Probabilistic Models of RNA Secondary Structure

James W. J. Anderson

St. Catherine's College College, Oxford



A thesis submitted to the University of Oxford for the degree of Doctor of Philosophy in the Division of Mathematical and Physical Sciences

Trinity Term, 2013

Probabilistic Models of RNA Secondary Structure

James W. J. Anderson

St. Catherine's College College, Oxford
University of Oxford

Thesis submitted for the degree of Doctor of Philosophy
at the University of Oxford

Trinity Term, 2013

Abstract

This thesis develops probabilistic models of RNA secondary structure. The first chapter introduces RNA secondary structure prediction, in particular stochastic context-free grammars (SCFGs), and considers a novel method for automated design of SCFGs. Many SCFGs are found with a similar predictive quality as those commonly used for RNA secondary structure prediction. The second chapter discusses the effect alignment quality, evolutionary distance between sequences, and number of sequences in an alignment have on RNA secondary structure prediction. By combining statistical alignment and SCFG models we can, in a statistically sound setting, average structure predictions over the space of alignments to decrease loss created by poor alignments. The third chapter incorporates additional biological information about RNA secondary structure formation into the decoding of the SCFG posterior distribution. Combining iterative helix formation, phylogenetic modelling, and a distance function between alignment columns leads to the an improvement in the accuracy of comparative RNA secondary structure prediction. Finally, appendices briefly discuss further work concerning probabilistic models of RNA secondary structure which may be of interest to the reader.

Acknowledgements

There are many people without whom this work would not have been possible.

Firstly I would like to thank the Department of Statistics and all the staff there, particularly Maddy Mitchell, Jennie McKenzie, and Jan Boylan. Also St. Catherine's College, the EPSRC for thesis funding, and all those who gave me travel or conference funding in order to learn and develop ideas.

Secondly, those I have worked with. There have been many people I have worked with who have contributed thoughts, code, or criticism. Students I have helped supervise on RNA projects and would like to thank include Tallulah Andrews, Maria Astefanoaei, Preeti Arunapuram, Amarendra Badugu, Stephen Cristiano, Ingolfur Edvardsson, Michael Golden, Diana Gratie, Pierre Haas, Tomas Hyland, Charles Lake, Leigh-Anne Mathieson, Cyrus Rich, Elena Sizikova, Adi Steif, Ashley Teufel, Vladimir Volykin, and William Weir. Enjoyable times were had working on RNA projects with Joe Herman, Ádám Novák, Joe Staines, and Paula Tataru. Particular thanks go to Rune Lyngsø and Zsuzsanna Sükösd, who have been very involved with exploring ideas with me, discussing new ones, and giving feedback on this thesis.

Lastly I would like to thank my supervisor Jotun Hein, who has provided me with many interesting discussions, a wealth of opportunities to travel and teach, and perhaps more of a free reign than might typically be given to doctoral students to develop ideas. I hope he will be as pleased as I am once this thesis is submitted.

Declaration

I declare that this thesis contains my own original work except where otherwise stated, with reference to the literature. The research was undertaken at the Department of Statistics, University of Oxford.

James W. J. Anderson
Oxford '13

Preface

The majority of work in this thesis has been published, mostly in peer-reviewed academic journals. A notable amount of effort has been made in the writing of these publications, to create quality research, express material clearly, and to discuss methodological details. Furthermore, I have been told that examining theses is significantly easier if large sections are taken from peer-reviewed published material. Consequently, it seems to make sense for all parties to compose this thesis as not much more than a collection of published papers.

I have also been told that examining a shorter thesis is, in general, more enjoyable than examining a longer one. With this in mind, I have decided to focus on three main ideas I have developed in probabilistic modelling of RNA secondary structure. To create a more complete picture, and for potential reader interest, I have created appendices for additional projects I have worked on in the course of my doctoral studies. Most of these have also been published in peer-reviewed academic journals.

A note on the manuscript
To ensure this manuscript remains coherent, numbering of figures, tables, algorithms, equations, and sections may have been altered from the published works. Formatting of equations may have changed slightly, and a common referencing style has been created. Permission has been obtained where necessary for reproduction of material.

James W. J. Anderson
Oxford ‘13

Contents

1 Evolving Stochastic Context-Free Grammars **1**

1.1 RNA Secondary Structure Prediction 2

1.1.1 Thermodynamic Models 3

1.1.2 Probabilistic Models 5

1.2 Background on Stochastic Context-Free Grammars 7

1.2.1 Stochastic Context-Free Grammars 7

1.2.2 Normal Forms 10

1.2.3 Ambiguity and Completeness 11

1.2.4 Algorithms for SCFGs 13

1.2.5 Parameter Inference 16

1.2.6 Secondary Structure Prediction 19

1.3 Evolving Stochastic Context-Free Grammars for RNA Secondary Structure Prediction 25

1.3.1 An Evolutionary Algorithm for Grammars 27

1.3.2 Data 31

1.3.3 Results and Discussion 39

1.3.4 Conclusions and further work 48

2 Averaging Variances in Comparative RNA Secondary Structure Prediction 51

2.1 Variances in Comparative RNA Secondary Structure Prediction 53

2.1.1 Data . . . 55

2.1.2 Results of Variance Analysis . . . 60

2.2 Averaging RNA Secondary Structure Predictions over Alignments . . . 64

2.2.1 Projecting Alignments . . . 65

2.2.2 Averaging over Alignments . . . 66

2.2.3 Benchmarking . . . 71

2.3 Discussion, Conclusions, and Further Work . . . 72

3 Kinetic Folding of RNA Using Stochastic Context-Free Grammars 77

3.1 Oxfold: Kinetic Folding of RNA using Stochastic Context-Free Grammars and Evolutionary Information . . . 79

3.1.1 Folding Kinetics: Iterative Helix Formation . . . 80

3.1.2 Bayesian Weighting: The Distance Function . . . 85

3.2 Results . . . 88

3.2.1 Benchmarking Data and Parameters . . . 88

3.2.2 Benchmarking Statistics . . . 93

3.3 Discussion . . . 94

3.3.1 Co-transcriptional Folding . . . 95

3.3.2 Pseudoknots . . . 97

3.3.3 Computational Complexity . . . 98

3.3.4 Non-Canonical Basepairs 99
3.4 Summary 100
3.4.1 Future Perspectives 102
3.4.2 Further Work 103

A Frnakenstein: Inverse Folding of RNA **106**

B Information Entropy for SCFGs **108**

C Predicting Alignment and Prediction Quality **111**

D An n-free-passes CYK Algorithm for Error-correction and Non-canonical Base-pairs **114**

Chapter 1

Evolving Stochastic Context-Free Grammars

In this chapter RNA secondary structure prediction and stochastic context-free grammars (SCFGs) are introduced. The problem of RNA secondary structure prediction is briefly motivated, and a number of key implementations of RNA secondary structure prediction are mentioned. Background material for SCFG-based RNA secondary structure prediction is given, which discusses the definition of SCFGs, normal forms, ambiguity and completeness, and some of the algorithms used in parameter inference and prediction. The main part of the chapter, though, is concerned with automated SCFG design for RNA secondary structure prediction, using an evolutionary algorithm to search the space of SCFGs. Much of the introductory material is taken from Anderson et al. (2012), entitled "Evolving Stochastic Context-Free Grammars for RNA Secondary Structure Prediction", published in BMC

Bioinformatics, and Anderson (2013), a book chapter entitled "Stochastic Context-Free Grammars and RNA Secondary Structure Prediction".

1.1 RNA Secondary Structure Prediction

Understanding the folding of RNA secondary structure is important to better understanding biological phenomena. "RNA can play key roles in gene expression, encouraging or discouraging expression of a given gene, or making post-transcriptional or -translational modifications. Structural flexibility can be important in roles like chaperoning (de Almeida Ribeiro et al. 2012), splice site recognition (Gahura et al. 2011), translation (Loh et al. 2012), and as riboswitches (Vitreschak et al. 2004). Naturally then, by predicting RNA secondary structure, we will have a better idea of the function of the RNA. With the advent of next-generation sequencing and transcriptomics, regulatory effects can be looked for by genome-wide prediction of RNA secondary structure (Washietl et al. 2005).

Similarly, we would like to understand RNA folding for RNA sequence design. If there is a specific RNA secondary structure we require in some cellular mechanism, for example in building bio-chemical structures like the DNA box (Andersen et al. 2009), we will need to know what sequences could be synthesised in a laboratory to fold into the required RNA structure. Computational applications like Frnakenstein (Lyngsø et al. 2012) use RNA folding programs to generate sequences likely to fold into these desired structures. RNA folding methods have also been used in RNA gene finding (Meyer 2007), as structural

components can be detected at the genomic level." (Anderson 2013).

Starting in the mid-1970s, attempts have been made to predict RNA secondary structure. Pipas & McMahon (1975) summed over all possible secondary structures with respect to free-energy functions, motivated by the fact that RNAs typically fold into thermodynamically stable structures. Nussinov et al. (1978) formulated the problem in terms of loop matchings, and predicted a secondary structure using dynamic programming. Since then, RNA secondary structure prediction methods have fallen roughly into two catagories: energy based thermodynamic models, and stochastic context-free grammar (SCFG) based probabilistics models.

1.1.1 Thermodynamic Models

The thermodynamic method assigns a free energy to each RNA secondary structure, motivated by bio-physical knowledge of thermodynamic stability. Although much of the method is beyond the scope of this thesis, it is useful to introduce, since implementations of the thermodynamic method like RNAfold (Hofacker et al. 1994) are used, particularly in Chapter 2.

"The thermodynamic method uses energy parameters, which must be gained experimentally (Mathews et al. 1999). The energy parameters describe the thermodynamic stability that different nucleotides will exhibit forming different structures. In particular, we must consider energy contributions from different combinations of base pairs. For example, if an AU base-pair is next to a GU base-pair, there will be a different energy contribution than if the

AU base-pair is next to another AU base-pair. These so-called stacking energies mean that a large number of parameters must be found, over a hundred, which is why many of the thermodynamic models are based on the same underlying energy models.

Once we have all of these energy parameters, [typically we find the secondary] structure with the minimum free-energy, through dynamic programming, and this is the structure predicted.

Since, for each secondary structure, we have an associated energy, this naturally leads to a Boltzmann probability distribution over secondary structures. To calculate the probability of a secondary structure S, we have

$$P(S) = \frac{1}{Q} e^{-F(S)/kT} \tag{1.1}$$

where $F(S)$ is the free-energy of S, T is the temperature, k a constant, and Q a normalising factor. To calculate Q, one needs to calculate the partition function (McCaskill 1990), and to find the free-energy of a given secondary structure, one simply sums over all the loops and stems, finding the free-energy of the smaller components, using the free-energy parameters determined experimentally. However, if we are only interested in calculating the minimum free-energy structure, this can be done without calculating the full partition function (Hofacker et al. 1994). Determining the partition function and minimum free-energy structure can be done in cubic time, the same complexity as the SCFG method predicts the maximum probability structure.

The main implementations of the thermodynamic method are UNAfold (Markham et al. 2008) and RNAfold, as part of the Vienna package (Hofacker et al. 1994). They are essentially the same method, but with slightly different energy models. They predict the minimum free-energy structure, in the same way that SCFGs can predict the maximum probability structure. Also like SCFG methods, they place a probability distribution on RNA secondary structures, so various different posterior decoding methods have been implemented to obtain slightly different prediction methods. Sfold (Ding et al. 2005) samples from the Boltzmann distribution generated by the free-energy model, and generates a consensus structure from the samples. Kinefold (Xayaphoummine et al. 2005) uses kinetic folding concepts in combination with the free-energy model to try and more closely echo biological folding mechanisms." (Anderson 2013).

1.1.2 Probabilistic Models

"The use of SCFGs in RNA secondary structure prediction was based on the success of Hidden Markov Models (HMMs) in protein and gene modelling (Krogh et al. 1994). Any attempt to apply HMMs to RNA secondary structure was prevented by the long–range interactions in RNA (Rabiner 1989). SCFGs, being generalisations of HMMs, offer a solution. This was first exploited by (Sakakibara et al. 1994) and then developed by others (e.g. (Lefebvre 1996))." (Anderson et al. 2012). The main implementation of SCFG-based methods have been in combination with an evolutionary model (Knudsen & Hein 1999, Knudsen & Hein 2003), using an alignment of homol-

ogous sequences instead of a single sequence. Comparative prediction models are discussed in more detail in Chapter 2.

The most simple SCFG prediction model for single sequences uses a context-free grammar, then predicts either the most likely structure, or the maximum expected accuracy (MEA) structure. Within this basic framework, several different SCFGs have been implemented for the RNA secondary structure prediction on a single sequence. Knudsen & Hein (1999) use a manually-designed SCFG, shown to be effective in comparison with other manually-designed SCFGs (Dowell & Eddy 2004) and automated-designed SCFGs (Anderson et al. 2012), demonstrated below. Attempts have also been made at creating SCFGs similar to the thermodynamic model (Nebel & Scheid 2011), and the recent TORNADO (Rivas et al. 2012) provides a comprehensive framework for SCFG-based modelling of RNA secondary structures.

There are a variety of methods which use SCFG methods outside this simple framework. These might define a probability distribution on RNA secondary structures, but then use an alternative method to decode, or augment the basic SCFG method with further probabilistic models. Doose & Metzler (2012) use a Bayesian sampling technique to sample helices from the posterior distribution generated by SCFGs. CONTRAfold (Do et al. 2006) generalises SCFGs using conditional log-linear models. Oxfold (Anderson et al. 2013*a*), described in Chapter 3, attempts to emulate folding kinetics to decode the posterior in a more biologically realistic way. There are clearly many interesting continuing avenues of research.

Although there are many similarities between the SCFG and thermodynamic methods, evaluating score functions over subsequences, there are some key differences. SCFGs, as learned probability distributions, can be tailored to specific RNA families (Nawrocki et al. 2009) and can capture features of folding captured by a thermodynamic model. However this also gives a bias to generating structures the SCFG has been trained from, which may be a disadvantage in artificial sequence design. Typically SCFGs require many fewer parameters, can predict non-canonical base-pairs, and combine easily with molecular evolution models. In single-single sequence (Anderson et al. 2012) and comparative (Gardner & Giegerich 2004) benchmarks, thermodynamic methods tend to slighly outperform SCFG-based methods.

1.2 Background on Stochastic Context-Free Grammars

In this section, we introduce background material on SCFGs, normal forms for context-free grammars, notions of ambiguity and completeness, and briefly discuss some algorithms for SCFGs. It is likely that much of this will already be familiar to the reader.

1.2.1 Stochastic Context-Free Grammars

"A context–free grammar G (henceforth abbreviated to "grammar") is a 4–tuple (N, V, P, S) consisting of a finite set N of non–terminal variables; a

finite set V of terminal variables that is disjoint from N; a finite set P of production rules; and a distinguished symbol $S \in N$ that is the start symbol. Each production rule replaces one non–terminal variable with a string of non–terminals and terminals.

One possible grammar, generating strings which may be interpreted as addition/multiplication expressions using only the number 1, may be represented thus:

$$S \to F + S | F$$

$$F \to 1 | (S) | F * F.$$

Note that each instance of S (standing for sum) generates a sum of $n \geq 1$ terms $(F + F + ...F)$, and each F (standing for factor) generates a 1, a product of terms $(F * F)$, or a whole expression within parentheses. It should be clear that many different expressions might be generated: 1+1+1, (1+1*1)+1+(1), and so forth. Formally, the grammar has non–terminal variables S, F, terminal variables $(,), +, *, 1$, production rules $S \to F + S, S \to F, F \to 1, F \to (S)$ and start symbol S. The order in which the production rules are used forms the *derivation* of a string. One valid derivation would be $S \Rightarrow F \Rightarrow (S) \Rightarrow (F) \Rightarrow (1)$, generating the string '(1)' and using the sequence of production rules $S \to F, F \to (S), S \to F, F \to 1$.

A SCFG is a grammar with an associated probability distribution over the production rules which start from each $T \in N$. Beginning with the start symbol and following production rules sampled from the relevant distribu-

tion, a string of terminal variables can be produced (if the grammar terminates). Choosing nucleotide symbols or the three characters used in dot–parenthesis notation as terminal variables, SCFGs can be constructed which produce strings corresponding to nucleotide sequences or secondary structures." (Anderson et al. 2012).

In the past, the "main SCFG used was the Knudsen and Hein grammar (Knudsen & Hein 1999, Knudsen & Hein 2003), hereafter denoted KH99. KH99 was a remarkable simple grammar:

$$\begin{array}{lcccc} S & \rightarrow & L & | & LS \\ & & .131 & & .869 \\ L & \rightarrow & s & | & dF\hat{d} \\ & & .895 & & .105 \\ F & \rightarrow & dF\hat{d} & | & LS \\ & & .788 & & .212 \end{array} \tag{1.2}$$

with d, $\hat{d}$, indicating paired nucleotides, and s indicating an unpaired nucleotide. The grammar generated stem and loop regions, and, although originally designed to be paired with an evolutionary model, performs well on single sequences too (Dowell & Eddy 2004, Anderson et al. 2012). Particularly, it has only 3 production rule parameters to be estimated from data, which means it is less likely to suffer from overfitting." (Anderson 2013).

1.2.2 Normal Forms

"To develop algorithms for analysing sequences under grammatical models, it is convenient to restrict the grammar to a normal form, with only a few possible types of productions. The normal form most commonly used is Chomsky Normal Form (CNF) (Chomsky 1956), as every context–free grammar is equivalent to one in CNF. However, a grammar in CNF cannot introduce the corresponding parentheses of paired nucleotides in a single production, and therefore does not capture structure in a straightforward manner. Thus [we use] double emission normal form (so called because paired bases are emitted simultaneously)[, created in (Anderson et al. 2012),] which is able to capture the fundamental features of RNA secondary structure: branching, unpaired bases, and paired bases. For any combination of non–terminals (T, U, V) we allow only rules of the following form:

$$T \to UV$$

$$T \to .$$

$$T \to (U)$$

This normal form allows the development of the structural motifs commonly found in RNA. For example (where V_i correspond to non–terminals) base–pair stacking can be generated by rules of the type $V_1 \to (V_1)$, hairpins by $V_1 \to (V_2)$, $V_2 \to V_2V_3|V_3V_3$ and $V_3 \to .$, and bulges by $V_1 \to (V_2)$, $V_2 \to V_3V_1$ and $V_3 \to .|V_3V_3$.

Furthermore, with the exception of the ability to generate empty strings,

this normal form allows the expression of dependencies of any context-free grammar producing valid structures." (Anderson et al. 2012).

"As a result of [only considering SCFGs in this normal form], many grammars already established in RNA secondary structure prediction cannot be exactly replicated, since they are not initially in the above normal form. For example, the KH99 contains the rule $S \to L$. As this normal form is an extension of CNF, any context-free grammar can be converted to this normal form, maintaining paired terminal symbol emissions. For example if $S \to L$ and $L \to .|(F)$, the forbidden rule $S \to L$ would be replaced by $S \to .|(F)$. This produces the same strings, and a given probability distribution for stochastic grammars can even be conserved. However, the transformation will often change the set of parameters in the model, which may result in different predictions when the production rule probabilities are inferred." (Anderson et al. 2012).

1.2.3 Ambiguity and Completeness

"A grammar is said to be *ambiguous* if it produces more than one derivation for a given structure (Reeder et al. 2005). If structure A has one derivation with probability 0.3, and structure B two derivations, each with probability 0.25, the [derivation with highest probability corresponds to] structure A, while structure B is more probable. This may reduce the quality of predictions [when finding the most probable structure]. The posterior decoding, though, sums over all derivations in prediction, so might be affected less by grammar ambiguity.

Nine grammars are tested in (Dowell & Eddy 2004), of which two are ambiguous. They find that [finding the most probable derivations does result in choosing] suboptimal structures, and that the ambiguous grammars perform poorly relative to the unambiguous grammars. Consequently, efforts have been made to avoid ambiguity (Reeder et al. 2005). The conclusion that ambiguous grammars are undesirable is not necessarily justified. The ambiguous grammars in Dowell & Eddy (2004) are small, with at most two non–terminal variables, and one might expect them to be ineffective regardless. Their poor predictive quality may be due to deficiency in design rather than ambiguity.

We define a grammar to be *complete* if it has a derivation for all possible structures which have no hairpins shorter than length two. Clearly the ability to generate all structures is desirable for a grammar. However, one should be careful not to overemphasise this desirability, even for a complete grammar, once parameters have been inferred, converting it to a SCFG, it is unlikely that all structures have a probability significantly different from zero for any sequence. Similarly, the posterior decoding is not prone to grammar incompletness in the same way that [considering the most probable structure] is, since, in theory, after obtaining the probabilities of unpaired and paired positions, structures which cannot be derived with the grammar can still be predicted. One can perform a heuristic test for completeness by testing on a sample of strings. Ambiguity can also be checked for heuristically (Reeder et al. 2005), but determining grammar ambiguity and completeness is undecidable (Hopcroft & Ullman 1979).

Practically, it is very difficult to ensure both unambiguity and completeness. A complete, unambiguous grammar cannot be simply modified without compromising one of the properties. Adding any production rules (if they are ever used) will create ambiguity by providing additional derivations. Equally, removing production rules will create incompleteness (unless the rule is never used in a derivation), as the original grammar is assumed unambiguous." (Anderson et al. 2012).

1.2.4 Algorithms for SCFGs

"We now move on to some of the main algorithms used for SCFG prediction. Throughout, we assume that SCFGs are in double emission normal form. There are two main sets of algorithms which are used for RNA secondary structure prediction with SCFGs. I will first introduce the algorithms, and then discuss how each one can be applied.

The Cocke-Younger-Kasimi (CYK) algorithm (Younger, 1967) was originally used to determine whether or not a context-free grammar could derive a given string. However, it is easily adapted to be useful for SCFGs, and determines the maximum probability of a derivation of the SCFG for a given string. It is a dynamic programming algorithm, so backtracking through the dynamic programming table will give the SCFG derivation which produced this maximum probability. This derivation can then be used to find the structure with the maximum probability derivation.

For each non-terminal U, string s, and string indices i and j, we define

$$C(U,i,j) = \begin{cases} \mathrm{P}\left[U \to .\right]\ \mathrm{P}[s[i] = .] & \text{if i = j} \\ \max\left\{ \max\limits_{U \to VW} \max\limits_{i \le k < j} \mathrm{P}[U \to VW]\ C(V,i,k)\ C(W,k+1,j), \right. & \\ \left. \max\limits_{U \to (V)} \mathrm{P}[U \to (V)]\ C(V,i+1,j-1)\ \mathrm{P}[s[i],s[j] = ()] \right\} & \text{if i < j} \\ 0 & \text{otherwise} \end{cases}$$

$C(U,i,j)$ is then the maximum probability of U generating the subsequence $s[i]...s[j]$, and in particular, $C(S,1,|s|)$ gives the maximum probability of the SCFG generating the sequence. To compute this table, start with subsequences of length one, then length two, and so on, until you have computed the CYK table for the entire string. Pseudocode for the algorithm can be found in Algorithm 1. As can be seen in the pseudocode, we can find $C(S,1,|s|)$ in time $O(|P||s|^3)$ and space $O(|V||s|^2)$, as we compute the $C(U,i,j)$ values in order of sub-sequences of s of increasing length.

