Viral RNA secondary structure prediction

Zsuzsanna Sükösd

PhD thesis
This thesis is submitted to Science and Technology at Aarhus University, Denmark, in order to fulfill the requirements for obtaining the PhD degree in Nanoscience, with specialization in Bioinformatics. The studies have been carried out under the supervision of Jørgen Kjems at the Interdisciplinary Nanoscience Center and co-supervision of Christian Storm Pedersen at the Bioinformatics Research Centre.
## Contents

**Summary in English and Danish**  iv  
**Preface**  vi  
  Short outline  vi  
  Acknowledgements  vii  
**Publications**  ix  
**Abbreviations**  xi  

### 1 Introduction  1  
  1.1 RNA as a macromolecule  1  
  1.2 Experimental methods for determining RNA secondary structure  6  
  1.3 Dynamic programming for RNA secondary structure prediction  8  
  1.4 The nearest-neighbour thermodynamic model  10  
  1.5 SCFGs for RNA folding  14  
  1.6 Comparison of SCFG-based and thermodynamic methods  20  
  1.7 RNA molecular evolution  20  
  1.8 Structural evolution and phylo-SCFGs  25  
  1.9 Constraining prediction methods  29  
  1.10 The HIV-1 genome as a case study  31  
  1.11 Bibliography  32  

### 2 Multithreaded comparative RNA secondary structure prediction  
  **using stochastic context-free grammars**  43  
  2.1 Problem statement  43  
  2.2 Results  43  
  2.3 Notes  44
2.4 Bibliography ........................................... 44

3 PPfold 3.0: fast RNA secondary structure prediction using phylogenetic and auxiliary data 55
3.1 Problem statement ................................... 55
3.2 Results ................................................ 55
3.3 Notes .................................................. 56
3.4 Bibliography ......................................... 56

4 Evaluating the accuracy of SHAPE-directed RNA secondary structure predictions 59
4.1 Problem statement ................................... 59
4.2 Results ................................................ 59
4.3 Notes .................................................. 61
4.4 Errata and clarifications .............................. 62
4.5 Bibliography ......................................... 62

5 Characterizing RNA secondary structure space using information entropy 74
5.1 Problem statement ................................... 74
5.2 Results ................................................ 75
5.3 Notes .................................................. 75
5.4 Bibliography ......................................... 76

6 The higher-order structure of the HIV-1 genome 86
6.1 Problem statement ................................... 86
6.2 Results ................................................ 87
6.3 Notes .................................................. 87
6.4 Bibliography ......................................... 88

A Quantifying variances in comparative RNA secondary structure prediction 111
A.1 Summary .............................................. 111

B RNA secondary structure prediction on an ad-hoc peer-to-peer network infrastructure 126
B.1 Summary .............................................. 126

C An n-free-passes CYK algorithm for error-correction and the prediction of non-canonical base-pairs in RNA secondary structure 141
C.1 Summary .............................................. 141

Index .......................................................... 154
Summary in English and Danish

English summary

The prediction of the structure of large RNAs, such as the genome of the HIV virus, remains a particular challenge in bioinformatics, due to the computational complexity and low levels of accuracy of state-of-the-art algorithms. This dissertation presents several contributions in this field, including theoretical developments and the practical application of the theory to biological data. To address the performance issues arising for large amounts of data, the PPfold program was developed, which is a parallelized and improved re-implementation of the pfold algorithm for RNA secondary structure prediction. PPfold was then extended with a probabilistic framework for incorporating experimental data, such as from high-throughput RNA probing experiments, into structure predictions. This enabled the combination of comparative and data-driven RNA secondary structure prediction in a single consistent framework. Nevertheless, prediction quality remained an important issue, as neither stochastic nor thermodynamic prediction algorithms are robust against poor quality in the input data. This highlighted the need to quantify input data quality and estimate its effect on the accuracy of structure prediction. To this end, the thesis includes a systematic study of thermodynamic SHAPE-directed RNA secondary structure prediction, as well as the application of information entropy to characterize RNA secondary structure folding space. Finally, these research results could be applied to produce a phylogenetically and experimentally supported secondary structure for the genome of the HIV-1 virus.
Dansk resumé

This thesis summarises the work I have done during my PhD studies at Aarhus University, Denmark, under the supervision of Jørgen Kjems at the Interdisciplinary Nanoscience Center and Christian Storm Pedersen at the Bioinformatics Research Centre.

During these four years, I have focused on aspects RNA secondary structure prediction relevant for large viral RNAs, and co-authored 10 manuscripts, 7 of which have been published in peer-reviewed journals, one is in preparation for submission to a peer-reviewed journal, and two have been published as technical reports. The main research content in this thesis consists of 5 of these manuscripts, in which I have been the lead author. Each of these papers is presented in its entirety, accompanied by a short description of the research problems, the main results, and additional notes. Some of the remaining manuscripts, to which I have made major contributions, are included in the appendices.

Permissions from the publishers have been obtained wherever required for the reproduction of any copyrighted material in this thesis.

Short outline

RNA secondary structure prediction has a long and interesting history. It is a prime example of a research area in bioinformatics that requires a highly interdisciplinary approach, integrating mathematical, computational and molecular biology techniques. In Chapter 1, my goal is to give an introduction to this interdisciplinary field of research, setting my own work into context.

My PhD project was originally motivated by the realization that previously existing algorithms for RNA secondary structure prediction could not be applied successfully to large viral RNA genomes, due to excessive runtimes and other com-
putational constraints. My initial work centered on re-implementing one of these algorithms, adapting it to the modern computer’s multithreaded processing abilities, as well as exploring possibilities for executing it on a computer network. This work is described in Chapter 2 and Appendix B.

Simultaneously with the developments in computing technology, a breakthrough took place in molecular biology techniques for probing RNA structures as well, when large-scale RNA structure probing became possible using high-throughput technologies. This in turn led to a need for improved methods of using this newly available molecular data in structural predictions. My next research direction was therefore to design, implement and test a flexible probabilistic model for using structure-probing data, producing a method, PPfold 3.0, that allowed the combined use of quantitative experimental data and evolutionary information for the first time. This work is presented in Chapter 3.

It has also been known for a long time that the currently available models for RNA structure do not scale well to large RNAs. Small changes in input parameters result in erratic prediction outcomes, and structure predictions are often inaccurate. Unfortunately, it has proved exceedingly difficult to improve the accuracy of RNA secondary structure predictions, in part because it is unclear what exactly affects prediction accuracy and to what extent. So in collaboration with groups at Georgia Tech and Oxford University, we set out to quantify the effect of various factors affecting prediction quality, including the quality of the structure probing data (Chapter 4) and the quality of the sequence and alignment input (Appendix A, Appendix C).

Recognizing the need to report information about the expected reliability of a particular prediction, I was fascinated by the concept of information entropy, which is an elegant measure for the “spread” of a probability space. I found a way to compute this quantity for the probabilistic model implemented in PPfold and showed that the information entropy delivers complementary information to the reliability scores reported by PPfold. This is described in Chapter 5.

Finally, drawing on my experiences from the above research, I analyzed the higher-order secondary structure of the RNA genome of the HIV-1 virus. Several new structural elements, including long-distance interactions, could be proposed. However, the results also indicate that HIV-1 generally has a flexible RNA genome, and may not be among viruses with a genome-scale ordered RNA structure. The (currently unpublished) manuscript describing these results is found in Chapter 6.

**Acknowledgements**

During my PhD, I have been fortunate to learn from and work together with many outstanding scientists from the computational, mathematical and molecular biology communities.

First of all I would like to thank my supervisors Jørgen Kjems and Christian Storm Pedersen, as well as my mentor Ebbe Sloth Andersen, for their continued support throughout my PhD: for helping me develop independence, for putting me
in touch with their broad network, and especially for the freedom they have allowed me to pursue the research directions that I have found most interesting.

Similarly, I would like to thank Christine Heitsch, for supporting me during my stay at her lab Georgia Tech, for our productive collaboration that extended well beyond the 5 months and for teaching me so much about not just science but also life in general. I am also grateful to many other present and past members of Christine’s group, particularly Shel Swenson and Svetlana Poznanovik, for their friendship and the many useful discussions. Also at Georgia Tech, I would like to thank Steve Harvey and David Bader for inspiring conversations.

Special thanks go to Bjarne Knudsen at CLC bio, who not only has an uncanny ability to understand my countless badly formulated questions (even those that I didn’t understand myself), but also always took the time to very pedagogically answer them. Also at CLC bio, Morten Værum, Mikkel Nygaard Ravn and Alex Andersen have been great sources of expert Java (and general programming) knowledge, and I am looking forward to joining them as a colleague very soon.

At the IT University of Copenhagen, I have enjoyed collaborating with Jakob Bardram and his Pervasive Interaction Technology lab in the Mini-Grid project, and have fond memories of our pizza-rich (and yet productive) meetings with Aurelien Tabard, Juan Hincapie-Ramos, Neela Narayanan Venkataraman and Timus Abell.

At the University of Oxford, I have been fortunate to collaborate with Jotun Hein’s group, particularly James Anderson, who has been a great colleague, a kind of scientific buddy and a source of inspiration in the last two years of my PhD. Also from Jotun’s lab, I would like to thank Ádám Novák for our smooth collaboration, even though we somehow never managed to meet in person in the end.

At Copenhagen University, I have enjoyed working together with with Jan Gorodkin and Stefan Seeman on my final HIV project – I would like to thank them for the time they have devoted to it, and for making me feel so welcome in their lab every time I visited. At MOMA in Aarhus, I would like to thank Jakob Skou Pedersen and Sudhakar Sahoo, for introducing me to the joys (and pains) of dealing with real-life genome data, and letting me be a part of the COAT project for a few months.

Students I have supervised or co-supervised and would like to thank include Mathias Hansen and Mads Krogh Jensen from Aarhus University, and David Georg Korczynski, Peter Nicolai Skovgaard, Elias la Cour and Peter Bertelsen from ITU. They have probably taught me more than I taught them!

I am grateful to my friends and colleagues at the Kjems lab and at BiRC too, for the many enjoyable hours we have spent together, and my friends from the PhD House Activity Group and the PhD Association at SciTech for the nice PhD-dreams they have helped turn into reality.

Lastly, I would like to thank my family, for always providing me with a home environment that encouraged curiosity and learning, for all the good advice, late-night conversations and long phone calls, and for their invaluable support during my entire PhD.

To Adam, because he saw way too much of how this thesis came about, and has chosen to marry me anyway.
The following list contains the manuscripts, conference presentations and software that I have contributed to during my PhD.

**Published, peer-reviewed papers**


Preface

Manuscripts in preparation

1. The higher-order structure of the HIV-1 genome. Sükösd et al., to be submitted (Chapter 6)

Technical reports

1. RNA secondary structure prediction on an ad-hoc peer-to-peer network infrastructure – The Collaborative Mini-Grid Project Sükösd et al. (Appendix B)

2. An n-free-passes CYK algorithm for error-correction and the prediction of non-canonical base-pairs in RNA secondary structure Anderson et al. (Appendix C)

Conference presentations

1. PPfold 3.0: Fast RNA secondary structure prediction using phylogeny and auxiliary data. Poster presentation at ECCB’12 (European Conference on Computational Biology), and oral presentation at ESCS’12 (European ISCB Student Council Symposium), both in Basel, Switzerland, September 2012.


Book chapter


Software

1. PPfold: a parallelized and improved version of the pfold program, extended with probabilistic support for experimental probing data. (I wrote all of this code.)

2. GTfold: a parallelized implementation of the thermodynamic model for RNA secondary structure prediction, written by others at Georgia Institute of Technology. (I have extended it with support for hard and soft constraints.)
Abbreviations

Abbreviations that are used in this thesis are listed below, in alphabetical order.

CBC compensating base changes
CYK algorithm Cocke-Younger-Kasami algorithm
EM algorithm expectation maximization algorithm
GORS genome-scale ordered RNA structure
GTR model General Time-Reversible Model
HIV Human immunodeficiency virus
KH grammar Knudsen-Hein grammar
MFE minimum free energy
ML maximum likelihood
RNA ribonucleic acid
SCFG stochastic context-free grammar
SHAPE selective 2'-hydroxyl acylation analyzed by primer extension
RNA secondary structure prediction has a long and interesting history, and is a prime example of a bioinformatics research area where mathematical, computational and molecular biology approaches are integrated in an effort to understand biological function. With this chapter, my goal is to give a brief introduction to this interdisciplinary field of research, and set my own work into context. It has not been my purpose to provide an exhaustive presentation of any subject area, as many excellent, in-depth reviews have been published elsewhere on every topic discussed here. Instead, I have focused on aspects of direct relevance to my research, and attempted to draw attention to the lesser-discussed connections between the different subject areas.

1.1 RNA as a macromolecule

Over the past two decades, it has increasingly been recognized that RNAs not only serve as messenger molecules between DNA and proteins, but also play diverse regulatory and enzymatic roles in cells, appear as the genomes of many viruses, and can even be nano-engineered for particular purposes, such as logical gates. The versatile nature of RNAs is, in a large part, due to the structural conformations they adopt in vivo.

RNA and DNA are chemically very similar; they are both polymers of nucleotides. Each nucleotide consists of a sugar ring with a nucleobase attached to the 1' position and a phosphate group on the 5' position. The sugar is deoxyribose in DNA, and ribose in RNA. Adjacent nucleotides are joined through a phosphodiester bond between the 5' phosphate group of one nucleotide and the 3' hydroxyl group of another one. This gives rise to the sugar-phosphate backbone of nucleic acids, which has a 5'-to-3' directionality. (Figure 1.1(a))
Chapter 1. Introduction

Figure 1.1: The structure of nucleic acids. (a) The sugar-phosphate backbone is formed by phosphodiester bonds between adjacent nucleotides. Three “edges” of hydrogen donors can be identified on each base. (b) Hydrogen bonds between nucleobases give rise to basepairing. The most common basepairs are formed on the Watson-Crick edge of G-C and A-T (in DNA) or A-U (in RNA) nucleotides on antiparallel strands. (Based on images from Wikimedia Commons)

The most commonly occurring nucleobases in RNA are alanine (A), guanine (G), cytosine (C) and uracil (U). In DNA, uracil is substituted by thymine (T), which has an added methyl group. Cytosine, uracil and thymine are pyrimidines: they are based on the pyrimidine ring system, which includes 2 nitrogens. Alanine and guanine are purines, because they contain the pyrimidine unit fused to an additional, imidazole ring. The sequence of these bases, as they occur in DNA or RNA, defines the primary structure (or sequence) of the nucleic acid chain.

The three-dimensional (tertiary) structure of an RNA molecule forms as a result of complex interactions between the atoms of the RNA and their environment. The major forces within the RNA molecule are the covalent bonds of the RNA backbone and bases, hydrogen bonds (forming especially between bases), electrostatic forces (forming especially on the backbone due to the negatively charged phosphate groups), and Van der Waals forces between all atoms in close proximity to each other. In addition, the cellular environment provides a large number of interacting factors, including other RNAs and proteins, which chaperone the folding of the RNA. Importantly, the RNA is folded simultaneously with its transcription, in a 5'-to-3' direction. The “native” fold is, therefore, also controlled by the kinetics of transcription.

The subject of this thesis is only a subset of these interactions: the hydrogen bonding between bases, also known as basepairing, which bend the RNA backbone. Each base has a unique set of hydrogen bond donors and acceptors, arranged on three “edges”. Leonidis and Westhof [1.1] have classified these as Watson-Crick, Hoogsteen and Sugar edges (Figure 1.1(a)), and basepairs can be categorized into six groups.
RNA as a macromolecule

on the basis of which edges are involved in the hydrogen bonding. Furthermore, in each case the local strand orientation can be parallel or antiparallel, determined by the orientation of the bases in relation to the 5'-3' directionality of the chain. In total, this gives rise to 12 families of possible basepairs.

The widely known double-helix structure of DNA is formed by G-C and A-T basepairs, taking place on the Watson-Crick edges of nucleotides on antiparallel strands. These are known as Watson-Crick basepairs between complementary nucleotides, and are illustrated in Figure 1.1(b). The vast majority of basepairs found in RNA are also Watson-Crick G-C and A-U basepairs. The “wobble” G-U basepair, which is also formed on Watson-Crick edges, is commonly found in RNA as well. These three are called canonical basepairs, whereas any other basepairs are considered non-canonical.

