for all sequences at positions i and j are retrieved and for each sequence, it is determined whether a 'legal' base pair is formed (GC, CG, AU, UA, GU, UG). Second, the more sequences that contain a legal base pair, the more credible the base. The presence of consistent (a standard base-pair is preserved) and compensatory (strength of base-pair conserved) mutations also lend credibility to the base pair. Third, symmetric base pairs are more credible than non-symmetric base pairs. A base pair, (i, j) is symmetric if j is the most frequent pairing partner of i in the sequence set and vice-versa. Lastly, pseudo-entropy is calculated for all the base pairs. Low entropy yields more credibility. The sorted list of base pairs is then scanned from the top - removing all base pairs that conflict with a higher ranking one. Base pairs below a frequency threshold are also removed.

The main advantages to this method is efficient computation: the algorithm only computes structures for conserved regions and therefore it can be used on long sequences (more than 10Kb) provided they have few conserved regions. The major drawbacks are the alignment step which only considers sequence information. In the absence of sequence conservation, the multiple alignment will be a weak starting point. Also the output (mountain plots) does not provide a probabilistic quantity with which to evaluate the results.

2.8 Summary

We have outlined the major areas of computational RNA sequence and secondary structure analysis related to the work we present in Chapter 3. The following chapter summarises this related work and introduces and describes a new approach to detecting RNA motifs in unaligned RNA sequences.

Chapter 3

The Disco Algorithm

3.1 Assessment of related work

We have outlined the major areas of computational work related to processing RNA sequences at the sequence and secondary structure level. In Chapter 2, we introduced algorithms to predict conserved motifs, predict consensus secondary structures, detect new ncRNAs and align RNA sequences. These algorithms introduce all of the major concepts that are needed to understand the algorithm we present in this chapter.

The challenge of discovering RNA motifs in unaligned sequences has been met with a diversity of approaches (see Section 2.7), but there is no one superior method for tackling this problem. As mentioned in Section 2.2.2, CMs offer a powerful probabilistic formalism to model a set of related RNA sequences. They are similar to HMMs in their construction. HMMs have been shown to be sensitive protein sequence alignment tools [5, 63, 64]. CMs are particularly robust if given an adequate multiple alignment and accurate consensus secondary structure [20, 21, 34]. Therefore in order to use CMs in motif discovery of unaligned RNA sequences, we must first try to construct a good multiple alignment and consensus secondary structure that represents the motif. As previously mentioned, the SLASH algorithm spends $O(N^4L^4)$ time doing this step. This time complexity makes SLASH only

feasible for small sets of short sequences. With the exception of SLASH, no other methods described in Section 2.7 make use of CMs to detect motifs.

Another technique introduced in Chapter 2 is expectation maximisation (EM) for iterative refinement of a CM (see Section 2.2.2). None of the tools outlined in Section 2.7 use iterative refinement techniques. In motif finding in DNA and protein sequences, EM forms the basis of a large body of work [2, 3, 7, 41, 42] and we explore its use in the RNA motif finding problem.

We formulate the problem as follows: consider a set of N unaligned RNA sequences S where a portion of the sequences M is expected to contain a motif of width approximately W. The optimal CM C that represents the motif consists of the optimal multiple alignment of the instances of the motif in S and the corresponding consensus secondary structure. We measure the quality of a CM with a score C_{score} which is the sum of the bit scores of the best alignment using the INSIDE algorithm of the CM to each sequence in S. The precise problem then is to discover the CM C with the maximum C_{score} given S and W.

To solve this problem, we designed an algorithm that improves on the complexity of SLASH for 'initialising' a CM and includes an iterative refinement phase using EM that improves this initial CM. The algorithm has a $O(W^3 \cdot L + L^2 \cdot W^2 + L^3)$ run-time where W is the expected width of the motif and L is length of the longest sequence in the input data (see Section 3.2.3 for further details). Since EM is only guaranteed to converge on local optima, we do not expect to always detect C, however we hope that given a good starting point, EM will converge on a model that is composed of a majority of the instances of the motif. The remaining sections of this chapter outline key concepts and ideas that we introduce, and describe our algorithm, called DISCO in detail.

Before describing our algorithm, we need to consider some key concepts that we used to motivate our method.

3.1.1 Global vs local sequence alignments

Several algorithms presented in Chapter 2 input and process global alignments. Global alignments consider the entire length of each sequence when computing alignments. This is not applicable for finding motifs, since they are expected to only cover a portion of each sequence in a set of RNA sequences. However, it may yet be advantageous to use tools that process global alignments but in a local context. Consider a multiple alignment made of a portion of each sequence. The alignment is global in the sense that it is a multiple sequence alignment, but it is local in the context of the input data. This 'local multiple alignment' can take advantage of tools that process global alignments. This idea is used in [19]. Recall that a first step is run FOLDALIGN on a portion of the input sequences to detect conserved elements, then use CMs to detect these elements in the remainder of the sequences. We employ a related use of CMs in our approach. We also use Alifold, which also operates on a global alignment, by providing it with a local multiple alignment for which we want to predict the secondary structure.

3.1.2 Using pairwise and multiple sequence alignment to initialise a CM

Our research tests several methods for determining a good alignment and secondary structure of a motif to be used as input to building a CM. In particular, we explore the use of $O(W^2)$ alignment techniques based on the Needleman-Wunsch algorithm [51] (see Section 3.2.1) to produce a multiple alignment suitable for CM initialisation, where W is the width of the motif. This $O(W^2)$ algorithm overcomes prohibitive running time of SLASH and provides a method to achieve a 'coarse-grained' multiple alignment and consensus structure to initialise a CM that is later refined.

3.1.3 Expectation Maximization for CM refinement

Expectation maximization (EM) is used to improve the quality of the CM by repeatedly aligning the CM to the input data and re-estimating its parameters and secondary structure based on those alignments. This idea was originally introduced in [16] and we test its validity in this work. We expect that the alignment presented to the CM will improve with iterative refinement. Our work represents the first use to our knowledge of EM in the RNA motif finding domain.

3.2 Proposed new algorithm: DISCO

The DISCO algorithm takes as input a set of unaligned RNA sequences and finds a CM that represents a motif shared by the input sequences. The algorithm outputs a multiple sequence alignment of the instances of the motif and a consensus secondary structure. The DISCO algorithm is best described in two phases, the initialisation phase and the refinement phase. The goal of the initialisation phase is to use pairwise and multiple sequence alignment of subsequences of width W, combined with secondary structure prediction using compensatory mutations to initialise a CM that represents the motif. The refinement phase uses expectation maximisation to iteratively refine the CM using the INSIDE alignment algorithm. The algorithm is depicted in pseudocode in Algorithm 1.

3.2.1 Initialisation phase

Sliding window secondary structure prediction

We first enumerate all windows of width W in the input data and predict the secondary structure of each window using the implementation of Zuker's algorithm as published in Andronescu *et al.* [1]. This step gives us a dot-bracket representation of each W-mer in the input, where each position in the W-mer is assigned a character in the alphabet '(', '.', ')'. Matched '(' and ')' indicate base-paired positions and '.'

indicates unpaired positions.

Pairwise alignment of W-mers

The next step is to pairwise align the W-mers. Each W-mer is aligned to every other W-mer using the Needleman-Wunsch optimal alignment algorithm. This is done in one of three ways:

1. 'sequence': using sequence information only with a RIBOSUM85-60 [34] scoring matrix (see Figure 3.1)

2. 'structure': using the dot-bracket representation of the secondary structure only, with a scoring matrix (called DISCOSUB) that is similar to Pavesi *et al.* [53] (see Figure 3.2)

3. 'combination': using a combination of 1) and 2) that uses RIBOSUM85-60 for unpaired nucleotides that align, and DISCOSUB for paired nucleotides that align

The algorithm was implemented in this way in order to test the properties of the input data (sequence, structure or combination) that contained the strongest signals for CM initialisation (see Chapter 4). The entries in the DISCOSUB matrix were determined using intuition and should be considered arbitrary. This is discussed further in Chapter 6.

To avoid an $O(L^2)$ number of pairwise alignments, we introduced a filter to reduce computational effort while maintaining accuracy. For each W-mer, we calculate its 'dot-composition' (DC), meaning the proportion of unpaired nucleotides in its secondary structure. We ignore all W-mers with a DC of greater than a threshold d. The remaining W-mers are called anchors. Furthermore, we do not align any two W-mers if their DC differ by more than 20% (arbitrarily selected). The highest k scoring W-mers that align to each anchor W-mer (W_a) is stored in sorted order according to alignment score in an array H with $H[1] = W_a$.

Multiple alignment of a set of W-mers

Each set H from the previous step is converted to a multiple alignment using a progressive alignment technique (see Algorithm 3). First, the alignment of W_a to $H[2]$ is converted to a profile alignment P. Each column of P is represented by q-dimensional vector P_i containing the frequency of occurrence of the alphabet `'A','C','G','U','-'` (or `'(','.',')','-'`), at a position i in the alignment, where '-' represents a gap in the alignment and q is the number of characters in the alphabet. P is then updated by aligning $H[3]$ to P so that P now contains a profile alignment of W_a, $H[2]$ and $H[3]$. At this step, the dynamic programming matrix for the alignment calculates a score based on aligning a single sequence to a profile. The score S_{ij} for aligning position i of the W-mer w to position j of the profile P is calculated as $\sum_{P_{j\alpha}} M_{P_{j\alpha} w_i}$ where $P_{j\alpha}$ is the frequency of character α in column j of P and w_i is the character at position i of w and M is the scoring matrix (one of RIBOSUM85-60 or DISCOSUB). P is similarly updated until all W-mers in H have been aligned. At the end of this step, we have a multiple alignment of the highest k scoring pairwise W-mers to W_a. We store a fixed number l of the highest scoring multiple alignments. These are then passed to the refinement phase.

Prediction of consensus structure from multiple alignment

A consensus structure for each of the l multiple alignments that are kept in the previous step is predicted using Alifold [28] (see Chapter 2 for a description of Alifold). We now have multiple alignments and corresponding secondary structures - the necessary inputs for creating CMs. A CM for each of the l multiple alignments and secondary structures is then initialised using the `cmbuild` routine from the Infernal package [15], and the intitialised CMs are refined in the refinement phase, described next.

3.2.2 Refinement phase

Expectation Maximisation

Using the initialised CM, we apply the INSIDE alignment algorithm to align the CM to each sequence in the input with the `cmsearch` routine from Infernal. Using the gapped representation of the best scoring 'hit' for each sequence, a new multiple alignment is created. The observed insertions and deletions are all relative to the same CM, making it possible to construct the multiple alignment as follows. In the case of a deletion in the gapped representation of the 'hit', the gap is simply maintained and the sequence is added to the multiple alignment. In the case of an insertion in the gapped representation of the 'hit', a gap is inserted in every other 'hit' at that position and the sequence is added to the alignment. We score the resultant multiple alignment as the sum of the bit scores for each hit. The bit score is a log-odds score that is the difference of the likelihood of the hit aligning to the CM (calculated by the INSIDE algorithm) and the likelihood of random sequence aligning to the CM. As in the initialisation phase, a secondary structure from this new alignment is then predicted with Alifold and a new CM is built from the multiple sequence alignment and consensus secondary structure. The refined CM is realigned to the sequences to generate a new multiple alignment and a new consensus secondary structure. This process is repeated until the score of the multiple alignment no longer improves, or a maximum number of iterations is reached. The pseudocode for this step is shown in Algorithm 5.

Output

The algorithm outputs the highest scoring CM detected in the refinement phase.

3.2.3 Complexity

The worst case time complexity of DISCO is $O(W^3 \cdot L + L^2 \cdot W^2 + L^3)$ where W is the user-inputted expected width of the motif and L is the length of the longest sequence in the input data. The $W^3 \cdot L$ term is from the predictive folding step of each W-mer in the data, shown in Algorithm 2, line 3. The $L^2 \cdot W^2$ term is from the pairwise alignment of each W-mer to every other W-mer (shown in Algorithm 2, line 11). The maximum number of pairwise alignments is $((L - W) \times N)^2$, however due to the threshold d introduced above, we expect that in practice, the running time for this step will be substantially better than $O(L^2 \cdot W^2)$. Finally, the L^3 term comes from the refinement phase in which the INSIDE algorithm (the `cmsearch` method) is run repeatedly over the input data (see Algorithm 5, line 9).

3.2.4 Input and output

The algorithm takes a set of unaligned sequences in FASTA format as input. A sample output is shown in Figure 3.3. The output contains the score, multiple alignment and consensus secondary structure produced by the most likely CM found by the algorithm. The index of the parent sequence (by location in the input data) of each sub-sequence and its position in its parent sequence are also given in the output.

3.2.5 Parameters

The adjustable input parameters are presented in Table 3.1.

Required parameters

A key parameter is W, the width of the motif. Another key parameter is a - the method of sequence alignment. If users expect a strong secondary structure signal, they can choose the 'structure' method, or they can choose the 'sequence' method if they expect the motif to be highly conserved at the sequence level.

Running-time parameters

There are several running-time enhancing parameters. As the dot-composition threshold d is lowered, fewer W-mers will be considered for pairwise alignment. o is the overlap used to enumerate the W-mers in the input data. W-mers are enumerated by sliding a W-sized window across each sequence. The overlap parameter alters how many positions to overlap when sliding the window to the next position. For example, if $W = 10$ and $o = 9$, the window slides one position and all W-mers in the input are enumerated. However, if $W = 10$ and $o = 5$ the sliding window steps skips over five positions before enumerating the next W-mer. This has a profound effect on the number of pairwise alignments that are performed in the initialisation phase, reducing the number by a factor of $(W - o)^2$. In addition, k - the maximum number of W-mers used to create a multiple alignment is a key parameter that we will discuss in Chapter 4 and Chaper 6. Finally, l is the maximum number of models on which the refinement phase is run. Reducing l reduces the number of times the INSIDE algorithm is run (which is $O(L^3)$).

Matrix parameters

Other matrices can be used in place of RIBOSUM85-60 or DISCOSUB. They must be in the same format as depicted in Figure 3.1 and Figure 3.2.

3.2.6 Implementation

The algorithm is implemented in the C/C++ programming language. All functions are implemented in C, but the main executable file is implemented in C++ due to a dependency on a C++ library. All source code is available by request from the author.

Algorithm 1 Pseudocode of DISCO algorithm. The procedure $DISCO(I, d, k, l, a, T)$ returns a CM C representing a conserved motif in the unaligned sequences I. d is the dot-composition threshold, k is the maximum number of W-mers to be included in a multiple alignment in the initialisation phase (see Algorithm 2), l is the maximum number of high-scoring multiple alignments to pass to the refinement phase, a is the method of sequence alignment used in the initialisation phase and T is the maximum number of iterations to use in the refinement phase. *ExpectationMaximisation* is shown in Algorithm 5.

```
1: procedure DISCO(I,d,k,l,a,T)
2:     Cset ← Initialisation(I, d, k, l, a, T)
3:     maxS_c ← −∞
4:     for all C ∈ Cset do
5:         (C′) ← ExpectationMaximisation(C, I)
6:         if score(C′) > maxS_c then
7:             maxS_c ← score(C′)
8:             maxC ← C′
9:         end if
10:    end for
11:    Return maxC
12: end procedure
```

Algorithm 2 Pseudocode of Initialisation procedure. Parameters are as described in Algorithm 1. Note that *DotComposition*() refers to the proportion of unpaired nucleotides of the sequence, *size*() refers to the number of entries in the set, *sort*() sorts the entries in descending order by score and *last* refers to the index of the lowest scoring entry in the set. *cmbuild* is described in [15] and Alifold is described in [28]. *Align* is the Needleman-Wunsch pairwise alignment algorithm described in [13]. *MultipleAlign* is shown in Algorithm 3.

```
1: procedure INITIALISATION(I,d,k,l,a,T)
2:     for all W-mer w ∈ I do
3:         Fold(w)
4:     end for
5:     maxS_p ← −∞ , Cset ← {}
6:     for all W-mer w ∈ I do
7:         maxS ← −∞ , H_w[1] ← w
8:         for all W-mer x ∈ I do
9:             if DotComposition(w) < d then
10:                if DotComposition(w) − DotComposition(x) < 20 then
11:                    A ← Align(w, x)              ▷ Pairwise alignment of w and x
12:                    if score(A) > maxS then
13:                        maxS ← score(A)
14:                        if size(H_w) < k then
15:                            H_w ← growArray(H_w, x)          ▷ appends x to H_w
16:                        else
17:                            H_w[last] ← x
18:                        end if
19:                        sort(H_w)
20:                    end if
21:                end if
22:            end if
23:        end for
24:        (P_w) ← MultipleAlignment(H_w)
25:        if score(P_w) > maxS_p then
26:            SS ← Alifold(P_w)   ▷ Predict the 2ndary struct from the alignment
27:            C ← cmbuild(P_w, SS)                          ▷ Build a new CM
28:            maxS_p ← score(P_w)
29:            if size(Cset) < l then
30:                Cset ← {Cset, C}
31:            else
32:                Cset[last] ← C
33:            end if
34:            sort(Cset)
35:        end if
36:    end for
37:    Return Cset
38: end procedure
```

Algorithm 3 Pseudocode for creating a multiple alignment from a set of sequences ordered with the first sequence as an anchor sequence and the rest of the sequences sorted in descending order according to how well they pairwise align to the anchor sequence. The $ProfileAlign$ procedure is implemented exactly as described in [13]. It returns the score and profile alignment of P and P' where P_i is a column vector containing the frequency of each character in the 'alphabet' of the sequences (eg 'ACGU-')

```
1: procedure MULTIPLEALIGNMENT(H)
2:     P ← Sequence2Profile(H[1])
3:     P' ← Sequence2Profile(H[2])
4:     P ← ProfileAlign(P, P')
5:     for i ← 3, i < size(H) do
6:         P' ← Sequence2Profile(H[i])
7:         P ← ProfileAlign(P, P')
8:     end for
9:     Return P
10: end procedure
```

Algorithm 4 Pseudocode for converting a sequence into a profile

procedure SEQUENCE2PROFILE(S)
  for all positions $i \in S$ **do**
    for $j \leftarrow 1, j <= size(alphabet(S))$ **do**
      $P_{j,i} \leftarrow 0$
    end for
    $index \leftarrow index(S_i)$
    $P_{index,i} \leftarrow 1$
  end for
  Return P
end procedure