Algorithm 1 CYK algorithm

for $i = 1$ **to** $|s|$ **do**
  for $U \in$ V **do**
    $C(U,i,i) = \mathrm{P}\left[U \to .\right]\ \mathrm{P}[s[i] = .]$
for $l = 1$ **to** $|s| - 1$ **do**
  for $i = 1$ **to** $|s| - l$ **do**
    for $U \in$ V **do**
      $$C(U,i,i{+}l) = \max\left\{ \max_{U \to VW} \max_{i \le k < l} \mathrm{P}[U \to VW]\ C(V,i,k)\ C(W,k+1,l), \right.$$
      $$\left. \max_{U \to (V)} \mathrm{P}[U \to (V)]\ C(V,i+1,l-1)\ \mathrm{P}[s[i],s[l] = ()] \right\}$$

The Inside and Outside algorithms (Lari and Young, 1990) are similar to the CYK algorithm, but instead of calculating the maximum probability of a derivation, they calculate the total probability of generating subsequences. First we define the Inside table, for each non-terminal U, string s, and string indices i and j:

$$I(U,i,j) = \begin{cases} \mathrm{P}\left[U \to .\right]\ \mathrm{P}[s[i] = .] & \text{if i = j} \\ \sum\limits_{U \to VW} \sum\limits_{i \le k < j} \mathrm{P}[U \to VW]\ I(V,i,k)\ I(W,k+1,j) & \\ \quad + \sum\limits_{U \to (V)} \mathrm{P}[U \to (V)]\ I(V,i+1,j-1)\ \mathrm{P}[s[i],s[j] = ()] & \text{if i < j} \\ 0 & \text{otherwise} \end{cases}$$

and similarly, the Outside table:

$$O(U,i,j) = \begin{cases} 1 & \text{if U = S,} \\ & i = 1,\ j = |s| \\ \sum\limits_{V \to UW} \sum\limits_{k > j} \mathrm{P}[V \to UW]\ O(V,i,k)\ I(W,j+1,k) & \\ \quad + \sum\limits_{V \to WU} \sum\limits_{k < i} \mathrm{P}[V \to WU]\ O(V,k,j)\ I(W,k,i-1) & \\ \quad + \sum\limits_{V \to (U)} \mathrm{P}[V \to (U)]\ O(V,i-1,j+1)\ \mathrm{P}[s[i-1],s[j+1] = ()] & \text{if i < j} \\ 0 & \text{otherwise} \end{cases}$$

$I(U, i, j)$ then gives the total probability of U generating the subsequence $s[i]...s[j]$, and $O(U, i, j)$ the total probability of generating everything except the subsequence $s[i]...s[j]$. Hopefully, considering the derivation trees required to produce the subsequence $s[i]...s[j]$, it is intuitive why the tables are named 'Inside' and 'Outside'. The pseudocode for the Inside algorithm is analogous to the CYK algorithm pseudocode. The Outside is computed in a similar manner, but instead starting from subsequences of length $|s|$, and working down to subsequences of length one. Again, the Inside and Outside algorithms are computed in time $O(|P||s|^3)$ and space $O(|V||s|^2)$." (Anderson 2013).

1.2.5 Parameter Inference

"For parameter estimation, we require a set of RNA sequences with known secondary structures, that is, in the language of machine learning, we use supervised training. Interestingly, if you try to train the SCFG in an unsupervised manner, that is just using RNA sequences, the SCFG will choose parameters which enable it to produce long strings at low 'cost'. To illustrate this, consider the start variable for KH99. The probabilities for the production rules $S \rightarrow LS$ and $S \rightarrow L$ given are 0.869 and 0.131 respectively (Knudsen & Hein 1999). However, consider also a SCFG with rules $S \rightarrow LS$, $S \rightarrow L$ and $L \rightarrow .$, with rule probabilities 0.99, 0.01, and 1. This latter SCFG will generate strings solely of unpaired bases with very high probability- much higher than KH99 does- and this is, in fact, the sort of SCFG you get when using unsupervised training, as you are looking to max-

imise the probability of the data. For this reason, we need known secondary structures for parameter estimation.

We need also the nucleotide probabilities for SCFG parameter estimation. Considering only single sequences, we simply take the frequency at which nucleotides appear in paired and unpaired positions and use these frequencies as probabilities. For example, if we see 100 AU base-pairs out of 500 total base-pairs, then the probability of seeing an AU base-pair would be 1/5.

Now that we have the nucleotide probabilities and our SCFG algorithms, we are interested in determining the production rule probabilities which will be useful in producing the most accurate structure predictions. We can do this parameter estimation using either the CYK algorithm, or by using the Inside and Outside algorithms.

For parameter estimation with the CYK algorithm, we first consider unambiguous grammars. Recall that if an unambiguous grammar can derive a string, then it will only have one way of doing so. So here, we use the CYK algorithm instead to generate the known structure, initially using uniform rule probabilities, and an indicator function in the algorithm to determine whether or not the algorithm is generating the known structure. Thus, by keeping track of the rule frequencies used in the derivation to generate this known structure, we will be able to count the total number of rules used over the whole training data set. Again, we just take the frequencies of rules observed. Taking KH99 as an example, if we saw 200 $S \to LS$ rules used, and 100 $S \to L$ rules used, our rule probabilities would be 2/3 and 1/3 respectively.

For ambiguous grammars, we no longer have a single derivation for each structure. So, if we want to estimate parameters on the data set, we must choose at random a single derivation. This is done in the backtracking step of the CYK algorithm: if there is not a single production rule which generates a given subsequence, one is chosen at random. Of course, implementing this does not require knowing whether or not a SCFG is ambiguous a priori, since picking randomly will still be correct for unambiguous grammars. Again, by keeping track of production rules, you can determine production rule probabilities.

For the Inside and Outside algorithms, we used Inside-Outside training (Lari & Young 1990). This is an implementation of expectation-maximisation training (Dempster et al. 1977). First we randomly initialise the rule probabilities, then use these to calculate the expected number of rule uses. From the expected number of rule uses, we re-estimate the rule probabilities. This process is then iterated until convergence of rule probabilities. The Inside and Outside algorithms can be used to calculate the expected rule frequencies." (Anderson 2013).

"There are good reasons why one might want to use one variety of training over the other. The CYK training method is significantly quicker: not only does the algorithm take less time (practically, not in complexity) than Inside-Outside, but there is only a single iteration required. Thus, for very large SCFGs, or if many are being trained, this may be of benefit. On the other hand, Inside-Outside training deals with ambiguous SCFGs much better, considering all derivations they can make instead of a random one. Similarly,

Inside-Outside training is more focused on getting the overall probability distribution correct, as opposed to the simply a maximum-likelihood style estimator the CYK algorithm uses.

We also might wonder how stable the parameter estimation procedure is, that is, if we choose slightly different parameters, might the SCFG predict entirely different structures? This was in particular an important question when considering designing grammars with an automated approach (Anderson et al. 2012). When the data set for parameter estimation was made to be a random subset of the whole data set, the stability of parameterisation could be tested. In fact, with only a handful of sequences, the parameters of small SCFGs varied very little, but with slightly larger SCFGs, the variation was more- not surprising, given there are more production rules to estimate. With a data set of around 100 sequences, though, the variation was very small. This idea has been further confirmed by looking at disturbances in the sampling probabilities of a heavyweight SCFG, and its corresponding prediction accuracy, and finding that the structures produced do not vary particularly (Scheid & Nebel 2012). Thus, we can be reasonably confident when estimating SCFG probabilities that we will have reached a reasonable parameterisation." (Anderson 2013).

1.2.6 Secondary Structure Prediction

"Now we have our SCFG with a set of parameters found from known structures, we want to begin predicting structures for other sequences. Again, the CYK, Inside, and Outside algorithms are used in a similar way to the

parameter estimation.

Prediction with the CYK algorithm is straightforward. Firstly, we fill in the CYK table using the algorithm above. The value $C(S, 1, |s|)$ gives the maximum probability of the SCFG generating the sequence, and it is the derivation corresponding to this probability which we use for the predicted structure. This derivation, as with the parameter estimation, is found by backtracking through the CYK table. We could also use a modified version CYK algorithm to determine the n-most probable structures (Chappelier & Rajman 1998).

Prediction with the Inside and Outside algorithms is a little more difficult. From the Inside and Outside tables, one can derive the base-pair probability matrix $B(i, j)$, the probability of i pairing with j:

$$\mathrm{B}(i,j) = \frac{\sum_{U \to (V)} O(U,i,j)\mathrm{P}[U \to (V)]\mathrm{P}[s[i], s[j] = ()]I(V, i+1, j-1)}{I(S, 1, |s|)} \quad (1.3)$$

that is, by summing over all rules of type $U \to (V)$.

Once we have the base-pair probability matrix, we have a probability distribution over secondary structures, and we can do posterior decoding to obtain our predicted structure. At this point we must decide what we are looking to maximise. Typically, expected positions correctly predicted or expected base-pairs correctly predicted are chosen as the criterion to maximise over. For example, if we are looking to maximise the expected number of positions

correctly predicted, we consider

$$E(i,j) = \max \begin{cases} \frac{1}{2}\left(1 - \sum\limits_{1 \le k \le |s|} B(i,k)\right) + E(i+1,j) & (1) \\ B(i,j) + E(i+1,j-1) & (2) \\ \max\limits_{i+1<k<j} \left[B(i,k) + E(i+1,k-1) + E(k+1,j)\right] & (3) \end{cases} \tag{1.4}$$

$E(i,j)$ is the maximum expected number of positions correctly predicted for the subsequence $i...j$. Case (1) considers this maximum having been achieved from i being unpaired, case (2) considers this maximum having been achieved from i being paired with j, and case (3) considers this maximum having been achieved from i being paired with some k in between i and j. $E(1,|s|)$ is then the maximum expected number of positions correctly predicted for the entire sequence, and the structure which produces this maximum, found by backtracking through the $E(i,j)$ table, is the structure predicted.

There are some key differences between the CYK and Inside-Outside methods of prediction. The CYK algorithm, predicting structures with the highest probability, predicts structures with a larger proportion of base-pairs. This is in some sense unsurprising when considering the double emission normal form, as forming a base-pair uses only a single rule for creating two positions. Hence the CYK algorithm can predict the structure 'quicker', which typically comes with a higher probability, and so the sensitivity of CYK predictions tends to be relatively higher. Inside-Outside, on the other hand, will consider probability contributions from all derivations. Therefore, struc-

tures predicted by the Inside-Outside algorithm tend to have relatively higher PPV, only predicting base-pairs where the majority of derivations (weighted by probability mass) 'agree' on a base-pair being present. In particular, when considering predictions in the context of comparative modeling, this difference can be more pronounced." (Anderson 2013).

Pseudoknots

"However, one key feature of the basic SCFG method is the inability to predict pseudoknotted structures (Brown & Wilson 1996). A pseudoknot is an RNA secondary structure containing at least two stem-loop structures in which half of one stem is intercalated between the two halves of another stem, for example the structure ((((...[[[...))))...]]]. RNA structures with pseudoknots in them are highly common (Andronescu et al. 2008). They occur in virtually all classes of RNA, and have important biological function (Chen & Greider 2005)." (Anderson 2013).

We can still predict pseudoknots using a grammatical framework. Rivas & Eddy (2000), for example, use a grammar (but not a context-free grammar) that corresponds to the pseudoknot folding algorithm in Rivas & Eddy (1999). Additionally, methods such as Oxfold (Anderson et al. 2013*a*), which use the probability distribution generated by a SCFG, but not the SCFG algorithms themselves, can easily be adapted to predict pseudoknots, as in Doose & Metzler (2012). As SCFGs cannot derive pseudoknotted structures, this thesis only considers non-pseudoknotted structures.

Accuracy Metrics

Throughout this thesis metrics will be used to evaluate the accuracy of RNA secondary structure predictions, and alignment quality. "There are a wealth of available metrics on RNA secondary structure (Freyhult et al. 2005, Gardner & Giegerich 2004). Here we use sensitivity, positive predicted value (PPV), and F-score (the harmonic mean of sensitivity and PPV). Defining true positives (TP) as the number of base-pairs correctly predicted, false positives (FP) as the number of true base-pairs not predicted, and false negatives (FN) as the number of base-pairs predicted which are incorrect, we have

$$\begin{aligned} \text{Sensitivity} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{PPV} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{F-score} &= \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FN} + \text{FP}} \end{aligned}$$

The strength of these RNA secondary structure accuracy metrics is that they are easy to interpret, and make it straightforward to compare methods across different datasets. An F-score of 1 would represent a structure prediction that was completely correct, and an F-score of 0 a structure prediction that only predicted incorrect base-pairs."(Anderson et al. 2013*c*)

"To analyse variances of RNA secondary structure as alignment quality varies, we calculate a similarity score that measures how close a sample alignment is to the reference alignment. We use an alignment metric, taken from (Schwartz et al. 2004), which is generalised to an alignment method in

(Schwartz & Pachter 2007).

Let a_{s_1,s_2} be an alignment of a sequence s_1 of length n to a sequence s_2 of length m. Each column of a_{s_1,s_2} can be expressed as pairs of the form (s_1^i, s_2^j), $(s_1^i, -)$, and $(-, s_2^j)$. We define

$$\begin{aligned} H(a_{s_1,s_2}) &= \{(i,j)|(s_1^i, s_2^j) \in a_{s_1,s_2}\}, \\ I(a_{s_1,s_2}) &= \{j|(-, s_2^j) \in a_{s_1,s_2}\}, \text{ and} \\ D(a_{s_1,s_2}) &= \{i|(s_1^i, -) \in a_{s_1,s_2}\}, \end{aligned}$$

sets which represent 'homology', 'insertion' and 'deletion' respectively. Given these sets, we define the distance between two alignments $a^k_{s_1,s_2}$ and $a^\ell_{s_1,s_2}$ of two sequences s_1 and s_2 to be

$$\begin{aligned} d(a^k_{s_1,s_2}, a^\ell_{s_1,s_2}) &= n + m - 2|H(a^k_{s_1,s_2}) \cap H(a^\ell_{s_1,s_2})| \\ &\quad -|D(a^k_{s_1,s_2}) \cap D(a^\ell_{s_1,s_2})| \\ &\quad -|I(a^k_{s_1,s_2}) \cap I(a^\ell_{s_1,s_2})|. \end{aligned} \tag{1.5}$$

For example, consider the case $a^k_{s_1,s_2} = a^\ell_{s_1,s_2}$. Then we have $n = m$, $|H(a^k_{s_1,s_2}) \cap H(a^\ell_{s_1,s_2})| = n$, $|D(a^k_{s_1,s_2}) \cap D(a^\ell_{s_1,s_2})| = 0$ as there are no deletions, and $|I(a^k_{s_1,s_2}) \cap I(a^\ell_{s_1,s_2})| = 0$ as there are no insertions. This gives the distance between the alignments as zero, as would be expected.

Equation 1.5 can then be generalised to sequence alignments with more than two sequences. Assuming now that a^k and a^ℓ are alignments of m sequences

$s_1, \ldots, s_m$ of lengths $n_1, \ldots, n_m$, we have

$$d(a^k, a^\ell) = \sum_{p=1}^{m-1} \sum_{q=p+1}^{m} d(a^k_{s_p,s_q}, a^\ell_{s_p,s_q}), \tag{1.6}$$

that is, summing all the pairwise alignment distances from Equation 1.5. This alignment metric is then normalised and subtracted from 1 to produce a similarity score

$$SS(a^k, a^\ell) = 1 - \frac{d(a^k, a^\ell)}{(m-1)\sum_{t=1}^{m} n_t} \tag{1.7}$$

The denominator of the fraction, $(m-1)\sum_{t=1}^{m} n_t$, is the normalising constant, the maximum that the alignment distance $d(a^k, a^\ell)$ can be. The similarity score is bounded by 0 and 1, with 1 indicating that the sample alignment is identical to the reference alignment."(Anderson et al. 2013*c*)

1.3 Evolving Stochastic Context-Free Grammars for RNA Secondary Structure Prediction

We now move on to considering an evolutionary algorithm to search the space of SCFGs, to try and optimise RNA secondary structure prediction with respect to the SCFG used. Much of this work is taken from Anderson et al. (2012), entitled "Evolving Stochastic Context-Free Grammars for RNA Secondary Structure Prediction", published in BMC Bioinformatics.

"While KH99 was effective, it seems to have been chosen relatively arbitrarily, in that there is little discussion about the motivation behind the choice of production rules. This problem was addressed by (Dowell & Eddy 2004), in which nine different lightweight SCFGs were evaluated on a benchmark set of RNA secondary structures. The set of grammars they tested, however, was by no means exhaustive. All of these grammars were constructed by hand and there was little motivation for their production rules (except for the extension to stacking grammars: production rules such as $P^{b\hat{b}} \rightarrow aP^{a\hat{a}}\hat{a}$ for nucleotides a,$\hat{a}$,b,$\hat{b}$). This suggested that a computational search of a large space of grammars might find stronger grammars, which is what [was attempted in (Anderson et al. 2012)].

Evolutionary approaches have already been implemented for HMMs. Indeed, Won et al. (2007) used one to find the best HMM for protein secondary structure prediction. A mild improvement was found compared to HMMs which had been constructed by hand. It is hard to tell quite how conclusive the results were since the limited size of the data set forced training and testing to be done on the same data. Given the size of the data set, overtraining may have caused unreasonably high quality predictions. Clearly though, the method is potentially very powerful." (Anderson et al. 2012).

In Anderson et al. (2012), "only *structure generating grammars* are considered (i.e. those which have terminal variables in $\{(,), .\}$). Strings generated by these grammars uniquely define secondary structures, with a dot '.' representing an unpaired nucleotide and an opening parenthesis indicating pairing with its corresponding closing parenthesis. Once a structure is generated, a

generative model for sequences is to stochastically allocate nucleotides to each site according to the frequency of occurrence (paired and unpaired) in some set of trusted sequences and structures." (Anderson et al. 2012).

1.3.1 An Evolutionary Algorithm for Grammars

"With the double emission normal form, for m non–terminal variables there are $2^{m^3+m^2+m}$ grammars (m^3 production rules of type $T \to UV$, m^2 of type $T \to (U)$ and m of type $T \to .$). An evolutionary algorithm would allow for efficient exploration of the space of grammars in the above normal form. The way that the algorithm searches the space is determined by the design of the initial population, mutation, breeding and selection procedure. To find effective grammars, these must be well designed. " (Anderson et al. 2012).

Initial population

"When forming the initial population, the size of the space of grammars quickly becomes problematic. The space is clearly large, even for small m, so the population size cannot approach that usually afforded in evolutionary algorithms (Grefenstette 1986). We start with an initial population of small grammars, and use mutation and breeding rules which grow the number of non–terminal variables and production rules. Our initial population

comprised sixteen grammars, of the form:

$$S \rightarrow \left\{ \begin{array}{c|c|c|c|c|c} SS & SB & BS & BB & (S) & \\ - & - & - & - & & \end{array} \right. .$$

$$B \rightarrow .$$

where between zero and four of the $S \rightarrow UV$ rules were excluded. We also tried initial populations containing the SCFGs from (Dowell & Eddy 2004) to consider examining SCFGs close to these." (Anderson et al. 2012).

Mutation

"Mutations constitute the majority of movement through the search space, so are particularly important. They give the grammar new characteristics, allow it more structural freedom, and add production rules which may be used immediately or may lie dormant. For non–terminals $V_i \in N$, and corresponding production rules P_{V_i}, the allowed stochastic mutations were:

- The start variable (and corresponding production rules) change,
- A production rule is added or deleted,
- A new non–terminal variable V' is added along with two new rules that ensure that V' is reachable and that $P_{V'}$ is not empty,
- A non–terminal variable is created with identical rules to a pre–existing one,
- A production rule of the form $V_i \rightarrow V_j V_k$ is changed to $V_i \rightarrow V_j V_l$,

$V_i \to V_l V_k$ or $V_i \to V_l V_p$, or production rule of the form $V_i \to (V_j)$ is changed to $V_i \to (V_k)$.

This form of mutation is very basic, but allows many structural features to develop over generations. The rate of mutation determines movement speed through the search space and development of these structural features. Adding rules too slowly prevents grammars from developing structure, while too many results in a lot of ambiguity and thus creates ineffective grammars. Deleting rules almost always results in a worse grammar. To aid the grammar design, especially in consideration of facets of the normal form, the rule $B \to .$ was kept constant in the evolutionary process.

More complex mutation is clearly possible. The derivation could be used to find the rules used more often and make mutations of those rules more or less likely. A model for simultaneous mutations could be developed, which might be able to make use of expert understanding of RNA structure, in combination with an evolutionary search. We have found the above model to give sufficient mobility in the search space, and therefore did not investigate other extensions." (Anderson et al. 2012).

Breeding

"The breeding model forms a grammar which can produce all derivations of its parent grammars. The grammar G formed from breeding G_1 and G_2 has start symbol S, non–terminals V_1, V_2, ..., V_n and W_1, W_2, ..., W_m, B, terminals $., (,)$ and production rules

- $P_S = P_{S_1} \cup P_{S_2}$,
- For V_i: P_{V_i} where all occurrences of S_1 are replaced with S,
- For W_i: P_{W_i} where all occurrences of S_2 are replaced with S.

This breeding model was chosen to keep the size of the grammar relatively small, whilst allowing expression of both bred grammars to be present in derivations." (Anderson et al. 2012).

Selection

"We grow the population in each generation by introducing a number of newly mutated or bred grammars, then we pare it back to a fixed population size by stochastic elimination. We determine the probability of elimination of a grammar by the inverse of some fitness measure. Fitness functions we use include mountain metric distances (Moulton et al. 2000) between the predicted structure and the trusted structure, sensitivity (True Positives/(True Positives + False Negatives)) and positive predictive value (PPV) (True Positives/(True Positives + False Positives)). We follow the definition of accuracy in (Dowell & Eddy 2004), where a base pair constitutes a true positive if it is present in both the predicted and true structures, false positive if present in only the predicted structure, and false negative if present in only the trusted structure. We also tried these accuracy measures in combination with other factors in the fitness function, including largest correctly predicted length, inability to predict structures, and cost for grammar complexity." (Anderson et al. 2012).

1.3.2 Data

"We took data from RNASTRAND (Andronescu et al. 2008), a collection of other databases (Andersen et al. 2006, Westbrook et al. 2003, Brown 1999, Cannone et al. 2002, Griffiths-Jones et al. 2005, Sprinzl et al. 1998, Westbrook et al. 2003). We filtered the data set so that the sequences and structures could ensure reliability of predictions. We removed identical sequences and disregarded synthetic data and sequences with ambiguous base pairs. We further cleaned the data to filter out any sequences with greater than 80% base pair similarity with another structure (the standard used in Dowell & Eddy (2004)). Furthermore, we removed all sequences with pseudoknots as it is well established that SCFGs cannot predict pseudoknots (Brown & Wilson 1996).

The spectrum of sequence length, is of particular significance in selecting data. The CYK and training algorithms are of cubic order in the length of the string, so we decided to use large training and test sets with small strings. Longer strings require longer derivations, thus they have a larger weight in the parameter training, which might lead to overtraining. Equally, if one omits longer strings, poorer predictions may result from overtraining on the shorter strings. We found the trained parameters highly sensitive to the choice of training data set, and struggled to balance this with computational efficiency.

We used a final data set from a variety of families, consisting of 369 sequences with corresponding structures. There was a total of 57,225 nucleotides with

12,126 base pairs. As with Dowell & Eddy (2004) and Knudsen & Hein (1999,2003), sensitivity and PPV were used as the measures of structure fit, as well as F–score. . . , a more balanced measure of sensitivity and PPV. We investigated the practice of dividing data into training and test sets. (Lari & Young 1990) randomly generate training sets of 200 and 400 strings for parameter training. Dowell & Eddy (2004) use a training set of 278 structures and test set of 403 total structures. Meanwhile, Knudsen & Hein (1999) used a considerably larger training set, with 2273 structures, and a much smaller test set of just four sequences. Our training and test sets used in the evolutionary algorithm were disjoint sets of size 109, chosen at random from those sequences in our data of length less than 200 nucleotides. The remaining data form our evaluation set, on which grammars' sensitivities and PPVs are reported.