In DNA, basepairing happens between two complementary strands. RNA, on the other hand, lacks a complementary strand, so basepairing typically happens within the same molecule\(^1\). This causes parts of the RNA molecule to fold back up on themselves, leading to local helical structures arranged in a complex topology. A major factor in the stability of helices is the ability of the aromatic rings of adjacent basepairs to form noncovalent interactions known as stacking\(^2\). A run of adjacent basepairs, accordingly, is known as a basepair stack.

The pattern of basepairing within an RNA molecule is known as its secondary structure. Pseudoknotted secondary structures contain at least two stem-loop structures in which half of one stem is intercalated between the two halves of another stem. Mathematically, basepairs \(i-j\) and \(i-j'\) are pseudoknotted with respect to each other if \(i<i'<j<j'\). Nested secondary structures contain no pseudoknots.

RNA secondary structures can be represented visually in a number of different ways; the most important ones are illustrated in Figure 1.2. The tertiary structure of the RNA forms on top of the secondary structure scaffold, and the secondary structure is also used in the prediction of tertiary structure (Figure 1.3) and in RNA structure design \([1.2]\).

The complete secondary structure of an RNA molecule can be simplified by describing it as a collection of structural motifs or elements. The most basic building block is a basepair. A sequence of adjacent basepairs, or a basepair stack, is also known as a helix. Unpaired nucleotides are categorized into various loops. A hairpin loop is enclosed by exactly one basepair. Many hairpins with exactly 4 unpaired bases are especially stable; these are known as tetraloops. An internal loop is enclosed by exactly two basepairs. A bulge is a special type of internal loop, where the length of one side of the internal loop is zero. A multi-loop contains more than two basepairs. Finally, an external loop is an unpaired region that is not enclosed by a basepair. External loops are sometimes classified into free ends, which are found at the end of the molecule, and joints, which are flanked (but not enclosed) by basepairs. (Figure 1.4(a))

\(^1\)There are, however, notable exceptions to this, such as the binding of miRNAs to messenger RNAs.

\(^2\)Stacking can also happen at the first unpaired nucleotides at the end of helices.
Figure 1.2: The most common representations of secondary structure. In this example, the secondary structure of *Saccharomyces cerevisiae* 5s rRNA is drawn. (a) Dot-bracket representation: matching parentheses indicate basepairing partners. (b) Classical representation: the topology of both the RNA backbone and the basepairs is shown. In some cases, nucleotides can be drawn as circles. (c) Linear arc diagram: the sequence is shown on a horizontal line, and each basepair is indicated with an arc. Arc lengths are correlated with basepairing contact distance. (d) Circular arc diagram: a space-saving version of an arc diagram, where the sequence is plotted on a circle. Arc lengths are not correlated with basepairing contact distance. (e) Dot plot: the sequence is plotted on both the horizontal and the vertical axis, and dots indicate basepairs. (Figures generated with JViz.RNA [1.3])

Mathematically, an RNA secondary structure can also be thought of as an ordered, rooted tree. (Figure 1.4(b)) A computational advantage of this representation is that it allows quick, systematic traversal of the secondary structure, in such a way that secondary structure elements become immediately apparent. This is useful when scoring or comparing secondary structures.

A fundamental simplifying assumption of RNA structure determination is that the primary sequence of nucleotides in an RNA encodes its secondary structure. This is only partially justified: not only the tertiary structure, but also the secondary structure of RNAs is strongly influenced by the kinetics of transcription, protein scaffolds and chaperones in the cellular environment, as well as physical factors, such as variations in pH, temperature and ion concentrations. Nevertheless, the assumption simplifies the study of RNA folding greatly in two ways. Firstly, it allows the use of *in vitro* experimental methods to study native RNA folding, because it
RNA as a macromolecule

**Figure 1.3:** Illustration of the workflow of tertiary RNA structure prediction using the secondary structure, with human telomerase RNA as an example. (a) An alignment of the wild-type human telomerase RNA pseudoknot region (pseudoknot bases are depicted in red), derived from the complete Rfam entry for vertebrate telomerase, RF00024. (b) Secondary structure prediction with KNetFold, including the pseudoknot. (c) The secondary structure representation derived from KNetFold. (d) The automatically generated 3D structure of the predicted wild-type region. The boxed region is believed to be critical for telomerase function and was further refined with manual manipulation and molecular dynamics simulations. (e) An overlay of the predicted optimized structure after molecular dynamics simulations of the wild-type (red) and a mutated version of the telomerase RNA (grey with the mutated bases in blue). Figure from [1.4].

It is assumed that RNAs spontaneously recover their native secondary structure in solution. Secondly, it allows the use of *ab initio* computational prediction methods to predict the secondary structure from the primary sequence. A major discussion in this thesis is the optimization of RNA secondary structure prediction, despite the shortcomings of the above assumptions.
1.2 Experimental methods for determining RNA secondary structure

Experimental methods of RNA structure determination can broadly be grouped into biophysical methods, chemical probing and enzymatic probing. Biophysical methods include Atomic Force Microscopy (AFM) [1.6], Nucleic Magnetic Resonance (NMR) [1.7], X-ray crystallography [1.8] and Small-Angle X-ray Scattering (SAXS) [1.9], which are mostly used to study the 3-dimensional (tertiary) structure of RNAs. These methods will not be discussed in detail here.

Chemical and enzymatic probing methods are more common for studying secondary structures. These methods are typically based on the so-called “protection principle”. Bases, or less commonly, the RNA backbone, are targeted by chemical reagents, causing context-dependent formation of adducts or cleavage of the RNA. Bases involved in basepairing, for example, will generally be less accessible to chemical modifications than unpaired bases. In a subsequent reverse transcription step, adducts or cleavage in the RNA will stop elongation, leading to truncated sequences. RT-PCR followed by gel electrophoresis then reveals the position of adducts, and
indicates basepaired and unpaired regions.

A wide range of reagents have been used for chemical probing. They typically target bases of the RNA and are also usually base-specific. For example, DMS (dimethyl sulphate) methylates A’s and C’s that are single stranded, paired at the end of a helix, or next to a G-U pair. CMCT modifies single-stranded U’s and G’s. Kethoxal attacks unpaired G’s. On the other hand, in hydroxyl-radical probing the sugar-phosphate backbone is cleaved at every residue, providing uniform cleavage at both paired and unpaired residues in a given RNA secondary structure. However, protein binding sites and sites buried in tertiary folds are protected from the reaction.

Enzymatic probing makes use of naturally occurring RNases, more specifically the endoribonuclease subclass, which cleave the RNA within the sequence rather than at the ends. RNase A, for example, is specific for single-stranded C and U residues, whereas RNase V1 is non-sequence specific for double-stranded RNA. A comprehensive review of the traditional RNA probing methods is given by Ehresmann et al. [1.10].

A recent development has been a move toward high-throughput methods for RNA probing, making use of next-generation sequencing technologies instead of gel electrophoresis. Kertesz et al. [1.11] reported a transcriptome-wide profiling study of the RNA secondary structure of the budding yeast *Saccharomyces cerevisiae*, based on deep sequencing fragments of RNAs that were treated with structure-specific endonucleases. The RNases V1 and S1 were used simultaneously, as V1 is dsRNA-specific and S1 is ssRNA-specific. The log ratio between the number of sequence reads obtained for each nucleotide in the V1 and S1 experiments (the PARS score) provided a quantitative measure at nucleotide resolution, representing the degree to which a nucleotide was in a double- or single-stranded conformation. Higher PARS scores denote a higher probability for a nucleotide to be in a double-stranded conformation.

The FragSeq method [1.12] used nuclease P1 instead, which specifically cleaves single-stranded nucleic acids. Probing the entire mouse nuclear transcriptome in two cell types, single-stranded RNA regions in multiple non-coding RNAs with known structure could be mapped. While the method developed by Kertesz et al. [1.11] compares digestion by two nucleases with opposing actions, FragSeq compares digestion by one nuclease to a control run without nuclease treatment.

In an alternative approach, the SHAPE technique (SHAPE: selective 2'-hydroxyl acylation analyzed by primer extension) [1.13] targets the 2'-OH of the ribose ring in a sequence-independent [1.14] fashion. A number of possible SHAPE reagents have been used, including NMA (N-methylisatoic anhydride) [1.15–17], BzCN (benzoyl cyanide) [1.18] and 1M7 (1-methyl-7-nitroisatoic anhydride) [1.19]. In this case, the rate of reaction is related to the flexibility of the backbone in these positions, which in turn is correlated with the structural context of the nucleotide. (Figure 1.5) In general, the backbone at basepairing nucleotides is less flexible than at unpaired nucleotides. Similarly to traditional probing methods, the SHAPE reagent causes adduct formation and fall-off of the reverse transcriptase enzyme. However, the amount of reaction at any position of the RNA was initially measured quantitatively.
by capillary electrophoresis (SHAPE-CE, [1.20]), and later by sequencing (SHAPE-Seq, [1.21]), increasing the throughput of the experiment. The raw read counts from the sequencing step are normalized, giving rise to dimensionless ‘reactivities’[1.13, 22, 23]. Normalization and its effects are discussed in Chapter 4.

For RNAs larger than a few nucleotides, the results of experimental probing methods are usually combined with computational models for RNA folding, as described in Section 1.9. A major research subject in my PhD has been the analysis of existing ways of doing this, as well as the development and application of a novel way of integrating experimental data into a computational method for improved predictions of RNA secondary structure (see Chapters 3 and 4, and Appendix A).

1.3 Dynamic programming for RNA secondary structure prediction

Computationally, the prediction of RNA secondary structure is an optimization problem. Given a sequence of nucleotides, we wish to identify a “best” (non-pseudoknotted) secondary structure, by maximizing or minimizing an objective function. The earliest algorithms for RNA secondary structure prediction were based on the observation that canonical basepairs spontaneously form in solution.

One of the first computational methods to be published simply listed all possible helical regions, identified compatible helices and computed the free energy of the possibilities in the last step [1.24]. This solution did identify the correct cloverleaf structure for several tRNAs, but was highly inefficient. In 1978, Nussinov et al. [1.25] developed an efficient algorithm to maximize the number of canonical basepairs in an RNA sequence, based on dynamic programming. Here, I describe the simplest version of this algorithm, to illustrate the basic principles of the computational
Dynamic programming for RNA secondary structure prediction

**Figure 1.6:** Basepair maximization: There are two structural possibilities for a subsequence from positions $i$ to $j$. The two positions can pair with each other, or there can be bifurcation.

Dynamic programming is a “divide-and-conquer” computational method for solving complex optimization problems by breaking them down into subproblems. It is applicable in situations where the optimal solution is a combination of optimal solutions to subproblems, and can therefore be written in terms of recursion relations. RNA secondary structure prediction by basepair maximization is an example of such a situation.

Consider a string of nucleotides $x_1 \ldots x_n$, representing the RNA sequence of length $n$. The maximal number of basepairs for the substructure between positions $i$ and $j$ can be written as the following recursion:

$$F(i, j) = \max \left\{ \text{score}_{-}\text{pair}(x_i, x_j) + F(i + 1, j - 1) \text{ paired}, \max_{i+1<k<j-1}\{F(i, k) + F(k+1, j)\} \text{ bifurcation} \right\}$$  \hspace{1cm} (1.1)

These correspond to the two situations depicted in Figure 1.6. In both cases, the maximum possible score for a subsequence $x_i \ldots x_j$ depends on the maximum possible score for shorter subsequences. Bifurcation involves dividing the sequence into two parts, each of which is optimized separately. Assuming only canonical pairs are possible,

$$\text{score}_{-}\text{pair}(x_i, x_j) = \max \left\{ 1 \text{ if } (x_i, x_j) \text{ is canonical}, 0 \text{ otherwise} \right\}$$  \hspace{1cm} (1.2)

Furthermore, as no basepairs are possible for subsequences of length 2 or lower, for $j - i <= 2$, $F(i, j) = 0$. This forms the initialization condition in the recursion. The value $F(1, n)$ is the maximal number of canonical basepairs possible for the RNA sequence.

From the above recursion relations, a naive recursive algorithm will compute the maximum number of canonical basepairs possible in the RNA sequence. However, it would be an extremely inefficient algorithm: the score for every smaller substructure is computed every time it is needed for a larger substructure. A more efficient algorithm is obtained by working “bottom-up”: starting from the shortest possible subsequences, progressively increasing sequence length and saving the sub-results into a table for later recall. As the pseudo-code for this shows (Algorithm 1), there is a direct correspondence between the recursion relations and the algorithm.

At the end of the algorithm, $F[1, n]$ contains the maximal number of canonical basepairs possible for the RNA sequence. When it is known, the actual secondary structure producing this can be obtained by backtracking in the table.
Algorithm 1 A variant of the Nussinov algorithm

\begin{algorithm}
\begin{algorithmic}
\For{$1 \leq i, j \leq n$}
\State $F[i, j] = 0$
\EndFor
\For{$i = 1$ to $n$}
\For{$j = i + 1$ to $n$}
\State $F[i, j] = \max(F[i, j], \text{score\_pair}(x_i, x_j) + F[i + 1, j - 1])$
\For{$k = i + 2$ to $j - 1$}
\State $F[i, j] = \max(F[i, j], F[i, k] + F[k + 1, j])$
\EndFor
\EndFor
\EndFor
\end{algorithmic}
\end{algorithm}

In the above algorithm, the most time-consuming step is the population of the $F(i, j)$ table in line 6, where a linear-time operation has to be carried out $O(n^3)$ times due to the three embedded loops that depend on $n$. The time complexity of the Nussinov algorithm is therefore $O(n^3)$. This is generally also the case for other RNA secondary structure prediction algorithms without pseudoknots. As $F(i, j)$ is a 2-dimensional table, the space complexity of the algorithm (and related algorithms) is $O(n^2)$. Note that in order to model pseudoknots, it is not enough to only consider shorter subsequences in the evaluation of the score of a sequence, and secondary structure prediction including pseudoknots has been proven to be an NP-hard problem [1.26]. Consequently, no $O(n^3)$ algorithms exist that predict pseudoknots, and the existing algorithms ranging from $O(n^4)$ to $O(n^6)$ in are restricted to particular types of pseudoknots for solvability in polynomial time [1.27-29].

Historically, both the time and the space complexity of dynamic programming algorithms for RNA structure prediction have been prohibitively high for the prediction of large RNA secondary structures. Recent developments in computer technologies have meant both that a very large amount of RAM is now available in most machines, and that multiple CPUs can be utilized to speed up computations, provided that the algorithms are adapted to multithreaded execution. Many algorithms for RNA secondary structure prediction have been parallelized in the past [1.30-33]. My own contribution in this area has been the parallelization and improvement of pfold, a dynamic programming algorithm for RNA secondary structure prediction (see Chapter 2).

1.4 The nearest-neighbour thermodynamic model

The basepair-maximization algorithm presented above has the major disadvantage that it considers all basepairs to be equally likely, and the scoring system is overly simple. A more complex model has emerged on the basis of a thermodynamic description of RNA folding, at the same time as allowing the same dynamic programming principles to be applied. Here, I will give a brief overview of the thermodynamic
The nearest-neighbour thermodynamic model

model, with special focus on its rarely discussed physical assumptions.

Fundamentally, the thermodynamic model describes RNA as a system at constant pressure and temperature at thermodynamic equilibrium. Such a system is known as an *isothermal-isobaric ensemble*, or $NpT$-ensemble. The second law of thermodynamics states that the system is in equilibrium when entropy is maximum. It can be shown that for an $NpT$-ensemble, this is fulfilled when the thermodynamic potential known as *Gibbs free energy* is minimum. The Gibbs free energy of a system at temperature $T$ and pressure $p$ is defined as:

$$G(p, T) = U + pV - TS \quad (1.3)$$

where $U$ is the system’s internal energy, $V$ is its volume and $S$ is its entropy.

As all energy scales are arbitrary, the Gibbs free energy of an unfolded RNA (“random coil”) is by convention 0 kcal/mol, and the change in Gibbs free energy due to a transition from this unfolded state to a folded state is the Gibbs free energy of the folded state. As RNA folding is a thermodynamically favourable (spontaneous) process, the Gibbs free energies of natural folded states are negative.