Algorithm 5 Pseudocode for $ExpectationMaximisation(C, I, T)$ where C is a CM, I is the input set of unaligned RNA sequences and T is the maximum number of iterations. *hits2multipleSequenceAlignment* simply creates a multiple sequence alignment from the best scoring hits from each sequence. This is possible since all alignments are to the same CM and hence have all insertions and deletions relative to the same model.

```
1: procedure EXPECTATIONMAXIMISATION(C, I, T)
2:     maxScore ← −1
3:     score_C ← 0
4:     t ← 0
5:     while score_C > maxScore or t < T do
6:         score_C ← 0
7:         hits_C ← {}
8:         for all s ∈ I do
9:             hits_s ← cmsearch(C, s)
10:            if size(hits_s) > 0 then
11:                maxHit_s ← max(score(hits_s))
12:                hits_C ← {hits_C, maxHit_s}
13:                score_C ← score_C + score(maxHit_s)
14:            end if
15:        end for
16:        if score_C > maxScore then
17:            maxScore ← score_C
18:            MSA ← hits2multipleSequenceAlignment(hits_C, C)
19:            SS ← Alifold(MSA)
20:            C ← cmbuild(MSA, SS)
21:        end if
22:        t ← t + 1
23:    end while
24:    Return C
25: end procedure
```

Parameter	Description
W	Expected width of motif
a	Sequence alignment method ('sequence', 'structure', 'combination')
d	Dot-composition threshold for pairwise alignment step
o	Overlapping nucleotides in W-mer enumeration
k	Maximum number of W-mers to include in multiple alignment step
l	Number of models on which to run refinement phase
T	Maximum number of iterations in refinement phase
m	Scoring matrix for alignment using sequence method
b	Scoring matrix for alignment using structure method

Table 3.1: Parameters of the DISCO algorithm

RIBOSUM85-60

	A	C	G	U
A	2.22			
C	-1.86	1.16		
G	-1.46	-2.48	1.03	
U	-1.39	-1.05	-1.74	1.65

Figure 3.1: The unpaired nucleotide portion of the empirically derived RIBOSUM85-60 [34] substitution matrix. This matrix is used in the 'sequence' and 'combination' alignment methods described in Section 3.2.1.

```
DOTBRACKET-2.0

     (         )         .
(    3.0
)   -1.5       3.0
.   -1.5      -1.5       1.5
```

Figure 3.2: The DISCOSUB matrix showing the substitution scores for aligning characters from the dot-bracket representation of secondary structure. Matched parentheses are given a score of 3.0, matched 'dots' or unpaired nucleotides are given a score of 1.5 and all mismatches are given a score of -1.5. This matrix is used in the 'structure' and 'combination' alignment methods described in Section 3.2.1.

```
SCORE: 559

                  ......(((.(((((......)))))))).....
   0         47 G-T-GGTCGCGTCAACAGTGTTTGATC-G-AACA-CCTGT
   1         12 GAT-TCTTGCTTCAACAGTGTTTGAACGG-AATT-TCTTT
   2          5 G-T-TCTTGTTTCAACAGTGATTGAACGG-AACT-CCTCT
   3          9 G-TTACCTGCTTCAACAGTGCTTGAACGGCAAC--CTTCT
   4         27 G-T-TCTTGCTTCAACAGTGATTGAACGG-AACT-CCTCT
   5         23 G-T-TCTTGCTTCAACAGTGTTTGAACGG-AAC--CCTCT
   6        160 G-T-TCTTGCTTCAACAGTATTTGAACGG-AAC--CCTCT
   7       1305 G-T-TCCTGCGTCAACAGTGCTTGGACGG-AAC--CGGCC
   8          2 G-T-TCCTGCTTCAACAGTGCTTGGACGG-AAC--CCGGC
   9         12 G-T--CCTGCTTCAACAGTGCTTGAACGG-AAC--CCGGC
  10         28 G-TCTCTTGCTTCAACAGTGTTTGGACGG-AACA-GATCC
  11        948 G-TTTCCTGCTTCAGCAGTGCTTGGACGG-AAC--CCGGC
  12          3 G-TCTCCTGCTTCAACAGTGCTTGGACGG-AGC--CCGGT
  13          9 G-TGTCTTGCTTCAACAGTGTTTGAACGG-AACAGAC-CC
  14       1189 G-T-ACTTGCTTCAACAGTGTTTGAACGG-AACAGAC-CC
  15        398 G-TATCTTGCTTCAACAGTGTTTGGACGG-AACAGAC-CC
```

Figure 3.3: Sample output of the DISCO algorithm. The score is given on line one of the output file. The next line is empty, followed by the consensus structure of the multiple alignment. The remaining lines are the multiple alignment of the sub-sequences used to construct the CM. The aligned sequences are preceded with the index of the parent sequence of the sub-sequence and the start position of the sub-sequence in the parent sequence.

Chapter 4

Experiments

4.1 Major questions and new ideas

We set out to learn a CM that accurately models a motif within a set of unaligned RNA sequences. To explore the performance of our method we posed major questions about the inherent properties of the data and how they might be exploited to accomplish this task.

4.1.1 Question 1: Which properties more strongly represent a motif embedded in a set of unaligned RNA sequences?

All of the papers mentioned in this chapter comment that both sequence and secondary structure signals must be taken into account in RNA sequence analysis. However, there is a lack of consensus on which properties - the sequence or the secondary structure emit the stronger signals. We compared three different alignment strategies for initialising a CM: a) sequence alone, b) structure alone and c) a combination of sequence and structure. The details of these alignment algorithms are presented in Section 3.2.1.

4.1.2 Question 2: Can a CM be initialised using only a few sequences?

To minimise the cost of creating a multiple alignment to initialise a CM, we explored the idea of only using a subset of the sequences to create the multiple alignment. We wanted to test whether a relatively crude multiple alignment created from a subset of the input sequences was of high enough quality to create a CM that could then recover the remaining motifs in the iterative refinement phase.

4.1.3 Question 3: Can a crude secondary structure filter be used to filter out subsequences not expected to be an instance of the motif?

Before constructing the multiple alignment, our algorithm first folds each subsequence of length W in the input using a dynamic programming algorithm that uses a thermodynamic energy model. This step produces a dot-bracket representation (see Section 3.2.1) of the secondary structure of each subsequence representing the base-paired nucleotides and the unpaired nucleotides. We used this representation to filter out subsequences in the data that had more than a minimum proportion of their nucleotides unpaired in their secondary structure. This was done by simply counting the '.' in the dot-bracket representation of the W-mer and dividing by W (recall the parameter d from Chapter 3). Given that the subsequent step is to pairwise align all the remaining subsequences using the methods introduced in 4.1.1, we expect this filtering step to improve the run-time of the algorithm. We tested different thresholds to assess how the filter affected accuracy.

4.2 Data

We used microRNAs and UTR elements as test data sets for the DISCO algorithm. We selected miRNA families and UTR element families from the Rfam

database using the keyword searches 'microRNA' and 'UTR' on the Rfam website (http://www.sanger.ac.uk/Software/Rfam/). MicroRNAs and UTR elements were selected in light of their important role in post-transcriptional gene regulation (see Chapter 1). In addition, miRNAs and UTR elements were of ideal size (30-100 bp) to prototype our algorithm.

We used the Rfam seed alignments as 'ground truth' alignments for testing the DISCO algorithm. The seed alignments are curated multiple alignments of individual members of an RNA family. The consensus secondary structure is annotated on this multiple alignment. Rfam uses these seed alignments and secondary structures to construct CMs, which are then used to search large genomic databases for other members of the family. There are several advantages to using data from Rfam. These are outlined below:

- The seed alignments from Rfam are hand curated and nearly all sequences included in the seed alignments have been experimentally determined and published in the literature. All Rfam records are tagged with Pubmed identifiers which point to the original papers that describe the RNA molecules.

- Nearly all the sequences included in the seed alignments are flanked by genomic sequence or UTR sequence. This makes it possible to extract a larger 'super-sequence' that contains a member of the seed alignment within it. This is essential for testing, since our algorithm is designed to detect shared substructures in a set of sequences.

- All sequences included in the Rfam seed alignments have EMBL/GenBank accession numbers. This makes data retrieval fairly straightforward, where otherwise this can be an onerous task. We used the Atlas integrated database for data retrieval [61].

An example Rfam seed alignment in Stockholm format (see [15]) is given in Figure 4.1.

From the initial set of families retrieved with the keyword searches, we removed all families with fewer than four members, with more than twenty members, with length more than 151 and UTR element families whose members extended into coding sequence. The last criterion reflects our opinion that protein coding sequences have distinct properties that would confound their analysis. We did not impose any taxonomic filters. Tables 4.1 and 4.2 list and describe some characteristics of the nine UTR data sets and seventeen miRNA data sets used in this analysis. Using the larger 'parent' sequences given by the GenBank accession numbers in the seed alignments, we constructed the test data sets as follows: for UTR data, the entire UTR in which the seed sequence was embedded was extracted; for miRNA data, the miRNA plus 200 nucleotides upstream and downstream of the miRNA were extracted. In some cases, extracting 200 nucleotides was not possible due the proximity of the miRNA to an end of the sequence. In such cases, we extracted as much flanking sequence as possible to the end of the sequence.

4.3 Preliminary experiments

A set of eight Rfam seed alignments (RF00047, RF00104, RF00129, RF00172, RF00180, RF00237, RF00241, RF00256) was used to evaluate three different parameters:

- Alignment method: structure alone, sequence alone, combination (a=0,1,2)
- Number of sequences used to construct multiple alignment for initialisation (k=2-7)
- Dot-composition threshold (d=0.45,0.50,0.55,0.60,0.65)

These experiments were designed to reveal the best parameters for running the algorithm on microRNAs and UTR elements and were designed to address the questions outlined in Sections 4.1.1, 4.1.2 and 4.1.3.

4.4 Fixed parameter experiments

We estimated the optimal parameters from the results (see Chapter 5) of the preliminary experiments and ran the DISCO algorithm on the remaining data sets using those optimal parameters. For these experiments, we used the 'sequence' method for alignment, $k = 6$, $d = 0.40$ for miRNA data and $d = 0.55$ for UTR data. For data sets with < 6 sequences, we used $k = N$, where N is the number of sequences in the input data. A maximum of $T = 10$ iterations was used for the refinement phase. W was set to the length of the seed alignment $+2$, and overlap, o was set to $W - 1$. l was set to 15.

4.5 Evaluation methods

Figure 3.3 shows an example of the DISCO output. It shows a score, a multiple alignment, a consensus structure and the positions of the motif instances in the parent sequence. The outputted score and multiple alignment for each run of the algorithm were used to calculate the measures of accuracy explained below.

4.5.1 Score

As mentioned in Chapter 3, the output of the algorithm is a consensus structure, a multiple sequence alignment and a score that reflects the quality of the multiple alignment. The score is a sum of the likelihood of the model given each sequence. The higher the score, the better the alignment. We assessed the correlation of the score to the measures listed below to determine whether a higher score meant better performance.

4.5.2 Sensitivity and positive predictive value

The score measures the quality of the alignment, but this is an insufficient measure on its own, since the algorithm may produce a very high scoring alignment that

does not contain members of the Rfam seed alignment. This could arise if the input data contained other regions of similarity that were more easily detectable than the members of the Rfam seed alignment. To get a quantitative measure of the accuracy of the final multiple alignment, we chose to use sensitivity (SENS) and positive predictive value (PPV) using three measures of accuracy. To define SENS and PPV, we first need to describe four other terms:

- true positives (TP): the number of pairs of nucleotides aligned in the output that were also aligned in the Rfam seed alignment
- false positives (FP): the number of pairs of nucleotides aligned in the output that were not aligned in the Rfam seed alignment
- true negatives (TN): the number of pairs of nucleotides unaligned in the output that were not aligned in the Rfam seed alignment
- false negatives (FN): the number of pairs of nucleotides unaligned in the output that were aligned in the Rfam seed alignment

SENS is defined as TP/(TP+FN), or the number of true positives over the total number of aligned nucleotides in the Rfam seed alignment. PPV is defined as TP/(TP+FP), or the number of true positives over the total number of aligned nucleotides in the output. Often, when measuring accuracy in tests of this nature specificity, defined as TN/(TN+FP), is used as a complementary measure to sensitivity. However, in this case TN is difficult to conceptualize as it represents all correctly predicted unpaired nucleotides, which makes little sense in this scenario.

Using PPV and SENS, we can now approximate the Matthews correlation coefficient [49] (CC) in the following way.

$$CC \approx \sqrt{SENS \cdot PPV} \tag{4.1}$$

This measure was originally used in [19] and in several subsequent papers [30, 32, 22]. Since we are interested in modeling and recovering motifs, our measures are slightly

different and focus on the aligned nucleotides and recovering specific nucleotides that are part of the motif. This is another advantage of using the Rfam seed alignments in that we have a nucleotide-level 'ground-truth' to compare our results against.

Overlapping nucleotides

In addition to aligned pairs of nucleotides, we can also measure the number of nucleotides from each sequence included in the output that are also in the Rfam seed alignment, or overlapping nucleotides. In this case TP is the number of nucleotides in the output that were also in the in Rfam seed alignment, FP is the number of nucleotides in the output that were not in the Rfam seed alignment, and FN is the number of nucleotides not in the output that were in the Rfam seed alignments. SENS, PPV and CC can then be calculated in the same way as explained in Section 4.5.2.

Sequence-level overlap

Finally, we can define success at a coarse level which gives a measure of whether the motif was found in each sequence or not. We define a TP in this scenario if 50% of the overlapping nucleotides in each sequence are TP nucleotides. FP and FN can be similarly defined.

4.5.3 Reported accuracy measures

In Chapter 5 we report accuracy based on aligned nucleotides, overlapping nucleotides and sequence-level overlap measures for each of the test data sets. Each accuracy measure consists of three separate values: sensitivity, positive predictive value and correlation coefficient. From here on, these will be referred to as: AS, $APPV$ and ACC for aligned nucleotides, NS, $NPPV$ and NCC for overlapping nucleotides and SS, $SPPV$ and SCC for sequence-level overlap. We first show the results for the preliminary experiments, and follow this with the results of the

fixed-parameter experiments. We also tested the effect of average pairwise sequence identity of the seed alignments and overall size of the input data on the results.

4.6 Comparison with RNAProfile

We compared our algorithm with RNAProfile. NS, $NPPV$, SS and $SPPV$ accuracy measures and running time were compared for the following data sets: RF00037, RF00130, RF00164, RF00180, RF00185, RF00239, RF00241 and RF00256. AS and $APPV$ measures were not compared because RNAProfile does not output aligned sequences. We chose four data sets from the UTR element group and four data sets from the miRNA group. We also chose data that included sets where DISCO failed to detect the motif (RF00130, RF00180, RF00185) and sets where DISCO performed well (RF00037, RF00164, RF00239, RF00241 and RF00256) in the fixed parameter experiments (see Tables 5.1 and 5.2). Ideally, all data sets used in the fixed parameter settings would have been used in the comparison with RNAProfile, but this was infeasible due to time constraints. Similarly, we would have liked to include Alidot and FOLDALIGN in this comparison. This work is being done to prepare a manuscript for publication.

The parameters used to run DISCO were as specified in Section 4.4 and RNAProfile parameters are given in Table 4.3. For RNAProfile the length-based parameters l_R and L_R were set similarly to the DISCO W parameter. L_R was set to the length of the Rfam seed alignment $+2$ and l_R was set to length of seed alignment -20. The '20' was chosen based on the default differential between l_R and L_R. The length-specific parameters for RNAProfile were set to confer prior knowledge of the width of the motif to RNAProfile in order to achieve a fair comparison with DISCO. Default values were used for all other RNAProfile parameters.

Only subsequences reported by RNAProfile with positive fitness values were included as predicted motif instances. This decision was based on the notion reported in Pavesi *et al.* [53] that subsequences with negative fitness values should be

Rfam id	Description	Num	Length	%id
RF00031	Selenocysteine insertion sequence	19	88	40.60
RF00032	Histone 3' UTR stem-loop	13	26	73.03
RF00037	Iron response element	16	30	84.36
RF00109	Vimentin 3' UTR protein-binding region	12	94	81.65
RF00164	Coronavirus 3' stem-loop II-like motif (s2m)	16	43	84.52
RF00176	Tombusvirus 3' UTR region IV	17	92	93.52
RF00180	Renin stability regulatory element (REN-SRE)	13	37	89.61
RF00185	Flavivirus 3' UTR pseudoknot	14	102	91.82
RF00214	Retrovirus direct repeat 1 (dr1)	19	95	89.47

Table 4.1: Description and characteristics of the UTR test data sets showing the number of members in the seed alignment (Num), the length of the seed alignment (Length) and the mean percent pairwise nucleotide identity of the members in the seed alignment (%id)

Rfam id	Description	Num	Length	%id
RF00027	let-7 microRNA precursor	12	90	70.88
RF00051	mir-17 microRNA precursor family	4	82	72.77
RF00052	lin-4 microRNA precursor	9	74	70.26
RF00053	mir-7 microRNA precursor	6	93	67.13
RF00075	mir-166 microRNA precursor	11	151	59.78
RF00076	mir-181 microRNA precursor	4	76	80.22
RF00103	mir-1 microRNA precursor family	7	80	70.01
RF00130	mir-192/215 microRNA precursor	4	110	70.39
RF00131	mir-30 microRNA precursor	4	72	82.55
RF00239	mir-124 microRNA precursor family	6	87	73.31
RF00241	mir-8/mir-141/mir-200 microRNA precursor	9	81	64.60
RF00246	mir-135 microRNA precursor family	5	91	71.56
RF00247	mir-160 microRNA precursor family	7	137	66.48
RF00248	mir-148/mir-152 microRNA precursor family	5	88	73.24
RF00251	mir-219 microRNA precursor family	7	76	83.87
RF00256	mir-196 microRNA precursor family	14	96	74.06
RF00364	mir-BART2 microRNA precursor family	8	62	92.85

Table 4.2: Description and characteristics of the miRNA test data sets showing the number of members in the seed alignment (Num), the length of the seed alignment (Length) and the mean percent pairwise nucleotide identity of the members in the seed alignment (%id)

```
AE003516.3/109215-109275    GUCUUUGGUUAUCUAGCUGUA.UGAGUGA.UAAAUA..ACGU.CAUAAAG
AC005316.1/63325-63386      CUCUUUGGUUAUCUAGCUGUA.UGAGUGC.CACAGA.GCCGU.CAUAAAG
AF155142.1/45725-45784      AUCUUUGGUUAUCUAGCUGUA.UGAGUGU.AUUGG...UCUU.CAUAAAG
AC116051.4/129891-129952    AUCUUUGGUUAUCUAGCUGUA.UGAGUGG.UGUGGA.GUCUU.CAUAAAG
Z81467.1/13319-13384        AUCUUUGGUGAUUCAGCUUCAAUGAUUGGCUACAGGUUUCUUUCAUAAAG
#=GC SS_cons                ...<<<<<<<<<<<<<<..<.<<<.............>.>>>..>>

AE003516.3/109215-109275    CUAGCUUACCGAAGUU
AC005316.1/63325-63386      CUAGAUAACCGAAAGU
AF155142.1/45725-45784      CUAGAUAACCGAAAGU
AC116051.4/129891-129952    CUAGAUAACCGAAAGU
Z81467.1/13319-13384        CUAGGUUACCAAAGCU
#=GC SS_cons                >>>>>>>>>>>>>...
//
```

Figure 4.1: Seed alignment of RF00237 in Stockholm format. This format consists of a multiple alignment with secondary structure annotation given in the bottom line. Also note the GenBank accession numbers and coordinates that are provided to facilitate straightforward retrieval of flanking sequence for the test data sets.

suspected to come from sequences not containing an instance of the motif.