As well as measuring performance on our own data, we have used results obtained with the Dowell & Eddy (2004) data. This helped us to verify the performances of the new grammars, to check our results against earlier work, and to verify that KH99′ is a good representation of KH99." (Anderson et al. 2012).

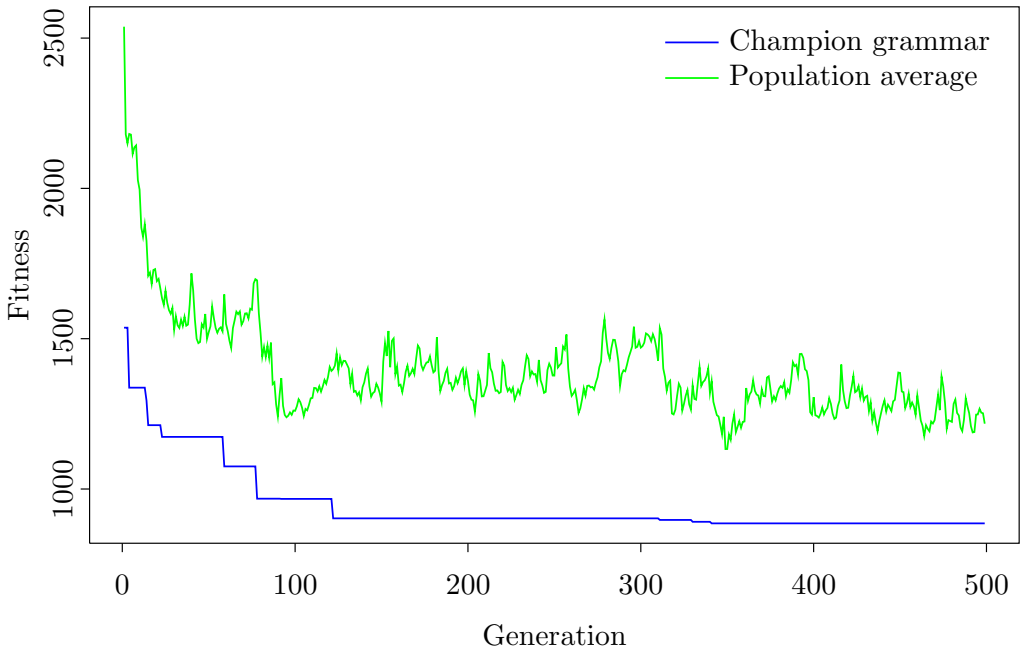


Figure 1.1: The change over generations in average fitness of population, and the fitness of the best SCFG. Here, a lower fitness is more desirable, the SCFG predicting better secondary structure. Many improvements to both the whole population and best SCFG are made in the first 100 or so generations. After this, the best SCFG does not become much better, but the average population fitness continues to fluctuate. Clearly the algorithm continues to explore alternative SCFGs and tries to escape the local optimum. Figure and caption taken from Figure 1, Anderson et al. (2012).

$$
\begin{array}{lll}
 & A \rightarrow BA|.|(C) \\
KH99' & B \rightarrow .|(C) \\
 & C \rightarrow BA|(C) \\
 & \\
 & A \rightarrow AA|BA|.|(A)|(C) \\
GG1 & B \rightarrow .|(C) \\
 & C \rightarrow BA|(C) \\
 & \\
 & A \rightarrow AA|AB|BA|BB|CB|BC|.|(B)|(C) \\
GG2 & B \rightarrow . \\
 & C \rightarrow AA|AB|BA|BB|BC|CA|CB|.|(A)|(B)|(C) \\
 & \\
 & A \rightarrow AB|BA|BB|AA|DD|(A)|(B)|(C)|(D) \\
 & B \rightarrow . \\
GG3 & C \rightarrow AA|.|(D) \\
 & D \rightarrow CD|BD|(A)|(C) \\
 & \\
 & A \rightarrow CC|CB|BC|EC|(A)|(E) \\
 & B \rightarrow . \\
 & C \rightarrow CB|BB|(A) \\
GG4 & D \rightarrow GC|(C) \\
 & E \rightarrow AB|CD|. \\
 & F \rightarrow AB \\
 & G \rightarrow FB \\
 & \\
 & A \rightarrow DA|CC|.|(B) \\
 & B \rightarrow . \\
 & C \rightarrow AA|HF|(G) \\
 & D \rightarrow .|(E) \\
GG5 & E \rightarrow (F) \\
 & F \rightarrow FB|BF|AA|.|(A)|(F) \\
 & G \rightarrow (E) \\
 & H \rightarrow BG \\
 & \\
 & A \rightarrow DE|AB|BA|AH|.|(F)|(H) \\
 & B \rightarrow . \\
 & C \rightarrow (H) \\
 & D \rightarrow BB|AC \\
GG6 & E \rightarrow .|(H) \\
 & F \rightarrow FB|CF|. \\
 & G \rightarrow GH|(H)|(C) \\
 & H \rightarrow FA|AF|HH|(B)|(H)
\end{array}
$$

Figure 1.2: A sample of the good grammars found in the evolutionary search. Taken from Anderson et al. (2012).

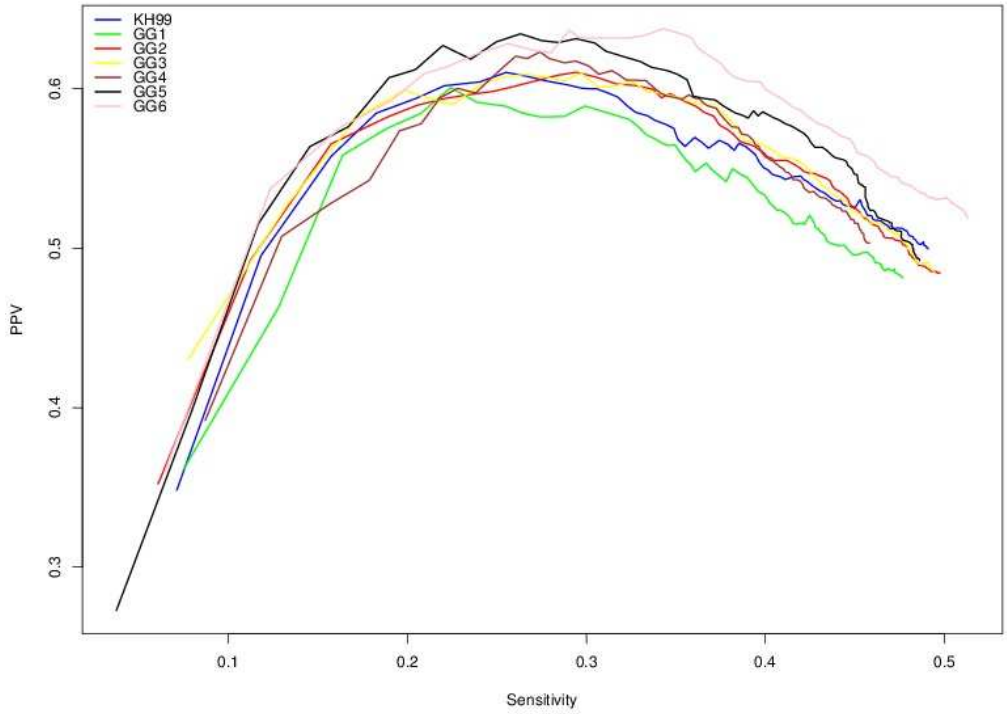


Figure 1.3: A graph showing how sensitivity and PPV change for grammars when the posterior decoding parameter γ is changed to alter the frequency of basepair prediction. The parameter γ ranged from 0.05 to 5 in increments of 0.05 for each grammar, and the sensitivity and PPV were measured on the full evaluation set. Grammar GG6 shows further strong performance here, having the largest area underneath the graph. Figure and caption taken from Figure 2, Anderson et al. (2012).

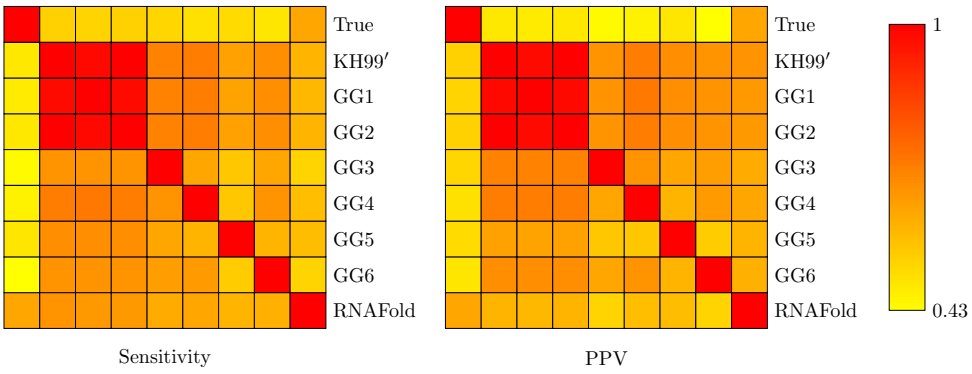


Figure 1.4: A heat map showing the relative sensitivities and PPVs of the different prediction methods, or between prediction method and known structure. KH99′ and GG1, produce very similar structures which is not surprising, given they were found by changing one and two rules of KH99′ respectively. Otherwise, the methods have relative sensitivities/PPVs of approximately 0.5–0.6, which is as expected, given they are all designed to predict RNA secondary structure. However, it is clear that they are markedly distinct from KH99′ in their structure predictions. Figure and caption taken from Figure 3, Anderson et al. (2012).

Grammar	KH99′	GG1	GG2	GG3	GG4	GG5	GG6	KH99	UNAfold	RNAfold	Pfold
Our data											
Sensitivity	0.496	0.505	0.408	0.413	0.474	0.469	**0.526**				
PPV	0.479	0.481	**0.551**	0.550	0.454	0.467	0.479				
F–score	0.478	0.441	0.473	0.470	0.461	0.339	**0.488**				
DE data											
Sensitivity	0.465	0.466	0.372	0.379	0.408	**0.487**	0.465	0.47	0.558	0.558	0.39
PPV	0.406	0.405	0.643	**0.646**	0.344	0.432	0.376	0.45	0.501	0.495	0.69
F–score	**0.480**	0.468	0.466	0.472	0.430	0.479	0.451				

Table 1.1: The sensitivities, PPVs, and F–scores of grammars GG1–GG6 and KH99′ on the evaluation set and on the on the Dowell and Eddy (2004) data, as well as the benchmarked sensitivities and PPVs of UNAfoldv3.8 and RNAfoldv1.85 on the Dowell and Eddy (2004) data. Results for KH99 and Pfold are taken from Dowell and Eddy (2004). Table and caption taken from Table 1, Anderson et al. (2012).

	Grammar	KH99′	GG1	GG2	GG3	GG4	GG5	GG6	Best
	Grammar found by		Local	IO	IO	CYK	CYK	CYK	
CYK	Sensitivity	0.496	0.505	0.330	0.374	0.474	0.469	**0.526**	0.675
	PPV	0.479	**0.481**	0.258	0.322	0.454	0.467	0.479	0.585
	F–score	**0.478**	0.441	0.426	0.435	0.461	0.339	0.461	0.622
IO	Sensitivity	0.387	0.392	0.408	**0.413**	0.373	0.404	0.410	0.450
	PPV	0.552	0.517	0.551	0.550	0.566	0.556	**0.583**	0.584
	F–score	0.461	0.443	0.473	0.470	0.449	0.471	**0.488**	0.493

Table 1.2: The sensitivities, PPVs, and F–scores of grammars GG1–GG6 and KH99′ on the evaluation set, using different methods of training and testing. ‘CYK’ indicates that the CYK algorithm was used, and ‘IO’ that the Inside and Outside algorithms were used. The column ‘Best’ was calculated by selecting, for each structure, the prediction with the highest F–score, and then recording the sensitivity, PPV, and F–score for that prediction. It is perhaps not surprising that the ‘best’ predictions for CYK are better than the ‘best’ predictions for IO, as IO is in some sense averaging over all predictions. One might expect the predictions to be more similar than those from CYK, as seen by comparing IO values for GG6 and ‘best’, giving less increase when considering those with best F–score. Table and caption taken from Table 2, Anderson et al. (2012).

1.3.3 Results and Discussion

"Figure 1.1 shows a typical realisation of the search process. The average fitness (here, smaller fitness is desirable) of the population is shown, as well as the fitness of the best recorded (champion) grammar. The average fitness of the population falls consistently as stronger grammars are found for approximately 100 generations, and then only minor improvements to the champion grammar were found. However, the population fitness continues to fluctuate as areas around the local optimum are searched.

Across all our experiments, over 300,000 grammars were searched. A number of strong grammars were found using both CYK and IO training and testing, denoted GG1–GG6. KH99′ is KH99 in the double emission normal form. [A sample of the best grammars found can be seen in Figure 1.2, and results] on the sensitivity, PPV, and F–score of each grammar can be found in Table 1.1, in addition to the benchmark with the Dowell & Eddy (2004) data, and results on different training and testing methods can be found in Table 1.2. Table 1.2 also gives the scores of the combined best prediction, calculated by selecting, for each structure, the prediction with the highest F–score, and then recording the sensitivity, PPV, and F–score for that prediction.

This shows grammars with very different structures perform well on the same (full evaluation) data set. KH99′ is still a strong performer, but we have shown that there exist many others which perform similarly (these GG1–GG6 form just a subset of the good grammars found in the search).

GG1 is KH99′ with two rules added, $A \rightarrow AA$ and $A \rightarrow (A)$. These rules

were used infrequently (probabilities 0.007 and 0.047 respectively). The mild improvement in functionality allows for an additional fraction of base pairs to be correctly predicted.

GG2 and GG3 were found using the posterior decoding version of the evolutionary algorithm. They have a high density of rules, that is many rules for each non–terminal variable. Particularly, GG2 has almost all of the rules it is possible for it to have, given $B \to .$ is kept constant through the evolutionary algorithm. Given this, it is not surprising that they performed poorly using the CYK training and testing. However, with posterior decoding, the base–pair probabilities are still effective for good predictions.

GG4 has only two variables (A and C) used almost exclusively in producing base pairs. It then uses various exit sequences to generate different sets of unpaired nucleotides and returns to producing base pairs. Finally, GG5 and GG6 are typical of larger grammars we have found with complex structure. It is not obvious to us how their structure relates to their success in secondary structure prediction. GG4, GG5, and GG6 were all found using the CYK version of the evolutionary algorithm, and perhaps their complex structure can be accredited to this. GG6 is a strong performer throughout, particularly when considering F–score.

Most grammars achieved lower predictive power on the Dowell and Eddy dataset. The difference in performance between KH99 and KH99′ is small and confirms that the representation of KH99 as KH99′ is a good one. Particularly noteworthy is the performance of GG3 and GG5. GG3 has had a considerable increase in PPV, likely due to the posterior decoding prediction

method. Given many of the structures in the Dowell and Eddy dataset contain pseudoknots, other grammars score poorly trying to predict pairs where there are not, in contrast to GG3. By predicting fewer base pairs, GG3 gains higher PPV as more of them are correct, but lower sensitivity. GG5 is a grammar which was unremarkable in its results on the original data set, however, it has outperformed the rest of the grammars on the benchmark set and is the only grammar with improved sensitivity when compared to the RNASTRAND dataset.

Figure 1.3 shows the sensitivity–PPV curve for grammars KH99′ and GG1–GG6. This was produced using the posterior decoding method by varying the parameter γ[, the balance between predicting paired and unpaired bases]. GG6 again proves to do slightly better than the other grammars, having the largest area underneath the curve. KH99′ does not distinguish itself much from the other grammars, being in the middle in terms of area underneath the curve.

Overall, the grammars found in the evolutionary search still perform well because they are not overadapted to deal with the original data. Determining which is best depends on the measure of strength of prediction, whether the size of the grammar is a concern, ability to approximate structures with pseudoknots effectively, and so on. However, it is clear that a selection of effective grammars has been found. Results shown by UNAfold and RNAfold continue to be superior to those produced by SCFG methods.

We also checked that the grammars obtained from the evolutionary algorithm do not merely produce similar structures to KH99′ by using different deriva-

tions. To do this we define the relative sensitivity of method A with respect to method B as the sensitivity of method A as a predictor of the structures produced by method B. The relative PPV is defined in a similar manner. We then compared the predictions of the grammars by building a heat map of the relative sensitivities and PPVs (Figure 1.4), using the evaluation set. As expected, KH99′ and GG1 predict almost identical structures, as they are highly similar. Similarly, it is perhaps not surprising that GG2 and GG3 have very similar predictions given they produce structure through posterior decoding. The rest of the methods have sensitivity and PPV relative to other prediction methods of approximately 0.4 – 0.6. As they are designed to predict RNA secondary structure using the same training set, one would expect some similarity in the predictions, although not as much as with KH99′ and GG1. This is confirmed by our results, suggesting that the new grammars produce different kinds of structures which are good representations of RNA secondary structure." (Anderson et al. 2012).

Further analysis of KH99

"To test the local features of the space, we evaluated variations of KH99′ against the full data set. Where a single rule was deleted, only one grammar had prediction accuracy of the order of KH99′. This is the grammar without the rule $A \to (C)$ (which would have been used only infrequently in KH99′, with probability 0.014). However, it is clear that deleting rules has a strong negative effect on the predictive power of KH99′, given that no others have sensitivity greater than 0.25. Of course, this might be expected given that

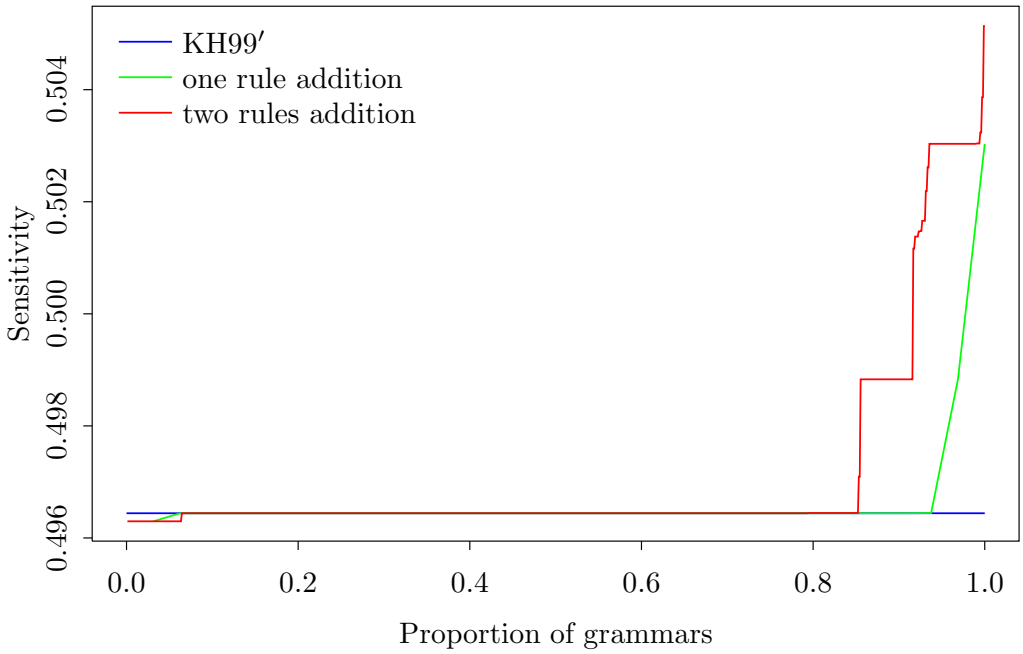


Figure 1.5: Summary of the effects of adding one (giving 32 grammars) or two (giving 961 grammars) production rules to KH99′. The plot shows the cumulative proportion of grammars with given sensitivity. The grammars' sensitivity is mostly still equal to the sensitivity of KH99′, with only a few outliers. GG1 was the top outlier for two production rule added. In this sense the space is reasonably flat. Figure and caption taken from Figure 4, Anderson et al. (2012).

this SCFG has been constructed manually, and it is therefore unlikely to have unnecessary production rules.

With addition of rules, the number of grammars to check quickly becomes large. With one production rule added, 32 grammars must be evaluated, with two added this increases to 496. A similar local search for larger grammars would be impractical, since there are many more grammars with one or two altered production rules (for GG6, there are 584 grammars with only one new production rule, and 170,236 with two). Ambiguity of tested grammars had little or no effect. Results of this local search can be seen in Figure 1.5." (Anderson et al. 2012).

Brute force search

"In addition to the evolutionary algorithm, we have run a brute force search to evaluate small grammars which might be effective. One of the main points of emphasis of Dowell & Eddy (2004) was to look for *lightweight* grammars. In searching exhaustively for a small grammar, computational problems are quickly encountered. The search is feasible for grammars with only two non–terminals in addition to the rule $B \rightarrow$. (16,384 grammars) but not for 3 non–terminals (over 500 billion grammars). Whilst the search in three non–terminals may be worthwhile, especially since KH99 contains only 3 non–terminals, it was far from computationally feasible for the purposes of this paper." (Anderson et al. 2012).

"The brute force search illustrated how, with this normal form, larger gram-

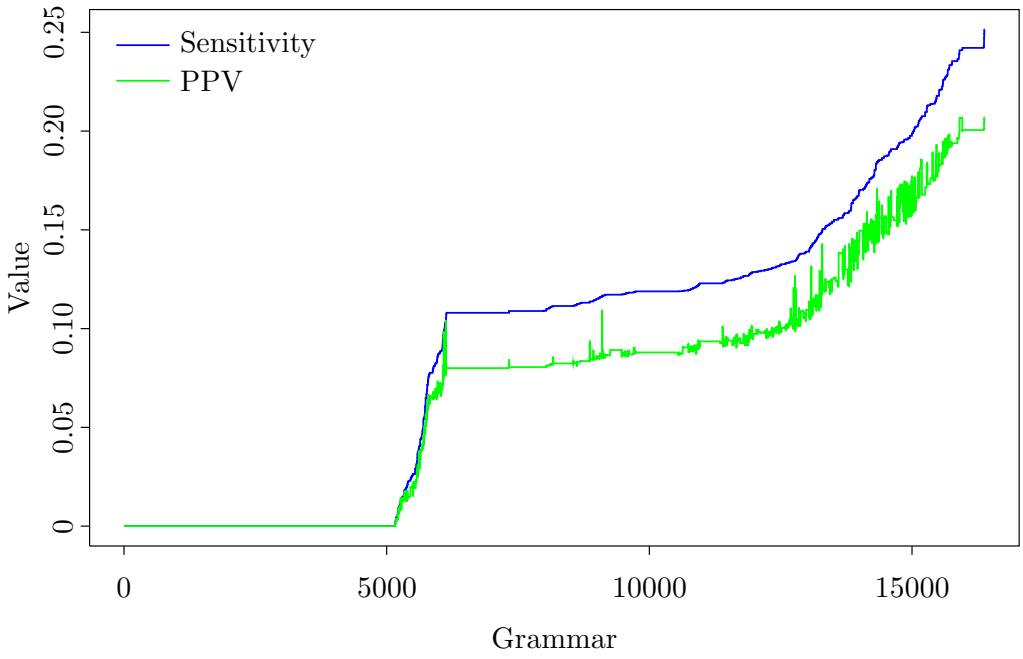


Figure 1.6: The distribution of sensitivity and corresponding PPV of grammars with at most 2 nonterminal variables. Approximately one quarter of grammars have sensitivity 0, as many cannot produce long strings. It is only the larger grammars that start to predict long strings which might correspond to structure. However, the prediction quality is still poor by both measures. Figure and caption taken from Figure 5, Anderson et al. (2012).