Gibbs free energy is an extensive thermodynamic variable, and is therefore additive. In practice, this means that if two systems at equilibrium are joined together, the Gibbs free energy of the combined system is the sum of the Gibbs free energies of the two systems, provided that they do not interact with each other (and thus do not influence each other’s energy). In the thermodynamic model for RNA folding, this is assumed in the *nearest-neighbour approximation*, which states that the free energy of any structural element depends only on the structural element itself and the adjacent base pairs. Thus, the Gibbs free energy of a large, folded RNA structure can be calculated by simply adding up the Gibbs free energies of its structural elements (basepairs, helices and loops). The problem of thermodynamic optimization over an RNA sequence can therefore be formulated in terms of thermodynamic optimization over shorter subsequences, allowing the development of dynamic programming algorithms for computing the *minimum free energy* (MFE) structure, similarly to the basepair maximization algorithm presented in Section 1.3.

The simplest form of the MFE algorithm, described by Zuker and Stiegler [1.34] in 1981, computes the the optimal substructure between positions $i$ to $j$, by considering two possible energies:

- $W(i, j)$: the minimum free energy of the substructure in all possible configurations. If $j - i <= 4$, $W(i, j) = 0$, because no basepairs form for very short sequences.

- $V(i, j)$: the minimum free energy of the substructure provided that positions $i$ and $j$ form a pair. If positions $i$ and $j$ cannot pair, then $V(i, j) = \infty$.

If $i$ and $j$ form a pair, it can be in the context of a hairpin, an internal loop, or something else (e.g. a basepair stack or multi-loop). The contributions of the different kinds of loops are computed using experimentally measured values. The free energy of small hairpin loops is taken directly from experiments, and larger
hairpins have an added penalty. In the case of the internal loop, the loop itself has an energy contribution, as well as the structure enclosed by the second basepair between $i'$ and $j'$, where $i < i' < j' < j$. Zuker and Stiegler [1.34] also expressed the free energy of all other situations with bifurcation: the sequence between positions $i$ and $j$ is divided into two parts at a position $i'$, such that $W(i+1, i') + W(i'+1, j-1)$ is minimum. By considering each of these possibilities, a recursion relation can be written up for $V(i, j)$:

$$V(i, j) = \min \left\{ \begin{array}{l}
\text{hairpin}(i, j) \\
\min_{i<i'<j} \{ \text{internal loop}(i, i', j', j) + V(i', j') \} \\
\min_{i+1<i'<j-1} \{ W(i+1, i') + W(i'+1, j-1) \} \end{array} \right\} \quad (1.4)$$

Similarly, $W(i, j)$ can be divided into three possibilities: the positions $i$ and $j$ can be paired with each other, or one of the positions can “dangle” as an unpaired nucleotide on one end (contributing 0 energy to the structure), or the entire substructure from positions $i$ to $j$ can be bifurcated into two parts, such that for some $i < i' < j$ the sum $W(i, i') + W(i'+1, j)$ is minimum. The recursion relation is thus:

$$W(i, j) = \min \left\{ \begin{array}{l}
V(i, j) \\
\min \{ W(i+1, j), W(i, j-1) \} \\
\min_{i<i'<j} \{ W(i, i') + W(i'+1, j) \} \end{array} \right\} \quad (1.5)$$

The MFE is given by $W(1, n)$. It is clear from the above recursion relations that because of the internal loop computation, the time complexity of the algorithm is increased to $O(n^4)$, where $n$ is the length of the sequence. However, Lyngsø et al. [1.35] used a heuristic to speed up internal loop calculations: restricting internal loop sizes to a maximum value reduces the time complexity to $O(n^3)$, as in the case of the Nussinov algorithm. Lyngsø et al. [1.36] were also the first to present the recursion relations in terms of 5 tables instead of just 2, simplifying the implementation.

The current versions of the thermodynamic model used in well-known programs such as mfold/UNAFold[1.34, 37], RNAfold (the Vienna RNA package)[1.38] and RNAstructure[1.39] are very advanced, with special considerations for multi-loops, coaxial stacking and dangling bases as well, and they use hundreds of experimentally measured thermodynamic parameters in the computation[1.40, 41].

The free energies of small structural elements can be measured experimentally. The process of folding a small oligonucleotide (chosen with the assumption that it has only one possible folded structure) is a reversible reaction:

$$\text{unfolded state} \rightleftharpoons \text{folded state} \quad (1.6)$$

The Gibbs free energy change $\Delta_r G$ is related to the equilibrium constant $K$ of this reaction:

$$\Delta_r G = RT \ln K \quad (1.7)$$

where $R$ is the gas constant and $T$ is the absolute temperature at which the reaction takes place.
The nearest-neighbour thermodynamic model

The equilibrium constant is measured using absorbance techniques at a known temperature or over a temperature range. From this, the Gibbs free energy of the reaction and hence the Gibbs free energy of the structure can be obtained. The Gibbs free energies of various structural elements are available in the Nearest Neighbor Database (http://rna.urmc.rochester.edu/NNDB/ref.html, [1.42]).

The Gibbs free energy of a structural configuration is related to the probability of finding the RNA in that configuration through the Boltzmann distribution. In the \( NpT \) ensemble, the probability of state \( i \) with Gibbs free energy \( \Delta G_i \) at temperature \( T \) is:

\[
p(i) = \frac{e^{-\Delta G_i/(k_B T)}}{Z(T)}
\]

where \( k_B = 0.001987 \) kcal/mol/K is the Boltzmann constant, and \( Z(T) \) is the normalizing constant known as the partition function: \( Z(T) = \sum_i e^{-\Delta G_i/(k_B T)} \).

Minimizing the Gibbs free energy in the MFE algorithm is, therefore, equivalent to maximizing the probability of the structural conformation. In cellular systems, we assume \( T = 37^\circ C = 310.15 \) K. A useful fact is that given two RNA secondary structures with energies \( \Delta G_1 \) and \( \Delta G_2 \), an estimate for the ratio of their probabilities is \( \frac{p_1}{p_2} \approx e^{-1.6(\Delta G_1 - \Delta G_2)} \). A difference of \( \sim 1.44 \) kcal/mol in free energy translates to a 10-fold probability difference. For comparison, the Gibbs free energy of a single G-C followed by C-G basepair stack is \( -3.42 \) kcal/mol, which translates to a 238-fold probability difference.

Statements can also be made about the entire structural ensemble using the partition function, which can also be evaluated using a dynamic programming algorithm. In fact, the main difference between the MFE algorithm and the partition function algorithm is that all minimizations are replaced by summing[1.41, 43]. Once the partition function is known for the whole RNA and all of its subsequences, the basepairing probabilities can be computed by considering the few possible forms of enclosing secondary structures for that basepair. Using stochastic backtracking, it is also possible to sample from the secondary structure ensemble according to the probability of the structure [1.44, 45].

The basic MFE algorithm has been implemented many times. Frustratingly for the end-user, the RNA secondary structure prediction problem is ill-conditioned, and small differences in implementation can result in large differences in prediction outcomes[1.46-50]. In such situations, it can be helpful to know what the exact differences are in the different implementations. Some of the differences between UNAfold and RNAfold are described in paper I have contributed to while at Georgia Tech [1.33].

\(^3\)The contributions of dangling bases are unclear when evaluating the partition function, but this only becomes relevant for implementations of very complex versions of the thermodynamic model.
1.5 SCFGs for RNA folding

Stochastic context-free grammars (SCFGs), also known as probabilistic context-free grammars (PCFGs), were first invented in the purely computational context of natural language-modelling, and to this day are widely used in computational linguistics. It is perhaps surprising that SCFGs found applications in RNA secondary structure modelling too—but RNA secondary structure and natural languages have many things in common. In both cases, we consider sequences (or sentences) of symbols (or words). The symbols have intrinsic meanings assigned to them, and interact with each other according to pre-defined rules (a grammar). In the process of applying the rules to a sentence, it acquires meaning beyond the bare sum of its parts.

Formally, a sequence generating grammar is a tuple $(V, \Sigma, P, S)$, where $V$ is a finite set of non-terminal symbols (variables, conventionally denoted by uppercase letters), $\Sigma$ is a finite set of terminal symbols (alphabet, conventionally denoted by lowercase letters), $P$ is a finite set of productions of the type $\alpha \rightarrow \beta$, where $\alpha \in (V \cup \Sigma)^*V(V \cup \Sigma)^*$ and $\beta \in (V \cup \Sigma)^*$, and $S \in V$ is the start symbol. A complete (left-most) derivation is a sequence of production rules, such that starting from the start symbol, and sequentially replacing all non-terminals (starting with the left-most ones) with a production rule emitting from that non-terminal, a string of terminal symbols is obtained.

Grammars can be classified on the basis of their production rules into the Chomsky-hierarchy[1.51]. Context-free grammars form a class in this hierarchy, and are defined by only having a single non-terminal symbol on the left-hand side of all of the production rules. Context-free grammars are convenient model for RNA secondary structure, where we assume context-independence: insertions of structural elements can happen anywhere without affecting the rest of the structure. This is equivalent to the nearest-neighbour approximation. Just as in the case of the nearest-neighbour thermodynamic model, context-free grammars are not naturally suited for modelling pseudoknotted structures.

A stochastic grammar is a grammar extended with a weight function $w : P \mapsto \mathbb{R}$, where $w$ for each left-hand side is a probability distribution over the possible right hand sides it can be replaced with. The probability of a derivation is the product of the probabilities of the productions used in that derivation. A stochastic context-free grammar (SCFG) is a context-free grammar that is stochastic. Derivations with higher probabilities are more consistent with the grammar than derivations with lower probabilities. The probability of a string under the grammar is the sum of the possible derivations of that string under the grammar.

An example of an SCFG is the Knudsen-Hein grammar, used in pfold [1.54, 55] and presented here in a slightly modified form. (Alternative productions from the

---

4The symbol $*$ is the Kleene star operator; it is a widely used regular expression and denotes any (possibly empty) string that is produced by concatenating elements drawn from the set. The elements can occur any number of times and in any order in this string.
SCFGs for RNA folding

same left hand side are separated by |.)

\[
\begin{align*}
S & \rightarrow L \mid LS \\
L & \rightarrow s \mid dFd' \\
F & \rightarrow dFd' \mid LS
\end{align*}
\]

In this grammar, the nonterminal symbols are \(S\), \(L\) and \(F\), and the terminal symbols are \(s\) (for single-stranded positions), \(d\) and \(d'\) (for double-stranded positions). In any emitted string, the double-stranded symbols \(d\) and \(d'\) are grammatically connected to each other, as the opening and closing sides of matching parentheses; indeed, this resembles the dot-bracket representation of RNA secondary structures. This grammar also makes use of bifurcation, in the rules \(S \rightarrow LS\) and \(L \rightarrow LS\). For brevity, we will refer to this grammar as the \textit{KH grammar}.

As an example, small stem-loop can be generated as:

\[
S \Rightarrow L \Rightarrow dFd' \Rightarrow dLSd' \Rightarrow dsSd' \Rightarrow dsLd' \Rightarrow dssd'
\]

There can be several equivalent grammars that describe the same language, and they can be converted to each other by rule transformations. It is possible to write up standards for writing equivalent grammars. These standards are called normal forms, and the most common normal form for context-free grammars is the \textit{Chomsky normal form}. A context-free grammar \((V, \Sigma, P, S)\) is in Chomsky normal form if all of its production rules are of the form:

- \(N \rightarrow AB\) (rules of this type are called bifurcations)
- \(N \rightarrow c\) for some terminal symbol \(c\)
- \(S \rightarrow \epsilon\)

where \(N, A, B \in V\), \(c \in \Sigma\), \(\epsilon\) is the empty string and \(A, B \neq S\). It can be proven that every context-free grammar can be written in Chomsky normal form. A grammar in Chomsky normal form is also always context-free. The transformation of a given context-free grammar to Chomsky normal form, however, is not always trivial.

A useful alternative normal form for SCFGs used in RNA secondary structure prediction is \textit{double emission normal form}, described by Anderson et al. [1.56]. In this form, only rules of the following types are allowed:

- \(\text{Type 1: } A \rightarrow c\)
- \(\text{Type 2: } A \rightarrow c_1Bc_2\)
- \(\text{Type 3: } A \rightarrow BC\)

where \(A, B \in V\), and \(c, c_1, c_2 \in \Sigma\). Anderson et al. [1.56] have shown that (apart from the generation of empty strings), all grammars in Chomsky normal form can be written in double emission normal form. By expressing grammars in double emission normal form, the corresponding parentheses of paired nucleotides can be introduced in a single production, and secondary structure is captured in a more straightforward
manner. The KH grammar, as presented above, is in neither Chomsky normal form nor double-emission normal form, but it is easily transformed into either by “expanding” the disallowed rules.

Assuming the probabilities in the grammar are known, a secondary structure generating grammar gives a probability distribution over secondary structures. As there is a one-to-one correspondence between derivations and secondary structures, the KH grammar is *semantically unambiguous*. Furthermore, any string of symbols \( (s, d \text{ and } d') \) can only be parsed in maximum one way, making the KH grammar *syntactically unambiguous* as well.

A more interesting situation is when the grammar is modified to emit an RNA sequence instead of a secondary structure. A modified form of the KH grammar emitting RNA sequences could be written as:

\[
S \rightarrow L \quad | \quad LS \\
L \rightarrow a | c | g | u \quad | \quad aFa | aFc | \ldots | uFu \\
F \rightarrow aFa | aFc | \ldots | uFu \quad | \quad LS
\]

The previous terminal symbols \( (s, d \text{ and } d') \) have now been replaced with the terminal symbols of the RNA sequence \( (a, c, g \text{ and } u) \). A small RNA string “acgu” can now be parsed in several different ways, corresponding to different secondary structures of this sequence. For example:\(^5\):

**unpaired:** \( S \Rightarrow LS \Rightarrow aS \Rightarrow aLS \Rightarrow acS \Rightarrow acLS \Rightarrow acgL \Rightarrow acgu \)

**stem-loop:** \( S \Rightarrow L \Rightarrow aFu \Rightarrow aLSu \Rightarrow acSu \Rightarrow acLu \Rightarrow acgu \)

We can see that there is still a one-to-one correspondence between derivations and secondary structures, so the grammar is still semantically unambiguous. However, now more than one derivation can exist for the same string, making the grammar syntactically ambiguous. This is very useful, because the stochastic version of the grammar can now be used to produce a probability distribution over possible secondary structures for the same RNA sequence. In the above example, assigning probabilities to the grammar rules will result in different probabilities for the single-stranded and stem-loop structures for the same string “acgu”.

The “best” structure for the sequence will be the structure with the maximum probability under the grammar. This optimization problem is very similar to finding the minimum free energy (MFE) structure under the thermodynamic model, and is also solved by considering all optimum (probability maximizing) substructures in dynamic programming. The algorithm is known as the Cocke-Young-Kasami (CYK) algorithm. Strictly speaking, the CYK algorithm only determines whether a sequence can be generated by a context-free grammar or not, and it is typically formulated for grammars in Chomsky normal form. In RNA secondary structure prediction, however, this use is rather relaxed and “CYK algorithm” also refers to

---

5 A note on notation: I use \( \rightarrow \) to indicate a rule of the grammar, \( \Rightarrow \) to indicate an actual step of the derivation, and \( \Rightarrow \) to indicate a derivational sequence that might contain more than one step.
modified versions of the algorithm, which compute the highest-probability parse for a sequence under an SCFG model. Pseudocode for a probability-computing version of this algorithm, applied to grammars in Chomsky normal form, is given below (Algorithm 2).

Algorithm 2

A modified form of the CYK algorithm applied to grammars in Chomsky normal form

\[
\text{for } i = 1 \text{ to } n \text{ do} \\
\quad \text{for each production } N \rightarrow c_i \text{ do} \\
\quad \quad P[i, i, N] = P(N \rightarrow c_i) \\
\text{end for} \\
\text{end for} \\
\text{for } j = 1 \text{ to } n \text{ do} \\
\quad \text{for } i = 1 \text{ to } j - 1 \text{ do} \\
\quad \quad \text{for } k = i \text{ to } j - 1 \text{ do} \\
\quad \quad \quad \text{for each production } N \rightarrow AB \text{ do} \\
\quad \quad \quad \quad P[i, j, N] = \max(P[i, k, A] \cdot P[k + 1, j, B] \cdot P(N \rightarrow AB), P[i, j, N]) \\
\quad \quad \end{for} \\
\quad \text{end for} \\
\text{end for} \\
\text{end for} \\
\]

At the end of the algorithm, \( P[1, n, S] \) contains the maximum parse probability. The parse with this probability, as in the case of the MFE algorithm, is found through backtracking. The time complexity of the CYK algorithm is also \( O(n^3) \), where \( n \) is the length of the sequence, and the space complexity is \( O(n^2) \).