To assess running time, we used the Unix `time` command to report the number of CPU seconds used. All running time analysis was performed on the identical machine with no other processes, except system processes running concurrently. Analysis was performed on an Intel Xeon processor at 2.4GHz with 1Gb of RAM.

Rfam id	l_R	L_R
RF00037	20	40
RF00130	90	112
RF00164	20	45
RF00180	20	40
RF00185	82	104
RF00239	67	89
RF00241	61	83
RF00256	76	98

Table 4.3: Parameters for RNAProfile in comparison experiment. The only parameters that were set were l_R and L_R. Default values were used for all other parameters.

Chapter 5

Results

5.1 Preliminary experiments

We ran the DISCO algorithm on eight sets of data to identify parameters giving the best results. We varied the method of alignment a (sequence, structure, combination), the dot-composition threshold d (0.45, 0.55, 0.60, 0.65) and the number of W-mers used to construct the multiple alignment k (2-7). In all, there were 788 runs in the preliminary experiments. The next few sections show the results of these experiments and provide the justification for the parameters that were chosen for the fixed parameter experiments.

5.1.1 Sequence method of alignment is superior

Figures 5.1 and 5.2 show the distributions of NS and $NPPV$ (see Section 4.5.3 for an explanation of NS and $NPPV$) of the cumulative results of all the runs for the sequence, combination and structure alignment methods. All the distributions in this chapter are shown as box-and-whisker plots[1]. The NS results (see Figure 5.1) show that the sequence method was significantly better than structure

[1]Box-and-whisker plots show distributions as a box with a line in the box indicating the median of the distribution, the top and bottom edges of the box indicating the third and first quartiles and the ends of the whiskers indicating the maximum and minimum values of the distribution. The points shown on the plots are considered outliers.

(Welch Two Sample t-test, t=12.84 and p=2.2E-16) and combination (Welch Two Sample t-test, t=12.44 and p=2.2E-16). The $NPPV$ results (see Figure 5.2) similarly show sequence to be significantly better than structure (Welch Two Sample t-test, t=12.84 and p=2.2E-16) and combination (Welch Two Sample t-test, t=12.44 and p=2.2E-16). Based on these results we chose a =sequence for the fixed parameter experiments.

5.1.2 $k = 6$ gives best results for NS and NPPV

Figures 5.3 and 5.4 show the distributions for the results of the a =sequence runs of the preliminary experiments for $k = 2$ to $k = 7$ (from this point on all analyses include only a =sequence runs due to poor performance of the structure and combination methods). By qualitative assessment of Figures 5.3 and 5.4, the results for $k = 6$ seem slightly better than for the other values of k. For NS, the median values increased with k, however the mean of $k = 6$ (0.55 ± 0.30) was higher than $k = 7$ (0.49 ± 0.36). For $NPPV$, the mean and median for $k = 6$ were 0.71 ± 0.35 and 0.84. $k = 5$ and $k = 7$ had comparable values for the mean and median (0.69 ± 0.38 and 0.84 for $k = 5$) and (0.63 ± 0.38 and 0.83 for $k = 7$). The results for $k = 6$ had the highest mean and the lowest standard deviation compared to the $k = 5$ and $k = 7$ results. There was a lack of statistically significant differences between $k = 5, 6, 7$, therefore we chose $k = 6$ based on highest mean and lowest standard deviation for the fixed parameter experiments. For input data sets with fewer than six sequences, we set $k = N$, where N was the number of sequences.

5.1.3 Dot-composition threshold

Figures 5.5 and 5.6 show the distributions for the results of the a =sequence runs of the preliminary experiments grouped by d. Although Figure 5.5 seems to indicate that $d = 0.50$ produced more accurate NS results, the distribution for $d = 0.50$ was not statistically significantly better than the distribution for $d = 0.45$ or $d = 0.55$.

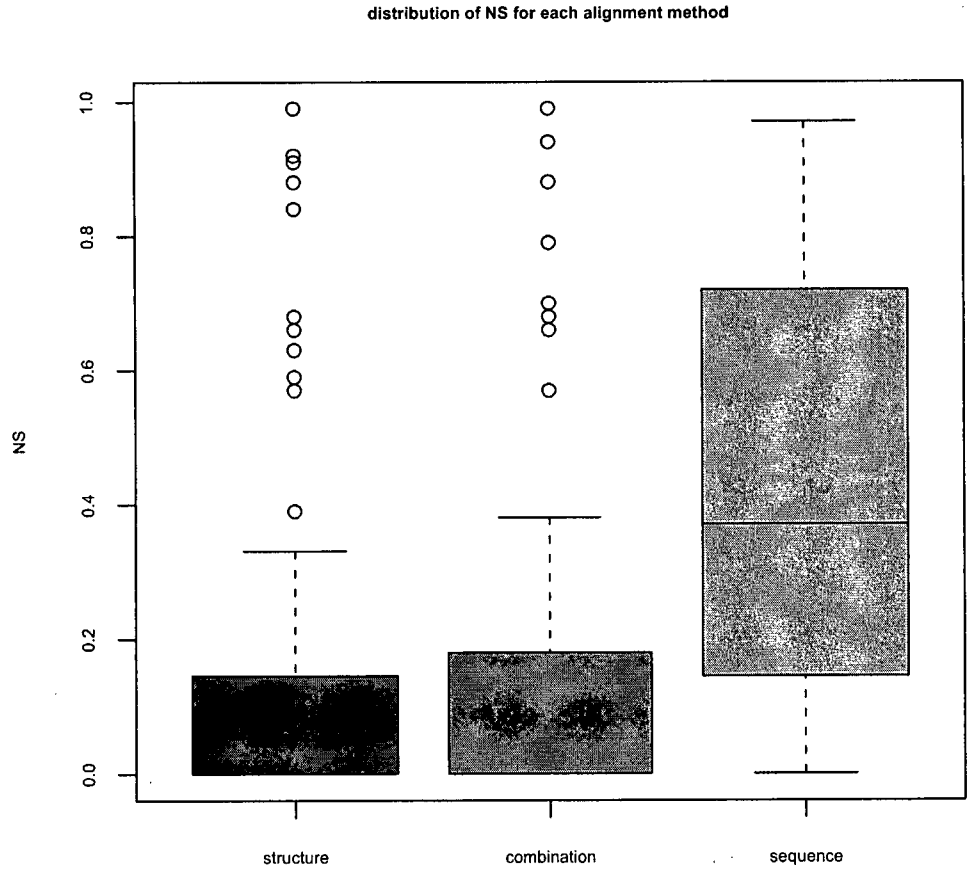

Figure 5.1: Box-and-whisker plots showing the distribution of NS over all runs for structure, combination and sequence alignment methods. The sequence method showed significantly better performance than the structure method (Welch Two Sample t-test, t=12.84 and p=2.2E-16) and the combination method (Welch Two Sample t-test, t=12.44 and p=2.2E-16).

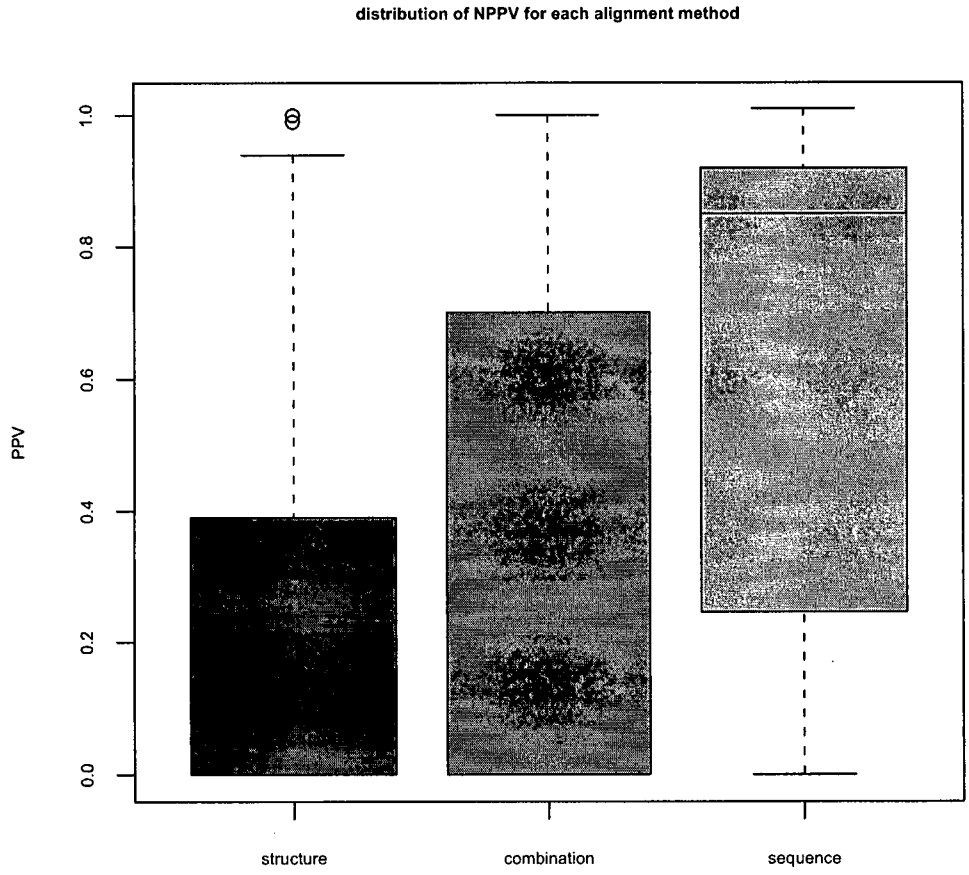

Figure 5.2: Box-and-whisker plots showing the distribution of NPPV over all runs for structure, combination and sequence alignment methods. The sequence method showed significantly better performance than the structure method (Welch Two Sample t-test, t=13.33 and p=2.2E-16) and the combination method (Welch Two Sample t-test, t=11.35 and p=2.2E-16).

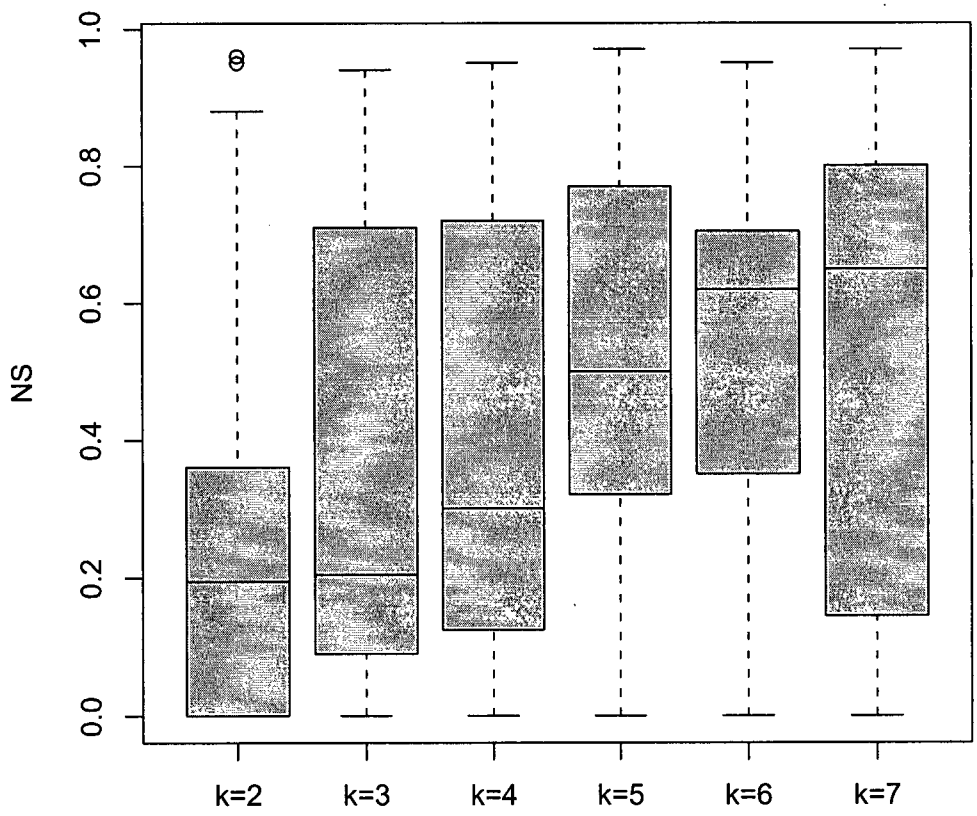

Figure 5.3: Box-and-whisker plots of all a =sequence runs showing the distribution of NS for $k = 2$ to $k = 7$. The median values increased with k, however the mean of $k = 6$ (0.55) was higher than $k = 7$. Also, the standard deviation of $k = 6$ (0.30) was lower than for $k = 7$ (0.36). These results show that $k = 6$ produced the best accuracy when measured with NS.

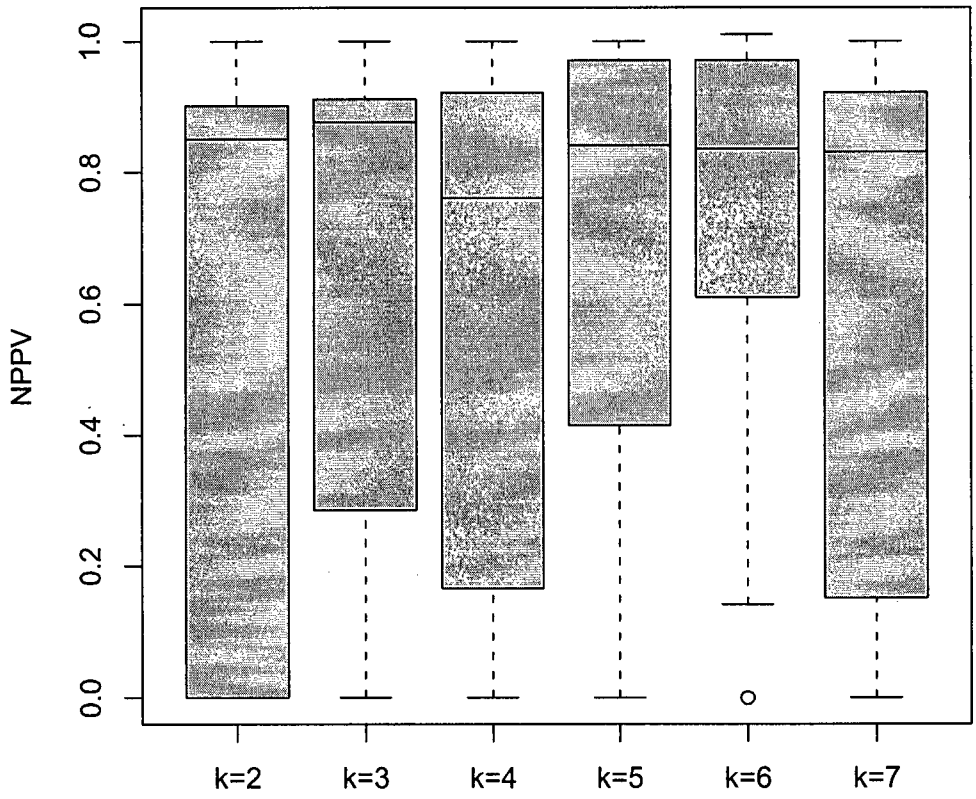

Figure 5.4: Box-and-whisker plots of all a =sequence runs showing the distribution of $NPPV$ for $k = 2$ to $k = 7$. Based on the results shown in Figure 5.3, we only compared $k = 6$ to $k = 5$ and $k = 7$. The mean, median and standard deviation for $k = 6$ were 0.71, 0.84 and 0.35. $k = 5$ and $k = 7$ had comparable values for the mean, median and standard deviation (0.69, 0.84 and 0.38 for $k = 5$) and (0.63, 0.83 and 0.38 for $k = 7$). The mean for $k = 6$ was highest and the standard deviation for $k = 6$ was lowest compared to $k = 5$ and $k = 7$. These results show that $k = 6$ produced the best accuracy when measured with $NPPV$.

The results for $NPPV$ showed no observable trend over the value of d. Given the lack of an obvious choice for d, we used biological intuition to set the value of d. For the miRNA data sets, we noted that the motifs are highly structured. We plotted the distribution of the proportion of unpaired nucleotides for the consensus structure of the 40 miRNA data sets available from Rfam (see Figure 5.7). Based on this data, we set $d = 0.40$ for the miRNA data. Ideally, we would have included $d = 0.40$ in the preliminary experiments (and therefore in Figures 5.5 and 5.6) to give a more rational basis for this choice. We initially thought that $d = 0.40$ would be too stringent, however by examining the distribution of proportion of unpaired nucleotides, we decided to choose $d = 0.40$ based on intuition.