Grammar	Ambiguity	Completeness
KH99′	No	Yes
GG1	Yes	Yes
GG2	Yes	Yes
GG3	Yes	Yes
GG4	Yes	No
GG5	Yes	No
GG6	Yes	Yes

Table 1.3: Ambiguity and completeness of KH99′ and GG1–GG6 grammars. All grammars found in the search were ambiguous. Some of the grammars found (GG4 and GG5), are incomplete but heuristically it seems that the structures that cannot be generated have little biological relevance. Table and caption taken from Table 3, Anderson et al. (2012).

mars are needed to provide effective prediction. Most small grammars will only be able to produce one type of string. Also, it suggested that the existing grammars are close to locally optimal and that the space around them is quite flat, demonstrating the need for intelligent searching methods. Figure 1.6 illustrates the distribution of sensitivity across the space of grammars with at most 2 non–terminal variables. No grammar has sensitivity higher than 0.25 and approximately one quarter of grammars have sensitivity 0 (those which cannot produce long strings)." (Anderson et al. 2012).

Ambiguity and Completeness

"One of the results of the search which we find most interesting is the ambiguity and completeness of GG1–GG6, shown in Table 1.3. All grammars found in the search were ambiguous, and still predicted structure effectively. In particular, ambiguous grammar GG1 performed better than KH99′, being a slight modification of it. Particularly, it is clear that GG2 and GG3 have

many different derivations for each structure, and their strong performance relies on this ambiguity, as they perform poorly when tested with CYK. GG5 demonstrates further that ambiguous grammars can even be effective at approximating structures with pseudoknots. The effectiveness of some ambiguous grammars is likely due to the prediction algorithm picking structures that, whilst perhaps suboptimal, are close to what the best prediction would be. Clearly there is room for a further investigation into quite why some grammars cope better with ambiguity than others.

Similarly, it might be surprising that some of the grammars found (GG4 and GG5), are incomplete. However, heuristically it seems that the structures that cannot be generated have little biological relevance (e.g. GG4 cannot generate "(...)(...)(...)(...)"). In some sense therefore, the incompleteness is permissible, as the grammar is still able to generate any relevant structure.

Our brute force search and search around KH99 demonstrate that intelligent searching methods are necessary, and overall, the method of evolving SCFGs for RNA secondary structure prediction proved effective. We found many grammars with strong predictive accuracy, as good or better than those designed manually. Furthermore, several of the best grammars found were both ambiguous and incomplete, demonstrating that in grammar design such grammars should not be disregarded." (Anderson et al. 2012).

1.3.4 Conclusions and further work

"The equally consistent inability to achieve any significant improvement on this level of performance, and the relative limited prediction correlation between the many good grammars found points to the inherent challenge of grammar design, or indeed to the limitations of SCFG based methods as a whole. It appears that the number of grammars able to achieve this level of performance is large, and may depend little on the overall grammar structure, and at the same time it appears that a performance improvement may be difficult or impossible to achieve with a grammatical approach." (Anderson et al. 2012).

Indeed, since publication of Anderson et al. (2012), further work has been done considering different SCFGs for RNA secondary structure prediction. Rivas et al. (2012) build a formal framework for considering a wide range of SCFGs, from KH99 to large SCFGs designed to emulate complex folding models like RNAfold and CONTRAfold (Do et al. 2006). They find the probabilistic analogues are good representations of the original folding methods, allowing for the advantages of a probabilistic framework, such as integration with evolutionary models. However, no substantial improvements in predictive accuracy were found, with the authors suggesting that data quality could be a limiting factor.

For single sequence RNA secondary structure prediction, it might therefore be reasonable to draw one of three conclusions:

I There is not another SCFG not yet evaluated which is a stronger predictor

of RNA secondary structure.

II SCFG prediction is about as good as it is going to get.

III The design of the SCFG is about as good as it is going to get, but the way we are parameterising or predicting structure is not optimal.

Although efforts have been made for lightweight SCFG by Dowell & Eddy (2004) and Anderson et al. (2012) as well as for for more complex SCFGs (Nebel & Scheid 2011, Rivas et al. 2012), not all SCFGs have been evaluated and so conclusion **I** cannot be ruled out completely. If one did want to continue searching for useful SCFGs, there are potential steps that could be taken to investigate this further. Since comparative prediction is frequently used, one could simply to add in a molecular evolution model to the evolutionary search to see if there are any grammars (both found in the search, or ones already found in Dowell & Eddy (2004) or Anderson et al. (2012)) which can outperform KH99 in a comparative setting. Much of the framework of the evolutionary search heuristic presented above would remain. However, it could be that adding in the molecular evolution model is actually unhelpful. Since data would need to consist of alignments, there would be many additional structural constraints which might make poor SCFGs predict better structures, and the molecular evolution model might have the same effect. In short, it could be harder to distinguish between the predictive quality of SCFGs.

Given the marginal improvements on KH99, it might be easy to accept conclusion **II**. However, note many other SCFGs equally as strong as those pre-

sented above were also found, and Anderson et al. (2012) showed the SCFGs found were not simply KH99 in disguise, nor were they predicting the same secondary structures. One might think to try and combine SCFGs to create a "meta"-SCFG, but, in some sense, this is what is attempted in the breeding stage of the evolutionary algorithm. Similarly, it is known that RNA secondary structures do not always adopt the minimum free-energy structure (analagously the maximum probability structure for SCFGs). Consequently, one might adopt the stance of **III**, and suggest that the SCFGs can capture the probability distribution of RNA secondary structure effectively, but do not select the correct structure as the predicted one. This is a broadly similar conclusion to the one reached in Rivas (2013), also trying to evaluate the current state of RNA secondary structure prediction.

Chapter 2

Averaging Variances in Comparative RNA Secondary Structure Prediction

Up to this point we have only considered single-sequence RNA secondary structure prediction. However, RNA secondary structure is often done in a comparative setting, considering homologous sequences of RNA, and postulating that they have a common ancestor. By using an evolutionary model, or simply the additional information given by considering homologous sequences, prediction accuracy improves (Knudsen & Hein 1999). Large data sets of alignments of RNA sequences are available for comparative prediction (Griffiths-Jones et al. 2005, Cannone et al. 2002).

A main strength of SCFG-based prediction methods is that they combine effectively with a molecular evolution models. This was first implemented

for a fixed alignment in Pfold (Knudsen & Hein 1999, Knudsen & Hein 2003), which assumes that columns (or pairs of columns) evolve indepedently following a general reversible model (Tavaré 1986), that is according to a continuous-time Markov chain. Rate matrices for unpaired and paired nucleotides are inferred from training data as a function of the frequencies of nucleotide and base-pair mutations observed, and then column probabilities can be calculated via the Felsenstein pruning algorithm (Felsenstein 1981). Gaps are typically treated as unknown nucleotides. Prediction is done as for single sequences, using the CYK algorithm or Inside-Outside algorithms.

Thermodynamic methods, on the other hand, do not combine effectively with molecular evolution models. Extensions like RNAalifold (Bernhart et al. 2008) of RNAfold allow the thermodynamic method to be used for fixed alignments, though. Prediction accuracy gains from additional structural constraints, since thermodynamic methods do not typically allow for non-canonical base-pairs, restricting the number of possible structures.

However, as detailed in Gardner & Giegerich (2004), structure prediction is only part of the problem. Some methods implement a simultaneous fold-and-align approach (Bradley et al. 2008), but these are often prohibitively expensive in computation or memory. RNA folding programs like Pfold and RNAalifold require an input of an aligned set of sequences. Therefore many factors can affect the structure prediction other than the folding model itself, like the number of homologous sequences aligned and how sequences might be sampled. In this chapter we discuss how to quantify many of these variances in comparative RNA secondary structure prediction. We create data

sets of alignment samples from a reference alignment using statistical alignment, predict RNA secondary structures using PPfold (Sükösd et al. 2012), a parallelised version of Pfold, and RNAalifold and use these to examine relationships between alignment distance and prediction quality. These data sets also allow us to examine how many homologous sequences are needed for a good prediction, and how these sequences might vary. Finally, a method is developed for trying to average predictions over possible alignments, and this method benchmarked against a number of alignment programs.

Much of the work here has been taken from Anderson et al. (2013*c*), a paper entitled "Quantifying Variances in Comparative RNA Secondary Structure Prediction" published in BMC Bioinformatics, and Arunapuram et al. (2013), a paper entitled "StatAlign2.0: Combining Statistical Alignment with RNA Secondary Structure Prediction" published in Bioinformatics.

2.1 Variances in Comparative RNA Secondary Structure Prediction

"Sequence alignment is a fundamental step in most comparative sequence analysis pipelines. The typical approach is to create a single, trusted multiple alignment of the sequences using methods based on an artificial scoring scheme and heuristics to find a highly scoring alignment (Chenna et al. 2003, Katoh et al. 2005). Although this methodology is successful when the alignment is well resolved, it has been shown in the context of various types of downstream analyses that the end result can be highly sensitive to

the choice of alignment (Wang et al. 2011, Chivian & Baker 2006, Gardner et al. 2005). RNA secondary structure prediction methods take a variety of approaches with respect to aligning RNA sequences. Some methods invoke a fold-and-align approach e.g. (Bradley et al. 2008), where alignment is done simultaneously with structure prediction. SCFG methods, although require an alignment as input, are usually combined with an evolutionary model, which can allocate probability to non-canonical base-pairs which often results from poor alignment. This can help make SCFG models more robust to poor alignment quality. Most modern methods e.g. (Doose & Metzler 2012), still consider prediction from a single, fixed, alignment, although some use alignment-free methods (Harmanci et al. 2011). However, even after considering poor alignment quality, there are many additional variances associated with poor comparative RNA secondary structure prediction.

Sequence selection is another important variance in comparative RNA secondary structure prediction. [For families of RNA with many known homologous sequences, we may also have to select a subset of these sequences to align, as many alignment programs have computational difficulties with aligning or predicting trees for a large number of sequences]. Alignment methods may produce poor alignments due to poor individual sequences, which will in turn produce poor structure predictions. There have been methods developed to select homologous sequence, particularly (Engelen & Tahi 2007) which is based on evolutionary models and structural constraints. This is then implemented in (Engelen & Tahi 2010), which shows strong results in RNA secondary structure prediction can be found by selecting useful

sequences."(Anderson et al. 2013*c*)

2.1.1 Data

To analyse variances in comparative prediction accuracy, we created two sets of alignment samples, to avoid introduing model-based biases. The first data set was created by using the statistical alignment model (Hein et al. 2000) implemented in StatAlign (Novák et al. 2008). StatAlign uses a modified verson of the TKF92 model (Thorne et al. 1992) (see (Miklos et al. 2008) for details), and uses Markov Chain Monte Carlo (MCMC) to sample alignments from the posterior distribution of alignments, phylogenies, and model parameters. Consequently we can use StatAlign to generate a large set of "representative" alignments from a given reference alignment. This data set was designed to represent alignments that might be obtained for applications of RNA secondary structure prediction, so variances that might be seen in applications of RNA secondary structure prediction might be characterised. It would have been desirable to also obtain alignment samples from other well-used alignment programs like Clustal (Sievers et al. 2011), but many of these are deterministic so sampling is not readily available.

If we only used alignment generated by StatAlign it may be some artefacts of the statistical alignment or MCMC framework may be interpreted as signal in the resulting predictions. Consequently we created a data set to assess structure prediction independently of alignment method. An MCMC framework was set up to sample alignments in a uniform manner. In particular, we were able to control the distance the sample alignment were from the

reference alignment, which would prevent skew in the analysis.

StatAlign Data Set

"To test factors relating to alignment quality and secondary structure prediction quality, we need a large number of alignment samples from trusted reference alignments with known secondary structures. We have created a curated RNA dataset based on the Rfam database (Griffiths-Jones et al. 2005) for the purposes of evaluating the framework. Alignments of homologous RNA sequences with known consensus secondary structure were extracted from Rfam seed alignments. From these, 50 RNA families with at least 50 sequences were randomly selected. From each family, in a pre-filtering step we removed divergent sequences with lots of long indels. Here, we defined insertion as consecutive non-gap characters of a sequence in the reference alignment which appear in columns where over 80% of the sequences have gaps. Deletions were defined analogously. Columns with fraction of gaps between 20% and 80% were regarded ambiguous and ignored. To over-penalise long indels we applied the superlinear score function $l \times \log_2(l+1)$ for indels of length l, indels being defined as above. Then, a sequence was removed from a family if its total indel score was beyond 20 *and* the difference between its indel score and the mean indel score in the family was beyond 3.7 times the standard deviation of the indel scores in family, i.e. if the sequence had significantly more and/or longer indels than what is representative of the rest of the family. Then, 50 sequences were selected at random, and further random selection was done to get pairs, triplets etc. up to 15 se-

quences in alignments. From these samples of known reference alignment, we could produce many different alignment samples using StatAlign (Novák et al. 2008). For each RNA alignment, 200 alignment samples were taken, and the reference alignments were also kept for comparison. We refer to this dataset throughout as the *StatAlign dataset*." (Anderson et al. 2013*c*)

Random Alignment Data Set

"We also wanted to measure the effect of alignment accuracy on secondary structure independently of alignment method. We created a dataset based on the RNA families of the StatAlign dataset, where alignments were sampled uniformly at various fixed distances from the reference alignment. Using the alignment distance measure in Eq. 1.6, we created a Metropolis-coupled MCMC framework that runs several parallel MCMC chains to take alignment samples from the target distribution

$$\pi(a) = \exp\left(-\frac{2\,|\,d(a, a^r) - \mathbb{d}\,|}{t}\right) \tag{2.1}$$

where a^r is the reference alignment, $\mathbb{d}$ is the target distance to get samples from and t is the temperature of the chain. To improve the mixing properties of the chains we allowed each chain to explore alignments that do not exactly match the specified target distance (with an exponentially decreasing probability, as described by Eq. 2.1) but then rejected non-exact matches when taking samples from the cold chain ($t = 1$).

The state space of the Markov chains is the set of all possible multiple alig-

ments of the input sequences. Alignments that represent the same set of homology statements, and only differ by the order of the alignment columns, are treated as different (e.g. alignments $\begin{matrix}A & -\\ - & A\\ - & B\\ B & -\end{matrix}$ and $\begin{matrix}A\\ -\\ -\\ B\end{matrix}$ of the sequences A and B). The following basic alignment rearrangement moves are iterated:

1. breaking an alignment column into two columns by moving one of its characters into a new column, placing gaps in every other row:

$$\begin{array}{c|c|c} C & \mathbf{A} & G\\ C & A & G\\ C & A & G \end{array} \overset{\overset{1.}{\searrow}}{\underset{2.}{\nwarrow}} \begin{array}{c|c|c} C & - \ \mathbf{A} & G\\ C & A \ - & G\\ C & A \ - & G \end{array}$$

2. the exact reverse of the previous, i.e. joining two compatible adjacent columns – one having gaps in all but one row, the other having a gap in that row – to form a single column (see above)

3. relocating one character of a column to a gap position of an adjoining column:

$$\begin{array}{c|c|c} C & A \ G & T\\ C & A \ G & T\\ C & \mathbf{A} \ - & T \end{array} \overset{3.}{\Longleftrightarrow} \begin{array}{c|c|c} C & A \ G & T\\ C & A \ G & T\\ C & - \ \mathbf{A} & T \end{array}$$

As the space of alignments is vast and the moves are very local, to get a reasonably good approximation of uniform sampling, the number of steps that have to be done is in the order of millions even for small inputs when a single chain is run. To this end, we created a special alignment representation where the above rearrangements can be carried out very efficiently

(essentially constant time, i.e. in time proportional to column size but independent of the alignment length, even when the cost of randomly selecting the column to alter is included). For moderate input sizes (5–10 sequences of length 300–500) this representation allowed us to take 1.5 million rearrangement steps in a second (using a Java 6 implementation on one core of a 2.4 GHz Intel i3 processor).

A single chain is sufficient for small alignments, but very slow mixing becomes an issue for practical alignment sizes. We have found that standard parallel tempering techniques effectively speed up mixing if the chain temperatures are chosen correctly. Because the optimal temperatures vary significantly depending on reference alignment characteristics and target distribution, a simple acceptance optimisation routine was added that tunes chain temperatures to achieve chain swap acceptance ratios between neighbouring chains around a pre-set value. We found ratios 0.7–0.8 to be the most effective. With this framework, running 8–10 parallel chains was best to maximise the speed of convergence to the uniform distribution as compared to a single chain with sampling times 8–10-fold.

The above described framework was utilized to create a dataset consisting of the RNA families of StatAlign dataset, where for each family and each selection of 5 representative sequences from the family, 10 samples were taken at a distance corresponding to a similarity of 0.98 to the reference alignment (see Eq. 1.7), then 10 samples at a similarity of 0.96 etc., down to a similarity ratio of 0.6. We refer to this dataset as the *Random Alignment dataset*."(Anderson et al. 2013*c*)

2.1.2 Results of Variance Analysis

"A common question in comparative RNA secondary structure prediction is how many sequences are required to get a good structure prediction. This is briefly addressed in (Knudsen & Hein 1999), but the sample size is quite small, and only total accuracy is considered. With 15 sequences in the alignment, we assume that no more evolutionary information can be gained by adding further sequences, but when fewer sequences are present, lack of information might yield a poorer structure prediction.

Instead of considering total accuracy, we wanted to quantify *relatively* how much accuracy is lost when fewer sequences are present. For example, if an alignment with 15 sequences is predicted with an average F-score of 0.5, and an alignment of the same family with 3 sequences is predicted with an average F-score of 0.4, then 80% of the accuracy will have been retained, that is the alignment with 3 sequences has a *relative F-score* of 0.8.

To investigate how many sequences are needed for a good structure prediction, we took the StatAlign dataset and considered the relative F-score for each family. Almost 100% of the possible F-score was achieved when the alignment contained 5 sequences, for both PPfold and RNAalifold. Interestingly, the accuracy of RNAalifold decreased slightly as the number of sequences got bigger. This is due to the increased number of non-canonical base-pairs in the alignments, which the thermodynamic method could not predict. PPfold, on the other hand, has a small probability for non-canonical base-pairs, so is not affected by these in the same way. Overall these results

suggest that 5 sequences are sufficient for approaching maximal predictive accuracy.

To consider the effect alignment quality has on RNA secondary structure prediction, we took the StatAlign and Random Alignment datasets and measured their similarity to the reference alignment using the similarity score above. Again, percentage of accuracy retained was calculated by normalising against the accuracy achieved on the reference alignment. Log-scale heatmaps showing the accuracy and percentage of accuracy retained for the StatAlign dataset (A) and Random Alignment dataset (B) for PPfold and RNAalifold can be seen in Figure 2.1. As expected, decreasing alignment quality decreases the accuracy of structure predictions. However, other patterns also emerge from these graphs.

Firstly, in the StatAlign dataset (Figure 2.1A), we observe a weaker correlation between alignment distance and accuracy than for the Random Alignment dataset (Figure 2.1B). This suggests that predictions are better for the StatAlign dataset on alignments far from the reference alignment. StatAlign looks to produce alignments with a high likelihood under an evolution model, which the random alignments do not consider, and so StatAlign's alignments could be considered more realistic. The expectation that StatAlign's alignments are more useful for RNA secondary structure prediction than random alignments is confirmed here.

Secondly, for the StatAlign dataset, we see a much higher correlation with the F-score than with the relative F-score. Some families of RNA consistently produce the same alignment, which skews the graphs. For example,

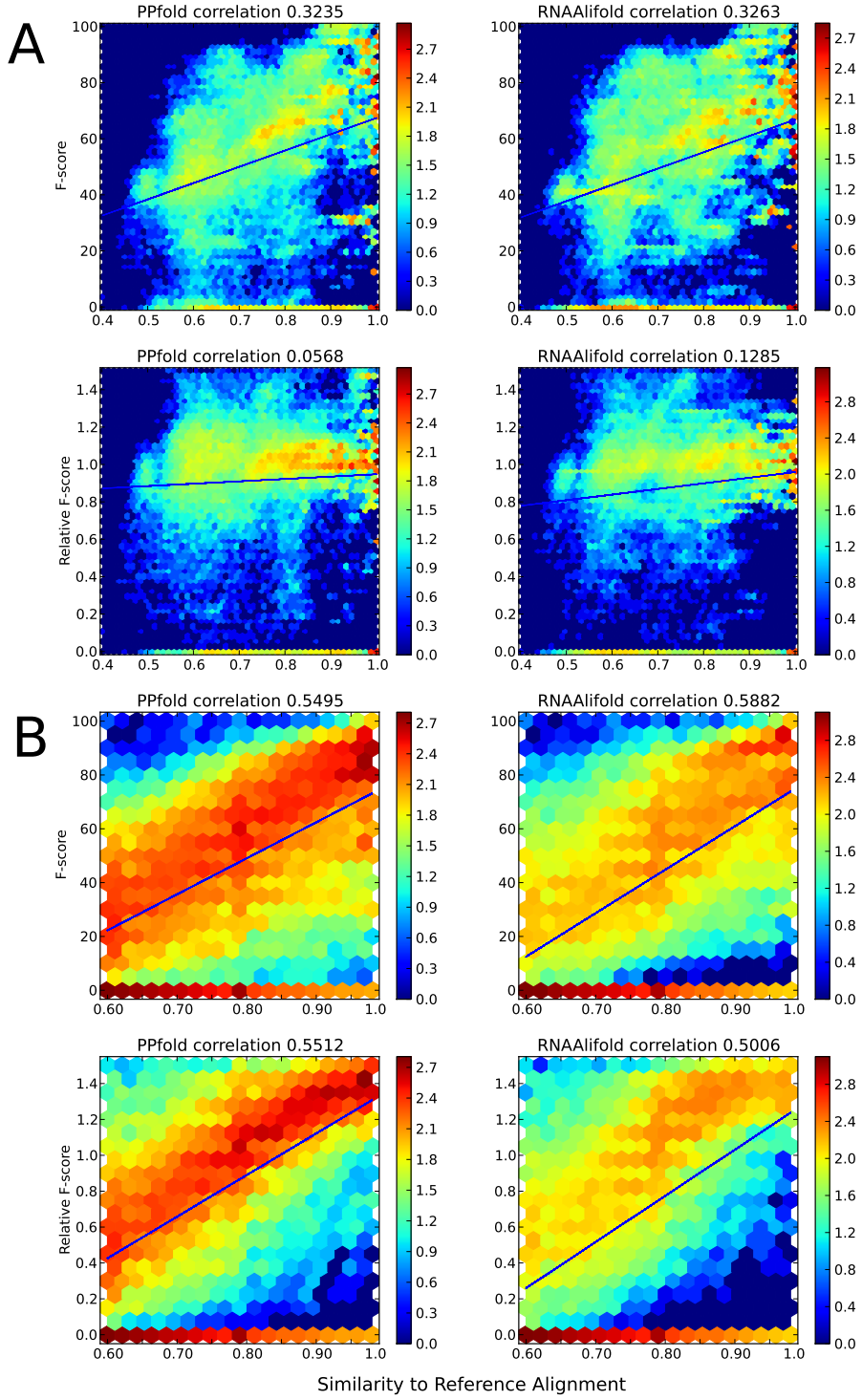


Figure 2.1: Performance accuracy on the StatAlign dataset (A) and Random Alignment dataset (B) for PPfold and RNAalifold when alignment quality is varied. Percentage of accuracy is determined by, for each family, normalising by the average accuracy on the reference alignment. The R^2 correlations given are determined by a linear regression model. Figure and caption taken from Figure 2, Anderson *et al.*, (2013*c*).

an alignment which consistently has similarity to the reference alignment of 0.9, and F-score 0.5 would give a relative F-score of 1 every time, and would support the correlations seen on each graph. Because we can control the spread of distances in the random alignment data set, we don't see this behaviour. As expected, variation comes more with families that produce more variable alignments. In the StatAlign dataset, this is obscured by those families which produce consistent alignments.