The partition function equivalent is the inside algorithm: it evaluates the total probability of producing the sequence (string) under the grammar. Given a sequence \( x_1 \ldots x_i \ldots x_j \ldots x_n \), the inside variables \( e \) are defined as:

\[ e(i, j, N) = P(N \Rightarrow x_i \ldots x_j) \tag{1.9} \]

\( e(i, j, N) \) is the sum of the probabilities of all the possible ways of generating \( x_i \ldots x_j \) from nonterminal \( N \). The inside algorithm is very similar to the CYK algorithm, except that maxima are replaced with sums.

Similarly, we define outside variables \( f \) as the sum of the probabilities of the possible ways of generating everything outside of subsequence \( x_i \ldots x_j \), given that \( N \) generates \( x_i \ldots x_j \):

\[ f(i, j, A) = P(S \Rightarrow x_1 \ldots x_{i-1}Nx_{j+1} \ldots x_n) \tag{1.10} \]

The outside variables are evaluated in the outside algorithm (Algorithm 3).

The algorithm is also illustrated in Figure 1.7. In the outside algorithm, we embed \( c_i \ldots c_j \) in a longer subsequence: \( c_i \ldots c_jc_{j+1} \ldots c_k \) (expressed from \( N \)), which we then split into two parts: \( c_i \ldots c_j \) (expressed from \( A \)) and \( c_{j+1} \ldots c_k \) (expressed from \( B \)). \( f(i, j, A) \) requires that \( c_i \ldots c_j \) is excluded from the complete sequence – so
Chapter 1. Introduction

Algorithm 3 The outside algorithm for grammars in Chomsky normal form

for all $N$
do
if $N = S$ then
  $f[1, n, S] = 1$
else
  $f[1, n, N] = 0$
end if
end for

for $i = 1$ to $n$
do
  for $j = n$ to $i$
do
    for each production $N \rightarrow AB$
do
      $f[i, j, A] += f[i, k, N] \cdot e[j + 1, k, B] \cdot P(N \rightarrow AB)$
    end for
  end for
end for

for $k = 1$ to $i$
do
  for each production $N \rightarrow BA$
do
    $f[i, j, A] += f[k, j, N] \cdot e[k, i - 1, B] \cdot P(N \rightarrow BA)$
  end for
end for

end for

A contribution to $f(i, j, A)$ from this particular splitting involves a combination of the probability of excluding all of $c_i \ldots c_k$ ($f(i, k, N)$), at the same time as expressing the extra part $c_{j+1} \ldots c_k$ ($e(j + 1, k, B)$). Similarly, we also consider the reverse situation, where $c_i \ldots c_j$ is embedded in $c_k \ldots c_{i-1}c_i \ldots c_j$, and $B \Rightarrow c_k \ldots c_{i-1}$ and $A \Rightarrow c_i \ldots c_j$. Again, for $f(i, j, A)$ we add the contribution from excluding the longer string $c_k \ldots c_j$ ($f(k, j, N)$) combined with including $c_k \ldots c_{i-1}$ ($e(k, i - 1, B)$).

The inside-outside variables can be used to derive useful probabilities under the SCFG model, such as the probability of a basepair between positions $i$ and $j$. For example, in the sequence-emitting version of the KH grammar, there are two types of rules that produce basepairs: $L \rightarrow aFa|aFc|\ldots|uFu$, and $F \rightarrow aFa|aFc|\ldots|uFu$. The total probability that one of these rules is used to parse the substring $x_i \ldots x_j$ is the probability that positions $i$ and $j$ form a basepair under the SCFG model:

$$P_d(i, j) = \frac{f(i, j, L)e(i + 1, j - 1, F)P(L \rightarrow x_iFx_j)}{e(1, n, S)} + \frac{f(i, j, F)e(i + 1, j - 1, F)P(F \rightarrow x_iFx_j)}{e(1, n, S)}$$  \hspace{1cm} (1.11)

In a more general case, the probability of $x_i \ldots x_j$ being derived from $N$ is given by the product $e(i, j, N)f(i, j, N)$. The probability that $x_i \ldots x_j$ is derived from an
initial application of the particular rule \( N \to AB \) is

\[
P(N \Rightarrow AB \Rightarrow x_i \ldots x_j) = \frac{f(i, j, N) \sum_{k=1}^{j-1} P(N \to AB) e(i, k, A) e(k + 1, j, B)}{e(1, n, S)}
\]

(1.12)

In order to be able to use a grammar for secondary structure prediction, its probabilities must first be trained. If only sequences are observed and their “correct” parses are not known, the inside-outside variables can be used to train the grammar in an unsupervised manner, in the expectation maximization algorithm (EM algorithm, [1.57]). Briefly, the grammar probabilities are set to some initial values, and the probability of each rule is re-estimated on the basis of the inside-outside variables computed for the observed sequence, as described above. In the next iteration, the revised SCFG probabilities are used. The process is repeated until convergence. The EM algorithm has been proven to find the SCFG parameters which maximize the likelihood of the observed sequence.

If a set of sequences are available where the parses (secondary structures) are also known, the situation is much simpler: the rules of the grammar can simply be trained by counting. The original KH grammar, in fact, was trained using known structures for tRNA and rRNA sequences [1.54, 55]. The basic assumption in such training is the consistency of RNA structures: we suppose that natural RNA structures “look alike”, so by learning to best describe the structure of known RNA sequences, the SCFG can model new, unknown sequences too.

Apart from pfold, a large number of SCFG-based methods have been published to model RNA secondary structures ([1.58–66] and others), many of which are also combined with an evolutionary model (see also Section 1.8). Interestingly, despite the simplicity of the KH grammar, it still appears to perform well compared to more complex grammars [1.56, 67].
1.6 Comparison of SCFG-based and thermodynamic methods

The SCFG-based approach to RNA secondary structure prediction (Section 1.5) and the thermodynamic approach (Section 1.4) are alike in many ways. The KH grammar can actually be thought of as a very simple thermodynamic model in itself: it gives scores between 0 and 1 to every basepair (distinguishing helix-extending basepairs from helix-opening ones), single-stranded nucleotide or loop. Conversely, Rivas and Eddy [1.68] have shown that the thermodynamic model can be converted to an SCFG by obtaining the probabilities of productions from the appropriate thermodynamic constants. As noted before, both SCFG-based and thermodynamic models are formulated in terms of the optimization of an objective function, which is expressed through recursion relations and implemented through dynamic programming algorithms, with the same $O(n^3)$ computational complexity.

At the same time, the two models are conceptually very different. Thermodynamic methods are based on a physical, energy-driven model for RNA folding, and obtain their parameters from calorimetric experiments. SCFG-based methods, on the other hand, are a form of machine learning, where structures observed in nature are used to model unknown ones, without assumptions about the physical process of RNA folding. SCFG-based methods are also inherently probabilistic. The advantage of this is that the entire well-developed theory of probability and statistics can be applied to SCFGs in a rigorous way. For example, Bayesian probabilities can be used to improve the quality of predictions, as described in Section 1.8 and Chapter 3. Even quantities such as the information entropy can be computed for the probability space generated for SCFGs (Chapter 5). Nonetheless, thermodynamic methods have the advantage that they are based on information that is much easier to relate to. Free energy contributions are clear, quantitative statements about the physical world that (at least in theory) can be tested experimentally, unlike the less tangible concept of probabilities, which are open to “interpretation”.

Lastly, both SCFG-based and thermodynamics-based RNA secondary structure prediction is ill-conditioned. Structures with very similar probabilities or free energies are not necessarily similar, and very similar structures can have very different probabilities or free energies. The definition of the “best” structure being the highest-probability structure has therefore been called into question. In pfold, the solution has been to optimize the expectation of correctly predicted positions, instead of simply finding the highest probability structure. This optimization is done through an additional specially designed recursion using a dynamic programming algorithm. Lu et al. [1.69] have recently applied a similar approach to a thermodynamic model as well.

1.7 RNA molecular evolution

RNA sequences, like all biological sequences, are under various evolutionary pressures. All functions of the RNA affect how it evolves: mRNA’s role to code for
proteins is just as important as functional RNA secondary structures. The competing evolutionary pressures make the phylogenetic analysis of RNAs especially difficult. Nevertheless, standard models for biological sequence evolution can be applied to a great degree. In this section, I outline the General Time-Reversible Model \((GTR \text{ model})\) for sequence evolution, as developed by Tavaré [1.70], and an algorithm for calculating the probability of an alignment given an evolutionary tree, as described by Felsenstein [1.71]. Both of these have been implemented in pfold and my re-implementation of it (PPfold, Chapters 2 and 3), so my treatment of the subject here is mostly focused on the aspects relevant to this implementation. Some details of specific relevance to RNA secondary structure are described in Section 1.8.

A phylogenetic tree represents the evolutionary relationships between sequences of an alignment. The leaves in the tree correspond to the observed sequences, whereas internal nodes are the common ancestors. The GTR model is time-reversible, and the phylogenetic trees we will be considering are unrooted and non-directed. This means that only changes from sequence to sequence are considered, without consideration to which sequence actually came first. As a result, any arbitrary node (leaf or internal node) can be designated as the reference (or root) sequence, from which the relationships to all other sequences are determined. The tree is uniquely described by its topology, and can most often be drawn in more than one equivalent way. The tree is not necessary binary. All branches of the tree are associated with a number, which represent the evolutionary distance between the two nodes connected by that branch.

Many methods exist for creating phylogenetic trees from an alignment. In the discussion below, it is assumed that the topology of the tree is already known (e.g. from an algorithm such as neighbour-joining[1.72]), and only the branch lengths need to be optimized. Furthermore, we consider only the maximum likelihood method, where trees are evaluated according to a model for evolution, which in this case is the GTR model. The evolutionary model reflects the mutation probabilities of nucleotides, and the probability of the alignment given the tree is the likelihood of the observed characters having evolved according to the model. The optimal tree is the tree with the highest likelihood.

The GTR model assumes that all positions in a sequence evolve independently, and that mutations from one nucleotide to another happen at uniform rates, only depending on nucleotide identity; they are thus independent of position in the tree or sequence, for example. These assumptions are quite restrictive, as rates of evolution might change with time, different sites might be under differential evolutionary pressures, and insertions and deletions in particular cannot be modelled by assuming independence at all sites. Nevertheless, these assumptions simplify computation and as we shall see later, some of the shortcomings can be accommodated in the final implementation anyway. In the following description, only one position of the sequence will be considered, but all positions are assumed to evolve in the same manner.

As the substitution rates are memoryless in the GTR model, the Markov property is fulfilled, and evolution is modelled as a continuous time-homogeneous Markov chain [1.70]. Let \(q_{ij}\) denote the instantaneous rate of mutation (substitution) from
Chapter 1. Introduction

nucleotide $i$ to nucleotide $j$ ($1 \leq i, j \leq 4$) at some position $S$ in the sequence. (Figure 1.8(a)) All mutation rates can be described in a rate matrix $Q$, which has the following properties:

$$\{Q\}_{ij} = q_{ij} = \lim_{t \to 0} P(S(t_0 + t) = j|S(t_0) = i) \quad (1.13)$$

$$\{Q\}_{ii} = -\sum_{i \neq j} q_{ij} \quad (1.14)$$

The rate matrix $Q$ is independent of time, but as Equation 1.13 implies, it can be interpreted as the time-derivative of a probability matrix, also known as transition matrix $P$, which does depend on the elapsed time interval $t$. Thus, we can define matrix $P(t)$ by:

$$\{P(t)\}_{ij} = p_{ij} = P(S(t_0 + t) = j|S(t_0) = i) \quad (1.15)$$

and

$$\{P(t)\}_{ii} = 1 - \sum_{i \neq j} p_{ij} \quad (1.16)$$

We note that the Markov property dictates $P(t + s) = P(t)P(s)$ and therefore:

$$P = QP' \quad (backward \ equations) \quad (1.17)$$

$$P'Q \quad (forward \ equations) \quad (1.18)$$

In either case the solution is:

$$P(t) = \exp(Qt) \quad (1.19)$$

In practice, the matrix exponentiation in Equation 1.19 is done with the help of the diagonalization of $Q$:

$$Q = UDU^{-1} \Rightarrow \exp(Qt) = U \exp(Dt)U^{-1} \quad (1.20)$$

where $D$ is a diagonal matrix and $\exp(Dt)$ is computed by exponentiating every entry on the main diagonal.

Time-reversibility in the GTR model implies that there exist a collection of positive numbers $\pi_k$ summing to 1 that satisfy the balance equations [1.70]:

$$\pi_i q_{ij} = \pi_j q_{ji} \quad (1.21)$$

The vector $\pi = [\pi_1, \pi_2, \pi_3, \pi_4]$ is the stationary distribution of the process, for which it is true that

$$\pi Q = 0 \quad (1.22)$$

where $0$ is the null-vector.

Felsenstein’s tree-pruning algorithm uses these results to compute the likelihood of the tree, as follows. We assume the sequence is known (observed) at the leaves of the tree. However, the sequence is not known at the internal nodes. Therefore, at every node $X$ in the tree, we define a probability distribution $p_X$ (the so-called Felsenstein vector) over nucleotides, describing the uncertainty of the nucleotide at that node. The algorithm then traverses the tree in post-order traversal, computing the Felsenstein vector at all internal nodes, all the way to the root, according to specific rules. The algorithm is illustrated in Figure 1.8(b)).
RNA molecular evolution

Figure 1.8: Illustration of the Felsenstein probabilities and algorithm. (a) Evolution from nucleotide $i$ to nucleotide $j$ (b) Calculating the probability of the phylogenetic tree at the designated root, using post-order traversal in the Felsenstein algorithm. Leaf nodes are marked with open circles. (c) Calculating the probability of the phylogenetic tree at an arbitrary node $X$, using bottom- and top-probabilities in the phylogenetic tree.

- **Felsenstein vector at leaves.** The Felsenstein vector at the leaves identifies the observed nucleotide. For example, if the nucleotide at a leaf $X$ is observed to be ‘A’, then $p_X = [1, 0, 0, 0]$.

- **Propagation upward along branch.** If $X$ is a non-root node of a tree, with a branch of length $t_X$ going towards the root of the tree, and $p_X$ is the Felsenstein vector at $X$, then the vector $\exp(Q t_X) p_X$ is the Felsenstein vector at the top of the branch going from $X$ (but not quite reaching its parent node).

- **Combination at parent nodes.** The Felsenstein vector of a parent node is calculated as the elementwise product of the Felsenstein vectors at the tops of the incoming branches. This is because different lineages are assumed to evolve independently.

The likelihood of the tree (given alignment column $s_i$) is the scalar product of
Chapter 1. Introduction

the Felsenstein vector at the root and the stationary distribution of nucleotides:

\[
\mathcal{L}(\text{tree}|s_i) = P(s_i|\text{tree}) = \sum_{\alpha \in \{A,C,G,U\}} P(s_i|\text{tree, root} = \alpha)P(\text{root} = \alpha|\text{tree}) \tag{1.24}
\]

\[
= p_{\text{root}} \cdot \pi \tag{1.25}
\]

Finally, due to the assumption of independent evolution at all positions of the alignment, the likelihood of the tree given the whole alignment is the product of the likelihoods at all positions:

\[
\mathcal{L}(\text{tree}) = \prod_{s_i} \mathcal{L}(\text{tree}|s_i) \tag{1.26}
\]

The fact that the position of the root is irrelevant is very useful when, for a given tree topology, we wish to identify the branch lengths that maximize the likelihood of the tree for a given alignment column. Felsenstein recognized that the root can be “relocated” in a convenient way for every branch length to be optimized. If the branch to be optimized is just above node \(X\), four probability vectors can be defined, as shown in Figure 1.8(c).

- The “down-bottom” vector is simply the Felsenstein vector at \(X\), and represents the total probability under node \(X\). It can be computed in a post-order traversal as described above: \(\text{down-bottom}(X) = p_X\).