The proportion of unpaired nucleotides was much more widely distributed for UTR elements (data not shown). We arbitrarily chose $d = 0.55$ (the middle value of our experiments) for the UTR element data. We recognise that this was not ideal, and we had hoped the preliminary experiments would provide a more rational basis for choosing d. We will discuss this further in Chapter 6.

5.2 Fixed parameter experiments

Table 5.1 shows score (SC) and accuracy measures (AS, $APPV$, ACC, NS, $NPPV$, NCC, SS, $SPPV$ and SCC) for the seventeen miRNA data sets. The mean, median and standard deviation of the distributions of these accuracy measures are also shown in Table 5.1 and the distributions themselves are shown in Figure 5.8.

Table 5.2 shows score (SC) and accuracy measures (AS, $APPV$, ACC, NS, $NPPV$, NCC, SS, $SPPV$ and SCC) for the nine UTR element data sets. The mean, standard deviation and median of the distributions of these accuracy measures are also shown in Table 5.2 and the distributions themselves are shown in Figure 5.9.

The DISCO algorithm detected the motifs for the majority of the data sets. For miRNA data, the mean and median NS were 0.66 ± 0.36 and 0.85 while the mean and median $NPPV$ were 0.76 ± 0.35 and 0.90. This indicates that on average,

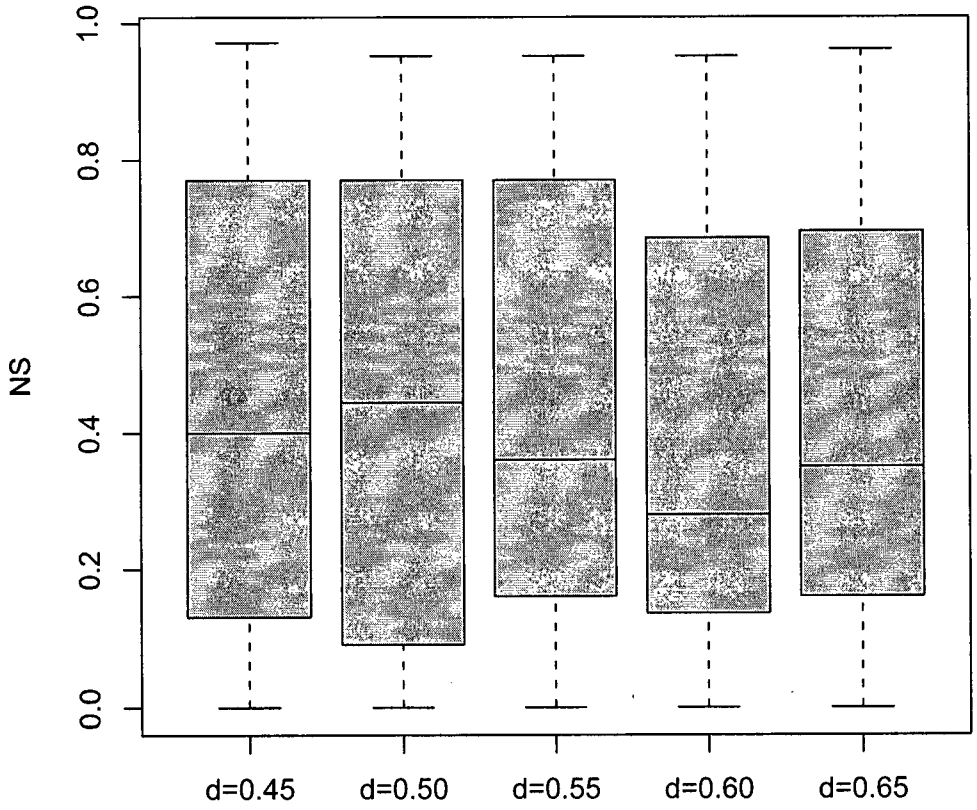

Figure 5.5: Box-and-whisker plots of all a =sequence runs showing the distribution of NS for $d = 0.45, 0.50, 0.55, 0.60, 0.65$. The mean and median values were (0.45, 0.40), (0.46, 0.45), (0.44, 0.36), (0.38, 0.28) and (0.42, 0.35) respectively. $d = 0.50$ had the highest values for NS, although the distribution $d = 0.50$ was not statistically significantly different from the distribution of $d = 0.45$ (Welch Two Sample t-test, t=-0.12, p=0.90) or from the distribution of $d = 0.55$ (Welch Two Sample t-test, t=-0.28, p=0.78).

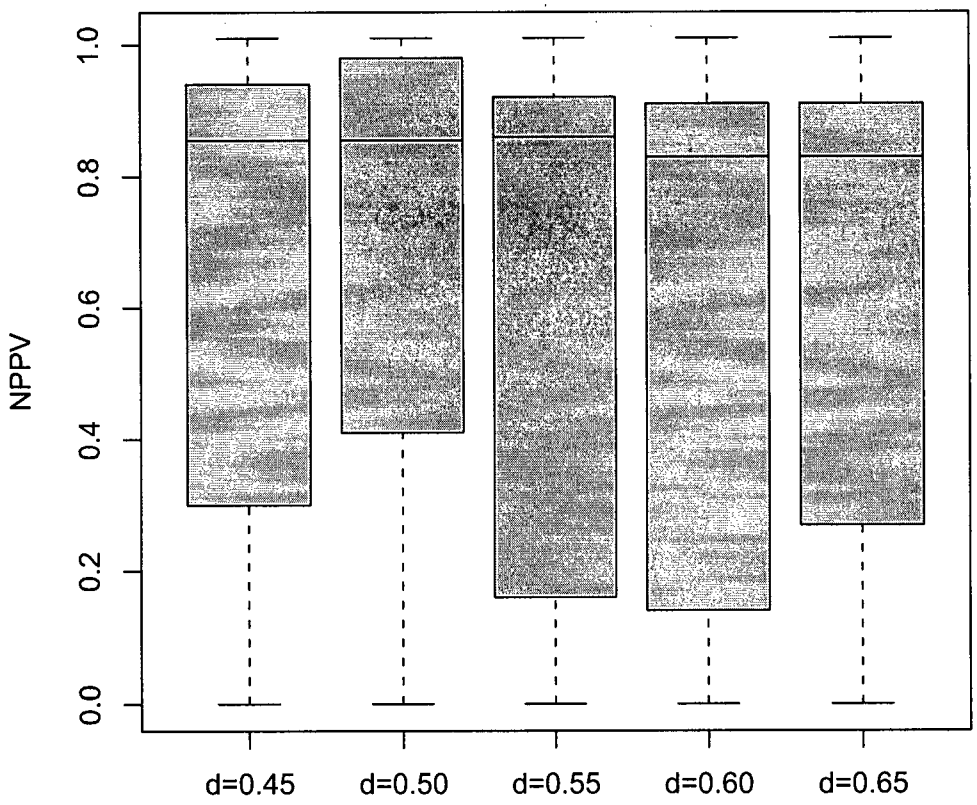

Figure 5.6: Box-and-whisker plots of all a =sequence runs showing the distribution of $NPPV$ for $d = 0.45, 0.50, 0.55, 0.60, 0.65$. The mean and median values were (0.65, 0.86), (0.67, 0.86), (0.65, 0.86), (0.61, 0.83) and (0.63, 0.83) respectively. No observable trend due to d was apparent from this data.

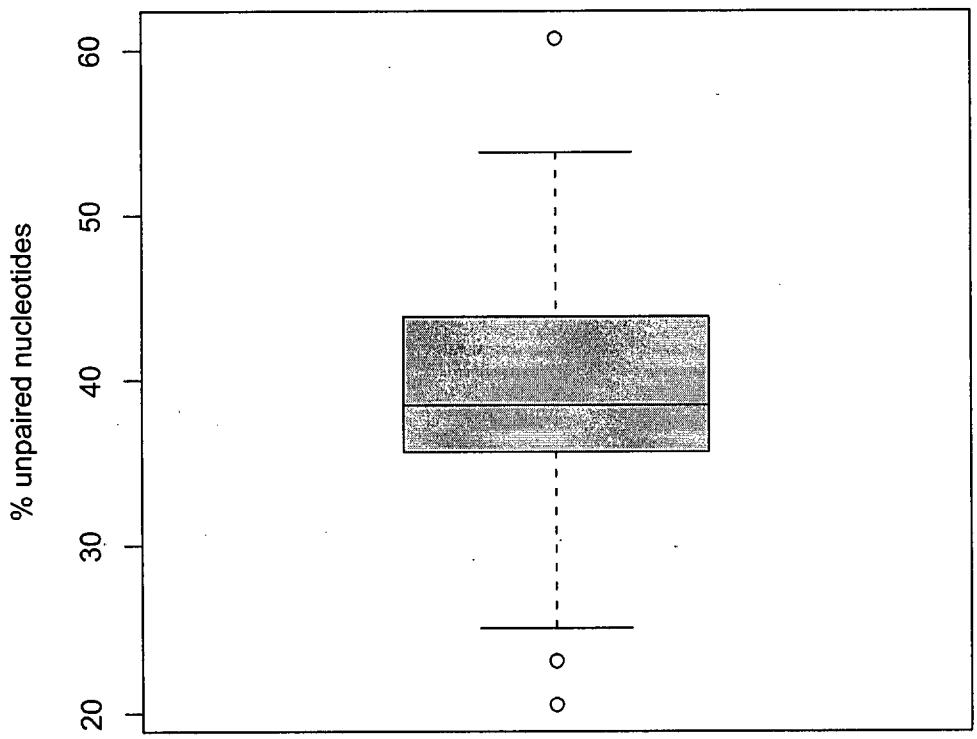

Figure 5.7: Box-and-whisker plot showing the distribution of the proportion of unpaired nucleotides in the consensus secondary structure of all 40 miRNA seed alignments from Rfam.

66% of nucleotides in the seed alignments were found in the best scoring CM and that 76% of the nucleotides found in the best CM were part of the seed alignment (see Table 5.1). DISCO recovered at least 67% of the seed sequences in twelve out of seventeen miRNA data sets (by the SS measure). The mean and median SS were 0.71 ± 0.40 and 0.86 and the mean and median $SPPV$ were 0.80 ± 0.39 and 1.00 respectively for the miRNA data. The mean and median AS were 0.46 ± 0.30 and 0.52 while the mean and median $APPV$ were 0.59 ± 0.36 and 0.79. The slightly lower performance by the AS and $APPV$ measures indicate that although the majority of motifs are being detected by DISCO, they are not necessarily accurately aligned in the output when compared to the seed alignments. This will be discussed further in Chapter 6. The large sources of error are most likely attributed to five of the data sets in which the motifs were essentially missed by the algorithm.

The results for the UTR element data sets, shown in Table 5.2 were less promising. The NS mean and median were 0.49 ± 0.48 and 0.57. The $NPPV$ mean and median were 0.45 ± 0.44 and 0.53. The performance was similarly poor by the other measures. The mean and median for both SS and $SPPV$ were 0.53 ± 0.51 and 0.83. The mean and median for AS were 0.46 ± 0.46 and 0.48. The mean and median for $APPV$ were 0.41 ± 0.49 and 0.49. This relatively poor performance and very large standard deviations are due the the fact that the algorithm completely missed the motif in four of the nine UTR element data sets. Table 5.2 shows that the data sets for which the motif was found show relatively high accuracy for all the measures. For example the NS mean of the remaining five data sets was 0.89 ± 0.18. The problems in detecting UTR elements will be discussed in Chapter 6.

5.2.1 Score is not an indicator of accuracy

Of great interest to us was whether the score of the CM (recall this was the sum of the bit scores of the best hit of each sequence aligned to the CM with the INSIDE algorithm) was a good indicator of accuracy. To test this, we first normalised the

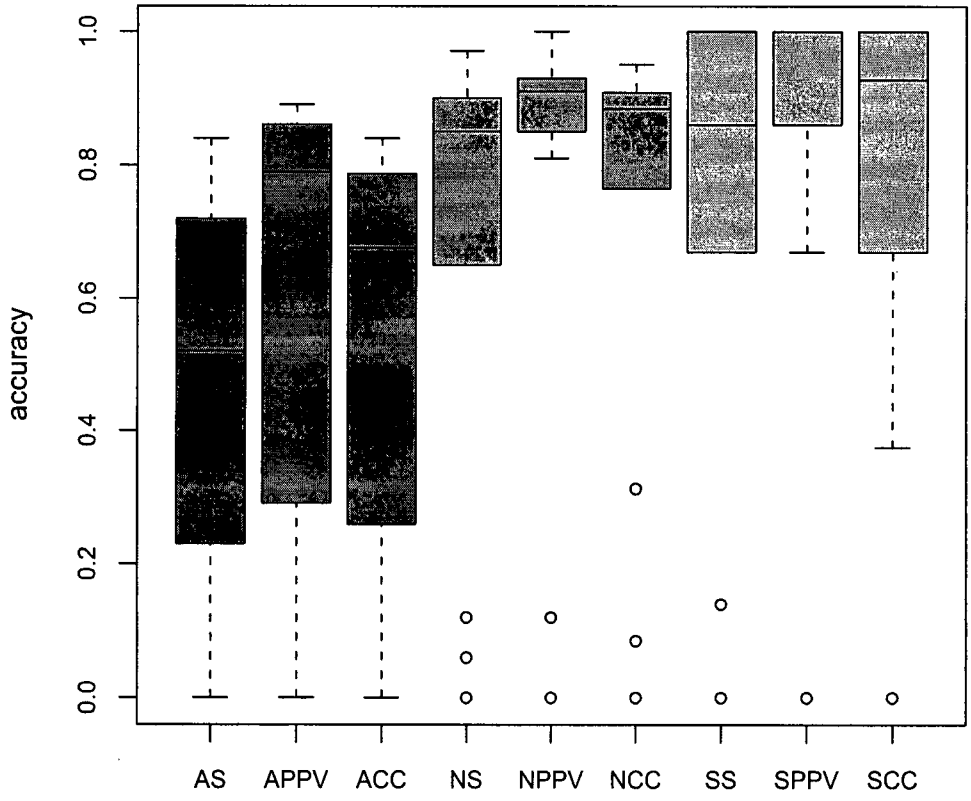

Figure 5.8: Distribution of accuracy results for seventeen miRNA test data sets by the AS, $APPV$, ACC, NS, $NPPV$, NCC, SS, $SPPV$ and SCC measures. In general, the algorithm performed well on the miRNA data by the S, $NPPV$, NCC, SS, $SPPV$ and SCC measures. The distributions for the alignment measures indicate that despite good recovery of the seed sequences, the alignments were not necessarily accurate.

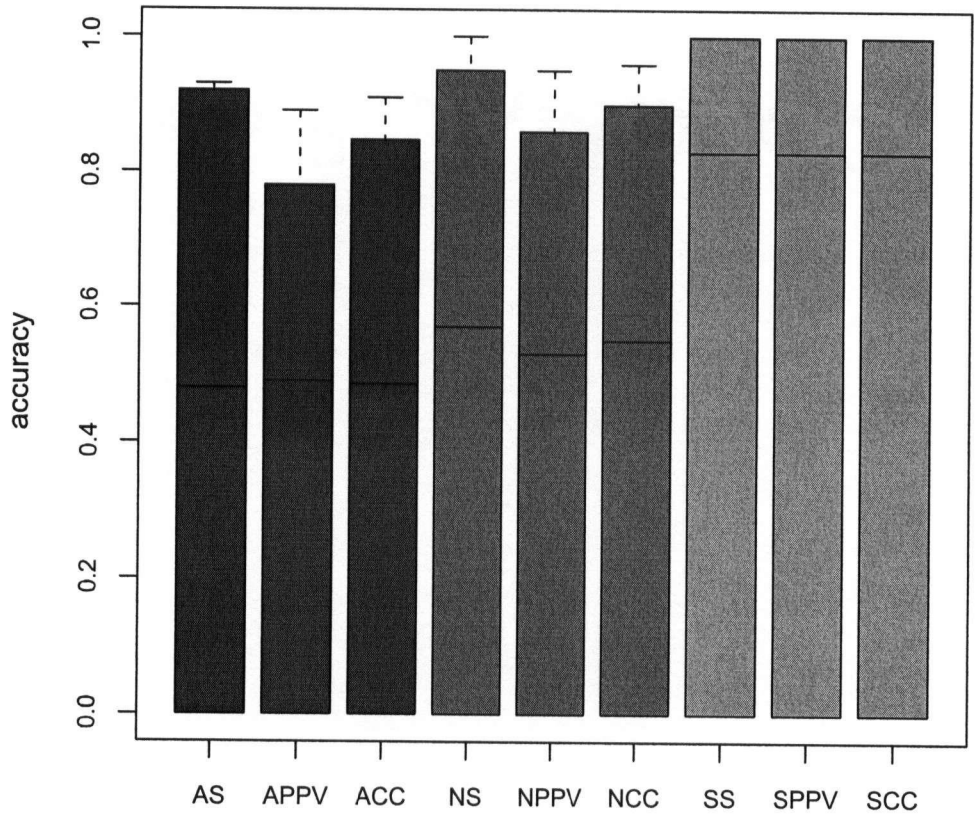

Figure 5.9: Distribution of accuracy results for nine UTR element test data sets by the *AS*, *APPV*, *ACC*, *NS*, *NPPV*, *NCC*, *SS*, *SPPV* and *SCC* measures. In general, the algorithm did not perform as well as it did on the miRNA data. This is indicated by the very wide distributions on all the measures. This was mainly due the the algorithm completely missing four of the nine motifs and therefore contributing 0 values to each of the accuracy measures in these cases. The algorithm however, performed quite well on the remaining five data sets.

score S by the number N of sequences in the input to give: $S' = S/N$. Normalisation was necessary since the score of a high scoring CM is expected to have a 'bit score' contribution from most of the sequences in the input. We then plotted S' against AS, $APPV$, NS, $NPPV$, SS and $SPPV$ for all of the fixed parameter results and tested each measure of accuracy for a statistical correlation with score using a Pearson's product moment correlation test. All measures were positively correlated with score and AS, $APPV$, NS and SS were statistically significantly correlated. Scatter plots of correlation against the accuracy measures along with the correlation coefficient (CC) and p-values of the correlation tests are shown in Figures 5.10, 5.11, 5.12, 5.13, 5.14 and 5.15. These results appear to indicate that score is an indicator of accuracy. Ideally the vertical axis intercept of the fitted line in Figures 5.10, 5.11, 5.12, 5.13, 5.14 and 5.15 would go through the origin (0,0) meaning that a CM with score of 0 had 0 true positives. This is not the case in our output, meaning that the algorithm is producing true positive results despite low-scoring output. Furthermore, there are six cases for which all accuracy measures are zero, meaning that the algorithm completely missed the motif. We investigated this data further by eliminating all score-accuracy measure pairs with zero values for the accuracy measures and replotting the data. The correlations were non-significant for all accuracy measures except for AS (correlation-coefficient = 0.57, p=0.01) (data not shown). These results demonstrate that for cases where the algorithm successfully identified the motifs, there is no correlation between score and accuracy. Therefore, we were unable to conclude that score was a positive indicator of accuracy. Cases where score is low, but accuracy is high need to be examined closely to determine why this occurs. We will discuss this further in Chapter 6.