Lastly, for the Random Alignment dataset (Figure 2.1B), we see many more zero quality predictions in the case of RNAalifold than in the case of PPfold. This is most easily seen by the larger intensity of red in the main body of the heatmap for PPfold. This suggests that PPfold "deals" with bad alignment quality better than RNAalifold, likely due to the molecular evolution model."(Anderson et al. 2013*c*)

"We also consider the effects of evolutionary distance on RNA secondary structure prediction quality. One might expect that there is a "sweet spot" for evolutionary distance– sequences too close to each other do not display enough co-variation to benefit the evolutionary model, but the evolutionary signal might be lost if the distance is too large. To investigate this, we measured evolutionary distance in the phylogenetic trees predicted by PPfold using four different measures:

Measure 1– Mean of all the evolutionary distances,

Measure 2– Standard deviation of all the evolutionary distances,

Measure 3– Maximum evolutionary distance,

Measure 4– Maximum difference between evolutionary distances.

All four measures would be expected to be correlated with sequence length, which is well known to correlate with predictive accuracy. To account for this, we considered relative evolutionary distance, similar to the relative predictive accuracy above. A measure was normalised by the average for that family and number of sequences, so that it could be seen whether an alignment had greater or less evolutionary distance than might be expected."(Anderson et al. 2013*c*)

When this was tested, all of the measures "correlated extremely poorly with predictive accuracy and relative predictive accuracy, measures 1 to 4 having R^2 correlations with relative predictive accuracy of 0.0259, 0.0373, 0.0303, and 0.0265 respectively (data not shown). This suggests that evolutionary distance is not an underlying factor for variation in RNA secondary structure prediction, and that other factors, like those seen in (Engelen & Tahi 2007), play a more important role in determining predictive accuracy."(Anderson et al. 2013*c*)

2.2 Averaging RNA Secondary Structure Predictions over Alignments

From the variances in prediction quality shown above, it can be seen that it would be particularly desirable to eliminate the reduction in prediction quality caused by having a poor quality alignment. Here we try to integrate

out alignment uncertainty by sampling over alignments using the posterior generated by statistical alignment models. By considering many possible alignments, we hope that the alignments which preserve structure well provide a strong signal to the model.

One simple way to attempt to integrate out alignment uncertainty would be to average all the alignments first, come up with a consensus alignment, and do structure prediction on that. Conveniently, StatAlign produces a consensus alignment using posterior column probabilities, and so this can be easily tested. However, since StatAlign does not have any underlying RNA secondary structure model, it builds the consensus alignment based only on the evolutionary and alignment models, so we might not expect a significant increase in accuracy. Instead, we might like to weight alignments sample based on some structural signal, for example by averaging base-pair probability matrices.

2.2.1 Projecting Alignments

The first problem we run into is that not all the alignments are the same length. "For example, if we have two alignments and corresponding secondary structures

```
((((.....))))        ((((....))))
CCCCAAAA-GGGG        CCCCAAAAGGGG
CCCCAAA-AGGGG        CCCCAAAAGGGG
```

we would not want to suggest that these alignments give two different secondary structures." (Anderson et al. 2013*c*)

There are methods of averaging alignment which do not depend on length (see Section 2.3), but these can be computationally expensive. Consequently, we use a projection method to give [features of structure prediction methods such as] alignment column pairing probability matrices the same dimension, so that we can average the matrices.

For a given set of input sequences, the sequence containing the greatest number of non-gap characters was chosen as the reference sequence. Each pairing probability matrix is projected by deleting columns and rows of the matrix corresponding to gap positions in the reference sequence, thus ensuring each matrix corresponding to an alignment sample has the same dimensions (a square matrix, with dimensions equal to the number of non-gap characters in the reference sequence). For example, we might start with an $(n+1)\times(n+1)$ matrix, delete row i and column i due to a gap in position i in the reference sequence, then end up with an $n \times n$ matrix as required." (Anderson et al. 2013*c*)

2.2.2 Averaging over Alignments

Using this projection method we are able to create two methods of averaging over alignments.

Base-Pair Averaging

The base-pair averaging method simply averages the posterior base-pair matrices produced by your alignment method. "Each alignment sample is folded using PPfold or RNAalifold (Bernhart et al. 2008), and a base-pairing probability matrix is produced. These matrices are averaged to obtain a single base-pairing probability matrix, and maximum posterior-decoding (as described in (Knudsen & Hein 2003)) is used to obtain a secondary structure. We refer to this method as the *base-pair averaging* method." (Arunapuram et al. 2013).

The advantage of this method is that is can be used for both probabilistic and thermodynamic models, as a base-pair probability matrix can be calculated using either the Inside-Outside algorithms or the full partition function (McCaskill 1990). In testing, we made attempts at weighting the base-pair probability matrices by, for example, the likelihood of the alignments, but all methods ended up essentially using only a single alignment. Moreover, the weightings were not mathematically meaningful, and so we discarded them. The base-pair averaging method is publicly available as part of StatAlign2.0.

Phylogenetic Averaging

"In a second approach, which also makes use of multiple alignment samples, we calculate phylogenetic probabilities as in (Knudsen & Hein 2003) by averaging over the phylogenetic probabilities of each of our alignment samples. PPfold then takes these probabilities and produces a secondary structure

prediction using the standard SCFG method. Given that this alignment requires an evolutionary model in order to produce phylogenetic probabilities, this second method of consensus structure prediction is only available for the SCFG method. We refer to this method as *phylogenetic averaging* method." (Arunapuram et al. 2013).

The "phylogenetic probabilities" are simply the evolutionary probabilities of columns given they are paired or unpaired. Here we present a brief proof that averaging over alignments in the way described above asymptotically converges to the phylogenetic probabilities considered over all possible alignments. The proof here is largely taken from Anderson et al. (2013*c*) where it is presented for calculating information entropy over alignment-space. The proof relies on two key facts: first, that the posterior probability of a structure can be written as a product of the SCFG production rule probabilities and the phylogenetic probabilities, and secondly, that the distribution we are sampling from is the population distribution, and so convergence happens through the weak law of large numbers. Throughout we consider only alignments of a fixed length, i.e. alignment samples have already been projected onto a reference sequence.

Define $\mathrm{P}_G[ss]$ as the SCFG production rule probability of a secondary structure ss, and $\mathrm{P}_T[ss]$ as the probabilities from the phylogenetic model. First note that the probability of a secondary structure can be written as

$$\begin{aligned} \mathrm{P}[ss] &= \mathrm{P}_G[ss]\mathrm{P}_T[ss] \\ &= \mathrm{P}_G[ss] \sum_{A \in \text{alignments}} \mathrm{P}_T[ss|A]\mathrm{P}[A], \end{aligned}$$

as for a fixed alignment length, the SCFG probabilities are constant for each alignment. Consequently, we can focus only on the evolutionary probabilities.

Similarly, the probabability of a given column given it is unpaired is

$$\mathrm{P}[c|c \text{ unpaired}] = \sum_{A \in \text{alignments}} \mathrm{P}[c|c \text{ unpaired in alignment } A]\mathrm{P}[A], \qquad (2.2)$$

There is no known efficient way to recurse over all possible alignments, so we cannot calculate this analytically. Instead we define a sample probability, and show that, in the sample-size limit, the alignment-sample information entropy tends to the true information entropy.

Consider alignment samples $a_1, \ldots, a_n$ from the space of all alignments of m sequences, sampled according to their probability. If we are using statistical alignment, as in StatAlign, we will be sampling alignments in this fashion. Then we have for a column c, once alignments have been projected to the same length, a sample phylogenetic probability P_S as the average of the sample phylogenetic probabilities:

$$\mathrm{P}_S[c|c \text{ unpaired}] = \sum_{i=1}^{n} \frac{1}{n}\mathrm{P}[c|c \text{ unpaired in alignment } a_i], \qquad (2.3)$$

with an analagous definition for columns given they are paired.

To show this sample probability converges to the true probability as sample

size tends to infinity, we first note that, rearranging Equation 2.3:

$$\mathrm{P}_S[c|c \text{ unpaired}] = \sum_{i=1}^{n} \frac{1}{n} \left(\sum_{A \in \, alignments} \mathrm{P}[c|c \text{ unpaired in alignment } A] 1_{\{A=a_i\}} \right), \tag{2.4}$$

with 1 being the indicator function. Hence

$$\mathrm{P}_S[c|c \text{ unpaired}] = \sum_{A \in \, alignments} \mathrm{P}[c|c \text{ unpaired in alignment } A] \left(\sum_{i=1}^{n} \frac{1}{n} 1_{\{A=a_i\}} \right).$$

Taking the limit $n \to \infty$, by the weak law of large numbers

$$\begin{aligned} \mathrm{P}_S[c|c \text{ unpaired}] &\to \sum_{A \in \, alignments} \mathrm{P}[c|c \text{ unpaired in alignment } A]\mathrm{P}[A] \quad \text{as } n \to \infty \\ &= \mathrm{P}[c|c \text{ unpaired}] \end{aligned}$$

as required. Since tree probabilities are the product of unpaired and paired column probabilities in the derivation, we can note that by algebra of limits the tree probabilities will converge in a similar fashion to the column probabilities. We now have a phylogenetic probability matrix, and can pass this to the CYK algorithm or Inside-Outside algorithms as before for secondary structure prediction. In practice (considering the consensus entropy results, see Section on entropy), we found convergence to happen reasonably quickly (≈1000 alignment samples) even for alignments with high sequence diversity. The phylogenetic averaging method is publicly available as part of StatAlign2.0.

2.2.3 Benchmarking

"To benchmark the methods, we took reference alignments from Rfam. In a pre-filtering step we discarded outlier sequences with many/long insertions and deletions from each family. Indels were first determined relative to a family consensus sequence, then a total mismatch score was calculated based on indel lengths, and sequences that had significantly larger mismatch score than the family mean were deleted. From 44 families that remained, we took a random sample of 5 sequences, as Knudsen & Hein (1999) suggest this is adequate to capture the evolutionary information. We then re-aligned the sequences using MUSCLE (Edgar 2004), MAFFT (Katoh et al. 2005), Clustal Omega (Sievers et al. 2011), and StatAlign, and predicted structures on the resulting alignments. Predictive accuracy was assessed by F-score, the harmonic mean of sensitivity and positive predictive value (Knudsen & Hein 2003).

Table 2.1 gives the results of the benchmark. By considering multiple alignments, the average F-score is at least as good or better as many current alignment programs. In particular the averaging methods improve upon the structure prediction on the StatAlign consensus alignment. When sequences are easy to align, we might expect methods to perform similarly, but when alignment is difficult, a method which can consider multiple alignments will produce fewer poor predictions."(Arunapuram et al. 2013)

	PPfold	RNAalifold
Reference Alignments	0.710	0.746
MUSCLE Alignments	0.539	0.572
MAFFT Alignments	0.567	0.599
Clustal Omega Alignments	0.554	0.598
StatAlign Alignments	0.561	0.584
StatAlign: Base-pair averaging	0.562	**0.605**
StatAlign: Phylogenetic averaging	**0.614**	*NA*

Table 2.1: Average F-scores for RNA secondary structure prediction benchmarks on 44 re-aligned Rfam Alignments. Table and caption taken from Table 1, Arunapuram et al. (2013).

2.3 Discussion, Conclusions, and Further Work

This chapter presented material discussing the effect of alignment quality, evolutionary distance between sequences, and number of sequences in an alignment on RNA secondary structure prediction. Alignment quality was found to affect the predictive accuracy most, with evolutionary distance between sequences contributing little variation. Five homologous sequences could, on average, capture all of the evolutionary information. Lastly, two methods were presented for averaging over alignments for structure prediction. By averaging over alignments, on average the predictive quality increases.

There is of course an increased computational burden associated with predicting structure over a sample of alignments. With the base-pair averaging method, structure predictions are needed for each alignment sampled. The phylogenetic averaging, though, only requires the phylogenetic probabilities to be computed for each sample, then the CYK algorithm or Inside-Outside algorithm to produce the structure prediction once sampling is completed.

However, for longer sequences, one might expect larger variation in alignments, and thus a longer convergence time of the averaged probabilities might be expected. RNA secondary structure prediction methods require cubic time with respect to sequence length, so for long sequences this may become prohibitive.

Additionally, note the two methods presented above are not the only ways to average over alignments. Here we will briefly discuss two additional methods for averaging structure predictions over alignments, for interest.

The first method creates what we will call "profile alignments". If, in the third column of the first sequence we see an A in 90% of the alignment samples, and a G in the other 10%, then instead of calling it an A or a G, we just leave the probability distribution as the frequencies observed. So, when doing the Felsenstein pruning algorithm, we put $(0.9, 0, 0.1, 0)$ in the leaf vector to represent the frequencies observed (instead of the $(1, 0, 0, 0)$ we might put if we just saw an A). We can then calculate phylogenetic probabilities with the pruning algorithm as before, and feed these into the CYK algorithm or Inside-Outside algorithms. This method, however, gives non-zero probabilities to alignments which could not possibly exist with the sequences given. For example, consider an alignment that has a sequence with a single A in it, in column 1 in one alignment, then column 4 in another alignment. Then the column probability of columns 1 and 4, given they are paired, will contain a contribution from the non-zero probability that A pairs with an A, which is of course nonsense, considering the sequence itself. Thus this method averages the phylogenetic probabilities over an extremely large

set of alignments, of which the correct set of alignments is a subset.

Given that the set of alignments averaged over by this method is not correct, we might have expected predictions to be poor. However, this was not the case. Heuristically testing this method over a test set of sequences, the structure predictions were slightly better (but probably not statistically significantly better) than for the maximum posterior decoding StatAlign alignments. This may be because the extra impossible alignments summed over do not alter the phylogenetic averages significantly, due to the extra alignments yielding similar phylogenetic probabilities as the originals. However, this does indicate that many of the alignments summed over do not affect the resulting prediction. Perhaps the main strength of the phylogenetic averaging method is to not consider the poor alignments, rather than focusing on the best ones.

The second method involves creating a directed acyclic graph (DAG) from the alignment samples. Each node in the DAG is a possible alignment column, and nodes are connected if they succeed or preceed each other as columns in one of the alignment samples. The graph edges are weighted by the frequencies the columns are seen with. Thus a path through the alignment DAG is then a valid alignment of the sequences, with the weights yielding the probability of the alignment. The DAG can also be viewed as an HMM emitting columns with probability 1. In particular, this means the alignment DAG has non-zero probability for alignments which were not seen in the samples.

The CYK algorithm and Inside-Outside algorithms can be adapted to jointly consider all possible paths in an HMM and all possible derivations of a SCFG

(Jagota et al. 2001). Thus one can jointly consider the space of alignments generated by the DAG and the SCFG for RNA secondary structure prediction. However, this can quickly become computationally expensive. As opposed to being cubic in the length of the sequence, the SCFGs algorithms are cubic in the number of nodes in the DAG, which would likely grow as more alignments are sampled. Averaging alignments through DAGs is current work at the time of writing, but I would be surprised if improvements were significant.

It is probably intuitive that having more sequences in your alignment will on average increase predictive accuracy, or that having a good quality alignment will, on average, yield a better prediction. Quantifying how structure prediction gets worse has led to work on predicting RNA alignment accuracy and RNA secondary structure prediction accuracy (see Appendix C). Although there is still a notable amount of variation unaccounted for, these results are useful for downstream analysis. Furthermore, by developing a method which is reliant on a statistical alignment and SCFG framework, we have shown averaging over alignments significantly increases prediction accuracy.

There are some clear paths forward for further work. Fully exploring the DAG model, and comprehensively comparing this to the two averaging methods presented here is a key next step, particularly as the DAG approach does not depend on having a fixed alignment length. It would also be interesting to find some criteria for which alignments were selected to be a part of the averaging process, particularly with a view of reducing computation. Some of the ideas in Engelen & Tahi (2007), with respect to sequence selection,

might extend to alignment selection, or one could even simultaneously sample sequences and alignments. Lastly, the alignment sampling approach could be combined with alternative SCFG models, like a model for co-transcriptional folding, or the model which emulates folding kinetics discussed in Chapter 3.

Chapter 3

Kinetic Folding of RNA Using Stochastic Context-Free Grammars

Recall from Section 1.3.4 conclusion **III**: "The design of the SCFG is about as good as it is going to get, but the way we are parameterising or predicting structure is not optimal." In this chapter we attempt to improve the method of prediction, trying to incorporate additional biological information about RNA secondary structure formation, particularly helix formation, into the decoding of the SCFG posterior distribution. Combining iterative helix formation, phylogenetic modelling, and a distance function between alignment colums leads to the an improvement in the accuracy of comparative RNA secondary structure prediction. Much of the work here has been taken from Anderson et al. (2013*a*), a paper entitled 'Oxfold: Kinetic Folding of

RNA using Stochastic Context-Free Grammars and Evolutionary Information" published in Bioinformatics.

"Most current approaches to RNA secondary structure prediction are static, insofar as they assess structures based on their constituent elements like base pairs and loops but with no contribution from the path followed to form these elements. The importance of folding mechanisms was noted by Tinoco et al. (1990), and Gultyaev et al. (1995) who studied the folding of intermediary stems. We note that "*the differences between real structures and the minimum energy states are believed to be determined mainly by defects in the energy rules used or by the existence of specific folding pathways capturing molecules in local minima*" (Gultyaev et al. 1995).

Just as comparative structure prediction is based on the observation that structure is important for function and hence conserved, since folding kinetics are important for either guiding or determining structure formation, we would expect evolution to exert selection on the kinetics, too. Previously, evolutionary models (Knudsen & Hein 1999, Knudsen & Hein 2003) and kinetic models (e.g. (Xayaphoummine et al. 2005, Danilova et al. 2006)) have been implemented, but have not been combined. It is therefore important to implement folding kinetics in an evolutionary framework.

In this [chapter], we work in the framework of the fundamental problem of predicting a consensus structure for a given, fixed alignment of RNA sequences. We incorporate folding kinetics, in a simplified way, into an evolutionary grammatical model in an iterative framework. Further, we introduce a distance function to incorporate information about the relationships

between different pairs of columns, thus adopting the second of the above approaches. The resulting model is benchmarked against PPfold (Sükösd et al. 2012), a parallelised implementation of the Pfold algorithm of Knudsen & Hein (2003). Additionally, we compare it to a thermodynamic model, RNAalifold (Hofacker et al. 2002, Bernhart et al. 2008)." (Anderson et al. 2013*a*).

3.1 Oxfold: Kinetic Folding of RNA using Stochastic Context-Free Grammars and Evolutionary Information

Anderson et al. (2013*a*) used much of the same framework given in Chapters 1 and 2 for RNA secondary structure prediction. The grammar KH99 was used (in double-emission normal form), with the probabilities taken from Knudsen & Hein (1999). Iterative helix formation required the use of base-pair probability matrices, determined by the Inside-Outside algorithm, and a MEA approach was used to determine the best structure (more details below). The basic evolutionary model from Knudsen & Hein (1999) was used, with the paired and unpaired rates being taken from the same. Gaps were treated as unknown nucleotides.

However, it was found that using a hybrid evolutionary model was more useful, partly for computational reasons. "For alignments of sequences or subsequences with high primary sequence conservation, the evolutionary model might miss "obvious" helices, since it introduces extra uncertainty. This is

especially relevant in iterative helix formation, because we only ever try to pair bases that have high posterior pairing probabilities. For this reason, rather than using the evolutionary model as in (Knudsen & Hein 2003), we use a mixed model: at the very start, we form basepairs with very high posterior pairing probabilities using the simpler heuristic of just multiplying the base-pairing probabilities for each sequence to obtain the prior base-pairing probabilities, and then switch to the full evolutionary model. The ability to mix different methods is a strength of the iterative approach." (Anderson et al. 2013*a*).

3.1.1 Folding Kinetics: Iterative Helix Formation

"The kinetics of RNA folding have been studied by (Craig et al. 1971), who determined the speed at which helices form. They showed that helices form quickly from a local basepair, in the sense that, once the first basepair of a helix has formed, nearby bases are more likely to pair.

This motivates emulating the kinetics of RNA folding in a simplified way by forming helices iteratively. Iterative helix formation that has also been used by (Harmanci et al. 2011). Once a suitable candidate basepair has been identified, a helix containing that basepair is formed." (Anderson et al. 2013*a*).

Local Helix Formation: Iterative MEA estimation

"We postulate that the first helix to form is the helix (without bulges) containing the basepair

$$(i_{\max}, j_{\max}) = \arg\max_{(i,j)} \left\{ \hat{\mathrm{P}}_{\mathrm{paired}}(i,j) \right\} , \tag{3.1}$$

where we use hats to denote the posterior pairing and unpairing probabilities obtained from the grammar. From a technical point of view, taking maximal probabilities in this way can be considered as a greedy approximation of the CYK algorithm (Durbin et al. 1998).

In the framework of iterative helix formation, the statistic corresponding to MEA estimation is the expected difference in the number of correctly predicted basepairs after pairing bases i and j,

$$\Delta(i,j) = \hat{\mathrm{P}}_{\mathrm{paired}}(i,j) - \frac{1}{2}\Big(\hat{\mathrm{P}}_{\mathrm{unpaired}}(i) + \hat{\mathrm{P}}_{\mathrm{unpaired}}(j) \Big) , \tag{3.2}$$

Just as the difference in equation (3.2) is naturally interpreted, by analogy with thermodynamic models, as a measure of the energy and therefore, of the stability of a basepair, the corresponding basepairing probabilities can be considered as a measure of the time it takes for that basepair to form. If we approximate basepair formation as a continuous Markov process with rate equal to the posterior pairing probability, the time until helix formation has the exponential distribution with mean equal to the inverse of the posterior pairing probability. With this interpretation in mind, equation (3.1) just

expresses the pairing of bases in the physical order.

At each iteration a new helix containing $(i_{\max}, j_{\max})$ and such that, for each basepair (i, j) in the helix, $\Delta(i, j) > 0$, is determined conditional on previously formed helices. By folding helices in one go, the fact that helices form quickly is taken into account. More helices are formed until $\Delta(i_{\max}, j_{\max}) < \delta$, for some threshold $\delta > 0$.

MEA estimation hinges on the assumption that the posterior base-pairing probabilities given by the grammatical model are equal to the probabilitity that a given basepair is correct. In fact, small positive values of the difference in equation (3.2) are not reliable, so requiring $\delta > 0$ might be expected to increase the positive predictive value of the algorithm. From a more physical point of view, to additionally require $\Delta(i, j) > \delta > 0$ for the basepair $(i_{\max}, j_{\max})$ is to incorporate the physics that once the first basepair is formed, nearby bases are more likely to pair. This local basepair needs to be "strong" enough for its dissociation time to be long enough for other basepairs to form. This also addresses the issue of the geometric and therefore unphysical distribution of helix lengths in the grammar (Knudsen & Hein 1999)." (Anderson et al. 2013*a*).