- The “down-top” vector is the Felsenstein vector at \(X\) propagated up to but not quite reaching the parent of \(X\), along a branch of length \(t_X\):

\[
\text{down-top}(X) = \exp(Q t_X) \text{down-bottom}(X) \tag{1.27}
\]

- The “up-top” vector can be thought of as the Felsenstein vector at \(X\) that we would obtain if we removed \(X\) and everything currently under it, and then set the top of the branch just over \(X\) to be the root of the tree. In other words, it is the combination of the Felsenstein vectors for the siblings of \(X\) with the nucleotide distribution of everything above \(X\). It is an elementwise product of “up-bottom” for the parent node and “down-top” for the siblings of \(X\):

\[
\text{up-top}(X) = \left( \bigotimes_{Y \in \text{siblings}} \text{down-top}(Y) \right) \odot \text{up-bottom}(\text{parent}) \tag{1.28}
\]

- The “up-bottom” vector is obtained from the “up-top” vector by propagating it along a branch of length \(t_X\): \(\text{up-bottom}(X) = \exp(Q t_X) \text{up-top}(X)\). As the root has neither siblings nor parent nodes, it’s “up-bottom” vector is \([1, 1, 1, 1]\). The “up”-vectors are calculated in pre-order traversal, starting from the root and out to the leaves of the tree.
Using these vectors, it is now possible to isolate the effect of any branch length on the overall probability of the tree. Around any node $X$, the likelihood of the tree can be expressed as:

$$L(\text{tree}|s_i) = [(\text{up-top}(X) \exp(Q_{tX})) \circ \text{down-bottom}(X) \cdot \pi]$$  \hspace{1cm} (1.29)

where $\circ$ represents elementwise multiplication, $\cdot$ is the scalar product, and $\pi$ is the stationary nucleotide distribution.

Given a tree topology, the maximum likelihood branch lengths can then be computed in an iterative fashion. The branch lengths are initialized to arbitrary values, and the “down” vectors are computed in all nodes by pre-order traversal, starting from the leaves. Next, the “up” vectors are computed for all nodes in post-order traversal, starting from the root. Now each branch is optimized separately using Equation 1.29, fixing all other branch lengths implicitly in the already computed “up” and “down” vectors. The process is repeated with the new branch lengths, until convergence at the maximum likelihood for the tree.

An important problem when implementing the above algorithm is that the multiplication of many probabilities results in numerical underflow. This is most commonly addressed by computing log-likelihoods instead, and replacing the product of likelihoods by sums of log-likelihoods.

Finally, it is worth noting that the Felsenstein algorithm is highly versatile due to the probabilistic representation of nucleotide identities at all nodes in the tree. Uncertainties in the sequences at the observed nodes (for example due to base ambiguities or sequencing errors), as well as alignment errors can be modelled by simply assuming a small probability for alternative nucleotide identities in the Felsenstein vectors for leaves. Even gaps can be modelled as completely uncertain nucleotides, assuming a uniform distribution over A,G,C,U, as has for example been done in fold [1.55]. In this way, some of the shortcomings of the GTR model can actually be accounted for in a probabilistic fashion in the final implementation of the algorithm.

### 1.8 Structural evolution and phylo-SCFGs

The models and algorithms described in Section 1.7 are generally applicable to the evolution of any biological sequences. The structural evolution of RNAs, however, is somewhat special, because (functional) structure is more conserved than nucleotide sequence, resulting in different patterns of evolution for paired and unpaired parts of the RNA molecule. Other functional roles of the RNA will also experience different evolutionary pressures – for example, due to the degeneracy of the genetic code, in coding regions the “wobble” positions are expected to undergo more rapid evolution. Comparative RNA secondary structure prediction is based on the principle that covariation will occur in interacting parts of the RNA sequence on evolutionary timescales, because the unique structure and function of RNA molecule is maintained through mutation and selection.
Chapter 1. Introduction

Figure 1.9: Illustration of sequence conservation and covariation. In each case, we observe a pair of columns in a hypothetical alignment, and examine if they support a hypothesis of basepairing between the corresponding positions in the sequences.

If the sequences are fully conserved (Figure 1.9, left), no information is provided about structure, as conservation might have occurred due to evolutionary pressures different from conserving secondary structure. If some sequences in the alignment support basepairing while others do not (Figure 1.9, middle), there is weak evidence for basepairing between the positions. If the nucleotides in the two positions both vary and at the same time consistently support canonical basepairing throughout evolution (Figure 1.9, right), it is known as compensating base changes (CBCs): a pattern highly supportive of basepairing between the positions. It has been said that helices where 2 or more CBCs occur in 2 or more helical positions are “phylogenetically proven”.[1.73, 74]

In fact, the first predictions of RNA secondary structure were based on looking for CBCs, long before other methods of structure prediction were available. In 1966, the structure of the yeast tyrosine tRNA was elucidated by comparing the sequence to an alanine tRNA. In 1980, a comparative structure was proposed for the E. coli 16S ribosomal rRNA as well [1.75], followed by a comparative structure for the 23S ribosome in 1981. Remarkably, when a crystal structure became available for the ribosome, 97-98% of the predicted pairs were found correct [1.76].

A disadvantage of looking for CBCs is that they can not detect non-canonical basepairs. In an approach that also allows detection of phylogenetic evidence for non-canonical basepairs, in 1992 Gutell et al. [1.74] quantified covariation using mutual information, which is defined for positions $i$ and $j$ in an alignment as:

$$M(i, j) = \sum_{b_x, b_y} f_{b_x, b_y} \ln \frac{f_{b_x, b_y}}{f_{b_x}f_{b_y}}$$  \hspace{1cm} (1.30)$$

where the frequency of each base at each position is $f_{b_x}$ and $f_{b_y}$, respectively, and $f_{b_x, b_y}$ is the frequency of the pairs of bases occurring at positions $x$ and $y$ in the same sequence. The mutual information is always positive, with $M(x, y) = 0$ only in the case of the frequency of position pairs being exactly predicted by the frequencies of the independent positions. $M(x, y)$ is maximum when the positions are highly variable and also completely correlated. Weak covariations are also detected by mutual information.
Mutual information has also been used in recently developed algorithms. For example, the MIfold package, written as a MATLAB toolbox, displays and predicts conserved RNA secondary structure including pseudoknots from an alignment [1.77], based on mutual information. The RNAalifold [1.78, 79] and the BayesFold [1.80] methods also incorporate scores based on the mutual information in order to improve prediction quality.

Mutual information, however, does not take into account the precise evolutionary relationships between the sequences in the multiple alignment, and is therefore less useful when the assumption of a “star-tree” relating the sequences to each other is inappropriate. In such cases, a more accurate likelihood-based approach can be applied instead. In pfold, for example, the GTR model and the Felsenstein algorithm are applied in a Bayesian fashion, coupled to the SCFG, producing what is known as a phylo-SCFG.

The main idea behind phylo-SCFGs is that the SCFG produces a prior probability distribution over secondary structures, each of which is then evaluated against the data (input alignment) using the evolutionary model. The probability of a secondary structure $\sigma$ can be written as a Bayesian probability [1.54]:

$$
P(\sigma|D,T,M) = \frac{P(D|\sigma,T,M)P(\sigma|T,M)}{P(D|T,M)} = \frac{P(D|\sigma,T,M)P(\sigma|M)}{P(D|T,M)}$$

where $D$ is the alignment, $T$ is the phylogenetic tree and $M$ is the SCFG model. $P(\sigma|M)$ is the prior probability of the secondary structure produced by the SCFG model. $P(D|T,M)$ is independent of the structure, and is therefore constant over all structures. The two models are coupled formally, by modifying the SCFG to emit alignment columns instead of single nucleotides.

Closely following the notation in my paper [1.81], consider RNA alignments of $k$ sequences ($k \geq 1$), with the $i$'th column denoted $c_i \in \Sigma^k = \{A,G,C,U, -\}^k \setminus \{-\}^k$. A stochastic context-free phylo-grammar (phylo-SCFG) on such alignments is a tuple $G = ((\Sigma^k, N, S, R), P)$, where:

- $\Sigma^k$ forms the (finite) set of terminal symbols
- $N$ is a finite set of nonterminal symbols, such that $\Sigma^k \cap N = \emptyset$
- $S$ is the start symbol, $S \in N$
- $R$ is a finite set of production rules, each rule of the form $A \rightarrow \alpha$, $A \in N$ and $\alpha \in (\Sigma^k \cup N)^*$
- $P$ is a function from $R$ to real numbers in the interval $[0,1]$

In the case of a phylo-grammar, the function $P$ for production rules that emit alignment columns are interpreted as the Bayesian probabilities equal to the product of “prior probabilities” that only depend on the type of rule being used, and a
Chapter 1. Introduction

“likelihood” factor that is derived from a phylogenetic model and is a function of the alignment columns.

Let us express the SCFG in double emission normal form (see Section 1.5), allowing the following rule types:

**Type 1:** \( A \rightarrow c \)

**Type 2:** \( A \rightarrow cBc' \)

**Type 3:** \( A \rightarrow BC \)

for \( A, B, C \in N, c, c' \in \Sigma^k \).

Type 1 rules correspond to the production of a single column of the alignment, and their probability can be expressed as

\[
P(A \rightarrow c) = P_G(A \rightarrow c)P_T(c|\hat{c})
\]  

(1.33)

where \( P_G(A \rightarrow c) \) only depends on \( A \), and \( P_T(c|\hat{c}) \) is the likelihood of observing column \( c \) under the phylogenetic model, assuming that it is unpaired in the consensus structure (denoted by \( \hat{c} \)). This likelihood is calculated using the Felsenstein algorithm described in Section 1.7.

Type 2 rules correspond to the production of two basepaired columns of the alignment, and their probability can be expressed as

\[
P(A \rightarrow cBc') = P_G(A \rightarrow cBc')P_T(c,c'|\hat{c},\hat{c}')
\]  

(1.34)

where \( P_G(A \rightarrow cBc') \) only depends on \( A \) and \( B \), and \( P_T(c,c'|\hat{c},\hat{c}') \) is the likelihood of observing column pair \( c, c' \) under the phylogenetic model, assuming that they are paired with each other in the consensus structure (denoted by \( \hat{c},\hat{c}' \)). This likelihood is also computed using the Felsenstein algorithm. However, the computation is done for column pairs rather than single columns this time (using a \( 16 \times 16 \) rate matrix for substitution rates for double nucleotides). Note that as all possible column pairs must be considered, the time complexity of this computation is \( O(n^2) \), where \( n \) is the length of the alignment.

Type 3 rules express bifurcation and correspond to dividing the alignment into two parts. As these rules do not depend on alignment columns, we simply have:

\[
P(A \rightarrow BC) = P_G(A \rightarrow BC) \cdot 1
\]  

(1.35)

where \( P_G(A \rightarrow BC) \) only depends on \( A, B \) and \( C \).

It is now clear that the probability of any particular derivation under a phylo-SCFG can be expressed as a product of two probabilities: a probability \( p_G \) that only depends on the types of rules used, and a probability \( p_T \) that only depends on the emitted alignment columns:

\[
p(d) = p_G(d)p_T(d)
\]  

(1.36)
Constraining prediction methods

where

\[
p_G(d) = \prod_{r_a} P_G(r_a)^{f_a(r_a)} \prod_{r_b} P_G(r_b)^{f_a(r_b)} \times \prod_{r_c} P_G(r_c)^{f_a(r_c)}
\]

(1.37)

\[
p_T(d) = \prod_{r_a} P_T(c|\hat{c})^{f_a(r_a)} \prod_{r_b} P_T(c, c'|\hat{c}^{'})^{f_a(r_b)}
\]

(1.38)

for \( r_a \in R \) of Type 1, \( r_b \in R \) of Type 2, and \( r_c \in R \) of Type 3.

Importantly, if the phylogenetic probabilities are pre-computed on the basis of the alignment, the division of rule probabilities into two components allows the inside-outside algorithm to be used to optimize the posterior probability of the structure in the joint phylogenetic and SCFG model in one iteration.

Variations on the phylo-SCFG theme have been applied in many other programs as well. Some examples include RNA-Decoder [1.61], which takes into account the known protein-coding context of RNAs, and Evofold [1.62], which exploits the differences of the substitution process in stem-pairing and unpaired regions. Infernal [1.63] is based on profile-SCFGs (also known as covariance models) and is used to search sequence databases for RNA structure and sequence similarities. The popular Rfam database [1.82, 83] of conserved RNA secondary structures families is based on Infernal.

The accuracy of comparative secondary structure predictions depends on the quality of the input alignment (Chapter A). Generating a good RNA alignment is usually not a trivial operation in the first place; most alignment methods rely on sequence-based multiple sequence alignments, which often misalign RNA structure. Specific alignment programs that incorporate RNA secondary structure also exist [1.84–92], but it is usually necessary to manually edit alignments using expert knowledge in a semi-automated fashion [1.93].

In general, however, comparative models for RNA secondary structure prediction are still much more accurate than simple MFE-based methods [1.94], because of the guided reduction in structure search space (see also Section 1.9). One way to think about this is that the probability mass is initially broadly spread over RNA secondary structures, but the evolutionary information focuses the probability mass into better defined “peaks” in probability space. To more exactly quantify this effect, I have found a way to compute the information entropy of the probability space generated by a phylo-SCFG over secondary structures for an RNA alignment. This work is described in Chapter 5.

1.9 Constraining prediction methods

To get a better sense for how “difficult” the RNA secondary structure problem is, it is useful to reflect on the number of possible secondary structures for an RNA sequence. After Stein & Waterman [1.95], assuming all nucleotide pairings are possible, the number of secondary structures \( S(n) \) of length \( n \) grows for large \( n \) as:
Chapter 1. Introduction

\[ S_n \sim 1.104 \cdot n^{-\frac{3}{2}} \cdot 2.618^n \]  

(1.39)

The space that secondary structure prediction methods need to search to find the optimal structure therefore grows exponentially with sequence length. To illustrate the scale of the problem, there are approximately 1.5 billion possible RNA structures of length 27!

This space can be substantially reduced by a few carefully-chosen constraints, as shown here for one example. Suppose that we require a single constraint, namely that nucleotide 1 forms a basepair with nucleotide \( \frac{n}{2} \) (assume this is an integer). Prohibiting pseudoknots, the number of possible secondary structures for each of the thus formed partitions of the RNA is \( S_{\frac{n}{2}} \), and the total number of possible structures is reduced to \( S_{\frac{n}{2}} < S_n \). More specifically:

\[ \frac{S_n}{S_{\frac{n}{2}}} \sim 8.832 \cdot n^{\frac{3}{2}} \]  

(1.40)

This is a significant reduction in the number of structures to search, even if the total number of available structures will still grow exponentially with sequence length. With multiple constraints, the number of possible structures can be further reduced, improving the outcome of any prediction algorithm. Constraining secondary structure prediction methods with carefully chosen constraints is therefore a widespread approach to improving the accuracy of predictions.

Known structural elements can be incorporated into the optimization process as “hard constraints”, which are necessarily satisfied in the optimal structure. Commonly used hard constraints include enforcing single-strandedness in regions of high chemical reactivity, and prohibiting basepairs between nucleotides further than 500-600 nucleotides apart. In contrast, soft constraints locally modify the objective function to reward or penalize particular structural configurations for the implicated regions, thereby directing the optimization. Examples of soft constraints include the use of evolutionary covariation information in comparative approaches (see Section 1.8), and the conversion of experimental data into pseudo-free energy variables to influence thermodynamic optimization.

In particular, a number of approaches have been proposed recently for incorporating SHAPE data into RNA secondary structure prediction methods [1.96–99]. In one of the first reports by Deigan et al. [1.96], the data were converted into pseudo-free energies in the RNAStructure program by converting the SHAPE value for nucleotide \( i \) into a pseudo-energy term using Equation 1.41:

\[ \Delta G_{\text{SHAPE}}(i) = m \ln(\text{SHAPE}(i) + 1) + b \]  

(1.41)

The term \( \Delta G_{\text{SHAPE}}(i) \) is added to the free energy change for nucleotide \( i \) in a nearest-neighbour base-pairing stack. The parameters, \( m = 2.6 \) and \( b = -0.8 \) kcal/mol, were originally determined by parameterizing against the \( E. coli \) 23S rRNA. Using this method, prediction accuracies of over to 95% have been reported for large ribosomal RNA sequences [1.96], but this result has been controversial [1.100–102]. The method has been applied in many studies (for example, [1.103–105]), and even
used to predict the secondary structure of a full HIV-1 genome [1.106]. However, the approach has also generated some controversy [1.100-102] and not many of the predicted novel structures in HIV-1 have been confirmed to date. In fact, a highly selective experimental mutational study of 16 of the novel predicted hairpins [1.107] could not confirm a significant biological role to any of them.