5.2.2 Testing the effects of properties of the input data

We wanted to test if the algorithm accuracy was sensitive to inherent properties of the input data. This was possible, since the data sets were of variable length and

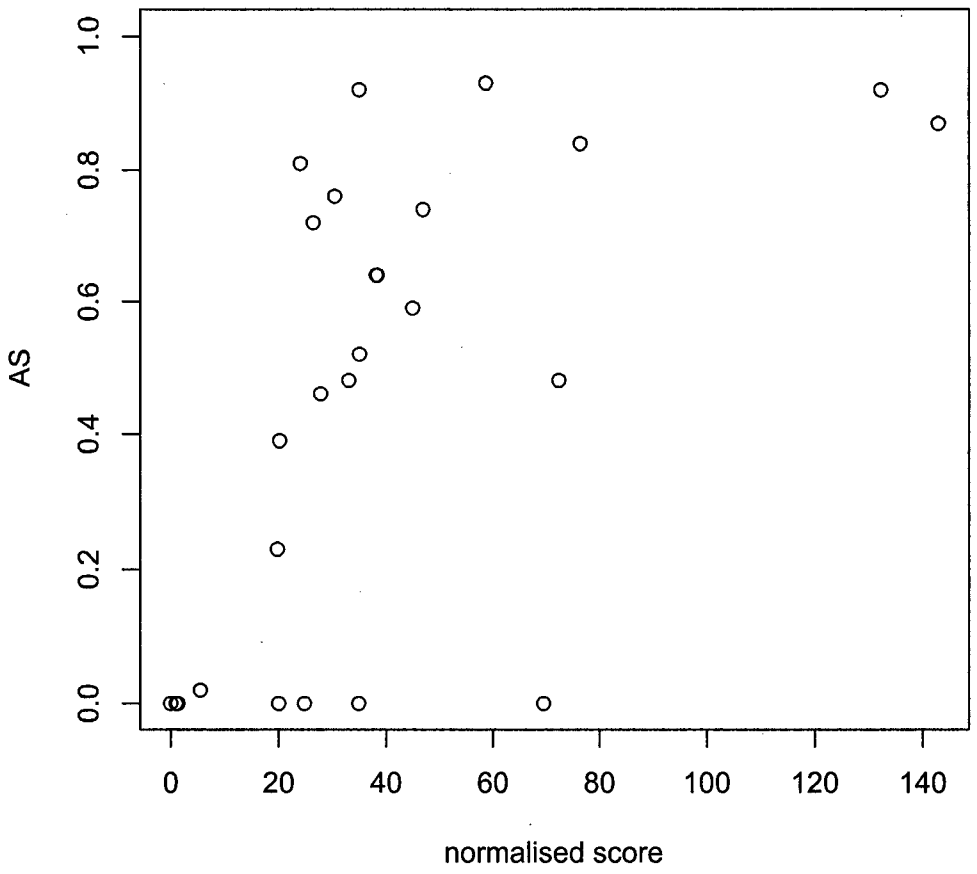

Figure 5.10: Scatter plot of AS against normalised score. AS was statistically significantly correlated with score (Pearson's product-moment correlation, correlation coefficient = 0.56, p=0.00).

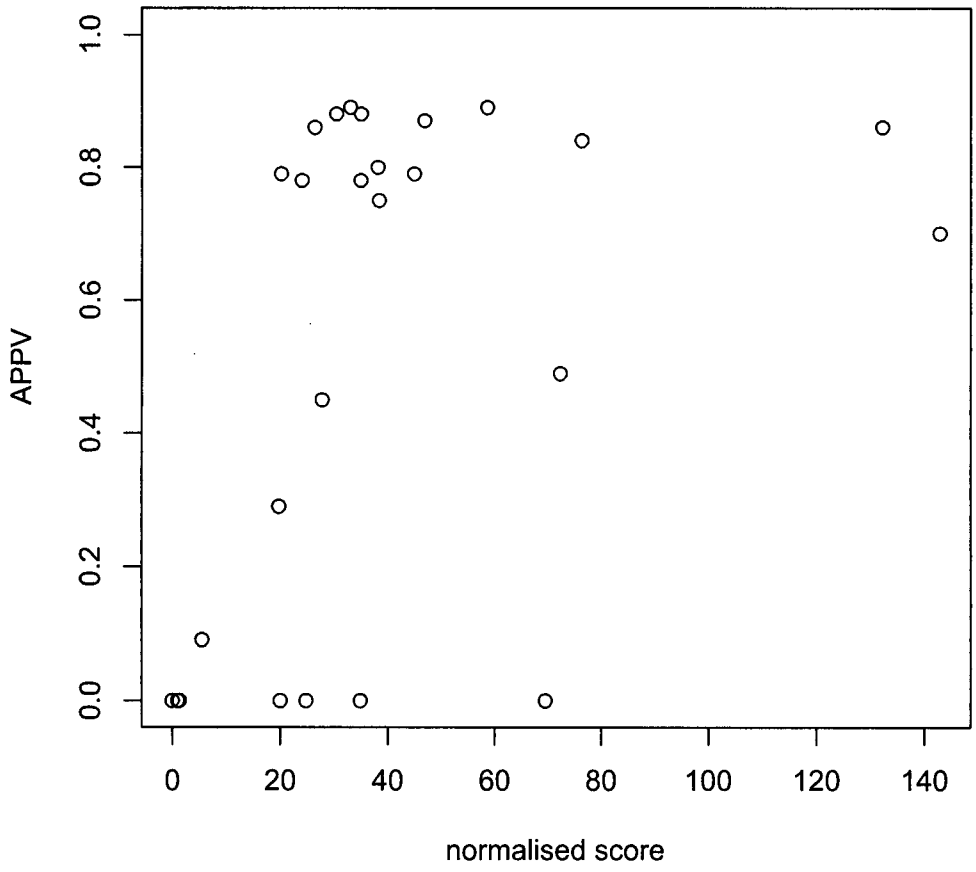

Figure 5.11: Scatter plot of *APPV* against normalised score. *APPV* was statistically significantly correlated with score (Pearson's product-moment correlation, correlation coefficient = 0.40, p=0.04).

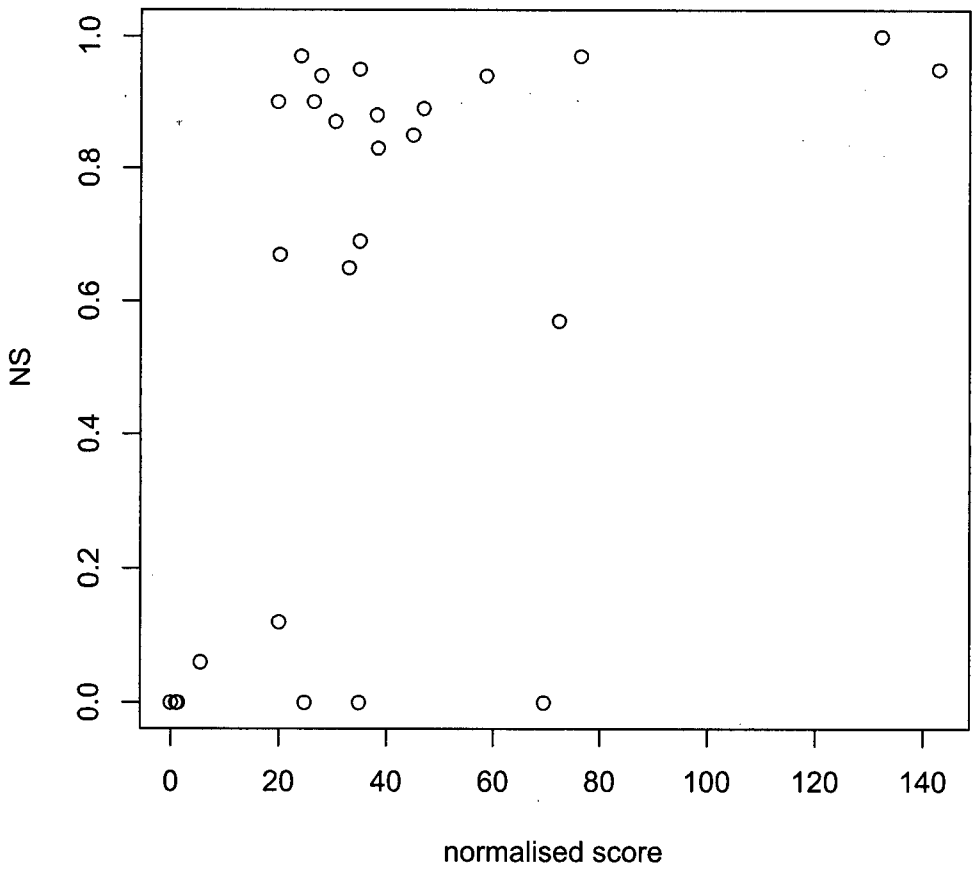

Figure 5.12: Scatter plot of NS against normalised score. NS was statistically significantly correlated with score (Pearson's product-moment correlation, correlation coefficient = 0.45, p=0.02).

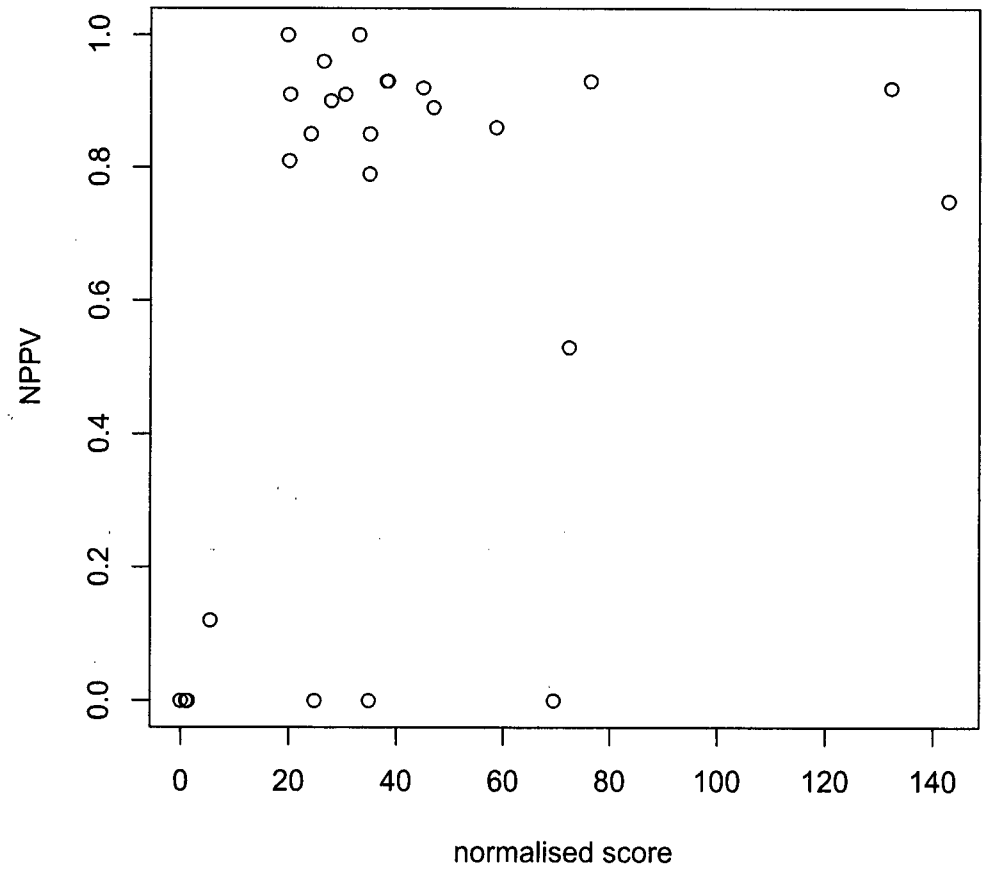

Figure 5.13: Scatter plot of $NPPV$ against normalised score. $NPPV$ was positively correlated with score, although the correlation was not statistically significant (Pearson's product-moment correlation, correlation coefficient = 0.29, p=0.14).

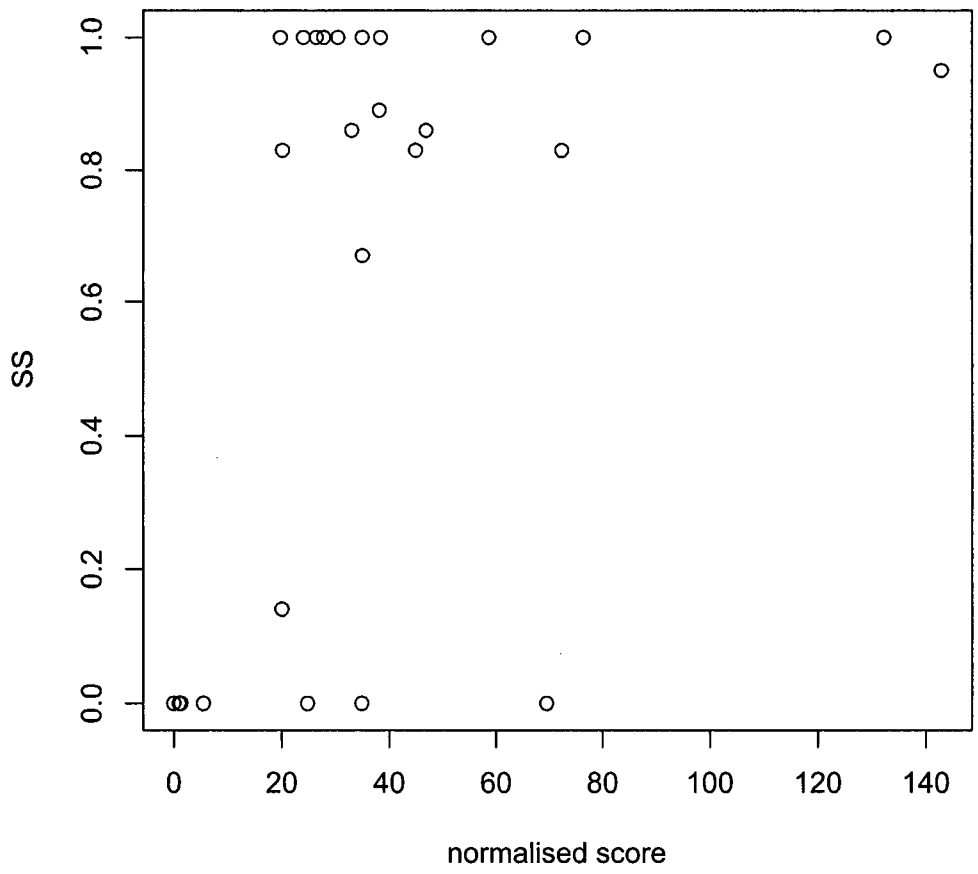

Figure 5.14: Scatter plot of SS against normalised score. SS was statistically significantly correlated with score (Pearson's product-moment correlation, correlation coefficient = 0.42, p=0.03).

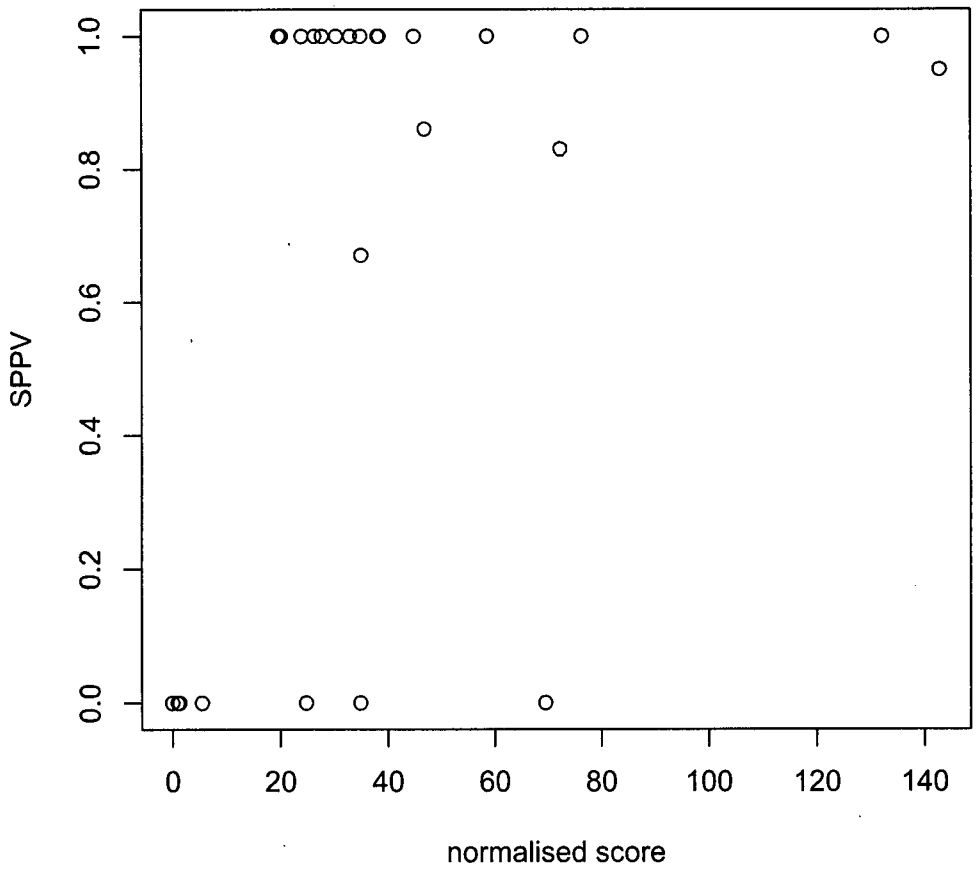

Figure 5.15: Scatter plot of $SPPV$ against normalised score. $SPPV$ was positively correlated with score, although the correlation was not statistically significant (Pearson's product-moment correlation, correlation coefficient = 0.36, p=0.07).

of varying sequence similarity (see Tables 4.1 and 4.2). We ran Pearson product-moment correlation tests to see if performance was affected by:

- average nucleotide percent pairwise identity (PID) of Rfam seed alignment members
- length of Rfam seed alignment
- length of input data
- number of sequences in the input data

None of the accuracy measures were significantly correlated with any of the data properties listed above (see Tables 5.3, 5.4, 5.5 and 5.6). All of the correlation coefficients were positive for the PID tests, suggesting a positive influence of PID on the results, however none of the tests yielded statistically significant results. Similarly, all of the correlation coefficients for the length of data test were negative, suggesting a negative influence of length of data on the results, however none of the tests yielded a statistically significant results. These results indicate that despite a mild bias towards high PID motifs and smaller input data, the algorithm can be run on different data sets with respect to size, number of sequences and PID and produce results that are independent of these properties. We will discuss the implications of these results with respect to the question posed in Section 4.1.2 in Chapter 6.