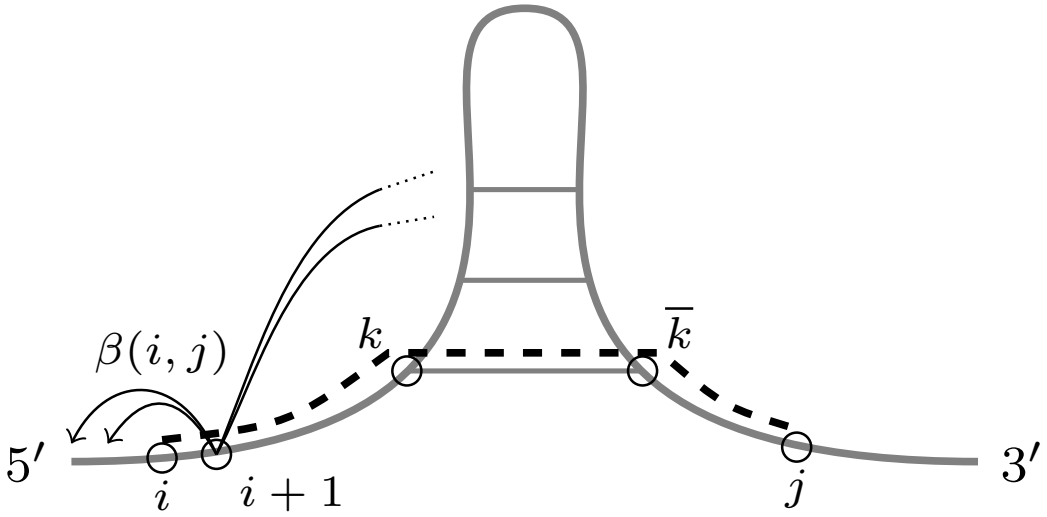


Figure 3.1: Representation of the distance function. The distance between two positions i and j in the sequence is the shortest distance moving along the structure, allowing shortcuts across stems. Intermediate positions are weighted with the probabilities of certain incompatible basepairs. Informally, this can be thought of as the shortest distance between two nodes of a weighted graph with edges between paired and adjacent positions. Figure and caption taken from Figure 1, Anderson *et al.*, (2013*a*).

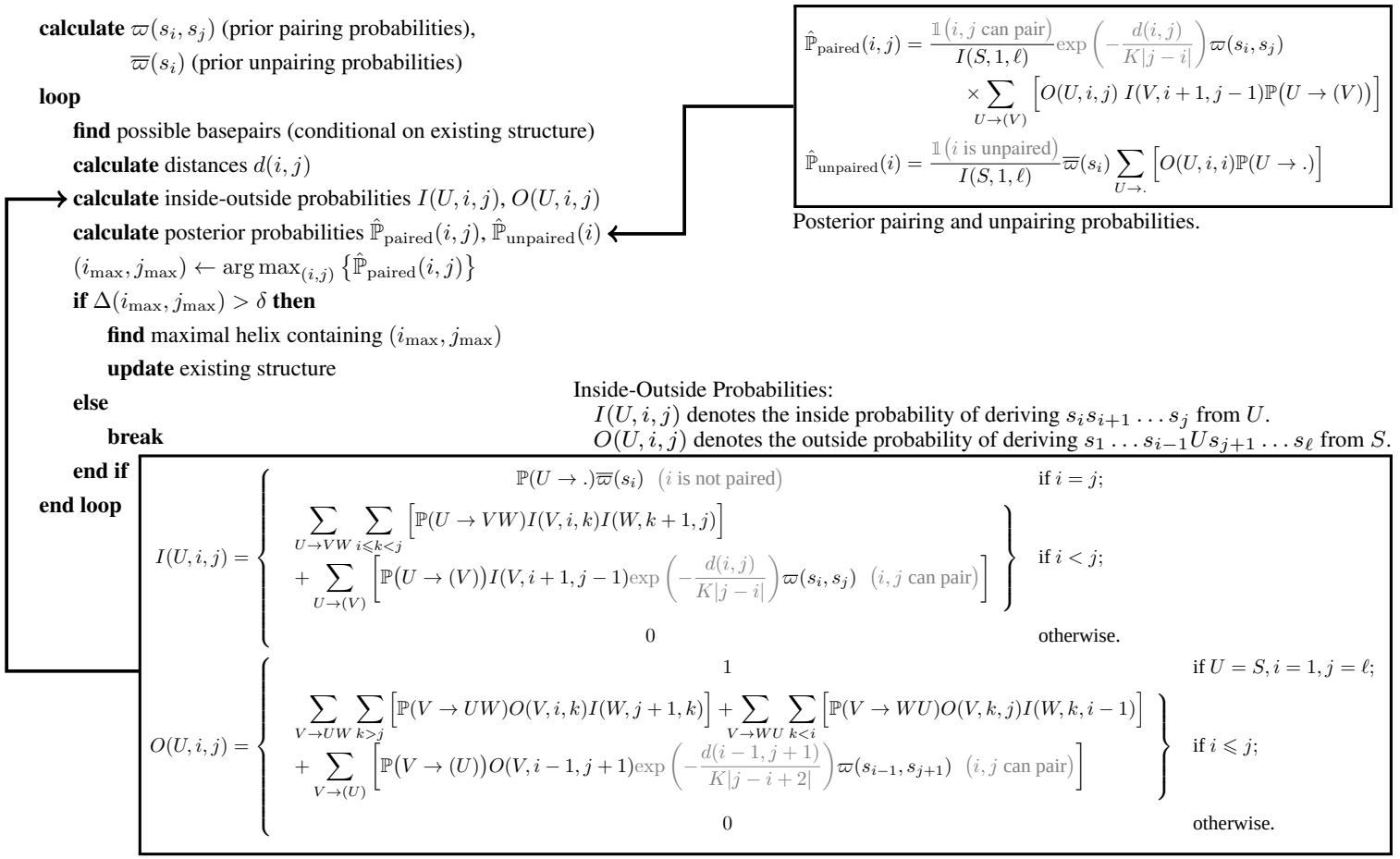


Figure 3.2: Simplified pseudocode summarising the full kinetic folding algorithm. The Inside-Outside and posterior probabilities are written in the double emission form of , and include the distance function and structural constraints. The modifications needed to take account of structural constraints and to introduce the distance function are shown in grey. Capital letters denote generic non-terminals, while lower-case letters denote column indices and s_i denotes the i-th column of the alignment; ℓ denotes the total sequence length. Figure and caption taken from Figure 2, Anderson et al (2013a).

3.1.2 Bayesian Weighting: The Distance Function

"In the model we have built up to this point, distinct columns and pairs of columns are, insofar as prior unpairing and pairing probabilities are concerned, assumed to be independent. This does not reflect biological fact, as interlacing structures prevent each other from forming and a column cannot pair with two different columns at the same time.

We note that this kind of crossing interaction may lead to pseudoknot formation. Standard SCFG approaches are unable to predict pseudoknots (Brown & Wilson 1996): in standard MEA estimation, one hopes that the model predicts the more stable of the two interlacing structures. In the iterative framework, the extra information is used to address, more generally, the most obvious drawback of the iterative approach: once a helix that is incompatible with the correct structure has been formed, the final prediction is likely to be poor. From a kinetic perspective, base pairs frequently blocked by other, transient, base pairs, would take longer to form, as the underlying continuous Markov process is only enabled during intervals where the base pair is not in conflict with other base pairs.

To include these physical dependencies between columns in the model, we look to penalise the pairing of two columns if there exist columns between that are likely to form a basepair incompatible with these two columns. Since standard SCFG approaches cannot model pseudoknots, here, "incompatible" does not only refer to basepairs that share a position with these two columns, but also to basepairs that would form a pseudoknot with these two columns.

Thus we discount the prior base-pairing probabilities by an exponential factor based on a distance function, so that they take the form

$$\mathrm{P}_{\mathrm{paired}}(s_i, s_j) = \exp\left(-\frac{d(i,j)}{K|j-i|}\right)\varpi(s_i, s_j)\ , \tag{3.3}$$

where s_i denotes the i-th column of the alignment, and where $d(i,j)$ is a distance function to be specified, K is a weighting parameter, and $\varpi(s_i, s_j)$ are the usual base-pairing probabilities, derived from an evolutionary model or the simple heuristic mentioned previously.

We choose a distance function such that two columns are "far away" from each other if there are columns between them that are likely to form a base-pair incompatible with these two columns. Each intermediate column k, with $i < k < j$, is given a weight equal to the probability of that column forming a basepair incompatible with $(k-1, j)$. Figure 3.1 shows an example of the distance function on a partially folded structure. Formally, we define, for $i \leq j$,

$$d(i,j) = \begin{cases} 0 & \text{if } i = j; \\ d(\overline{i}, j) & \text{if } i, \overline{i} \text{ pair and } i \leq \overline{i} \leq j; \\ \beta(i,j) + d(i+1, j) & \text{otherwise.} \end{cases} \tag{3.4}$$

For example, in Figure 3.1, we calculate the distance between i and j, following equation (3.4). We have $d(i,j) = \beta(i,j) + d(i+1,j)$ and $d(i+1,j) = \beta(i+1,j) + d(i+2,j)$ and so on. Further, $d(k,j) = d(\overline{k}, j)$, as k and $\overline{k}$ are paired. Continuing, $d(\overline{k}+1, j) = \beta(\overline{k}, j) + d(\overline{k}+1, j)$, and so on; finally, $d(j,j) = 0$.

Here, $\beta(i,j)$ is the probability that column $i+1$ forms a basepair that is incompatible with (i,j), so that, the events in question being disjoint,

$$\beta(i,j) = 1 - \hat{\mathrm{P}}_{\mathrm{unpaired}}(i+1) - \sum_{k=i+2}^{j-1} \hat{\mathrm{P}}_{\mathrm{paired}}(i+1,k) \; .$$

Thus the posterior pairing probabilities are used to guide the folding of the next iteration (and the posterior probabilities for the wholly unfolded sequence, without the exponential weighting factor, are used to calculate the distance function for the first iteration).

This completes the setup of the kinetic folding algorithm. A summary of the algorithm is given as pseudocode in Figure 3.2, which also shows the expressions for the Inside-Outside and posterior probabilities conditional on existing basepairs.

Since the distance function, as shown in Figure 3.1, allows shortcuts across stems, this distance function implements the physics that, once a basepair has been formed, nearby bases are more likely to pair (Craig et al. 1971). We note that the distance function imposes a certain hierarchy on substructures, making it more attractive to pair interior stems before pairing exterior ones.

It is important to note the effect of the denominator $K|j-i|$ in the exponential factor in equation (3.3): we do not penalise absolute distance between basepairing partners, but rather, the penalisation is relative, comparing the probabilistic weights β to the parameter K.

From a more fundamental point of view, this is a Bayesian approach: the observed "horizontal" relationships between the columns in the alignment

are used to update the prior information from the evolutionary model, which evaluates the "vertical" relationships between the sequences in the alignment. Thus these two parts of the model complement each other. " (Anderson et al. 2013*a*).

3.2 Results

"The full kinetic model for RNA secondary prediction, Oxfold, was benchmarked against PPfold (Sükösd et al. 2012), a parallelised implementation of the Pfold model of (Knudsen & Hein 1999, Knudsen & Hein 2003). We also evaluate the performance of RNAalifold (Hofacker et al. 2002, Bernhart et al. 2008), a thermodynamic model without evolutionary information." (Anderson et al. 2013*a*).

3.2.1 Benchmarking Data and Parameters

"For benchmarking purposes, we have created a curated RNA dataset based on the the Rfam database (Griffiths-Jones et al. 2005). Alignments of homologous RNA sequences with their consensus secondary structure were extracted from among those Rfam seed alignments that bear the "published" tag. From these, 50 alignments with at least 50 sequences were randomly selected. We note that random data selection ensures reliability of results. In a pre-filtering step we discarded outlier sequences with many/long insertions and deletions from each family, using a similar approach to that of PPfold (Sükösd et al. 2012), which does not consider columns with more than 75% of

gaps for pairing. Indels were first determined relative to a family consensus sequence, then a total mismatch score was calculated based on indel lengths, and sequences that had significantly larger mismatch score than the family mean were deleted. Further random selection was performed to reduce these to alignments of five sequences each; the results of Knudsen & Hein (1999,2003), suggest that this suffices to take into account the evolutionary information. Because of the computational complexity of the model (which we discuss further below), we restricted to 41 alignments of length up to 214, with an average length of 105.

The consensus secondary structures given in Rfam may be slightly different from the secondary structures that the individual sequences fold into. In this sense, we cannot say that our secondary structures are experimentally verified, but the approach of comparing predictions to these secondary structures is commonly used in analysis of comparative prediction methods. As with all benchmarks of this nature (Knudsen & Hein 1999, Bernhart et al. 2008), this should be taken into account.

It is known that grammar performance depends on data sets (Rivas et al. 2012). Consequently, it is important to monitor data set dependence, in particular to avoid overfitting. For these reasons, the grammar parameters and evolutionary trees used for benchmarking purposes were those of PPfold. In particular, the grammar underlying our present approach is essentially the simple grammar for which Rivas et al. (2012) did not find evidence of overfitting. The other parameters were chosen heuristically: the parameter δ was set to 0.5 to make the first, local basepair stable enough for its dissociation

time to be long enough for other basepairs to form. This is a trade-off between losing sensitivity at high values of δ and losing PPV at low values of δ. Similarly, K determines the amount of penalisation by the distance function; setting $K = 0.5$ leads to a maximum penalisation of about one order of magnitude." (Anderson et al. 2013*a*).

Algorithm	*F-Score*	*Sensitivity*	*PPV*
RNAalifold	0.704	**0.748**	0.689
PPfold	0.673	0.650	0.728
Iterative without evolutionary model, without distance function	0.684	0.698	0.694
Iterative without evolutionary model, with distance function	0.688	0.696	0.703
Iterative with evolutionary model, without distance function	0.698	0.666	0.780
Oxfold (full kinetic model)	**0.723**	0.688	**0.800**

Table 3.1: Comparison of the performance of different algorithms on the test data set of 41 alignments. The algorithms presented in this paper are compared to PPfold (Sukosd et al., 2012) and RNAalifold (Hofacker et al., 2002; Bernhart et al., 2008). See Methods section for details. The values shown are the averages of the F-score, sensitivity and PPV of the alignments in the test data set. The values shown in bold type are the maximum values in the respective column. Table and caption taken from Table 1, Anderson et al. (2013*a*).

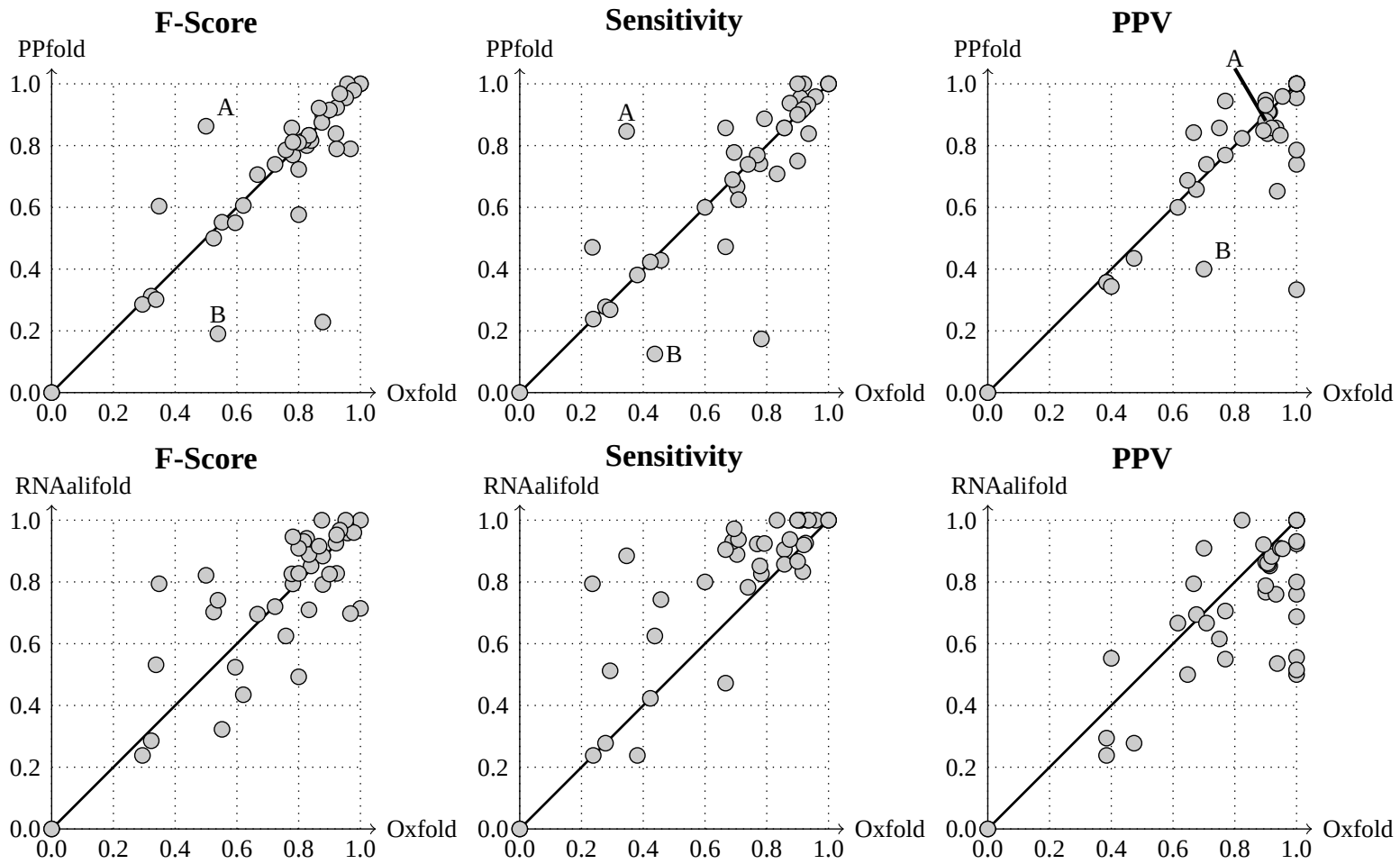


Figure 3.3: Comparison of the F-score, sensitivity and PPV of the consensus structures predicted by Oxfold (the full kinetic model presented in this paper) and those of PPfold (Sukosd et al., 2012) and RNAalifold (Hofacker et al., 2002; Bernhart et al., 2008) respectively, for the sequences in the test data set of 41 alignments. The outliers marked A and B are discussed in the text. Figure and caption taken from Figure 3, Anderson et al. (2013*a*).

3.2.2 Benchmarking Statistics

The predictions of the full kinetic model presented in this paper are compared to those of PPfold and RNAalifold inTable 3.1 and Figure 3.3. "We note that Oxfold performs better, on average, than PPfold in terms of averaged sensitivity, PPV and F-score. Additionally, it has a higher F-score and PPV than RNAalifold, though the thermodynamic model has a higher sensitivity. In particular, we observe that Oxfold has a noticeably higher PPV than PPfold and RNAalifold.

We also note that including an evolutionary model decreases the sensitivity of the algorithm, but greatly increases the PPV. Moreover, the distance function does not seem to be of much use without an evolutionary model. Both of these observations are compatible with the conclusion that the posterior probabilities without the evolutionary model are less reliable (in the sense that they somehow correlate with correctness) than those obtained with an evolutionary model. The fact that the iterative model without distance function, both with the mixed and the unmixed evolutionary model, have essentially the same PPV, gives further weight to this conclusion.

Also, the model with evolutionary information, but without distance function, performs mildly better than PPfold. This is notable, because this method is the iterative method applied to the standard PPfold model. By conditioning on known structure, the grammar model no longer has probability mass contributed from incompatible structures, which one might hope would lead to a better structure prediction.

Since PPfold and Oxfold share the same grammatical framework, outliers in Figure 3.3 can be attributed to the iterative method developed in this paper. We discuss two such outliers, marked A and B in Figure 3.3 (see supplementary data for the corresponding predictions). In outlier A, the prediction cutoff δ is not met by a basepair predicted by PPfold, upon which Oxfold terminates prediction (consequently, the PPVs are very similar, but the sensitivity of PPfold is higher). By contrast, outlier B is an example of an alignment where Oxfold performs noticeably better than PPfold. Without the distance function, Oxfold has zero F-score on this alignment; with this distance function, the F-score rises to 0.5, illustrating how the distance function facilitates the prediction of correct basepairs.

With the interpretation of the difference in equation (3.2) as a measure of the stability of a basepair, and of the posterior pairing probabilities as a measure of the inverse time it takes to form a basepair (discussed in the Methods section), the fact that this model works indicates that RNA folds a stable scaffold before less stable substructures with short dissociation times start to appear and disappear (rather than folding its functional secondary structure while such substructures appear and disappear)." (Anderson et al. 2013*a*).

3.3 Discussion

Here "we have incorporated kinetic effects into a grammatical model for RNA secondary structure prediction by iterative formation of helices and by taking into account some relationships between columns of an alignment by

means of a distance function. Conceptually, introducing a distance function is the analogue, at the level of the emission probabilities of the grammar, of including (albeit possibly different) information about the relationships between columns in the alignment by making the production rules of the grammar more complex. The performance of the kinetic model suggests that the dynamical aspects of RNA folding should not be disregarded in SCFG approaches to RNA secondary structure prediction." (Anderson et al. 2013*a*).

3.3.1 Co-transcriptional Folding

"Incorporating co-transcriptional effects into the model might therefore be a possible next step: Kramer & Mills (1981), have shown that RNA folds as it is being transcribed, usually in the 5' to 3' direction. Thus the 5' end of the RNA molecule is allowed to fold before the sequence has been entirely transcribed, resulting in intermediate structures that do not necessarily exist in the final, functional structure, since the speed of stem formation greatly exceeds the speed of transcription (Gultyaev et al. 1995). Moreover, Meyer & Miklos (2004), have demonstrated "*with statistical significance that co-transcriptional folding strongly influences RNA sequences in two ways: (1) alternative helices that would compete with the formation of the functional structure during co-transcriptional folding are suppressed and (2) the formation of transient structures which may serve as guidelines for the co-transcriptional folding pathway is encouraged*". (Gultyaev et al. 1995), for instance, have incorporated co-transcriptional effects into a thermodynamic model by using a genetic algorithm." (Anderson et al. 2013*a*).

Indeed, recently CoFold (Proctor & Meyer 2013) has incorporated co-transcriptional folding into a thermodynamic model of RNA secondary structure prediction. They use a very similar approach to Anderson et al. (2013*a*), using a scaling function γ of the distance d between nucleotides to create an "affected" free-energy, in a similar way the distance function in Section 3.1.2 affects the posterior base-pairing probabilities. The improvement shown by CoFold over RNAfold and the conservation of co-transcriptional folding pathways discussed above gives hope that incorporating co-transcriptional folding into the Oxfold framework will increase prediction accuracy.

Creating an analogue of the CoFold distance function for the SCFG framework would be one place to start, and to consider the effect phylogenetic models have on predictive accuracy. Alternatively, one could modify the distance function in Section 3.1.2 to create either a 5' or 3' bias. Neither of these would increase the computational complexity of the algorithm. Alternatively, considering increasing length of the sequence or alignment, and re-estimating the posterior distribution at every step, could provide a more biologically realistic framework.

Moreover, it would be desirable to analyse the modelling of co-transcriptional folding in more detail. The SCFG KH99 contains a direction bias. We could define KH99^T as follows

KH99 KH99^T

S	$\rightarrow$	L	**LS**	S	$\rightarrow$	L	**SL**
		.131	.869			.131	.869
L	$\rightarrow$	s	$dF\hat{d}$	L	$\rightarrow$	s	$dF\hat{d}$
		.895	.105			.895	.105
F	$\rightarrow$	$dF\hat{d}$	**LS**	F	$\rightarrow$	$dF\hat{d}$	**SL**
		.788	.212			.788	.212

with the emboldened symbols indicating the differences between the two SCFGs. This would essentially reverse the direction of the RNA sequence. Considering analysis like the predictive quality of KH99^T may give a idea on what can be gained from incorporating co-transcriptional folding within the SCFG framework, and to what extent directionality is already considered.