While it is widely expected that constraints improve the quality of predictions, these and other observations suggest that the nature and extent of this effect are poorly understood. With the increasing availability of high-throughput RNA probing data, there is an increasing need for a better characterization of the behaviour prediction methods when experimental data are used as constraints, especially in the case of very large structures, such as viral genomes. Some of my contributions in this aspect are included in this thesis (Chapter 4 and Appendix A).

1.10 The HIV-1 genome as a case study

The genome of the HIV-1 virus is a ∼10 kb-long RNA, and due to its length, it has been used as a hallmark demonstration of the performance of RNA secondary structure prediction methods [1.30, 33, 108, 109].

The RNA genome of HIV-1 is situated in the viral capsid as two non-covalent linked positive stranded RNAs, each with a total length of 10 kb. It encodes several layers of information required for the viral replication cycle both for instructing protein synthesis and as functional RNA elements. Each strand of proviral RNA is flanked by long terminal repeats (LTRs), with the “Unique 3’” region missing in the 5’ end, and the “Unique 5’” region missing at the 3’ end. Proteins are encoded in structural (gag, pol, env), regulatory (tat, rev) and accessory (nef, vif, vpr, vpu) genes that have up to three overlapping reading frames in the central part of the genome. HIV-1 is alternatively spliced, giving rise to proteomic diversity at different stages of its replication cycle. The HIV-1 RNA transcript is 5’-capped, 3’-polyadenylated and contains several well-characterized RNA elements. The 5’ UTR of the transcript contains important structural signals for gene expression [1.110, 111]). It begins with the highly conserved transactivator of region (TAR) hairpin which recruits the viral Tat protein [1.112, 113]. The polyA-hairpin contains an unused polyadenylation signal, but plays an important role in nuclear export, dimerization and packaging [1.114]. The primer binding site (PBS) primarily binds the host cell tRNA-Lys3. The packaging signal Psi (ψ) plays a role in RNA packaging and includes a major splicing donor site and the dimerization initiation site (DIS) that mediates RNA-RNA interactions between the two RNA strands through the formation of a “kissing loop”.

Further downstream, the gag-pol frameshift signal controls the ratio of Gag and Gag-Pol polyproteins, and a slippery sequence and an RNA stem are believed to be required for ribosomal frameshifting. Towards the 3’ end of the genome, the Rev response element (RRE) [1.115] is a highly structured, 350 nucleotide segment in the Env coding region. The RRE recruits the viral Rev protein, and facilitates the nuclear export of the unspliced viral mRNA.
A number of other structural studies have elucidated additional functional RNA structures in the HIV-1 genome, some of which have been reported to play roles in all aspects of the viral replication cycle, including modulation of ribosome processivity [1.106], alternative splicing [1.116], recombination-mediated gene swapping [1.117], protein evolution [1.118], dimerization [1.119, 120] and circularization of the genome [1.121, 122], as well as evading host defence mechanisms [1.123].

A detailed secondary structure model has been proposed for an entire HIV-1 genome, based on nucleotide-level probing data from the SHAPE technique and a thermodynamic structure prediction algorithm [1.106]. This first complete structural model supports many previously suggested structures, but also features a large number of previously uncharacterized secondary structures, and has raised several questions.

In Chapter 6, I present an alternative model for HIV-1 genome structure, based on a combined phylogenetic and SHAPE-driven prediction made with PPfold. This forms the final paper in my PhD dissertation.

1.11 Bibliography

Textbooks, which I have used in the preparation in this chapter but not cited directly, are included as the last items in the list.


Chapter 1. Introduction


Chapter 1. Introduction


Chapter 1. Introduction


Chapter 1. Introduction


Chapter 2

Multithreaded comparative RNA secondary structure prediction using stochastic context-free grammars

2.1 Problem statement

The motivation behind this project was that the popular pfold algorithm was not practically useful for predicting the secondary structure of HIV-1, for two main reasons. Firstly, the initially published version [2.1] experienced underflow due to multiplying probabilities, and the problem increased both with sequence length and sequence number in the alignment. Secondly, a “patched”, non-optimized (and unpublished) version of pfold, which fixed the underflow problems, took several weeks to produce a single folding the HIV-1 genome.

The decision was then made to develop a multithreaded version of pfold, named PPfold, in the context of the “PC Mini-Grids for RNA structure prediction” project, which was a collaboration between Aarhus University, IT University in Copenhagen and CLC bio. This placed particular design requirements on PPfold, most importantly:

- The programming language had to be Java, to enable it to also run as a plugin in the CLC Workbenches.
- It had to have a non-shared-memory design, to enable it to run on a mini-grid of computers.

2.2 Results

In PPfold, both the phylogenetic calculations (Felsenstein algorithm for computing single- and basepaired column likelihoods) and the inside-outside algorithm were
Chapter 2. Multithreaded comparative RNA secondary structure prediction using stochastic context-free grammars

distributed. The phylogenetic calculations were distributed by simply dividing the alignment columns between nodes. The inside-outside algorithm was distributed by exploiting the natural dependencies in the computations (Figures 3 and 4). The parallelization was implemented in an asynchronous execution pattern, in which jobs are queued for execution immediately after dependencies are completed. Both the phylogenetic and the SCFG parts of the algorithm scaled very well with the number of available cores (Figure 6), resulting in a significant runtime reductions on multicore machines (Table 1).

The floating-point underflow problem was solved by implementing a base-2, 64-bit, extended-exponent datatype. The datatype performed very well for large RNA alignments, both in terms of overall accuracy (Table 2) and numerical stability (Table 3).

PPfold is the first parallelized comparative RNA structure prediction algorithm to date. At the same time as the first release, PPfold was also made available as a free plugin to the CLC Workbenches.

2.3 Notes

Since its publication, two major updates have been released to PPfold. In version 2, the full pfold-like pipeline was completed with the neighbour-joining algorithm for generating a phylogenetic tree from an alignment, and the Felsenstein algorithm for the optimization of branch lengths. In version 3, the model was further extended with support for quantitative experimental data, and a user-friendly GUI was also developed. Version 3 was also published as a separate publication (Chapter 3).

Since 2011, PPfold has also been consistently distributed through CLC bio as a plugin to all versions of the CLC workbenches. PPfold (version 2) has been evaluated by the independent CompaRNA server [2.2], which provides continuous benchmarking of available RNA structure prediction methods. PPfold generally scores very highly among other methods in these tests. As an example, Figure 2.1 shows a benchmarking dataset derived from the RNAstrand database [2.3], but similar patterns are observed for other datasets too.

PPfold has also been incorporated into the StatAlign tool (version 2) [2.4], and used in a biological application by Rěblová et al. [2.5].

2.4 Bibliography


Figure 2.1: Comparison of PPfold with other methods, according to CompaRNA. The figure shows the “All RNAs from the RNAstrand dataset” test, the July 30, 2013 update from the CompaRNA website. 1987 sequences were used, with lengths varying between 20-30000 nt. Pseudoknotted structures and non-canonical basepairs were included. “Seed” alignments include all RNA sequences from the Rfam [2.6] seed alignment and the query RNA sequence. Alignments marked with “20” consist of the query sequence and 19 randomly selected sequences from the seed alignment, with an identity between 65 and 95% in comparison with the query. If the method requires an alignment input, CompaRNA also re-aligns the sequences with Infernal [2.7], using the covariance model for the given family. The Mathews Correlation Coefficient is a measure for the overall accuracy of the method, taking into account both positive predictive value (specificity) and sensitivity. http://iimcb.genesilico.pl/comparna/rankings/rnastrand/110/
Multithreaded comparative RNA secondary structure prediction using stochastic context-free grammars

Zsuzsanna Sükösd¹²*, Bjarne Knudsen³, Morten Værrum³, Jørgen Kjems¹² and Ebbe S Andersen¹²*

Abstract

Background: The prediction of the structure of large RNAs remains a particular challenge in bioinformatics, due to the computational complexity and low levels of accuracy of state-of-the-art algorithms. The pfold model couples a stochastic context-free grammar to phylogenetic analysis for a high accuracy in predictions, but the time complexity of the algorithm and underflow errors have prevented its use for long alignments. Here we present PPfold, a multithreaded version of pfold, which is capable of predicting the structure of large RNA alignments accurately on practical timescales.

Results: We have distributed both the phylogenetic calculations and the inside-outside algorithm in PPfold, resulting in a significant reduction of runtime on multicore machines. We have addressed the floating-point underflow problems of pfold by implementing an extended-exponent datatype, enabling PPfold to be used for large-scale RNA structure predictions. We have also improved the user interface and portability: alongside standalone executable and Java source code of the program, PPfold is also available as a free plugin to the CLC Workbenches. We have evaluated the accuracy of PPfold using BRaliBase I tests, and demonstrated its practical use by predicting the secondary structure of an alignment of 24 complete HIV-1 genomes in 65 minutes on an 8-core machine and identifying several known structural elements in the prediction.

Conclusions: PPfold is the first parallelized comparative RNA structure prediction algorithm to date. Based on the pfold model, PPfold is capable of fast, high-quality predictions of large RNA secondary structures, such as the genomes of RNA viruses or long genomic transcripts. The techniques used in the parallelization of this algorithm may be of general applicability to other bioinformatics algorithms.

Background

Recent years have seen an explosion in the amount of biological data available from large-scale genome sequencing projects, but this increase has not been met by a corresponding increase in single-core computer power to bioinformatically analyze this data. It is therefore predicted that the scientific community will face serious computational problems in the coming years in their efforts to interpret genomic data. The prediction of RNA secondary structure remains a particularly challenging problem, in a large part due to its computational complexity: even without pseudoknot prediction, the execution time of state-of-the-art algorithms scales as $O(L^3)$ or worse with the length of the sequence, $L$. One way to address this problem is by exploiting heuristics to reduce complexity, but this happens at the cost of accuracy in predictions, which is particularly detrimental in the case of already inaccurate algorithms. Another possibility is to apply emerging multithreading paradigms to more accurate algorithms, and obtain the precise results in a fraction of the time. RNA secondary structure prediction algorithms are typically based on either thermodynamic or stochastic context-free grammar (SCFG) models, and are implemented using dynamic programming. A recent review [1] gives an overview over existing tools. Previous attempts to parallelize RNA structure prediction algorithms have included thermodynamic prediction [2-5] and the SCFG-based profiling of RNAs [6], as well as...
massively parallel genetic algorithms [7] and hardware-accelerated folding on FPGA chips [8] and GPUs [9]. Despite improved runtimes, the accuracy of these algorithms remains low, due to models that may not be appropriate for very long sequences.

Here we focus on pfold [10,11], which couples a phylogenetic model to a SCFG to accurately predict the consensus structure of RNA alignments in $O(L^3)$ time [12] (Figure 1). Due to the combined approach, the pfold model is theoretically capable of obtaining high-quality predictions of large and biologically significant RNA structures, such as the genomes of RNA viruses. However, it has not been possible in practice to use pfold for such predictions: it is single-threaded, so it can take days to fold a long alignment, and it fails to predict large structures correctly due to floating-point underflow errors [12].

In this study, we address both of these issues, and create PPfold, an improved and multithreaded version of pfold. To our knowledge, PPfold is the first example of a multithreaded comparative RNA secondary structure prediction algorithm. We demonstrate its practical use by predicting the secondary structure of an alignment of 24 HIV-1 genomes.

![Figure 1 Comparison of RNA secondary structure prediction algorithms](image)

**Results and Discussion**

**Algorithm**

PPfold uses the same combined evolutionary and SCFG model as pfold [10,11]. A summary of this model is provided in the Methods section. Here we focus strictly on the parts of PPfold that present improvements on pfold.

**Multithreading the phylogenetic calculations**

After estimating the phylogenetic tree, the pfold algorithm calculates column- and column-pair likelihoods, based on post-order traversal through the tree. We have only distributed the calculation of column pair likelihoods, as this is the most time-consuming part with a time complexity of $O(L^3)$, where $L$ is the length of the alignment. It is desirable to distribute the calculations as evenly as possible, so all processing units have an equal workload. As all column-pairs are treated independently, a natural division for multithreading arises by grouping a number of column pairs together in such a way that there are as many groups with equal numbers of column pairs as processing units. However, a unique mapping from the number of groups, $n$, to the size of each group, $s$, does not exist.

Column-pair likelihoods are symmetric, so in total there are $\frac{1}{2}L(L-1)$ column pairs to calculate. For simplicity, we chose to distribute these on the basis of the first iterator column: to each group, we incrementally assign as many first iterator columns (and all their pairing columns) as possible, such that the total number of column pairs in all groups up to group number $k > 0$ does not exceed

$$s_k = k \left( \frac{L^2}{2n} - \frac{1}{2} \mod n \right)$$

This provides an approximately even distribution of workload to the processing units, and we observe a corresponding reduction in running time on multicore machines. (Figure 2)

**Multithreading the inside-outside algorithm**

The inside-outside algorithm fills two lower triangular matrices of dimension $L$ for each nonterminal symbol of the grammar, through dynamic programming. The Knudsen-Hein grammar contains 3 nonterminal variables, so in total there are 6 such matrices to be filled. The algorithm exhibits heavy dependencies, making its distribution into independent “jobs” nontrivial. In PPfold, we have chosen an asynchronous wavefront computational approach that exploits the geometry of the algorithm.

We divide the triangle into equal-sized parallelogram-shaped “sectors” (Figure 3). We will refer to the number of sectors in the first row of the triangle by $N$. The dependency of the sectors on each other in the inside
and outside parts of the algorithm is illustrated by Figure 4; the values for all nonterminals in each sector can be evaluated once all dependencies are completed. A “job” thus entails the evaluation of either the inside or the outside variables corresponding to a sector for all nonterminal variables in the grammar. The workload in jobs is not equally distributed, as illustrated by Figure 5.

By design, jobs are self-sufficient objects created only when their dependencies are completed: they contain all data necessary for the calculation of the values in the corresponding sector in order to also allow distribution to a non-shared memory framework. Asynchronous implementation makes it possible to execute jobs immediately after the necessary dependencies are completed, rather than waiting for all jobs in the same row to complete. (Additional File 1) This is ideal for a setting where executor units have different capabilities, such as a grid of personal computers.

It is important to note that multithreading is not possible for all parts of the algorithm: for example, the job at the top of the triangular matrix has to be executed by one processing unit without any simultaneous calculations. Therefore it is ideal to choose \( N >> u \), where \( u \) is the number of available processing units. In the limit \( N \rightarrow \infty \), the theoretical execution time on \( u \) processing units is reduced to \( 1/u \) of the execution time on one processing unit, and this is also what we observe in practice. (Figure 6) We note that this method of divisions is generally applicable to any bifurcating SCFG, and thus may be used for the parallelization of other algorithms also.

Memory use is optimized with a large number of divisions, where only the lower triangular matrices are stored. However, the space complexity of the algorithm remains \( O(L^3) \), and roughly 6 GB of memory are needed to fold a 10000 nucleotide-long alignment.

**Multithreading expectation maximization**

In contrast to many SCFG-based RNA secondary structure prediction programs, \( pfold \) returns the structure with the maximum number of expected correctly predicted nested structural elements, instead of the maximum likelihood estimate. To find this structure, it is necessary to calculate a matrix of expectation values for every fraction of the alignment, similarly to the inside algorithm. The details of this are described in the Methods section. As these calculations also contain bifurcations, they are distributed analogously to the inside algorithm.

**Underflow**

Floating-point underflow arises commonly in dynamic programming for the folding of long RNAs, due to the multiplication of several thousands of low probabilities with each other. It has effectively prevented the correct folding of large RNA alignments with \( pfold \), as the values of the inside-outside variables decrease both with the length of alignment the number of sequences in it. A common solution is to calculate with log-probabilities, implementing addition as a “log sum” function with a
lookup table. Other possible solutions include multiplying the rules of the grammar with a factor, such that underflow is reached more slowly, thus extending the foldable length of alignments, or multiplying a block of probabilities lower than a certain cutoff value by a scaling factor.