5.3 DISCO is more accurate, but considerably slower than RNAProfile

Figures 5.16, 5.17, 5.18, 5.19 compare NS, $NPPV$, SS and $SPPV$ accuracy measures of the best scoring model for DISCO against the best scoring model for RNAProfile. DISCO generally outperformed RNAProfile in accuracy for all measures. Mean NS was 0.56 with standard deviation 0.45 for DISCO and 0.41 with

standard deviation 0.26 for RNAProfile. Mean $NPPV$ was 0.56 with standard deviation 0.43 for DISCO and 0.48 with standard deviation 0.31 for RNAProfile. Mean SS was 0.58 with standard deviation 0.50 for DISCO and 0.48 with standard deviation 0.31 for RNAProfile. Mean $SPPV$ was 0.58 with standard deviation 0.50 for DISCO and 0.48 with standard deviation 0.31 for RNAProfile. In general, DISCO had better accuracy for RF00037, RF00164, RF00239, RF00241 and RF00256. RNAProfile had better accuracy for RF00130 and RF00185, two data sets where DISCO completely missed the motif. Both programs missed the motif in the RF00185 data set. Overall, we conclude that DISCO is more sensitive and has higher positive predictive value than RNAProfile.

Figure 5.20 shows the running time of DISCO and RNAProfile plotted against the size of the input data. RNAProfile had considerably faster running time than DISCO for all data sets by approximately one order of magnitude (mean log ratio of DISCO running time to RNAProfile was 1.46).

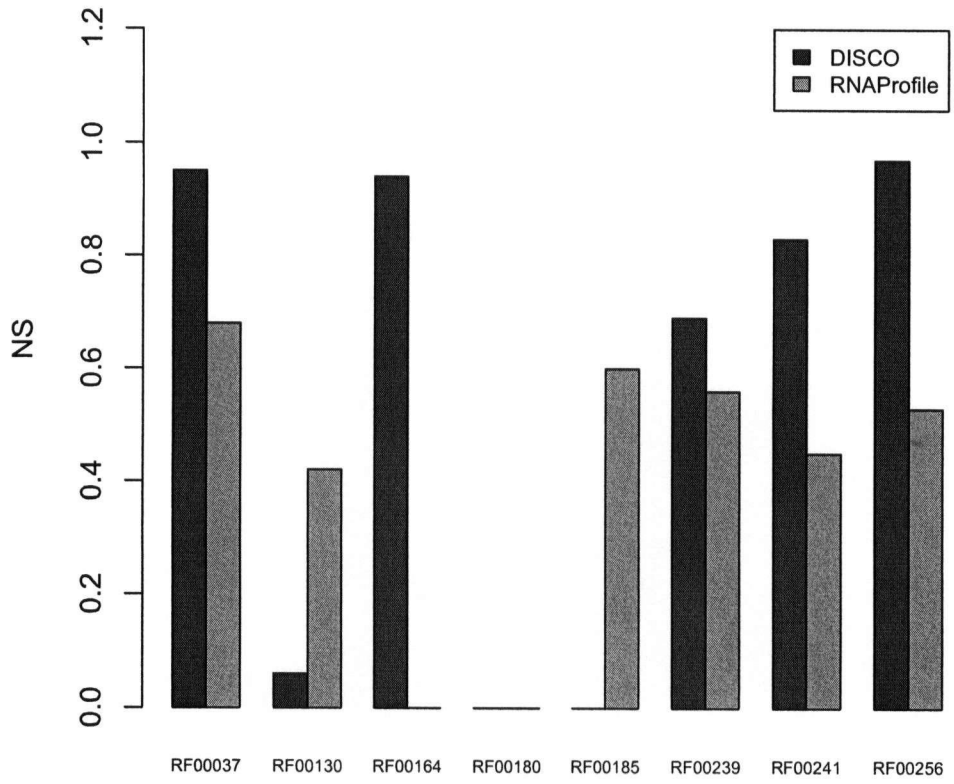

Figure 5.16: Comparison of NS between DISCO and RNAProfile. DISCO outperformed RNAProfile for the RF00037, RF00164, RF00239, RF00241 and RF00256 data sets, while RNAProfile outperformed DISCO for RF00130 and RF00185 data sets. Mean and standard deviation NS were 0.56 and 0.45 for DISCO and 0.41 and 0.26 for RNAProfile. By the NS measure, DISCO was more sensitive than RNAProfile.

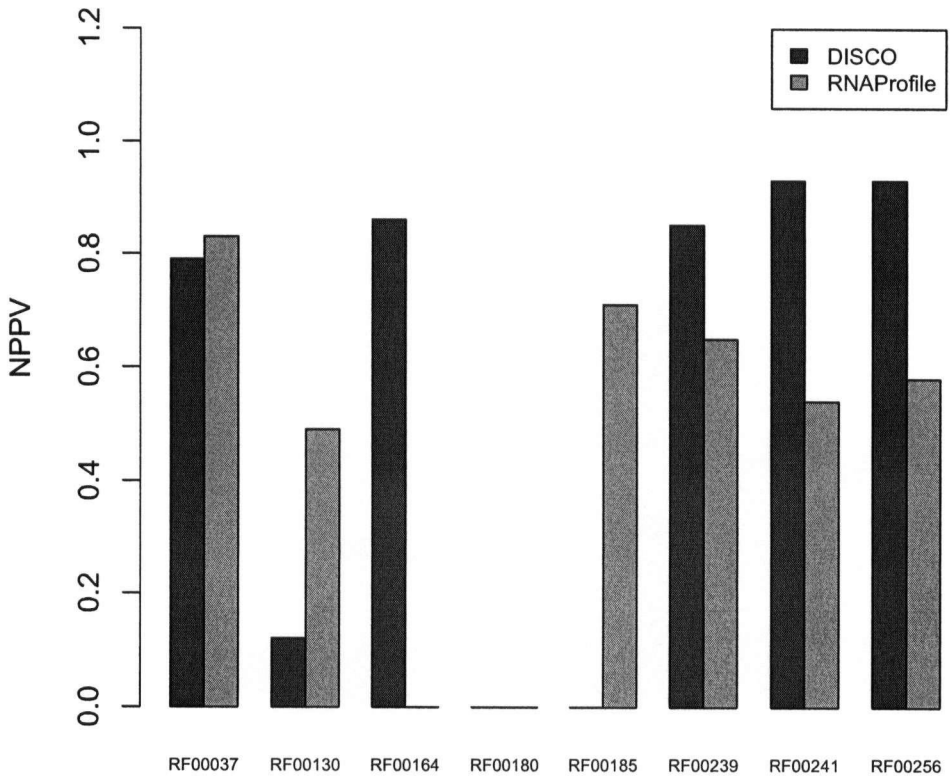

Figure 5.17: Comparison of $NPPV$ between DISCO and RNAProfile. DISCO outperformed RNAProfile for the RF00164, RF00239, RF00241 and RF00256 data sets, while RNAProfile outperformed DISCO for RF00037, RF00130 and RF00185 data sets. Mean and standard deviation $NPPV$ were 0.56 and 0.43 for DISCO and 0.48 and 0.31 for RNAProfile. By the $NPPV$ measure, DISCO had better positive predictive value than RNAProfile.

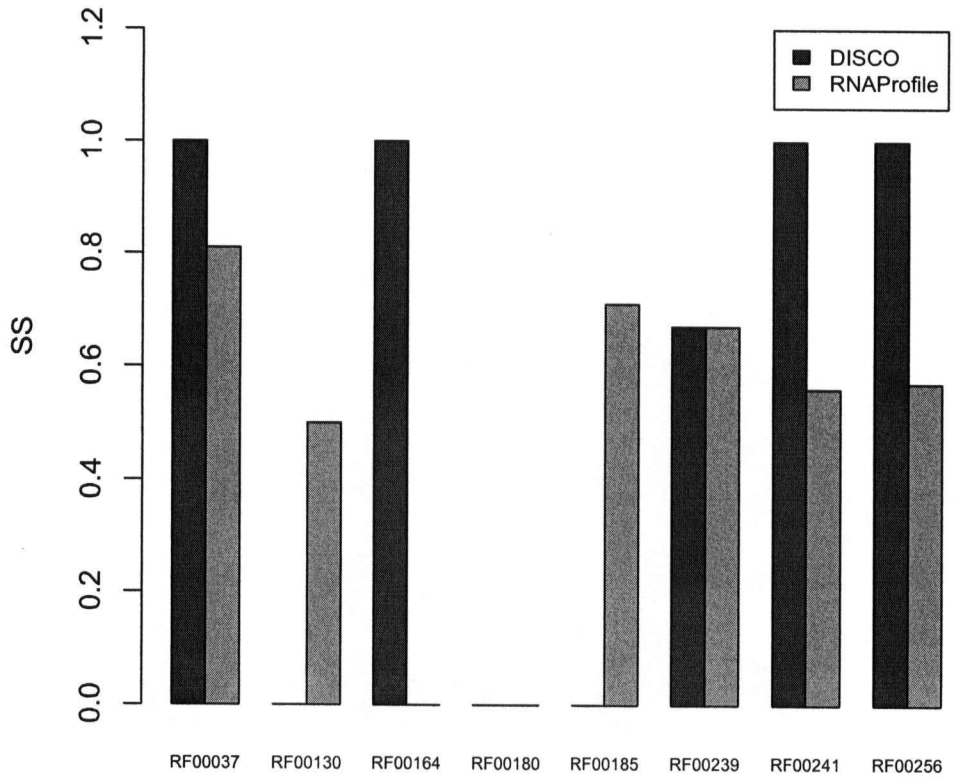

Figure 5.18: Comparison of SS between DISCO and RNAProfile. DISCO outperformed RNAProfile for the RF00037, RF00164, RF00241 and RF00256 data sets, while RNAProfile outperformed DISCO for RF00130 and RF00185 data sets. Mean and standard deviation SS were 0.58 and 0.50 for DISCO and 0.48 and 0.31 for RNAProfile. By the SS measure, DISCO was more sensitive than RNAProfile.

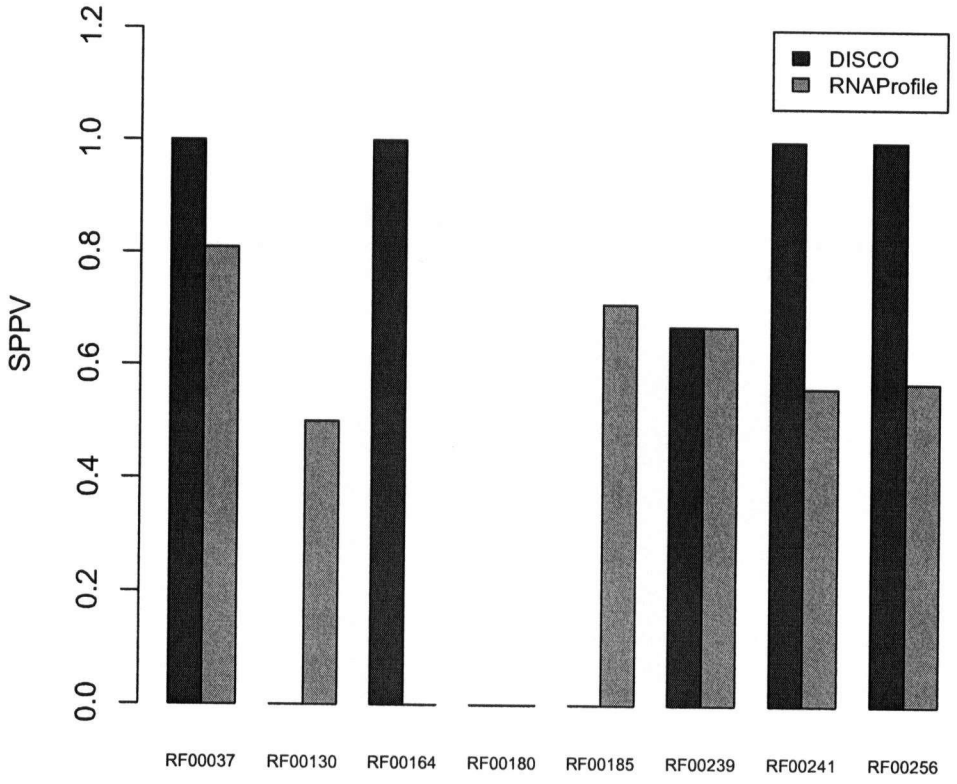

Figure 5.19: Comparison of $SPPV$ between DISCO and RNAProfile. DISCO outperformed RNAProfile for the RF00037, RF00164, RF00241 and RF00256 data sets, while RNAProfile outperformed DISCO for RF00130 and RF00185 data sets. Mean and standard deviation SS were 0.58 and 0.50 for DISCO and 0.48 and 0.31 for RNAProfile. By the $SPPV$ measure, DISCO was more sensitive than RNAProfile.

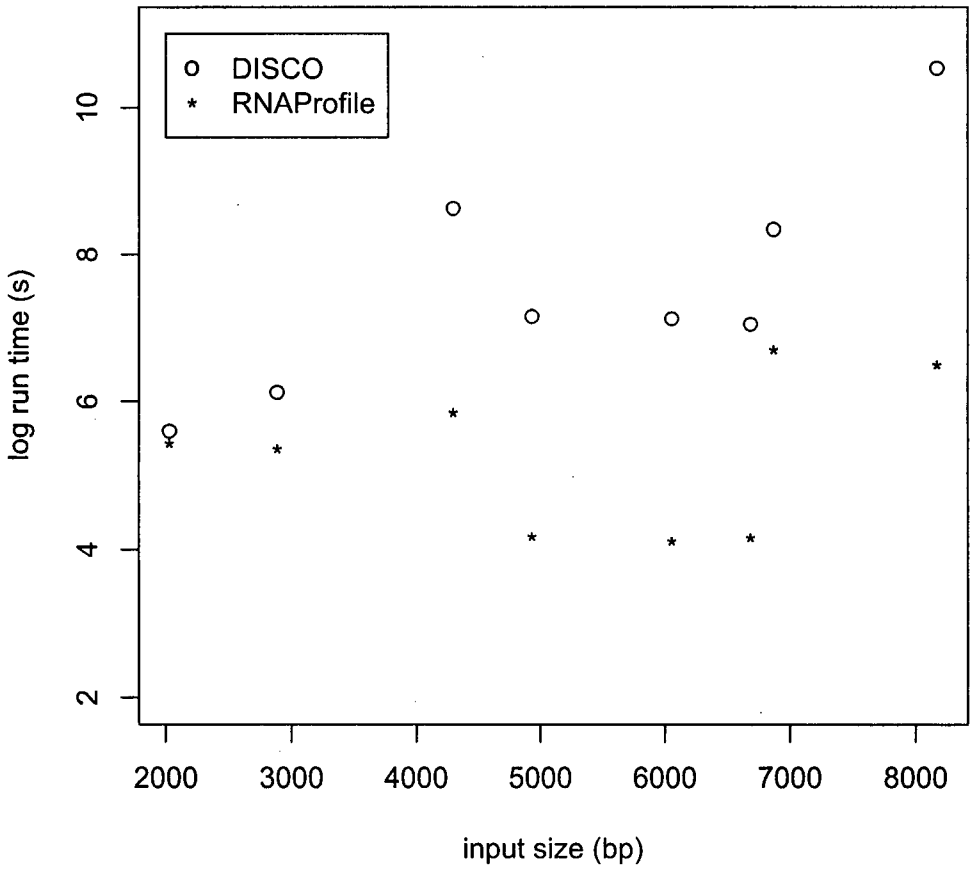

Figure 5.20: Running time vs size of input data for DISCO and RNAProfile. RNAProfile ran faster than DISCO for all data sets. The mean log ratio of DISCO to RNAProfile was 1.46 indicating on average, RNAProfile was faster by approximately one order of magnitude.

ID	SC	AS	APPV	ACC	NS	NPPV	NCC	SS	SPPV	SCC
027	539	0.59	0.79	0.68	0.85	0.92	0.88	0.83	1.00	0.91
051	79	0.23	0.29	0.26	0.90	1.00	0.95	1.00	1.00	1.00
052	343	0.64	0.80	0.72	0.88	0.93	0.90	0.89	1.00	0.94
053	121	0.39	0.79	0.56	0.67	0.91	0.78	0.83	1.00	0.91
075	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
076	96	0.81	0.78	0.79	0.97	0.85	0.91	1.00	1.00	1.00
103	231	0.48	0.89	0.65	0.65	1.00	0.81	0.86	1.00	0.93
130	22	0.02	0.09	0.04	0.06	0.12	0.08	0.00	0.00	0.00
131	111	0.46	0.45	0.45	0.94	0.90	0.92	1.00	1.00	1.00
239	210	0.52	0.88	0.68	0.69	0.85	0.77	0.67	0.67	0.67
241	345	0.64	0.75	0.69	0.83	0.93	0.88	1.00	1.00	1.00
246	132	0.72	0.86	0.79	0.90	0.96	0.93	1.00	1.00	1.00
247	140	0.00	0.00	0.00	0.12	0.81	0.31	0.14	1.00	0.37
248	152	0.76	0.88	0.82	0.87	0.91	0.89	1.00	1.00	1.00
251	328	0.74	0.87	0.80	0.89	0.89	0.89	0.86	0.86	0.86
256	1069	0.84	0.84	0.84	0.97	0.93	0.95	1.00	1.00	1.00
364	556	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M		0.46	0.59	0.52	0.66	0.76	0.70	0.71	0.80	0.74
Med		0.52	0.79	0.68	0.85	0.91	0.88	0.86	1.00	0.93
StD		0.30	0.36	0.32	0.36	0.35	0.35	0.40	0.39	0.39

Table 5.1: Results of fixed parameter experiments for miRNA data. The **ID** column shows the last three digits of the Rfam accession number of each miRNA data set. **M**, **Med** and **StD** rows show the mean, median and standard deviation of each measure of accuracy over all data sets.