3.3.2 Pseudoknots

"As mentioned previously, standard SCFG approaches cannot predict pseudoknots (Brown & Wilson 1996). Leaving the framework of stochastic context-free grammars, but still using a formal grammar, it is possible to predict pseudoknotted structures (Rivas & Eddy 2000). The iterative method presented in this paper could be adapted to predict pseudoknots either by adapting the definition of permissible basepairs or by using the methods described by Rivas & Eddy (2000). Similarly, we would expect biophysical folding mechanisms to be conserved in pseudoknotted structures as in non-pseudoknotted structures." (Anderson et al. 2013*a*).

3.3.3 Computational Complexity

"Standard SCFG algorithms for RNA secondary structure prediction have computational complexity $\mathcal{O}(\ell^3)$, where ℓ is the sequence length (Knudsen & Hein 1999). Hence the kinetic model presented in this paper has complexity $\mathcal{O}(\ell^4)$, and a co-transcriptional algorithm along the lines of the algorithm of Gultyaev et al. (1995) (i.e. folding longer and longer subsequences, starting at the 5' end) would have a complexity $\mathcal{O}(\ell^5)$, which makes the algorithms even more expensive than standard prediction approaches. Whereas it is straightforward, even intrinsic, to reuse computations for shorter subsequences in current methods for thermodynamic models, this may seem much more complex in a kinetic model, as the optimum pathway may completely change upon the elongation of a subsequence. However, when the aim is to include co-transcriptional effects, it is not unreasonable to assume that relevant models can be formulated allowing algorithms with complexity lower than $\mathcal{O}(\ell^5)$. This matters especially for co-transcriptional folding, since one expects transcriptional effects to be stronger for longer sequences." (Anderson et al. 2013*a*).

Computational complexity (and actual computational time) was a real concern throughout. For this reason we were not able to test the algorithm on much longer sequences than those in our data set. The problems associated with computational complexity are only to increase if this method is extended to co-transcriptional folding. Problems could be alleviated, for example, by predicting multiple high-reliability helices before re-estimating the distance function, by approximating the re-estimation of parts of the Inside-Outside

table using the previous values present, or by limiting the classes of structures considered (Holmes 2005). These are planned for future work.

3.3.4 Non-Canonical Basepairs

"The existence of non-canonical basepairs is a possible complication in RNA secondary prediction, for long, correct helices with non-canonical basepairs may appear less attractive than short, yet spurious helices. This observation ties the non-canonical basepairs issue somewhat to the geometric, and therefore unphysical, distribution of the helix lengths in the grammar (Rivas et al. 2012). Here, we have addressed that issue by making the pairing of the first basepair of a helix more expensive than that of later pairs (as discussed in the Methods section). The effect of more complex distributions of helix lengths has previously been studied by Dowell & Eddy (2004) and Rivas et al. (2012) who considered more complex grammars, allowing for stacking non-terminals in the grammar.

Nonetheless, RNAalifold (Hofacker et al. 2002, Bernhart et al. 2008) does not allow non-canonical basepairs in its default settings, whereas PPfold (Sükösd et al. 2012) associates very low pairing probabilities with non-canonical basepairs. Some gain in sensitivity might therefore be possible by allowing some non-canonical basepairs in pairs of alignment columns at low probability cost, but more insight into the role of non-canonical basepairs (and a corresponding model) may well be required." (Anderson et al. 2013*a*).

With this in mind, Appendix D attempted to increase the robustness of sin-

gle sequence RNA secondary structure prediction with the CYK algorithm to non-canonical base-pairs. The n free-passes algorithm introduced allowed n non-canonical base-pairs to be predicted with a higher probability than with the basic SCFG model. The method had mixed success, as many alternative sequences produced high-probability structures, and often base-pairs different to the desired non-canonical base-pairs were produced. Given these results, I would conjecture that for comparative prediction the uncertainty allowed for by the evolutionary model allows sufficiently for most types of non-canonical base-pairs (or, indeed, sequencing errors). I would also conjecture that there is already considerable uncertainty in the single sequence model, and allowing more uncertainty for non-canonical base-pairs will not increase prediction accuracy.

3.4 Summary

In this thesis aspects of probabilistic RNA secondary structure models are discussed. The first chapter introduces RNA secondary structure prediction, and in particular stochastic context-free grammars (SCFGs), and considers a novel method for automated design of SCFGs. Many SCFGs are found with a similar predictive quality as those commonly used for RNA secondary structure prediction. The second chapter presents material discussing the affect of alignment quality, evolutionary distance between sequences, and number of sequences in an alignment on RNA secondary structure prediction. By combining statistical alignment and SCFG models we can, in a statistically

sound setting, average structure predictions over the space of alignments to decrease loss created by sub-par alignments. The third chapter sees incorporation of additional biological information about RNA secondary structure formation, particularly helix formation, into the decoding of the SCFG posterior distribution. Combining iterative helix formation, phylogenetic models, and a pseudo-distance function between nucleotides which is re-estimated after each helix forms, the accuracy of comparative RNA secondary structure prediction is improved.

A number of conclusions on RNA secondary structure have already been mentioned. Firstly, it seems unlikely that the underlying SCFG can be improved upon significantly, as discussed in Section 1.3.4. At this point many SCFGs have been tested, and whilst a number of SCFGs have been found that predict different sets of structures, their success seems to have a ceiling. All of the SCFGs found in the automated search were optimised in terms of sensitivity and PPV, and there is no guarantee that their predictions are biologically realistic in terms of an entire structure. We will need to improve the underlying folding model– either the way we define folding space, or the way we move through it– to improve RNA secondary structure prediction.

Secondly, when folding homologous sequences in a comparative setting, it is not surprising to conclude that a poor alignment contributes to poor structure predictions, and that we need several sequences to fold in order to gain adequate evolutionary information. By combining statistical alignment and SCFG models we can, in a statistically sound setting, average structure predictions over the space of alignments to decrease loss created by sub-par

alignments.

Lastly, we can improve the underlying probabilistic folding model by incorporating further biological information. By considering RNA folding kinetics in an evolutionary framework we can decode the posterior distribution produced by the SCFG in an alternative manner to improve comparative prediction. Given the success of CoFold (Proctor & Meyer 2013), it is likely that including a co-transcriptional aspect will also produce success.

3.4.1 Future Perspectives

Future developments in RNA secondary structure prediction will need to come from using additional information in the folding models. The probabilistic framework of SCFGs allows for biological ideas to be incorporated in a straightforward manner. This is well summarised in Rivas (2013): "there seems to be a performance ceiling that affects all methods with complex architectures, ceiling that impacts all scoring schemes with remarkable similarity. This suggests that modeling RNA secondary structure by using intrinsic sequence-based plausible foldability will require the incorporation of other forms of information in order to constraint the folding space and to improve prediction accuracy. This could give an advantage to probabilistic scoring systems since a probabilistic framework is a natural platform to incorporate different sources of information into one single inference problem."

It is hard to see that major improvements in RNA secondary structure modelling will come from slight model tweaks, additional data, or meta-methods.

RNA folds for reasons, and I don't think current models have quite captured these. It may be that for successful models we need to move away from the general folding problem, focusing more on specific families or RNAs. It may be specific genomic regions near the RNA that need to be modelled, considering RNA-protein interactions, for example. It may be that we need to consider further biological information, like three dimensional RNA structure. In my opinion, under the current folding paradigms, the approximate ceiling we have seen from the many attempts at the general RNA secondary structure prediction problem is here to stay. Also, as experimental techniques develop, particularly genome-wide RNA secondary structure determination (Kertesz et al. 2010), it may be there is less demand for improvements in RNA secondary structure prediction.

3.4.2 Further Work

Many potential topics of future work have already been mentioned. Firstly, it would be desirable to incorporate pseudoknot prediction into the Oxfold framework (see Section 3.3.2). Secondly, there is potential for improvement of Oxfold through incorporating co-transcriptional folding (see Section 3.3.1), especially given the improvement seen by CoFold (Proctor & Meyer 2013). Lastly, as discussed in Section 2.3, we could combine Oxfold with statistical alignment techniques to reduce poor predictions introduced by bad alignment.

One area which has not received much attention in this thesis is the evolutionary model, or indeed the alignment model. We have seen the im-

portance of a good alignment quality (Anderson et al. 2013*c*), and we are able to estimate alignment quality from alignment samples and the statistical alignment model (see Appendix C). Alignment quality for RNA could be increased through classifying RNA based on projected alignment quality, then using multiple alignment or structure prediction methods based on this classification. More advanced evolutionary models (Holmes 2004, Bradley & Holmes 2009) which consider correlated evolution beyond the basic base-pair correlation may also improved structure prediction, and this could tie to alignment as well. Another example combines the evolutionary model with an HMM to better take account of insertion-deletion events as in Fang et al. (2008). The HMM considers evolution in structured and unstructured regions, and results suggest that this yields better accuracy than the SCFG approach with a simple evolutionary model.

Similarly, three-dimensional structure prediction techniques are still developing but will likely become more prominent as technology and data advance. Particularly, the possibility of extending SCFG prediction to three-dimensional structures will be an interesting task. Similarly, as RNA roles in gene expression become better understood, structural features of RNA may well be used in combination with network inference techniques to better understand gene regulatory networks.

Lastly, attention should also be paid to the software resulting from this work. Currently all of which is open-source and publicly available[1]. Section 3.3.3 briefly discusses improvements that could be made to make Oxfold faster.

[1] `http://www.stats.ox.ac.uk/~anderson/code.html`, `http://statalign.github.com/statalign/`

Further algorithmic developments, such as sequence selection, or, like PPfold, considering SHAPE data would also create further functionality. Additionally, Oxfold is currently only operable via command line. Creating a GUI, allows for easy use and interpretation of results. A web server could be created, resulting in an easy interface for users which does not require program installation, software compatability, or time spent managing program updates or configuration. Online tutorials and demonstrations can make the software easier to use, process output, and link to external resources. These will increase impact, but more importantly usefulness for the bioinformatics and biological community.

Appendix A

Frnakenstein: Inverse Folding of RNA

With the growing application of RNA and DNA structure design in synthetic biology and bio-nanotechnology, the development of software for sequence design is key, for example in producing lock-and-key type mechanisms (Andersen et al. 2009). Due to the high experimental cost, it is not practical to use experimental data to verify the folding of sequences into RNA structures, so search heuristics involving structure prediction programs must be relied upon. These search heuristics then, given a particular RNA secondary structure, find a sequence of base pairs that will fold into this structure. This is known as the RNA inverse folding problem. The sequences designed for are verified by RNA folding programs, typically RNAfold (Hofacker et al. 1994) is used. This allows the inverse folding problem to be addressed completely *in silico*, using predicted structures as a proxy for the true structure of a

designed sequence. Sequences produced by the computational methods can then be synthesised, with the expectation that they fold into the required structure.

Frnakenstein (Lyngsø et al. 2012, Anderson et al. 2013*b*), uses a genetic algorithm implementation to solve the inverse RNA problem. Frnakenstein uniquely uses base-pair probabilities and stability measures from the Boltzmann distribution (Hofacker et al. 1994), or stochastic context-free grammar posterior distribution (Sükösd et al. 2012), to guide mutations and crossover point selection, and proved most effective at structure design when compared to other inverse folding programs (Lyngsø et al. 2012). Many different positional fitness schemes are available to enable flexibility, and users can employ several different optimisation criteria to end up with sequences emphasising different factors. These are particularly important for multi-target inverse folding, which is useful for designing, for example, riboswitches. Frnakenstein is the only available software which designs for multiple structures as targets, necessary for designing, for example, riboswitches, which are known to have important genomic functions. Additionally, the program avoids the heavy C-G base pair bias seen in many inverse folding programs, thus producing more biologically realistic sequences. Frnakenstein 2.0 includes support for pseudoknotted structures using pknotsRG (Reeder et al. 2007), improved search heuristics using pareto optimal sets on compound fitnesses, support for PP-fold, sequence motifs, parallelisation, and a GUI for easy use. Frnakenstein 2.0 is open-source and publicly available.

Appendix B

Information Entropy for SCFGs

"The information entropy H of a probability distribution P with a countable set of events $\mathcal{X}$ is defined as:

$$H(\mathrm{P}) = -\sum_{x \in \mathcal{X}} \mathrm{P}[x] \log_2(\mathrm{P}[x]) \ . \tag{B.1}$$

The information entropy is a measure for the "spread" of the probability distribution, and has well-defined lower and upper bounds. The minimum entropy of 0 occurs when there is only one outcome with probability 1, and the maximum entropy of $\log_2(n)$ occurs when there are n possible outcomes, each with probability $1/n$, that is the uniform distribution. When the base of the logarithm is 2, the entropy is measured in bits. For a probability distribution, an entropy of k bits indicates that the expected value of the information content of observing a single outcome is k bits. In the context of secondary structure prediction, a low entropy therefore indicates that few

secondary structures dominate the probability space, whereas a high entropy indicates a more even probability distribution over possible secondary structures. Thus, information entropy is a useful single quantity to characterize the underlying probability distribution of secondary structures.

The information entropy of the probability distribution over the possible secondary structures generated by a phylo-SCFG can be obtained using expected rule frequencies, which can be computed using the inside-outside algorithm (Sükösd et al. 2013)" (Anderson et al. 2013*c*). This calculation holds for both ambiguous and unambiguous SCFGs. For interpretations about the spread of probability mass over RNA secondary structure space, unambiguous SCFGs must be used, as there is a one-to-one correspondence between SCFG derivations and RNA secondary structures. Information entropy can be useful for model analysis, and give potential reasons for poor predictions (Sükösd et al. 2013).

After calculating information entropy for phylo-SCFGs in Sükösd et al. (2013), we calculated a sample information entropy in (Anderson et al. 2013*c*), which we prove asymptotically converges to the true information entropy, using the same proof as in Section 2.2.2. "Figure B.1 gives information entropy for 3 different representative RNA families from the StatAlign dataset. On each graph, the information entropy for each of 1000 statistical alignment samples is given, as well as the alignment consensus entropy. The leftmost figure is one where the alignment samples were very similar, the rightmost figure where the alignment samples were very different, and the middle figure closer to the median value of alignment sample similarity." (Anderson et al. 2013*c*).

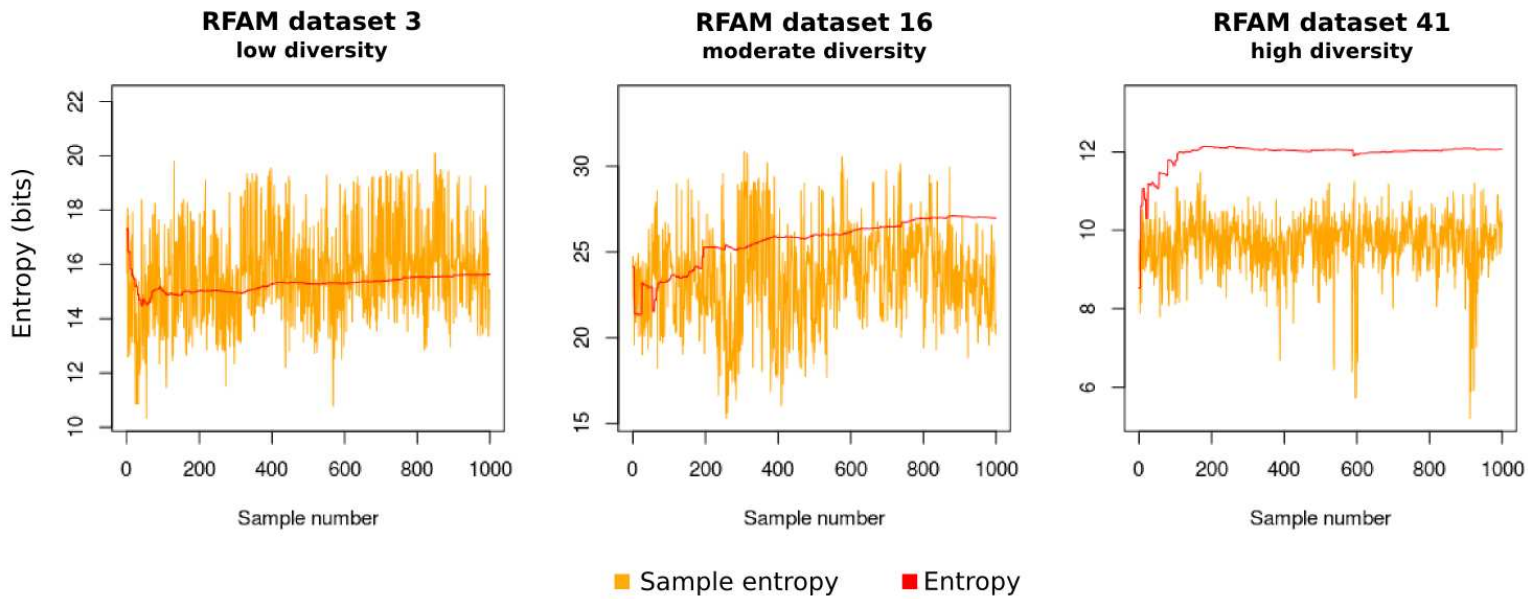


Figure B.1: Information entropy calculated for each alignment sample (yellow) and the alignment consensus entropy (blue). The leftmost figure is one where the alignment samples were very similar, the rightmost figure where the alignment samples were very different, and the middle figure closer to the median value of alignment sample similarity. See Methods section for details on entropy calculation. Figure and caption taken from Figure 4, Anderson *et al.*, (2013*c*).

Appendix C

Predicting Alignment and Prediction Quality

"Given the results concerning accuracy lost as alignment quality decreases, it would be desirable to be able to predict alignment quality, with the hope of predicting structure prediction quality. This has previously been attempted in Schwartz et al. (2004). First, the sequences were aligned with ClustalW (Chenna et al. 2003). The sequences were then re-aligned using 4 other programs (Align-m (Walle et al. 2004), MUSCLE (Edgar 2004), Prob-Cons (Do et al. 2005), and T-Coffee (Notredame et al. 2000)) and the similarity between the alignment generated using ClustalW to each of the 4 other alignments was measured. The maximum of the 4 similarities, $\max(g)$, was chosen as a predictor of alignment quality. The authors of Schwartz et al. (2004) detected a strong correlation between the true similarity (the similarity between the ClustalW alignment and a reference alignment) and $\max(g)$.

We implemented a modified version of this method. For a given set of input sequences we aligned with both AMAP (Schwartz et al. 2004) and with StatAlign, obtaining the maximum posterior decoding alignment (MPD alignment) from StatAlign. The similarity between the AMAP alignment and the MPD alignment was used as our predicted similarity measure. This produced a strong correlation between our predicted similarity and true similarity, with an adjusted R-squared value of 0.6524.

We also implemented another method, which calculates an estimate of the expected similarity score using posterior probabilities from the MPD alignment. For each column, we might expect that a posterior probability close to 1 would contribute a score of close to 1 to the similarity measure. So our predicted alignment distance is just the average of the column posterior probabilities. [This new method gave an] adjusted R-squared value of 0.8403. Our new predicted similarity can be calculated efficiently, and is a strong predictor of true similarity to the reference alignment."(Anderson et al. 2013*c*)

"Given strong correlations between the alignment quality and the structure prediction quality, we might expect that we could find a predictor of structure prediction accuracy. By "integrating out" alignment uncertainty, we may find a reliability score which makes the PPfold reliability score more, well, reliable. To test this, we predicted accuracy for one of the five-sequence alignments of each family and then tested the predicted accuracy against the true accuracy. The PPfold reliability score produced an adjusted R^2 score of 0.252 when considering correlation with the true F-scores.

For our new reliability score, we adjusted the PPfold reliability score to consider only base-pairs, as the F-score considers only base-pairs (i.e. ignored unpaired nucleotide reliabilities). We then performed linear regression with the average of the MPD column probabilities, the information entropy of the alignment space, and this pairs-only reliability score against the known F-score measure. This multiple regression improved the reliability score significantly. [F]or a randomly chosen five-sequence alignments from each family of the StatAlign dataset[, the] adjusted R^2 value with the new reliability measure improved to 0.496. We suggest that these results indicate that alignment quality will affect structure prediction quality, the overall predictive quality is determined by the actual structure prediction model, either the thermodynamic approach, or by using SCFGs."(Anderson et al. 2013*c*)

Appendix D

An n-free-passes CYK Algorithm for Error-correction and Non-canonical Base-pairs

"The n-free-passes algorithm is a modified version of the CYK algorithm, which allows non-canonical base-pairs to be predicted with a higher probability than otherwise indicated by grammar training (see Algorithm 2). To illustrate the general idea, consider the sequence GGGAAAAACAC, with secondary structure '(((.....)))'. Note the non-canonical base-pair G-A in positions 2 and 10. With no free-passes, the CYK algorithm with a typical RNA SCFG would choose the most likely structure, '.....', because the probability of the base-pair GA is very low. With one free-pass, the CYK algorithm is allowed instead to use its free-pass to draw from an alternative probability distribution, where the free-pass probability P'_d[GA] is higher than the stan-

dard probability P_d[GA], and the correct structure '(((.....)))' can become the most likely structure." (Anderson et al. 2013*d*).

"Figure D.1 gives three examples of known structures and corresponding predictions with an increasing number of free passes. Subfigure A is an example of a structure prediction which gets considerably worse when free-passes are allowed, Subfigure B an example of a structure prediction which gets considerably better when free-passes are allowed, and Subfigure C an example of a structure prediction which does not change much when free-passes are allowed." (Anderson et al. 2013*d*).

To test the algorithm we used data where natural non-canonical base-pairs occured, and data where we simulated non-canonical base-pairs. The results were in general mixed, with a slight improvement for the simulated data, but accuracy worsened for data with natural non-canonical base-pairs. "In the case where the sequences include naturally occurring non-canonical base-pairs or no non-canonical base-pairs at all, the original KH99 model is already as well trained as possible to predict the correct structure. The *n*-free-passes approach does not capture any additional non-canonical pairing signals. If it does add non-canonical base-pairs, these are chosen more-or-less randomly among the large number of available options, and will therefore in a large number of cases be incorrect. In the case where the sequences include sequencing errors, however, the original grammatical model is no longer appropriate. The appearing "non-canonical" base-pairs include the additional signal that an alternative, closely related sequence would be significantly more adequately explained by the model." (Anderson et al. 2013*d*).

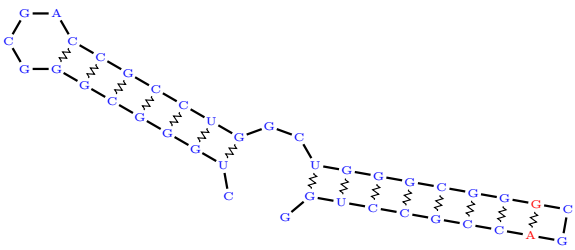
Algorithm 2 CYK Algorithm with at most n free-passes

for $m = 0$ **to** n **do**
  for $i = 0$ **to** $|s|$ **do**
    for $j = i$ **to** $|s|$ **do**
      $C(m, U, i, j) = 0$
      if $i = j$ **then**
        $C(m, U, i, i) = \mathrm{P}[U \to .]\ \mathrm{P}_u[s[i]]$
      if $i < j$ **then**

$$C(m, U, i, j) = \max \begin{cases} \max\limits_{0 \le l \le m} \max\limits_{i \le k < j} \max\limits_{U \to VW} [\mathrm{P}[U \to VW]\ C(l, V, i, k)\ C(m - l, W, k + 1, j)] \\ \max\limits_{U \to (V)} [\mathrm{P}[U \to (V)]\ \mathrm{P}_d[s[i]s[j]]\ C(m, V, i + 1, j - 1)] \\ \max\limits_{U \to (V)} [\mathrm{P}[U \to (V)]\ \mathrm{P}'_d[s[i]s[j]]\ C(m - 1, V, i + 1, j - 1)\ 1_{m>0}] \end{cases}$$

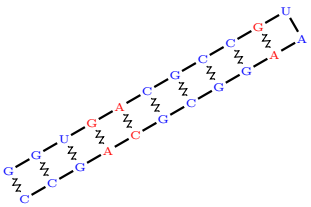
Pseudocode for the n-free-passes CYK algorithm for SCFGs in double-emission normal form is given in Algorithm 2. For a sequence s, $C(n, S, 1, |s|)$ then gives the maximum probability of deriving the sequence s with n free-passes. Backtracking can be done to find the secondary structure associated with this probability in the same way as with the original CYK algorithm. Algorithm and caption taken from Anderson et al. (2013*d*).