In PPfold we have taken an alternative approach and implemented an extended exponent datatype, consisting of a float “fraction” and an integer “exponent” (base 2) part. Together, 64 bits are used to store each number - the same amount of space as a double-precision floating point number, so the overall memory requirements of the algorithm are not increased substantially. For each nonterminal symbol (in the inside, outside and expectation parts of the algorithm), we store two 2-dimensional arrays: one for the exponents

Figure 4 Dependencies of sectors in the inside-outside algorithm. The geometry for job divisions is inspired by the dependency structure of the (a) inside and (b) outside algorithms. For illustration purposes, here the number of divisions of the sequence is 11, giving rise to 66 sectors in total (sectors are numbered starting from 0, in accordance with the algorithm implementation). In order to calculate values in the sector marked red, the indicated values in the coloured sectors must be known.

Figure 5 Workload of jobs in the inside-outside algorithm. The execution time is proportional to job height, as illustrated here for the (a) inside and (b) outside algorithms. Jobs executed last take longest (top job in inside algorithm, bottom row in outside algorithm). (The expectation value calculations are analogous to the inside algorithm, but each job takes comparatively shorter time.) The linear dependence on job height gives rise to the $O(n^3)$ time complexity.
and one for the fractions. For every operation, we combine numbers from these arrays using custom bitmasking and bitshifting methods.

Implementation

PPfold has been written in Java 5.0, and integrated into the CLC Workbenches using the CLC Developer Kit (version 3.31) API. The source code and executables are available for download at http://www.daimi.au.dk/~compbio/pfold/downloads.html.

PPfold consists of an "algorithm" package that can be compiled and run independently of the CLC Workbenches, and a "plugin" package that provides interfacing with the CLC Workbenches.

The "algorithm" package includes all classes that are involved in the processing of sequences and alignments and creating the final structures. It has no dependencies on any CLC classes, is capable of taking command-line arguments and provides a simple graphical user interface for the selection of input files. Export formats currently supported by PPfold include, .ct, .seq (with reliability scores) and white text, as well as basing error probability data.

The "plugin" package makes use of the CLC Developer Kit API such that PPfold becomes a full-featured plugin to the CLC Workbenches. Furthermore, integration into "minigrid-enabled" CLC Workbenches makes it possible to distribute calculations to a collaborative mini-grid of computers [13]. The details of this aspect of our work will be published elsewhere.

Testing and benchmarking

Performance

We have evaluated the speed of our algorithm for alignments of various sizes, with a varying number of cores and divisions of calculations. (Table 1) The algorithm is fast, scales well with the number of cores and makes the folding of long alignments practically possible.

Accuracy

We have replicated the BRaliBase I benchmarking tests [12]. PPfold performs as well as pfold for short sequences (tRNA, RNaseP), and significantly better than pfold for longer sequences (SSU, LSU), as it does not suffer from the underflow problem. (Table 2)

Folding of the HIV-1 genome

To demonstrate the speed and accuracy of our algorithm, we have folded an alignment of 24 full HIV-1 genomes.

Table 1 Performance

<table>
<thead>
<tr>
<th>Alignment</th>
<th>divisions</th>
<th>1 core (sec)</th>
<th>2 cores (sec)</th>
<th>4 cores (sec)</th>
<th>8 cores (sec)</th>
<th>pfold (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 × 500 nt</td>
<td>1</td>
<td>5.41</td>
<td>4.85</td>
<td>4.84</td>
<td>4.84</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5.74</td>
<td>3.06</td>
<td>2.15</td>
<td>2.16</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>3.70</td>
<td>2.26</td>
<td>1.25</td>
<td>0.92</td>
<td></td>
</tr>
<tr>
<td>20 × 1000 nt</td>
<td>1</td>
<td>31.8</td>
<td>32.0</td>
<td>31.1</td>
<td>30.4</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>46.6</td>
<td>27.9</td>
<td>19.0</td>
<td>19.6</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>35.7</td>
<td>18.3</td>
<td>9.7</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>30 × 3000 nt</td>
<td>1</td>
<td>1738</td>
<td>1640</td>
<td>1581</td>
<td>1464</td>
<td>368</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1476</td>
<td>878</td>
<td>642</td>
<td>652</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>842</td>
<td>429</td>
<td>217</td>
<td>123</td>
<td></td>
</tr>
</tbody>
</table>

The actual execution time of PPfold (including both the phylogenetic and SCFG parts) on a Dell Precision T7500 Workstation with Dual Quad Core Intel® Xeon® X5667 3.07 GHz CPU, 24 GB RAM, is shown, for alignments of different lengths, choosing different divisions, and enabling different number of cores to be used by PPfold by varying the size of the thread pool. A small number of divisions can in some circumstances result in disproportionately long runtimes, due to the higher number of extra (unnecessary) points that are present in the calculations. The algorithm is intended to be run using a high number of divisions on all architectures. For comparison, we also include the runtimes of the original pfold implementation (written in C), which suffers from underflow, making the results unreliable for alignments of these lengths.
The prediction of the full consensus secondary structure of a large viral genome alignment, including phylogenetically supported long-distance interactions, has not previously been

### Table 2 BRaliBase accuracy

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Program</th>
<th>Ref. basepairs</th>
<th>Pred. basepairs</th>
<th>Sensitivity, %</th>
<th>Selectivity, %</th>
<th>Correlation, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>tRNA (M)</td>
<td>PPfold</td>
<td>20</td>
<td>21</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>tRNA (M)</td>
<td>pfold</td>
<td>20</td>
<td>21</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>tRNA (H)</td>
<td>PPfold</td>
<td>20</td>
<td>21</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>tRNA (H)</td>
<td>pfold</td>
<td>20</td>
<td>21</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>tRNA M</td>
<td>PPfold</td>
<td>110</td>
<td>110</td>
<td>86.4</td>
<td>96.0</td>
<td>91.2</td>
</tr>
<tr>
<td>tRNA M</td>
<td>pfold</td>
<td>110</td>
<td>110</td>
<td>86.4</td>
<td>96.0</td>
<td>91.2</td>
</tr>
<tr>
<td>tRNA H</td>
<td>PPfold</td>
<td>110</td>
<td>110</td>
<td>86.4</td>
<td>96.0</td>
<td>91.2</td>
</tr>
<tr>
<td>tRNA H</td>
<td>pfold</td>
<td>110</td>
<td>110</td>
<td>86.4</td>
<td>96.0</td>
<td>91.2</td>
</tr>
<tr>
<td>SSU (M)</td>
<td>PPfold</td>
<td>468</td>
<td>436</td>
<td>74.4</td>
<td>86.1</td>
<td>82.2</td>
</tr>
<tr>
<td>SSU (M)</td>
<td>pfold</td>
<td>468</td>
<td>436</td>
<td>74.4</td>
<td>86.1</td>
<td>82.2</td>
</tr>
<tr>
<td>SSU (H)</td>
<td>PPfold</td>
<td>468</td>
<td>373</td>
<td>68.4</td>
<td>89.1</td>
<td>78.8</td>
</tr>
<tr>
<td>SSU (H)</td>
<td>pfold</td>
<td>468</td>
<td>373</td>
<td>68.4</td>
<td>89.1</td>
<td>78.8</td>
</tr>
<tr>
<td>LSU (M)</td>
<td>PPfold</td>
<td>839</td>
<td>838</td>
<td>58.2</td>
<td>62.5</td>
<td>60.3</td>
</tr>
<tr>
<td>LSU (M)</td>
<td>pfold</td>
<td>839</td>
<td>838</td>
<td>58.2</td>
<td>62.5</td>
<td>60.3</td>
</tr>
<tr>
<td>LSU (H)</td>
<td>PPfold</td>
<td>839</td>
<td>754</td>
<td>52.2</td>
<td>61.0</td>
<td>56.9</td>
</tr>
<tr>
<td>LSU (H)</td>
<td>pfold</td>
<td>839</td>
<td>754</td>
<td>52.2</td>
<td>61.0</td>
<td>56.9</td>
</tr>
</tbody>
</table>

The results of our tests on the BRaliBase I dataset, using the comparison script available at [http://projects.binf.ku.dk/pgardner/bralibase/bralibase1.html](http://projects.binf.ku.dk/pgardner/bralibase/bralibase1.html), are shown, both for *pfold* and *PPfold*. We carried out stem-extension separately on the *PPfold* predictions, as it is not implemented in *PFold* at present. M: medium-similarity, H: high-similarity.

Additionally, genomes using *PPfold* in 65 minutes on a Dell Precision T7500 Workstation with Dual Quad Core Intel® Xeon® X5667 3.07 GHz CPU, using 6 GB of memory. In addition to predicting the consensus structure within a practically reasonable timeframe, *PPfold* has also predicted a number of known RNA structures, including the TAR, poly(A), PBS, DIS, AUG hairpins, the gag-pol frameshift and the RRE region. (Figure 7) [14] The prediction of the full consensus secondary structure of a large viral genome alignment, including phylogenetically supported long-distance interactions, has not previously been
Table 3 Numerical stability

<table>
<thead>
<tr>
<th>Number of divisions</th>
<th>P (S → C_i,C_j)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>1.7616076 × 10^16</td>
</tr>
<tr>
<td>35</td>
<td>1.7614585 × 10^16</td>
</tr>
<tr>
<td>60</td>
<td>1.7610938 × 10^16</td>
</tr>
</tbody>
</table>

The values of the S variable from the inside algorithm representing a full-length HIV-1 genome alignment (P = C_i,C_j) stays stable correct to at least 3 significant digits, when carrying out the calculations in a different order (with different numbers of divisions).

Methods

Summary of the pfold model

Given an alignment, pfold creates a phylogenetic tree by neighbour-joining, then optimizes branch lengths using a maximum likelihood approach based on a general reversible evolutionary model described by Felsenstein [15]. Column-based likelihoods for unpaired nucleotides and basepairs are obtained from the stochastic context-free grammar:

\[ P(S \rightarrow C_i,C_j) = \prod_{i=1}^{k-1} P(S \rightarrow C_i,C_j) \]

The matrix of expectation values is defined by the following recursion relation:

\[ E_{ij} = \begin{cases} E_{i,j-1} + 2P_d(i,j) & \text{(Basepairing)} \\ E_{i,k} + E_{k,j}, & k \leq j \end{cases} \]

with initialization conditions \( E_{i,j} = P_d(i,j) \) for all \( i \), where the basepair probabilities \( P_d \) and unpaired base probabilities \( P_s \) are obtained from the inside-outside variables. The final structure returned to the user is obtained by backtracking in this matrix. The reader is advised to consult references [10] and [11] for more details on the pfold model and algorithm.

Additional material

Additional file 1: Animation of distribution of the SCFG calculations

The animation demonstrates the asynchronous wavefront computational approach, and was created on the basis of actual runtime data during the folding of a sequence of 460 nt, on a 2-core machine. The animation is divided into three parts: the inside (red/green), outside (yellow/blue) and expectation (cyan/magenta) calculations. The first colour represents jobs that are ready to be executed (because all their dependencies are fulfilled), and are therefore placed in a queue. At any time during the animation, the two jobs that entered the queue earliest are under execution (not shown), as the execution happens in a threadpool corresponding to the available number of cores (here, 2). When a job is finished, it changes to the second colour, and any newly available jobs (with finished dependencies) are pushed onto the queue.
Chapter 2. Multithreaded comparative RNA secondary structure prediction using stochastic context-free grammars


Cite this article as: Sükösd et al. Multithreaded comparative RNA secondary structure prediction using stochastic context-free grammars. BMC Bioinformatics 2011, 12:103.

Submit your next manuscript to BioMed Central and take full advantage of:
• Convenient online submission
• Thorough peer review
• No space constraints or color figure charges
• Immediate publication on acceptance
• Inclusion in PubMed, CAS, Scopus and Google Scholar
• Research which is freely available for redistribution

Submit your manuscript at www.biomedcentral.com/submit
3.1 Problem statement

The emergence of high-throughput RNA probing techniques has revolutionized RNA secondary structure prediction, by making it possible to study the structure of very large RNAs quantitatively and at single-nucleotide resolution. These experimental developments must be matched with statistically rigorous theoretical models to make use of the probing data. As described in Section 1.9, many approaches have been presented for incorporating experimental data into RNA structure predictions. However, before PPfold 3.0, phylogenetic and probing data had only been used independently. In this project, the purpose was to combine phylogenetic and data-driven RNA secondary structure prediction in a statistically well-justified framework.

3.2 Results

PPfold 3.0 extends the evolutionary analysis of the pfold model with a flexible probabilistic model for incorporating auxiliary data, such as data from structure probing experiments. The approach is based on a Bayesian expression for the posterior probability of the structure, where the effects of the SCFG model, the phylogenetic model and a likelihood model for the data can be separated. The model was also implemented in practice, and tested using ribosomal data. The testing results show that the accuracy of single-sequence secondary structure prediction using experimental data in PPfold 3.0 is comparable to RNAstructure, and that alignment structure prediction quality is improved even further by the addition of experimental data (Table 1).
3.3 Notes

An important result in this paper is that the effects of the experimental data can be separated out from the rest of the model, and the existing framework for optimization in the inside-outside algorithm can still be used with minor modifications and no increase in computational complexity.

For the specific application to SHAPE data, an important assumption in the paper is that the experimental data only depend on the alignment through the structure, i.e. \( P(D, H | \sigma, M_s, M_t) = P(D | \sigma, M_s) P(H | M_s, M_t, \sigma) \), where \( D \) is the alignment, \( H \) is the experimental data, \( \sigma \) is the secondary structure, \( M_s \) is the SCFG part of the model and \( M_t \) is the evolutionary part of the model. This assumption is only partially justified, as SHAPE data do exhibit some degree of sequence-dependence [3.1]. Similarly, Equations (3) and (4) only hold if the data are independent of nucleotide position, and data for two bases in the same basepair are uncorrelated. If the assumption does not hold for a particular type of probing data, the general model presented in Equation (2) can still be used – however, it is then necessary to develop an extended model for the covariance between alignment and probing data. This can be done using the “advanced track” option implemented in PPfold.

The probabilistic approach in PPfold has many advantages over the previously proposed ways of incorporating pseudo-energy terms into prediction methods. Firstly, the PPfold model is highly versatile and is able to handle an arbitrary number of experimental datasets, also from different experiments. This is increasingly important, as the volume of RNA mapping data from different kinds of structure probing experiments is growing, and probing results are even being systematized in databases such as the RNA mapping database [3.2]. Secondly, the PPfold model is statistically rigorous. The data and its properties are used directly, rather than through an inferred quantity with somewhat ad-hoc fitted parameters. In the cases where the experimental data contains a high amount of information, its effects do not “drown” in a sea of other parameter in the model – rather, the data is at the very core of the model and algorithm. In the cases where the experimental data is noisy, the other parts of the model naturally dominate.

3.4 Bibliography


PPfold 3.0: fast RNA secondary structure prediction using phylogeny and auxiliary data

Zsuzsanna Sükösd¹,²,³,*, Bjarne Knudsen⁴, Jørgen Kjems¹,² and Christian N.S. Pedersen¹,³,⁵

¹Interdisciplinary Nanoscience Center, Aarhus University, Ny Munkagade 120 and ²Department of Molecular Biology and Genetics, Aarhus University, C. F. Møllers Alle 8, DK-8000 Aarhus C, Denmark, ³CLC bio, Finlandsgade 10-12 and ⁴Department of Computer Science, Aarhus University, Aabougade 34, DK-8200 Aarhus N, Denmark

ABSTRACT
Summary: PPfold is a multi-threaded implementation of the Pfold algorithm for RNA secondary structure prediction. Here we present a new version of PPfold, which extends the evolutionary analysis with a flexible probabilistic model for incorporating auxiliary data, such as data from structure probing experiments. Our tests show that the accuracy of single-sequence secondary structure prediction using experimental data in PPfold 3.0 is comparable to RNAstructure. Furthermore, alignment structure prediction quality is improved even further by the addition of experimental data. PPfold 3.0 therefore has the potential of producing more accurate predictions than it was previously possible.

Availability and implementation: PPfold 3.0 is available as a platform-independent Java application and can be downloaded from http://birc.au.dk/software/ppfold.

Received on June 11, 2012; revised on July 6, 2012; accepted on July 27, 2012

1 INTRODUCTION
The Pfold package predicts (non-pseudoknotted) RNA secondary structure by combining a stochastic context-free grammar (SCFG) with an evolutionary model (Knudsen and Hein, 1999, 2003). PPfold is a recent multi-threaded re-implementation of Pfold (Sükösd et al., 2011). The Pfold model has been shown to result in highly accurate predictions when the input alignment is of high quality (Gardner and Giegerich, 2004). In a different approach, data from high-throughput, quantitative RNA structure probing methods have also recently been used in thermodynamic prediction methods to increase prediction accuracy (Deigan et al., 2009; Washietl et al., 2012). However, phylogenetic and probing data have only been used independently in RNA secondary structure prediction so far. Here, we present PPfold 3.0, which integrates these different sources of information at the level of the model. This is expected to increase prediction accuracy beyond what is possible with either evolutionary information or experimental data alone.