ID	SC	AS	APPV	ACC	NS	NPPV	NCC	SS	SPPV	SCC
031	19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
032	17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
037	559	0.92	0.78	0.85	0.95	0.79	0.87	1.00	1.00	1.00
109	868	0.48	0.49	0.48	0.57	0.53	0.55	0.83	0.83	0.83
164	938	0.93	0.89	0.91	0.94	0.86	0.90	1.00	1.00	1.00
176	2248	0.92	0.86	0.89	1.00	0.92	0.96	1.00	1.00	1.00
180	322	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
185	488	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
214	2715	0.87	0.70	0.78	0.95	0.95	0.95	0.95	0.95	0.95
M		0.46	0.41	0.43	0.49	0.45	0.47	0.53	0.53	0.53
Med		0.48	0.49	0.49	0.57	0.53	0.55	0.83	0.83	0.83
StD		0.46	0.41	0.43	0.48	0.44	0.46	0.51	0.51	0.51

Table 5.2: Results of fixed parameter experiments for UTR data. The **ID** column shows the last three digits of the Rfam accession number of each UTR element data set. **M**, **Med** and **StD** rows show the mean, median and standard deviation of each measure of accuracy over all data sets.

Method	*CC*	p
AS	0.29	0.15
APPV	0.14	0.51
ACC	0.22	0.28
NS	0.21	0.30
NPPV	0.04	0.85
NCC	0.14	0.49
SS	0.18	0.39
SPPV	0.08	0.68
SCC	0.15	0.47

Table 5.3: Correlation statistics measuring association between PID and accuracy. No accuracy measures were significantly correlated with PID, although all were positively correlated to some degree.

Method	*CC*	p
AS	-0.21	0.3
APPV	-0.15	0.47
ACC	-0.19	0.36
NS	-0.16	0.43
NPPV	0.01	0.95
NCC	-0.10	0.61
SS	-0.15	0.45
SPPV	0.00	0.99
SCC	-0.11	0.59

Table 5.4: Correlation statistics measuring association between length of the seed alignment and accuracy. The correlation coefficient (CC) of the Pearson product-moment test and associated p-value (p) are shown for each measure of accuracy. No accuracy measures were significantly correlated with the length of the seed alignment, suggesting that DISCO does not show a bias based on length of the motif.

Method	*CC*	p
AS	-0.03	0.88
APPV	-0.18	0.38
ACC	-0.10	0.61
NS	-0.20	0.33
NPPV	-0.34	0.09
NCC	-0.26	0.19
SS	-0.23	0.26
SPPV	-0.27	0.18
SCC	-0.25	0.22

Table 5.5: Correlation statistics measuring association between length of the input data and accuracy. The correlation coefficient (CC) of the Pearson product-moment test and associated p-value (p) are shown for each measure of accuracy. No measures were significantly correlated with the length of the input data, suggesting that DISCO does not show a statistically significant bias based on the size of the input. However, all measures had a negative CC which indicates a small effect of size on the results.

Method	*CC*	p
AS	0.12	0.55
APPV	-0.09	0.66
ACC	0.02	0.91
NS	-0.10	0.61
NPPV	-0.30	0.14
NCC	-0.19	0.35
SS	-0.14	0.51
SPPV	-0.20	0.33
SCC	-0.16	0.44

Table 5.6: Correlation statistics measuring association between number of sequences in the input data and accuracy. The correlation coefficient (CC) of the Pearson product-moment test and associated p-value (p) are shown for each measure of accuracy. No accuracy measures were significantly correlated with the number of sequences in the input data, suggesting that DISCO does not show a bias based on the number of sequences in the input.

Chapter 6

Discussion

We developed an algorithm called DISCO to detect the most likely covariance model representing a motif embedded in a given set of unaligned RNA sequences. We tested our algorithm on 26 data sets from Rfam from two categories of RNA molecules: miRNAs and UTR elements. The data sets we constructed consisted of selected members of the Rfam family flanked by genomic or UTR sequence so that each instance of the motif was embedded in a larger sequence. Our algorithm performed quite well for the miRNA data sets and showed a type of bi-modal distribution for the UTR elements where the motif was very accurately found, or it was not found at all (see Chapter 5). We found that the score of the CM produced by the algorithm was not correlated with our measures of accuracy, suggesting that score is not an indicator of performance. We also found that the measures of accuracy were not significantly correlated with any inherent properties of the input data, indicating that the algorithm has an unbiased performance with respect to sequence similarity of the motif instance, length of the motif instances, length of the input data and the number of sequences in the input data. A comparison with a similar algorithm, RNAProfile, showed that DISCO produced more sensitive output with higher positive predictive value.

6.1 Interpretation of results

6.1.1 Sequence information is more important than secondary structure in the initialisation phase

In Chapter 4 we posed three major questions that we hoped our algorithm and experiments would help to answer. First, we wanted to determine what properties of the data - sequence or secondary structure emitted the stronger signals for motif detection (see Section 4.1.1). The results of the preliminary experiments showed that the sequence method of alignment was far superior to the structure and the combination methods (see Figures 5.1 and 5.2). These results indicate that the sequence carries more information than the secondary structure and that sequence information is generally sufficient to create a crude multiple alignment to initialise a CM, which necessarily introduces secondary structure information in the refinement phase.

Surprisingly, there was no statistically significant correlation between accuracy and pairwise sequence identity of the motif sequences using the sequence method. This is counter-intuitive and merits further study. It would be important to find an empirically derived threshold of pairwise sequence identity of the motifs below which the sequence method accuracy degraded.

For the RF00185 test data set in the fixed parameter experiments, the accuracy was 0 for all methods. However, we re-ran the algorithm with the same parameters except we used the structure alignment method instead. The accuracy results were $NS = 1.00$, $NPPV = 0.88$, $SS = 1.00$, and $SPPV = 1.00$. Although not as extreme, a similar improvement in accuracy using the structure method was achieved for RF00180, where the sequence results for all measures were 0, but the structure method gave: $NS = 0.54$, $NPPV = 0.48$, $SS = 1.00$ and $SPPV = 1.00$. Score results were 322 and 578 for the sequence and structure method respectively. These two examples indicate that while the sequence method of alignment gave the

most accurate results in general, the structure method is superior for specific data sets. More work is needed to see if there are detectable properties in the data that could indicate the selective use of the sequence or structure alignment method.

6.1.2 Relatively few sequences can be used to initialise the CM

The second question we posed was whether a sufficiently good CM could be initialised using only a subset of the input sequences (see Section 4.1.2). We ran our algorithm on data sets which contained between four and nineteen sequences using a fixed value of $k = 6$ for all data sets where the number of sequences N in the input data was at least 6. For the remaining data sets, we set $k = N$. Recall that the parameter k is the maximum number of W-mers to include in the multiple alignment step of the initialisation phase of the algorithm (see Algorithm 2). There was no statistically significant bias detected when the accuracy measures were tested for correlation with N. This indicates that in general, the algorithm works equally well with a fixed k independent of N. This gives us good evidence that our method can present a good initialisation multiple alignment to build a CM that is capable of retrieving the remainder of the motifs in the input data through iterative refinement.

6.1.3 The unpaired nucleotide filter improves performance but does not compromise accuracy for miRNA data sets

Finally, we tested to see if a simple secondary structure filter could improve performance while maintaining accuracy (see Section 4.1.3). When designing the algorithm we were concerned with the $O(L^2 \cdot W^2)$ term of the run-time complexity where L is the length of the longest sequence in the input, and W is the given width of the motif. Recall that this term arises from the exhaustive pairwise alignment of all W-mers in the input. For large data sets, this step is very expensive, so we introduced a threshold measure to reduce the number of pairwise alignments performed. Only W-mers with a proportion of unpaired nucleotides lower than a user

inputted d were considered for pairwise alignment. Of major concern was whether this threshold eliminated W-mers that were motif instances in the data. For the fixed parameter experiments, we used a relatively stringent threshold of $d = 0.40$ for the miRNA experiments, and the results were satisfactory for most data sets (see Table 5.1). This gives us some indication that setting d to capitalise on specific structural properties of the motif can yield good results and improve run-time performance. This attribute of our algorithm is unique when compared to the other algorithms presented in Section 2.7. We view this as a strength of our system that it can be tuned to take advantage of the structural properties of the motif if they are known ahead of time. Recent work by Bonnet *et al.* [6] and Washeitl *et al.* [68] has shown that minimum free energy signals are detectable in certain types of RNAs and our algorithm is poised to take advantage of this information.

Some motifs, however, are highly unstructured, and would not be detectable with only a minimum threshold. Implementing a maximum threshold as well would provide a range of proportion of unpaired nucleotides for W-mers to be admitted into the search space. We believe this idea should be explored further and would further enhance our algorithm.

6.1.4 Relatively poor accuracy of aligned nucleotides

Recall from Figures 5.8 and 5.9 the relatively poor accuracy with respect to the AS and $APPV$ methods. Measuring accuracy of alignments in this way was perhaps not the right approach in hindsight. Consider that the dynamic programming algorithms for both the alignment methods we use in the initialisation phase and the INSIDE algorithm potentially have multiple paths in their traceback routines. Since we broke ties in the scoring matrix by choosing a matched pair over a gap, this could have introduced a bias.

6.1.5 Poor UTR element results

Table 5.2 shows that the algorithm completely missed the motif in four out of nine UTR data sets. Given that three of these four sets were UTRs in predominantly mammalian mRNAs, it is not too surprising that the 'sequence' alignment method for the initialisation phase presented non-motif sequences to the refinement phase. Considering the proximity of the UTRs to coding sequence, it is reasonable to assume that these sequences may be under evolutionary selection pressures to maintain their sequence. Indeed, Shabalina *et al.* [60] recently reported the existence of highly conserved sequences in UTRs, detected through a genome wide comparison of orthologous mRNAs from eukaryotic species. Highly conserved patterns at the sequence level would most certainly influence the performance of our algorithm, which is not specifically designed for mRNAs. Pedersen *et al.* [54] introduce a comparative method for finding and folding RNA secondary structures within protein-coding regions. This work is of specific interest to the problem of detecting UTR elements and should be carefully considered in any modifications to our work that deal with biases in mRNA sequences.

6.2 Improvements on other methods

We introduced three improvements on other methods in our algorithm. First, we used the powerful probabilistic framework offered by CMs to both model and detect motifs in our input data. With the exception of SLASH [19], none of the other methods described in Section 2.7 model motifs in this way. The use of CMs have a great advantage in that they offer a sensitive alignment algorithm (INSIDE) to search for an instance of a CM in a given sequence. With respect to our work, this has a two-fold benefit in that the INSIDE algorithm can be used in the refinement phase and that the output of DISCO can be easily used to detect instances of the predicted motif in other sequence databases of interest. Second, we reduced the worst-case

time complexity to initialise a CM from $O(N^4L^4)$ in SLASH to $O(L^2 \cdot W^2)$ where N is the number of sequences in the input, L is the length the longest sequence in the input and W is the user inputted approximate width of the motif. This theoretical improvement is enhanced by the threshold parameter d which in our algorithm will substantially reduce the $((L-W) \times N)^2$ maximum possible number of pairwise alignments performed in the initialisation phase. Third, we introduce iterative refinement using EM to the RNA motif discovery problem. None of the methods described in Section 2.7 use iterative refinement. Given its widespread use in the sequence motif finding domain, we believe the use of EM is a worthwhile technique and confers an advantage to our algorithm.

6.2.1 Comparison between DISCO and RNAProfile

Our algorithm shows better accuracy than RNAProfile (see Figures 5.16, 5.17, 5.18 and 5.19) yet is considerably slower that RNAProfile (see Figure 5.20). We attribute both the superior accuracy and slower running time to the use of CMs. The Needleman-Wunsch based alignment algorithm of RNAProfile considers each position of the sequences to be independently derived by definition. One strength of the CM INSIDE algorithm is the use of transition probabilities between the states in the CM data structure which correspond to basepairs or unpaired bases in the sequence. The transition probabilities introduce a dependence between these states, and although they are costly in running time, in our opinion the transition probabilities confer an advantage over the profiles and associated alignment algorithm used by RNAProfile . We did not investigate the exact nature of this advantage but we believe it merits further study. Furthermore, comparison with other algorithms such as Alidot and FOLDALIGN is on-going work that is being prepared for submission to a journal for publication.

6.3 Drawbacks and limitations of the DISCO method

We acknowledge several drawbacks to our approach. Perhaps the most significant limitation is the need to specify the width W of the motif. Tools such as CARNAC [66] and RNAProfile [53] require different input parameters. RNAProfile only requires the number of stems the motif is expected to have. CARNAC does not require any other input except the unaligned sequences. The need to specify W is a limitation, but it should be noted that the sequences that make up the outputted CM need not be exactly W nucleotides long. Recall that the INSIDE algorithm allows for insertions and deletions and so the multiple alignment used in the refinement phase of the algorithm is expected to contain variable-length sequences.

6.3.1 Limitations of covariance models

While providing a robust probabilistic framework for modeling sets of related RNA sequences, CMs have two notable limitations. First, the bifurcating tree structure underlying the CM is incapable of modeling pseudoknots. Our algorithm will most likely not be able to detect pseudoknots which are detectable with comRNA [32]. Second, the complexity of the INSIDE algorithm is $O(L^3)$ where L is the length of the sequence. This makes our algorithm prohibitively expensive to run on long sequences. However, Weinberg and Ruzzo [70] recently reported a method that can filter out sequences in a database to be searched with a CM in $O(L^2)$ time with no reduction in accuracy. Use of this method in the refinement phase should be considered as a potential optimisation.

6.3.2 Reliance on predictive folding

There are two parts of our algorithm where predictive folding is performed. In the initialisation phase, we use the Zuker algorithm to predict the fold of each W-mer based on thermodynamic energy. This method is known to have limited accuracy of about 73%, measured by proportion of correctly predicted base pairs [48]. This is an

acknowledged limitation in our approach, but the Zuker algorithm remains the state of the art for single sequence prediction of secondary structure. An additional source of error could come from the consensus structure prediction of the multiple alignment in the refinement phase using Alifold. We did not perform a rigorous evaluation of the accuracy of the consensus structure predictions and therefore we do not know to what extent or how frequently the consensus structure is incorrectly determined. This merits further study which should include a comparative evaluation of Alifold and Pfold at both the run-time and accuracy levels.

6.4 Potential improvements and future work

While our results were encouraging, there are several areas where the DISCO algorithm could be improved. In the initialisation phase, we tested three alignment methods. The 'sequence' alignment method was superior. For the 'structure' and 'combination' methods, we constructed a scoring matrix using intuition rather than empirical results. A rigourously derived scoring matrix for the 'structure' method would provide a more accurate comparison to the 'sequence' method which used a matrix, RIBOSUM85-60 that was derived using maximum likelihood methods under the BLOSUM model of evolution (see [34]). Given that for some data sets, the 'structure' method did outperform the 'sequence' method, we feel this further work has merit.

6.4.1 Multiple sequence alignment method

We used a crude heuristic to construct a multiple alignment (see Algorithm 3). This method is missing the guide tree creation step used by hierarchical multiple alignment methods such as Clustalw [65] and outlined in Durbin *et al.* [13]. We did not compare our multiple alignment algorithm to hierarchical methods, and so we do not know if our heuristic negatively affected accuracy of our alignment. An assessment of specific cases where this method of multiple alignment in introducing

error is necessary to determine if our multiple alignment method is adequate.

6.4.2 The use of priors when initialising the CM

A uniform Dirichlet prior was used to intialise both the transition and emission probabilities of the CM. The effect of different priors and the use of any other empirically derived statistics in the construction of the CM were not investigated. Given the numerous (almost 400) CMs now available in Rfam, it would be interesting to estimate a more data-driven prior from these existing sets. Furthermore, priors for specific types of RNAs (eg miRNAs) could be estimated and optionally used if the user had prior knowledge of the type of motif they were expecting to discover.

6.4.3 Using phylogenetic weighting

In the field of comparative genomics, a growing body of literature is reporting different models to incorporate phylogenetic distance in analysing sets of sequences where the individual sequences originate from different organisms. Knudsen and Hein [36] infer a phylogenetic tree using maximum likelihood methods and use the distances in the tree to help infer a consensus secondary structure using SCFGs. Using a similar approach of weighting the alignment scores in the initialisation phase is bound to more accurately reflect the similarity of the sequences and in effect normalise the scores by evolutionary distance. The work of Holmes [29] describes an evolutionary model for RNA structure and its use in constructing pair-SCFGs to align two homologous RNAs. Exploring the use of such evolutionary models for RNA sequences would undoubtedly add an beneficial layer of accuracy to detection of motifs in sequences from different organisms.

6.4.4 Optimisations

Given the relatively high complexity of our algorithm, we have enumerated a number of optimisations that would improve the running time. Recall that the first step

of the initialisation phase is to fold every W-mer in the input. Currently this is implemented as a 'sliding window' across each sequence in the input, moving one position at a time. As each successive fold is only different by one nucleotide, $W-1$ columns and rows of the dynamic programming matrices used for folding could be saved and used in the calculation of the secondary structure of the next 'window'. Another optimisation could be implemented in the pairwise alignment step. As only a fixed number of high-scoring W-mers are kept for each W-mer, it may be possible to tell early in the alignment process if the alignment score will be sufficiently high. Implementing this early detection would reduce the number of complete W^2 operations and offer a substantial speed-up for the cases where the sequences do not align well. Finally, our algorithm is very amenable to parallelization. It should be very straightforward to implement message passing using the Message Passing Interface (MPI), so that the algorithm could run on a distributed memory cluster. Parallelization could be achieved for the folding of W-mers, the pairwise alignment step and the iterative refinement step. Given a distributed memory cluster with x nodes, the algorithm theoretically would run faster by a factor proportional to x.