A



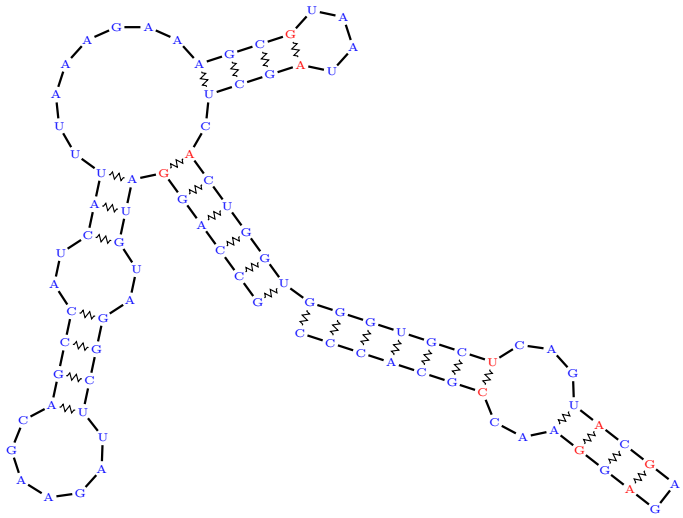
```
Sequence:          CUGGGCGGGCGACCGCCUGGCUGGGCGGGCCGACCGCCUGG
True Structure:    .(((((((....)))))))..(((((((..)))))))).
Zero Free-Passes:  (((((((((....)))))))))(((((((((....)))))))))
One Free-Pass:     (((((((((....)))))))))(((((((((....)))))))))
Two Free-Passes:   ((((((((((((((((((..))))))))))))))))))
```

B



```
Sequence:          GGUGAACGCCGUAAGGCGCAGCC
True Structure:    ((((((((((..))))))))))
Zero Free-Passes:  .....((((....))))).....
One Free-Pass:     ((((((((((....))))))))))
Two Free-Passes:   ((((((((((....))))))))))
```

C



```
Sequence:          GCCACGAUGUAGGCUUAGAAGCAGCCAUCAUUUAAAGAAAGCGUAAUAGCUCACUGGUGGGUGCUCAGUACGAGAGGAACGCACCC
True Structure:    (((((((((..((((.......))))..)))).........(((((....)))))..)))))).(((((((...((((..))))..)))))))
Zero Free-Passes:  (((((.....(((((.......))))).............(((((.....)))))..)))))).(((((.................)))))
One Free-Pass:     (((((.....(((((.......))))).............(((((.....)))))..)))))).(((((.................)))))
Two Free-Passes:   ((((((((((((((.......)))))))))).......(((((.....)))))..)))))).(((((.................)))))
Three Free-Passes: ((((((((((((((.......)))))))))).......(((((.....)))))..)))))).(((((.................)))))
Four Free-Passes:  ((((((((((((((..))))))))))).........(((((.....)))))..)))))).(((((.................)))))
```

Figure D.1: Non-canonical base-pairs are highlighted in red in both the sequence and graphical representation of the secondary structure. Figure taken from Anderson et al. (2013*d*).

Bibliography

Andersen, E. S., Dong, M., Nielsen, M. M., Jahn, K., Subramani, R., Mamdouh, W., Golas, M. M., Sander, B., Stark, H. & Oliveira, C. L. (2009), 'Self-assembly of a nanoscale DNA box with a controllable lid', *Nature* **459**(7243), 73–76.

Andersen, E. S., Rosenblad, M. A., Larsen, N., Westergaard, J. C., Burks, J., Wower, I. K., Wower, J., Gorodkin, J., Samuelsson, T. & Zwieb, C. (2006), 'The tmRDB and SRPDB resources', *Nucleic acids research* **34**(suppl 1), D163–D168.

••Anderson, J. W. J. (2013), *Stochastic Context-Free Grammars and RNA Secondary Structure Prediction*, To appear in 'Genome Analysis: Current Procedures and Applications', Horizon Press.

••Anderson, J. W. J., Haas, P. A., Mathieson, L.-A., Volynkin, V., Lyngsø R., Tataru, P. & Hein, J. (2013*a*), 'Oxfold: kinetic folding of RNA using stochastic context-free grammars and evolutionary information', *Bioinformatics* **29**(6), 704–710.

••Anderson, J. W. J., Lake, C., Teufel, A., Weir, W., Lyngsø R. & Hein, J.

(2013*b*), 'Frnakenstein 1.0: Inverse folding of rna secondary structures with pseudoknots. *In preparation*'.

••Anderson, J. W. J., Novák, A., Sükösd, Z., Golden, M., Arunapuram, P., Edvardsson, I. & Hein, J. (2013*c*), 'Quantifying Variances in Comparative RNA Secondary Structure Prediction', *In press.*

••Anderson, J. W. J., Sükösd, Z., Pedersen, C. S. & Hein, J. (2013*d*), 'An n-free-passes CYK algorithm for prediction of non-canonical base-pairs and error correction in RNA secondary structure. `http://www.stats.ox.ac.uk/~anderson/Downloads/Anderson_n_free_passes.pdf`'.

••Anderson, J. W. J., Tataru, P., Staines, J., Hein, J. & Lyngsø, R. (2012), 'Evolving stochastic context-free grammars for RNA secondary structure prediction', *BMC Bioinformatics* **13**(1), 78.

Andronescu, M., Bereg, V., Hoos, H. & Condon, A. (2008), 'RNA STRAND: The RNA secondary structure and statistical analysis database', *BMC Bioinformatics* **9**(1), 340.

••Arunapuram, P., Edvardsson, I., Golden, M., Anderson, J. W. J., Novák, A., Sükösd, Z. & Hein, J. (2013), 'StatAlign 2.0: combining statistical alignment with RNA secondary structure prediction', *Bioinformatics* **29**(5), 654–655.

Bernhart, S., Hofacker, I., Will, S., Gruber, A. & Stadler, P. (2008), 'RNAalifold: improved consensus structure prediction for RNA alignments', *BMC Bioinformatics* **9**(1), 474.

Bradley, R. K. & Holmes, I. (2009), 'Evolutionary Triplet Models of Structured RNA', *PLoS Comput Biol* **5**(8), e1000483.

Bradley, R. K., Pachter, L. & Holmes, I. (2008), 'Specific alignment of structured RNA: stochastic grammars and sequence annealing', *Bioinformatics* **24**(23), 2677–2683.

Brown, J. W. (1999), 'The Ribonuclease P Database', *Nucleic acids research* **27**(1), 314–314.

Brown, M. & Wilson, C. (1996), RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search., *in* 'Proc. Pac. Symp. Biocomput.', pp. 109–125.

Cannone, J. J., Subramanian, S., Schnare, M. N., Collett, J. R., D'Souza, L. M., Du, Y., Feng, B., Lin, N., Madabusi, L. V. & Møller, K. M. (2002), 'The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs', *BMC bioinformatics* **3**(1), 2.

Chappelier, J. & Rajman, M. (1998), 'A generalized CYK algorithm for parsing stochastic CFG', *Proc. of 5eme conference sur le Traitement Automatique du Langage Naturel.* pp. 153–157.

Chen, J.-L. & Greider, C. W. (2005), 'Functional analysis of the pseudoknot structure in human telomerase RNA', *Proceedings of the National Academy of Sciences of the United States of America* **102**(23), 8080–8085.

Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T. J., Higgins, D. G. & Thompson, J. D. (2003), 'Multiple sequence alignment with the Clustal series of programs', *Nucleic acids research* **31**(13), 3497–3500.

Chivian, D. & Baker, D. (2006), 'Homology modeling using parametric alignment ensemble generation with consensus and energy-based model selection', *Nucleic acids research* **34**(17), e112–e112.

Chomsky, N. (1956), 'Three models for the description of language', *Information Theory, IRE Transactions on* **2**(3), 113–124.

Craig, M. E., Crothers, D. M. & Doty, P. (1971), 'Relaxation kinetics of dimer formation by self complementary oligonucleotides', *Journal of Molecular Biology* **62**(2), 383–401.

Danilova, L. V., Pervouchine, D. D., Favorov, A. V. & Mironov, A. A. (2006), 'RNAKinetics: a web server that models secondary structure kinetics of an elongating RNA', *Journal of bioinformatics and computational biology* **4**(02), 589–596.

de Almeida Ribeiro, E., Beich-Frandsen, M., Konarev, P. V., Shang, W., Veerek, B., Kontaxis, G., Hmmerle, H., Peterlik, H., Svergun, D. I., Blsi, U. & Djinovi-Carugo, K. (2012), 'Structural flexibility of rna as molecular basis for hfq chaperone function', *Nucleic acids research* **40**(16), 8072–8084.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the Royal Statistical Society.Series B (Methodological)* pp. 1–38.

Ding, Y., Chan, C. Y. & Lawrence, C. E. (2005), 'RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble', *RNA* **11**(8), 1157–1166.

Do, C. B., Mahabhashyam, M. S. P., Brudno, M. & Batzoglou, S. (2005), 'Probcons: Probabilistic consistency-based multiple sequence alignment', *Genome research* **15**(2), 330–340.

Do, C. B., Woods, D. A. & Batzoglou, S. (2006), 'CONTRAfold: RNA secondary structure prediction without physics-based models', *Bioinformatics* **22**(14), e90–e98.

Doose, G. & Metzler, D. (2012), 'Bayesian sampling of evolutionarily conserved RNA secondary structures with pseudoknots', *Bioinformatics* **28**(17), 2242–2248.

Dowell, R. & Eddy, S. (2004), 'Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction', *BMC Bioinformatics* **5**(1), 71.

Durbin, R., Eddy, S. R., Krogh, A. & Mitchison, G. (1998), *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge university press.

Edgar, R. C. (2004), 'MUSCLE: multiple sequence alignment with high accuracy and high throughput', *Nucleic acids research* **32**(5), 1792–1797.

Engelen, S. & Tahi, F. (2007), 'Predicting RNA secondary structure by the comparative approach: how to select the homologous sequences', *BMC Bioinformatics* **8**(1), 464.

Engelen, S. & Tahi, F. (2010), 'Tfold: efficient in silico prediction of non-coding RNA secondary structures', *Nucleic acids research* **38**(7), 2453–2466.

Fang, X.-Y., Luo, Z.-G. & Wang, Z.-H. (2008), 'Predicting RNA secondary structure using profile stochastic context-free grammars and phylogenic analysis', *Journal of Computer Science and Technology* **23**(4), 582–589.

Felsenstein, J. (1981), 'Evolutionary trees from DNA sequences: A maximum likelihood approach', *Journal of Molecular Evolution* **17**(6), 368–376.

Freyhult, E., Gardner, P. & Moulton, V. (2005), 'A comparison of RNA folding measures', *BMC Bioinformatics* **6**(1), 241.

Gahura, O., Hammann, C., Valentova, A., Puta, F. & Folk, P. (2011), 'Secondary structure is required for 3' splice site recognition in yeast', *Nucleic acids research* **39**(22), 9759–9767.

Gardner, P. & Giegerich, R. (2004), 'A comprehensive comparison of comparative RNA structure prediction approaches', *BMC Bioinformatics* **5**(1), 140.

Gardner, P. P., Wilm, A. & Washietl, S. (2005), 'A benchmark of multiple sequence alignment programs upon structural RNAs', *Nucleic acids research* **33**(8), 2433–2439.

Grefenstette, J. J. (1986), 'Optimization of control parameters for genetic algorithms', *Systems, Man and Cybernetics, IEEE Transactions on* **16**(1), 122–128.

Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R. & Bateman, A. (2005), 'Rfam: annotating non-coding RNAs in complete genomes', *Nucleic acids research* **33**(suppl 1), D121–D124.

Gultyaev, A. P., van Batenburg, F. H. D. & Pleij, C. W. A. (1995), 'The computer simulation of RNA folding pathways using a genetic algorithm', *Journal of Molecular Biology* **250**(1), 37–51.

Harmanci, A., Sharma, G. & Mathews, D. (2011), 'TurboFold: Iterative probabilistic estimation of secondary structures for multiple RNA sequences', *BMC Bioinformatics* **12**(1), 108.

Hein, J., Wiuf, C., Knudsen, B., Moller, M. B. & Wibling, G. (2000), 'Statistical alignment: computational properties, homology testing and goodness-of-fit', *Journal of Molecular Biology* **302**(1), 265–279.

Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, L., Tacker, M. & Schuster, P. (1994), 'Fast folding and comparison of RNA secondary structures.', *Chemical Monthly* **125**(2), 167–188.

Hofacker, I. L., Fekete, M. & Stadler, P. F. (2002), 'Secondary structure prediction for aligned RNA sequences', *Journal of Molecular Biology* **319**(5), 1059–1066.

Holmes, I. (2004), 'A probabilistic model for the evolution of RNA structure', *BMC Bioinformatics* **5**(1), 166.

Holmes, I. (2005), 'Accelerated probabilistic inference of RNA structure evolution', *BMC bioinformatics* **6**(1), 73.

Hopcroft, J. E. & Ullman, J. D. (1979), *Introduction to Automata Theory, Languages and Computation.*, Addison-Wesley.

Jagota, A., Lyngsø R. & Pedersen, C. N. S. (2001), *Comparing a hidden Markov model and a stochastic context-free grammar*, Algorithms in Bioinformatics, Springer, pp. 69–84.

Katoh, K., ichi Kuma, K., Toh, H. & Miyata, T. (2005), 'MAFFT version 5: improvement in accuracy of multiple sequence alignment', *Nucleic acids research* **33**(2), 511–518.

Kertesz, M., Wan, Y., Mazor, E., Rinn, J. L., Nutter, R. C., Chang, H. Y. & Segal, E. (2010), 'Genome-wide measurement of RNA secondary structure in yeast', *Nature* **467**(7311), 103–107.

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

Knudsen, B. & Hein, J. (2003), 'Pfold: RNA secondary structure prediction using stochastic context-free grammars', *Nucleic acids research* **31**(13), 3423–3428.

Kramer, F. R. & Mills, D. R. (1981), 'Secondary structure formation during RNA synthesis', *Nucleic acids research* **9**(19), 5109–5124.

Krogh, A., Brown, M., Mian, I. S., Sjolander, K. & Haussler, D. (1994), 'Hidden markov models in computational biology: Applications to protein modeling', *Journal of Molecular Biology* **235**(5), 1501–1531.

Lari, K. & Young, S. J. (1990), 'The estimation of stochastic context-free grammars using the inside-outside algorithm', *Computer Speech and Language* **4**(1), 35–56.

Lefebvre, F. (1996), 'A grammar-based unification of several alignment and folding algorithms.', *Proc Int Conf Intell Syst Mol Biol.* **4**, 143–154.

Loh, E., Memarpour, F., Vaitkevicius, K., Kallipolitis, B. H., Johansson, J. & Sondén, B. (2012), 'An unstructured 5'-coding region of the prfA mRNA is required for efficient translation', *Nucleic acids research* **40**(4), 1818–1827.

••Lyngsø R., Anderson, J., Sizikova, E., Badugu, A., Hyland, T. & Hein, J. (2012), 'Frnakenstein: multiple target inverse RNA folding', *BMC Bioinformatics* **13**(1), 260.

Markham, N. R., Zuker, M., Keith, J. M. & Walker, J. M. (2008), *UNAFold*, Bioinformatics, Humana Press, pp. 3–31. Methods in Molecular Biology; SP: 3.

Mathews, D., Sabina, J., Zuker, M. & Turne, D. (1999), 'Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure', pp. 911–940.

McCaskill, J. S. (1990), 'The equilibrium partition function and base pair binding probabilities for RNA secondary structure', *Biopolymers* **29**(6-7), 1105–1119.

Meyer, I. & Miklos, I. (2004), 'Co-transcriptional folding is encoded within RNA genes', *BMC Molecular Biology* **5**(1), 10. 15298702.

Meyer, I. M. (2007), 'A practical guide to the art of RNA gene prediction', *Briefings in bioinformatics* **8**(6), 396–414.

Miklos, I., Novák, A., Dombai, B. & Hein, J. (2008), 'How reliably can we predict the reliability of protein structure predictions?', *BMC Bioinformatics* **9**(1), 137.

Moulton, V., Zuker, M., Steel, M., Pointon, R. & Penny, D. (2000), 'Metrics on RNA secondary structures', *Journal of Computational Biology* **7**(1-2), 277–292.

Nawrocki, E. P., Kolbe, D. L. & Eddy, S. R. (2009), 'Infernal 1.0: inference of RNA alignments', *Bioinformatics* **25**(10), 1335–1337.

Nebel, M. & Scheid, A. (2011), 'Evaluation of a sophisticated SCFG design for RNA secondary structure prediction', *Theory in Biosciences* **130**(4), 313–336.

Notredame, C., Higgins, D. & Heringa, J. (2000), 'T-coffee: A novel method for fast and accurate multiple sequence alignment', *Journal of Molecular Biology* pp. 205–217.

Novák, A., Miklos, I., Lyngsø R. & Hein, J. (2008), 'StatAlign: an extendable software package for joint Bayesian estimation of alignments and evolutionary trees', *Bioinformatics* **24**(20), 2403–2404.

Nussinov, R., Pieczenik, G., Griggs, J. R. & Kleitman, D. J. (1978), 'Algorithms for loop matchings', *SIAM Journal on Applied Mathematics* **35**(1), 68–82.

Pipas, J. M. & McMahon, J. E. (1975), 'Method for predicting rna secondary structure', *Proceedings of the National Academy of Sciences* **72**(6), 2017–2021.

Proctor, J. R. & Meyer, I. M. (2013), 'CoFold: an RNA secondary structure prediction method that takes co-transcriptional folding into account', *Nucleic acids research*.

Rabiner, L. R. (1989), 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proceedings of the IEEE* **77**(2), 257–286.

Reeder, J., Steffen, P. & Giegerich, R. (2005), 'Effective ambiguity checking in biosequence analysis', *BMC Bioinformatics* **6**(1), 153.

Reeder, J., Steffen, P. & Giegerich, R. (2007), 'pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows', *Nucleic acids research* **35**(suppl 2), W320–W324.

Rivas, E. (2013), 'The four ingredients of single-sequence RNA secondary structure prediction: A unifying perspective. `http://selab.janelia.org/publications/Rivas13/Rivas13-preprint.pdf`'.

Rivas, E. & Eddy, S. R. (1999), 'A dynamic programming algorithm for RNA structure prediction including pseudoknots', *Journal of Molecular Biology* **285**(5), 2053–2068.

Rivas, E. & Eddy, S. R. (2000), 'The language of RNA: a formal grammar that includes pseudoknots', *Bioinformatics* **16**(4), 334–340.

Rivas, E., Lang, R. & Eddy, S. R. (2012), 'A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more', *RNA* **18**(2), 193–212.

Sakakibara, Y., Brown, M., Hughey, R., Mian, I. S., Sjlander, K., Underwood, R. C. & Haussler, D. (1994), 'Stochastic context-free grammers for tRNA modeling', *Nucleic acids research* **22**(23), 5112–5120.

Scheid, A. & Nebel, M. (2012), 'Evaluating the Effect of Disturbed Ensemble Distributions on SCFG Based Statistical Sampling of RNA Secondary Structures', *BMC Bioinformatics* **13**(1), 159. M3: 10.1186/1471-2105-13-159.

Schwartz, A. S. & Pachter, L. (2007), 'Multiple alignment by sequence annealing', *Bioinformatics* **23**(2), e24–e29.

Schwartz, A. S., Myers, E. W. & Pachter, L. (2004), 'Alignment metric accuracy `http://arxiv.org/abs/q-bio/0510052`'.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Soding, J., Thompson, J. D. & Higgins, D. G. (2011), 'Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega', *Molecular Systems Biology*.

Sprinzl, M., Horn, C., Brown, M., Ioudovitch, A. & Steinberg, S. (1998), 'Compilation of tRNA sequences and sequences of tRNA genes', *Nucleic acids research* **26**(1), 148–153.

••Sükösd, Z., Knudsen, B., Anderson, J. W. J., Novák, A., Kjems, J. & Pedersen, C. (2013), 'Characterising RNA secondary structure space using information entropy', *BMC Bioinformatics* **14**, S22.

Sükösd, Z., Knudsen, B., Kjems, J. & Pedersen, C. N. S. (2012), 'PPfold 3.0: fast RNA secondary structure prediction using phylogeny and auxiliary data', *Bioinformatics* **28**(20), 2691–2692.

Tavaré, S. (1986), 'Some probabilistic and statistical problems in the analysis of DNA sequences', *Lect.Math.Life Sci* **17**, 57–86.

Thorne, J., Kishino, H. & Felsenstein, J. (1991), 'An evolutionary model for maximum likelihood alignment of dna sequences'.

Thorne, J. L., Kishino, H. & Felsenstein, J. (1992), 'Inching toward reality: An improved likelihood model of sequence evolution', *Journal of Molecular Evolution* **34**(1), 3–16.

Tinoco, I., Puglisi, J. & Wyatt, J. (1990), *RNA folding*, Nucleic Acids and Molecular Biology 4, Springer, pp. 205–226.

Vitreschak, A. G., Rodionov, D. A., Mironov, A. A. & Gelfand, M. S. (2004), 'Riboswitches: the oldest mechanism for the regulation of gene expression?', *Trends in Genetics* **20**(1), 44–50.

Walle, I. V., Lasters, I. & Wyns, L. (2004), 'Align-m- a new algorithm for multiple alignment of highly divergent sequences', *Bioinformatics* **20**(9), 1428–1435.

Wang, L.-S., Leebens-Mack, J., Wall, P. K., Beckmann, K., de Pamphilis,

C. W. & Warnow, T. (2011), 'The impact of multiple protein sequence alignment on phylogenetic estimation', *Computational Biology and Bioinformatics, IEEE/ACM Transactions on* **8**(4), 1108–1119.

Washietl, S., Hofacker, I. L., Lukasser, M., Huttenhofer, A. & Stadler, P. F. (2005), 'Mapping of conserved RNA secondary structures predicts thousands of functional noncoding RNAs in the human genome', *Nat Biotech* **23**(11), 1383–1390.

Westbrook, J., Feng, Z., Chen, L., Yang, H. & Berman, H. M. (2003), 'The Protein Data Bank and structural genomics', *Nucleic acids research* **31**(1), 489–491.

Won, K.-J., Hamelryck, T., Prgel-Bennett, A. & Krogh, A. (2007), 'An evolutionary method for learning HMM structure: prediction of protein secondary structure', *BMC bioinformatics* **8**(1), 357.

Xayaphoummine, A., Bucher, T. & Isambert, H. (2005), 'Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots', *Nucleic acids research* **33**(suppl 2), W605–W610.