Chapter 7

Conclusions

We designed and implemented an algorithm called DISCO to discover an optimal covariance model (CM) representing a motif expected to occur in a given set of unaligned RNA sequences. We were able to conclude that sequence information used to create a multiple alignment of motif sequences exhibits stronger signals than secondary structure information. We also demonstrated that a proportion of sequences of the input data could be used in the initialisation phase to crudely construct a CM. Most often the CM could retrieve the motifs in the remaining sequences in the refinement phase. Finally, we were able to use a simple filter that admitted only W-mers with a lower than d proportion of unpaired nucleotides into the search space without noticeable loss of accuracy.

Results on test data sets from Rfam showed that our algorithm performed well on miRNA data sets, however it showed inconsistent results on UTR element data sets. The score we used to measure the quality of the CM was not sufficiently correlated with accuracy measures we used to evaluate the performance of our algorithm, meaning that we were unable to conclude that score was a positive indicatory of accuracy. We were not able to detect any significant bias in our results with respect to pairwise sequence identity of the motif sequences, number of sequences in the input data, length of the input data or length of the motif. This suggests that

our algorithm can be applied generally to different types of data.

Our algorithm improves on competing algorithms in its use of covariance models for modeling motifs, a fast initialisation phase to generate the CM and the use of iterative refinement to improve the CM once initialised. In addition, we introduce a parameter that allows the user to tune the algorithm for data sets that exhibit specific structural properties if known before hand. The algorithm has a $O(W^3 \cdot L + L^2 \cdot W^2 + L^3)$ run-time complexity where W is the expected width of the motif, L is the length of the longest sequence in the input data, however we suggest several optimisations where this could be improved in Chapter 6. A comparison between RNAProfile and DISCO showed that DISCO was more sensitive and had higher positive predictive value than RNAProfile although due to the use of CMs, the running time was considerably slower.

The DISCO algorithm represents a new approach to detecting motifs in unaligned RNA sequences. We showed encouraging results with this prototype implementation and expect that, with the improvements suggested in Chapter 6, this algorithm could be widely applied to analysing sets of RNA sequences for conserved sequences and secondary structures. The work presented in this thesis is a step forward in computational RNA sequence analysis and we hope it will contribute to identifying novel functional RNA sequences.

Bibliography

[1] M. Andronescu, ZC. Zhang, and A. Condon. Secondary structure prediction of interacting RNA molecules. *J Mol Biol*, 345(5):987–1001, Feb 2005.

[2] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximizationm. *Machine Learning*, 21:51–83, 1995.

[3] T.L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol*, 2:28–36, 1994.

[4] I. Barrette, G. Poisson, P. Gendron, and F. Major. Pseudoknots in prion protein mRNAs confirmed by comparative sequence analysis and pattern searching. *Nucleic Acids Res*, 29(3):753–758, Feb 2001.

[5] A. Bateman, E. Birney, R. Durbin, SR. Eddy, RD. Finn, and EL. Sonnhammer. Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucleic Acids Res*, 27(1):260–262, Jan 1999.

[6] E. Bonnet, J. Wuyts, P. Rouzé, and Y. Van de Peer. Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. *Bioinformatics*, 20(17):2911–2917, Nov 2004.

[7] J. Buhler and M. Tompa. Finding motifs using random projections. *J Comput Biol*, 9:225–242, 2002.

[8] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, 268(1):78–94, Apr 1997.

[9] JL. Casey, MW. Hentze, DM. Koeller, SW. Caughman, TA. Rouault, RD. Klausner, and JB. Harford. Iron-responsive elements: regulatory RNA sequences that control mRNA levels and translation. *Science*, 240(4854):924–928, May 1988.

[10] FS. Collins, ED. Green, AE. Guttmacher, MS. Guyer, and . . A vision for the future of genomics research. *Nature*, 422(6934):835–847, Apr 2003.

[11] INTERNATIONAL HUMAN GENOME SEQUENCING CONSORTIUM. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, Oct 2004.

[12] RD. Dowell and SR. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5(1):71–71, Jun 2004.

[13] R. Durbin, S.R. Eddy, Krogh A., and Mitchison G. *Biological sequence analysis.* Cambridge University Press, 1998.

[14] SR. Eddy. Non-coding RNA genes and the modern RNA world. *Nat Rev Genet*, 2(12):919–929, Dec 2001.

[15] SR. Eddy. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics*, 3(1):18–18, Jul 2002.

[16] SR. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Res*, 22(11):2079–2088, Jun 1994.

[17] GB. Fogel, VW. Porto, DG. Weekes, DB. Fogel, RH. Griffey, JA. McNeil, E. Lesnik, DJ. Ecker, and R. Sampath. Discovery of RNA structural elements using evolutionary computation. *Nucleic Acids Res*, 30(23):5310–5317, Dec 2002.

[18] J. Gorodkin, LJ. Heyer, and GD. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res*, 25(18):3724–3732, Sep 1997.

[19] J. Gorodkin, SL. Stricklin, and GD. Stormo. Discovering common stem-loop motifs in unaligned RNA sequences. *Nucleic Acids Res*, 29(10):2135–2144, May 2001.

[20] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and SR. Eddy. Rfam: an RNA family database. *Nucleic Acids Res*, 31(1):439–441, Jan 2003.

[21] S. Griffiths-Jones, S. Moxon, M. Marshall, A. Khanna, SR. Eddy, and A. Bateman. Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res*, 33 Database Issue:121–124, Jan 2005.

[22] JH. Havgaard, R. Lyngso, GD. Stormo, and J. Gorodkin. Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. *Bioinformatics*, Jan 2005.

[23] CU. Hellen and P. Sarnow. Internal ribosome entry sites in eukaryotic mRNA molecules. *Genes Dev*, 15(13):1593–1612, Jul 2001.

[24] MW. Hentze, SW. Caughman, JL. Casey, DM. Koeller, TA. Rouault, JB. Harford, and RD. Klausner. A model for the structure and functions of iron-responsive elements. *Gene*, 72(1-2):201–208, Dec 1988.

[25] MW. Hentze and LC. Kühn. Molecular control of vertebrate iron metabolism: mRNA-based regulatory circuits operated by iron, nitric oxide, and oxidative stress. *Proc Natl Acad Sci U S A*, 93(16):8175–8182, Aug 1996.

[26] IL. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Res*, 31(13):3429–3431, Jul 2003.

[27] IL. Hofacker, M. Fekete, C. Flamm, MA. Huynen, S. Rauscher, PE. Stolorz, and PF. Stadler. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucleic Acids Res*, 26(16):3825–3836, Aug 1998.

[28] IL. Hofacker, M. Fekete, and PF. Stadler. Secondary structure prediction for aligned RNA sequences. *J Mol Biol*, 319(5):1059–1066, Jun 2002.

[29] I. Holmes. A probabilistic model for the evolution of RNA structure. *BMC Bioinformatics*, 5(1):166–166, Oct 2004.

[30] YJ. Hu. Prediction of consensus structural motifs in a family of coregulated RNA sequences. *Nucleic Acids Res*, 30(17):3886–3893, Sep 2002.

[31] A. Jasinska, G. Michlewski, M. de Mezer, K. Sobczak, P. Kozlowski, M. Napierala, and WJ. Krzyzosiak. Structures of trinucleotide repeats in human transcripts and their functional implications. *Nucleic Acids Res*, 31(19):5463–5468, Oct 2003.

[32] Y. Ji, X. Xu, and GD. Stormo. A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics*, 20(10):1591–1602, Jul 2004.

[33] B. John, AJ. Enright, A. Aravin, T. Tuschl, C. Sander, and DS. Marks. Human MicroRNA targets. *PLoS Biol*, 2(11):–1373, Nov 2004.

[34] RJ. Klein and SR. Eddy. RSEARCH: Finding homologs of single structured RNA sequences. *BMC Bioinformatics*, 4(1):44–44, Sep 2003.

[35] B. Knudsen and J. Hein. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, 15(6):446–454, Jun 1999.

[36] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res*, 31(13):3423–3428, Jul 2003.

[37] J. Krol, K. Sobczak, U. Wilczynska, M. Drath, A. Jasinska, D. Kaczynska, and WJ. Krzyzosiak. Structural features of microRNA (miRNA) precursors and their relevance to miRNA biogenesis and small interfering RNA/short hairpin RNA design. *J Biol Chem*, 279(40):42230–42239, Oct 2004.

[38] M. Lagos-Quintana, R. Rauhut, W. Lendeckel, and T. Tuschl. Identification of novel genes coding for small expressed RNAs. *Science*, 294(5543):853–858, Oct 2001.

[39] A. Lambert, A. Lescure, and D. Gautheret. A survey of metazoan selenocysteine insertion sequences. *Biochimie*, 84(9):953–959, Sep 2002.

[40] NC. Lau, LP. Lim, EG. Weinstein, and DP. Bartel. An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans. *Science*, 294(5543):858–862, Oct 2001.

[41] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[42] C.E. Lawrence and A.A. Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7:41–51, 1990.

[43] RC. Lee and V. Ambros. An extensive class of small RNAs in Caenorhabditis elegans. *Science*, 294(5543):862–864, Oct 2001.

[44] EA. Lesnik, R. Sampath, and DJ. Ecker. Rev response elements (RRE) in lentiviruses: an RNAMotif algorithm-based strategy for RRE prediction. *Med Res Rev*, 22(6):617–636, Nov 2002.

[45] TM. Lowe and SR. Eddy. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res*, 25(5):955–964, Mar 1997.

[46] TM. Lowe and SR. Eddy. A computational screen for methylation guide snoRNAs in yeast. *Science*, 283(5405):1168–1171, Feb 1999.

[47] TJ. Macke, DJ. Ecker, RR. Gutell, D. Gautheret, DA. Case, and R. Sampath. RNAMotif, an RNA secondary structure definition and search algorithm. *Nucleic Acids Res*, 29(22):4724–4735, Nov 2001.

[48] DH. Mathews. Predicting a set of minimal free energy RNA secondary structures common to two sequences. *Bioinformatics*, Feb 2005.

[49] BW. Matthews. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta*, 405(2):442–451, Oct 1975.

[50] JS. Mattick. Non-coding RNAs: the architects of eukaryotic complexity. *EMBO Rep*, 2(11):986–991, Nov 2001.

[51] SB. Needleman and CD. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, Mar 1970.

[52] R. Nussinov and AB. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc Natl Acad Sci U S A*, 77(11):6309–6313, Nov 1980.

[53] G. Pavesi, G. Mauri, M. Stefani, and G. Pesole. RNAProfile: an algorithm for finding conserved secondary structure motifs in unaligned RNA sequences. *Nucleic Acids Res*, 32(10):3258–3269, 2004.

[54] JS. Pedersen, IM. Meyer, R. Forsberg, P. Simmonds, and J. Hein. A comparative method for finding and folding RNA secondary structures within protein-coding regions. *Nucleic Acids Res*, 32(16):4925–4936, 2004.

[55] O. Perriquet, H. Touzet, and M. Dauchet. Finding the common structure shared by two homologous RNAs. *Bioinformatics*, 19(1):108–116, Jan 2003.

[56] S. Pfeffer, M. Zavolan, FA. Grässer, M. Chien, JJ. Russo, J. Ju, B. John, AJ. Enright, D. Marks, C. Sander, and T. Tuschl. Identification of virus-encoded microRNAs. *Science*, 304(5671):734–736, Apr 2004.

[57] E. Rivas and SR. Eddy. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics*, 16(7):583–605, Jul 2000.

[58] E. Rivas, RJ. Klein, TA. Jones, and SR. Eddy. Computational identification of noncoding RNAs in E. coli by comparative genomics. *Curr Biol*, 11(17):1369–1373, Sep 2001.

[59] LF. Sempere, S. Freemantle, I. Pitha-Rowe, E. Moss, E. Dmitrovsky, and V. Ambros. Expression profiling of mammalian microRNAs uncovers a subset of brain-expressed microRNAs with possible roles in murine and human neuronal differentiation. *Genome Biol*, 5(3):–862, 2004.

[60] SA. Shabalina, AY. Ogurtsov, IB. Rogozin, EV. Koonin, and DJ. Lipman. Comparative analysis of orthologous eukaryotic mRNAs: potential hidden functional signals. *Nucleic Acids Res*, 32(5):1774–1782, 2004.

[61] SP. Shah, Y. Huang, T. Xu, MM. Yuen, J. Ling, and BF. Ouellette. Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics*, 6(1):34–34, Feb 2005.

[62] TF. Smith and MS. Waterman. Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–197, Mar 1981.

[63] EL. Sonnhammer, SR. Eddy, E. Birney, A. Bateman, and R. Durbin. Pfam: multiple sequence alignments and HMM-profiles of protein domains. *Nucleic Acids Res*, 26(1):320–322, Jan 1998.

[64] EL. Sonnhammer, SR. Eddy, and R. Durbin. Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins*, 28(3):405–420, Jul 1997.

[65] JD. Thompson, DG. Higgins, and TJ. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–4680, Nov 1994.

[66] H. Touzet and O. Perriquet. CARNAC: folding families of related RNAs. *Nucleic Acids Res*, 32(Web Server issue):142–145, Jul 2004.

[67] DI. Van Ryk and S. Venkatesan. Real-time kinetics of HIV-1 Rev-Rev response element interactions. Definition of minimal binding sites on RNA and protein and stoichiometric analysis. *J Biol Chem*, 274(25):17452–17463, Jun 1999.

[68] S. Washietl, IL. Hofacker, and PF. Stadler. Fast and reliable prediction of noncoding RNAs. *Proc Natl Acad Sci U S A*, 102(7):2454–2459, Feb 2005.

[69] RH. Waterston, K. Lindblad-Toh, E. Birney, J. Rogers, JF. Abril, P. Agarwal, R. Agarwala, R. Ainscough, M. Alexandersson, P. An, SE. Antonarakis, J. Attwood, R. Baertsch, J. Bailey, K. Barlow, S. Beck, E. Berry, B. Birren, T. Bloom, P. Bork, M. Botcherby, N. Bray, MR. Brent, DG. Brown, SD. Brown, C. Bult, J. Burton, J. Butler, RD. Campbell, P. Carninci, S. Cawley, F. Chiaromonte, AT. Chinwalla, DM. Church, M. Clamp, C. Clee, FS. Collins, LL. Cook, RR. Copley, A. Coulson, O. Couronne, J. Cuff, V. Curwen, T. Cutts, M. Daly, R. David, J. Davies, KD. Delehaunty, J. Deri, ET. Dermitzakis, C. Dewey, NJ. Dickens, M. Diekhans, S. Dodge, I. Dubchak, DM. Dunn, SR. Eddy, L. Elnitski, RD. Emes, P. Eswara, E. Eyras, A. Felsenfeld, GA. Fewell, P. Flicek, K. Foley, WN. Frankel, LA. Fulton, RS. Fulton, TS. Furey, D. Gage, RA. Gibbs, G. Glusman, S. Gnerre, N. Goldman, L. Goodstadt, D. Grafham, TA. Graves, ED. Green, S. Gregory, R. Guigó, M. Guyer, RC. Hardison, D. Haussler, Y. Hayashizaki, LW. Hillier, A. Hinrichs, W. Hlavina, T. Holzer, F. Hsu, A. Hua, T. Hubbard, A. Hunt, I. Jackson, DB. Jaffe, LS. Johnson, M. Jones, TA. Jones, A. Joy, M. Kamal, EK. Karlsson, D. Karolchik, A. Kasprzyk, J. Kawai, E. Keibler, C. Kells, WJ. Kent, A. Kirby, DL. Kolbe, I. Korf, RS. Kucherlapati, EJ. Kulbokas, D. Kulp, T. Landers, JP. Leger, S. Leonard, I. Letunic, R. Levine, J. Li, M. Li, C. Lloyd, S. Lucas, B. Ma, DR. Maglott, ER. Mardis, L. Matthews, E. Mauceli, JH. Mayer, M. McCarthy, WR. McCombie, S. McLaren, K. McLay, JD. McPherson, J. Meldrim, B. Meredith, JP. Mesirov, W. Miller, TL. Miner, E. Mongin, KT. Montgomery, M. Morgan, R. Mott, JC. Mullikin, DM. Muzny, WE. Nash, JO. Nelson, MN. Nhan, R. Nicol, Z. Ning, C. Nusbaum, MJ. O'Connor, Y. Okazaki, K. Oliver, E. Overton-Larty, L. Pachter, G. Parra, KH. Pepin, J. Peterson, P. Pevzner, R. Plumb, CS. Pohl, A. Poliakov, TC. Ponce, CP. Ponting, S. Potter, M. Quail, A. Reymond, BA. Roe, KM. Roskin, EM. Rubin, AG. Rust, R. Santos, V. Sapojnikov, B. Schultz, J. Schultz, MS. Schwartz, S. Schwartz, C. Scott, S. Seaman, S. Searle, T. Sharpe, A. Sheridan, R. Shownkeen, S. Sims, JB. Singer, G. Slater, A. Smit, DR. Smith, B. Spencer, A. Stabenau, N. Stange-Thomann, C. Sugnet, M. Suyama, G. Tesler, J. Thompson, D. Torrents, E. Trevaskis, J. Tromp, C. Ucla, A. Ureta-Vidal, JP. Vinson, AC. Von Niederhausern, CM. Wade, M. Wall, RJ. Weber, RB. Weiss, MC. Wendl, AP. West, K. Wetterstrand, R. Wheeler, S. Whelan, J. Wierzbowski, D. Willey, S. Williams, RK. Wilson, E. Winter, KC. Worley, D. Wyman, S. Yang, SP. Yang, EM. Zdobnov, MC. Zody, and ES. Lander. Initial sequencing and comparative analysis of the

mouse genome. *Nature*, 420(6915):520–562, Dec 2002.

[70] Z. Weinberg and WL. Ruzzo. Exploiting conserved structure for faster annotation of non-coding RNAs without loss of accuracy. *Bioinformatics*, 20 Suppl 1:334–334, Aug 2004.

[71] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*, 9(1):133–148, Jan 1981.