*To whom correspondence should be addressed.

2 MODEL
Pfold combines two models: (1) a SCFG model $M_s$, which generates the prior probability distribution over secondary structures $\sigma$. The prior probabilities are denoted as $P(\sigma|M_s)$. (2) A phylogenetic model $M_z$, which computes the likelihood of the input alignment $D$, given the secondary structures. The likelihoods are denoted as $P(D|\sigma, M_z)$.

If an additional set of experimentally observed data, $H$, is available, the posterior probability of a secondary structure is

$$P(\sigma|D, H, M_z, M_s) = \frac{P(D, H|\sigma, M_z)P(\sigma|M_s)}{P(D|H, M_z, M_s)}$$

Expanding the expression and removing dependencies,

$$P(\sigma|D, H, M_z, M_s) = \frac{P(H|D, \sigma)P(\sigma|M_z)P(D|\sigma, M_s)}{P(D|H, M_z, M_s)}$$

where $P(D|\sigma, M_z)$ are the probabilities obtained from the phylogenetic part of the algorithm, and $P(\sigma|M_s)$ are the prior probabilities from the SCFG calculations. The quantity $P(H|D, \sigma)$ can be computed for any type of experimental data once a probabilistic model for its structure dependence is obtained. The posterior probability under the combined model can therefore be computed in the existing framework for optimization.

In the case of many chemical probing methods, for example Selective 2’-Hydroxyl Acylation Analyzed by Primer Extension (SHAPE: Wilkinson et al., 2009), the data are assumed to be independent of nucleotide identity and sequence position, and data values for nucleotides in the same pair are not correlated. For such data, $P(H|D, \sigma) = P(H|\sigma)$, which is readily obtained by measuring known structures with the method.

Let $H$ be the observed experimental value for position $i$, and $P(\sigma)$ and $P(H_i|\sigma)$ the likelihoods of the alignment (for single-stranded and base paired columns, respectively) calculated purely on the basis of the phylogenetic model. The combined likelihood of the alignment and data, $P(\sigma_i)$ (single-stranded case for alignment column $i$) and $P(\sigma_i, H_i)$ (base paired case for alignment columns $i$, $j$), can then be calculated as

$$P(\sigma_i) = P(\sigma_i)P(H_i|\sigma)$$

$$P(\sigma_i, H_i) = P(\sigma_i, H_i|\sigma)P(H_i|\sigma)$$
Chapter 3. PPfold 3.0: fast RNA secondary structure prediction using phylogeny and auxiliary data

3 IMPLEMENTATION
PPfold 3.0 has been written in Java 6 and consists of a single jar file. In addition to added support for experimental data, PPfold version 3.0 also features an intuitive graphical user interface. An arbitrary number of experimental data tracks can be added to each prediction. The following types of data tracks are currently supported:

1. Probing data can be given in the same format as in RNAstructure (Mathews et al., 2004). The distributions \( P(H) \) must also be given as histograms; a default distribution for SHAPE data is included in the application.

2. 'Hard' constraint data can be specified for a sequence in the alignment in the same format as in mfold (Markham and Zuker, 2008).

3. Advanced data tracks are also supported, where the \( P(H) / \) paired and \( P(H) / \) unpaired values are pre-computed by the user for some positions i of the sequence.

4 TESTING
We obtained SHAPE data for the Escherichia coli 16S and 23S rRNAs from the authors of Dugan et al. (2009) (personal communication), removed invalid data points and computed the \( P(H) / \) distribution histograms for paired and unpaired nucleotides at a resolution of 0.01 units. SHAPE data are currently only available for few sequences with known structures, so we used the data for the E. coli 16S rRNA for testing.

PPfold 3.0 is designed for structure prediction based on alignments. Nevertheless, the accuracy of single-sequence structure prediction for the E. coli 16S RNA sequence is greatly improved on the addition of SHAPE data and is comparable to that of RNAstructure (Table 1, upper block).

We also examined the effect of SHAPE data on the quality of predictions of various alignments: (a) a Clustal W2 sequence alignment of the highly divergent small ribosomal subunit (SSU) sequences from E. coli (accession number K00421) and Escherichia coli nuc1 (accession number X08487); (b) an R-coffee RNA alignment of the two same sequences; (c) the ‘SSU high similarity’ alignment from BAliBASE II (Gardner and Giegertich, 2004) and (d) the ‘SSU medium similarity’ structural alignment from BAliBASE II.

SHAPE data improved the quality of structure predictions for all alignments (Table 1, lower block). The highest quality predictions can be obtained when a high-quality alignment is combined with experimental data. In these cases, prediction accuracies exceed what has been observed for a single sequence with RNAstructure with additional data, or with PPfold without additional data. However, a single-sequence structure prediction using experimental data is more accurate than a prediction using a low-quality sequence alignment alone or in combination with experimental data. The quality of the input alignment must therefore still be considered when using PPfold 3.0.

In conclusion, PPfold 3.0 enables phylogenetic RNA secondary structure prediction in conjunction with experimental data and has the potential of producing highly accurate predictions.

ACKNOWLEDGEMENT
We would like to thank K. Weeks, from UNC Chemistry, for providing the experimental SHAPE data used in this work.

Funding: The Danish Council for Strategic Research under project number 09-061856.

Conflict of Interest: none declared.

REFERENCES

Table 1. Evaluation of the accuracy of single-sequence (upper block) and alignment (lower block) structure prediction

<table>
<thead>
<tr>
<th>Program</th>
<th>Input</th>
<th>Data</th>
<th>PPV (±0.005)</th>
<th>Sensitivity (±0.005)</th>
<th>F-measure (±0.005)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNAstructure</td>
<td>E. coli 16S</td>
<td>none</td>
<td>0.3246</td>
<td>0.3462</td>
<td>0.3351</td>
</tr>
<tr>
<td>RNAstructure</td>
<td>E. coli 16S</td>
<td>SHAPE</td>
<td>0.7221</td>
<td>0.7329</td>
<td>0.7275</td>
</tr>
<tr>
<td>PPfold</td>
<td>E. coli 16S</td>
<td>none</td>
<td>0.6358</td>
<td>0.2201</td>
<td>0.3270</td>
</tr>
<tr>
<td>PPfold</td>
<td>E. coli 16S</td>
<td>SHAPE</td>
<td>0.7560</td>
<td>0.6752</td>
<td>0.7133</td>
</tr>
<tr>
<td>PPfold</td>
<td>Clustal W2</td>
<td>none</td>
<td>0.3791</td>
<td>0.1789</td>
<td>0.2356</td>
</tr>
<tr>
<td>PPfold</td>
<td>Clustal W2</td>
<td>SHAPE</td>
<td>0.6590</td>
<td>0.5334</td>
<td>0.6016</td>
</tr>
<tr>
<td>PPfold</td>
<td>R-coffee</td>
<td>none</td>
<td>0.4519</td>
<td>0.2009</td>
<td>0.2781</td>
</tr>
<tr>
<td>PPfold</td>
<td>R-coffee</td>
<td>SHAPE</td>
<td>0.6745</td>
<td>0.5562</td>
<td>0.6096</td>
</tr>
<tr>
<td>PPfold</td>
<td>SSU (high)</td>
<td>none</td>
<td>0.8379</td>
<td>0.6538</td>
<td>0.7610</td>
</tr>
<tr>
<td>PPfold</td>
<td>SSU (high)</td>
<td>SHAPE</td>
<td>0.8616</td>
<td>0.7714</td>
<td>0.8140</td>
</tr>
<tr>
<td>PPfold</td>
<td>SSU (med)</td>
<td>none</td>
<td>0.8245</td>
<td>0.7436</td>
<td>0.7964</td>
</tr>
<tr>
<td>PPfold</td>
<td>SSU (med)</td>
<td>SHAPE</td>
<td>0.8721</td>
<td>0.8013</td>
<td>0.8552</td>
</tr>
</tbody>
</table>

PPV: positive predictive value; true positives as a fraction of predicted pairs. Sensitivity is the true positives as a fraction of reference base pairs. True positives are the correctly predicted base pairs. F-measure: the harmonic mean of PPV and sensitivity, and provides a measure for overall accuracy.
4.1 Problem statement

With the rapid developments in the quantitative, high-throughput methods for probing even very large RNA secondary structures, the accuracy of data-driven RNA secondary structure prediction for large RNAs is a potential concern. Even though remarkably high accuracies have been reported for SHAPE-directed RNA secondary structure prediction [4.1], only few known, large RNA secondary structures have been probed to date, and a wider assessment of SHAPE-directed RNA structure prediction for large RNAs has yet to take place.

The accuracy of SHAPE-directed RNA secondary structure prediction depends on many factors. Firstly, the SHAPE chemistry itself is highly stochastic in nature, and the various data analysis and normalization protocols required to produce the position-specific, normalized, unitless “reactivities” are rather complicated [4.2]. Secondly, the thermodynamic model itself is very complicated too, with hundreds of thermodynamic parameters [4.3] and small perturbations in parameters can result in erratic prediction outcomes [4.4]. A combination of the model with the experimental data is even less understood.

In this project, we wanted to separate the effects of the intrinsic uncertainties in the data from the effects of the ill-conditioned nature of the thermodynamic optimization, and investigate the behaviour of the method with simulated datasets for long biological sequences that fold into topologically very similar structures. By simulating data, we could precisely control the statistical noise in the data, and observe its effects on thermodynamic optimization.

4.2 Results

A diverse range of 16S/18S ribosomal sequences formed our dataset, as these sequences are relatively long (697 to 1562 nt), and their well-studied comparative
secondary structures are available and topologically very similar to each other.

Using experimental SHAPE data for the \textit{E. coli} 16S and 23S ribosomal sequences, we generated empirical distributions for the SHAPE values at nucleotides of different structural groups of RNA. For each position in the target RNA sequence, knowing the secondary structure category of the position, we generated a random simulated SHAPE value from the appropriate distribution by inverse transform sampling. For each sequence, we repeated this process 1000 times, producing 1000 simulated SHAPE datasets. We then predicted the secondary structure corresponding to each simulated dataset in a SHAPE-directed model using GTfold [4.5]. The accuracies of predictions produced in this way varied greatly, but some important patterns could be identified.

We found that using a model that distinguished unpaired, "helix-end" and stacked nucleotides, the very high accuracy of the SHAPE-directed \textit{E. coli} secondary structure prediction [4.1] could be replicated by the simulated datasets, well within statistical variation (Table 1). Such high accuracies, however, were not observed for all of our test sequences (Figure 2), even though the average accuracy for data-directed predictions always improved over the MFE structure. The amount of improvement was weakly correlated with MFE accuracy.

Interestingly, we found a correlation between the accuracy of the MFE and the similarity between the SHAPE-directed and MFE predictions (Figure 3). The reason for this is that accurate MFE base pairs are typically preserved in a data-directed prediction, whereas inaccurate ones are not (Figure 4, Table 2). While the reasons for this are unclear, we could conclude from this that the positive predictive value of common base pairs is consistently higher than the directed prediction accuracy.

The above results have two important implications. Firstly, a comparison of the MFE and a single data-directed prediction provides information about the most reliable basepairs. Secondly, for a sequence with an unknown structure, the similarity of the MFE and a single SHAPE-directed prediction can be used to broadly identify whether the MFE structure prediction is expected have an unusually high accuracy (in case of >60-70% similarity between the MFE and the data-directed prediction), an unusually low accuracy (in case of <30% similarity), or average quality (for intermediate similarity scores).

To investigate if the variations in prediction accuracies could have arisen due to over-fitting the SHAPE model to the \textit{E. coli} sequence, we used simulated SHAPE datasets to try to fit the SHAPE model to the best-performing (\textit{H. volcanii}) and the worst-performing (\textit{E. cuniculi}) sequences as well. We found that a very high accuracy could be obtained for a wide range of parameters when fitting to the \textit{H. volcanii} sequence, but no parameter combination resulted in a high accuracy for the \textit{E. cuniculi} prediction. We can conclude that there is no evidence that the parameters in the Deigan et al. [4.1] model were overfitted, and that the SHAPE data simply do not contain enough information to guide the thermodynamic method to a correct prediction of the \textit{E. cuniculi} structure.

To see if it was even theoretically possible to produce a good structure prediction for the \textit{E. cuniculi} sequence using quantitative data in the same pseudo-free energy model, we simulated data from theoretical distributions with a greater separation...
4.3 Notes

Even though it has been reported that SHAPE data for the same position are highly reproducible across several independent replicas [4.6], there is great variation for nucleotides within the same structural configuration (Figure 1 in the paper). It is this inter-nucleotide variation that is relevant for all existing methods for incorporating SHAPE data, which all assume that the reported SHAPE reactivities only depend on structural configuration. From the point of view of SHAPE-directed secondary structure prediction, it is irrelevant why the statistical noise in the SHAPE distributions for the same structural configuration arises in the first place.

The true strength of our method is therefore that our simulated datasets accurately reflect the inherent uncertainty of real-world SHAPE data, in the most relevant way from the point of view of thermodynamic optimization. Hence the uncertainties we observe in the final structure predictions also reflect the uncertainties of SHAPE-directed RNA secondary structure predictions in the model developed by Deigan et al. [4.1].

The implications of the patterns shown in Figures 2 and 3 and Table 2 are particularly powerful. In a biological application, we are typically given a single RNA sequence of unknown structure, and a set of SHAPE values for this sequence. We can then make two predictions: one with the SHAPE data in the thermodynamic method (the directed prediction), and one without (the MFE prediction). Calculating the similarity between these two predictions, we can identify the position of the sequence on the vertical axis of Figures 2 and 3. The strong positive correlation shown in Figure 3 allows us to identify a rough range of expected accuracy for the MFE prediction. Using Figure 2, we can get a rough idea of the probable accuracy of the directed prediction too. Finally, by doing a basepair-by-basepair comparison of the directed and the MFE structures, we can actually identify a set of basepairs in the predictions that are most likely to be correct.

Another important conclusion is that the data-directed model for RNA secondary structure prediction has great potential even in the case of the worst-predicted sequences, but it requires a better separation in the data distributions for unpaired and paired nucleotides. Even if it is not possible to obtain such well-separated distributions for one single probing method, a combination of data from independent probing methods could realistically be used.

In the development of future experimental methods, a comparison of the distributions for the different structural groups (as in Figure 1), could be highly useful to give an indication of the information content of the experimental data. Rigorous
statistical measures to measure the distinction between the two distributions, such as the Jensen-Shannon divergence (Equation 4.1), could also be used for this purpose. The Jensen-Shannon divergence between two probability distributions $P$ and $Q$ (with probability density functions $p(x)$ and $q(x)$, respectively) is defined by:

$$JSD(P|Q) = \frac{1}{2} D_{KL}(P|Q) + \frac{1}{2} D_{KL}(Q|P) \quad (4.1)$$

where $D_{KL}(P|Q)$ is the Kullback-Leibler divergence:

$$D_{KL}(P|Q) = \sum_x -p(x) \log q(x) - (-p(x) \log p(x)) = \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (4.2)$$

### 4.4 Errata and clarifications

A couple of minor points have been raised since the publication of this paper, none of which affect the main results and conclusions.

The function in Figure 1(a) should read:

$$P(X = x) = \frac{1}{\lambda} e^{-x/\lambda} \quad (4.3)$$

The paper does not mention that the SHAPE distributions in Figure 1 were derived from the pseudoknotted comparative secondary structures for *E. coli*, downloaded from the Comparative RNA website [4.7]. Whether choosing the pseudoknotted or non-pseudoknotted structure as reference, the parameters of the distribution fits are very similar (equal correct to 1 s.f.).

In the evaluation of the directability of the NNTM optimization, we used a Gaussian distribution. This distribution is entirely hypothetical and does not reflect the actual nature of the SHAPE reaction mechanisms; its only purpose was to demonstrate that if a future probing method (or combinations of methods) could obtain a greater separation of the SHAPE distributions for paired and unpaired nucleotides, even the otherwise poorly predicted secondary structures could be improved.

### 4.5 Bibliography


Evaluating the accuracy of SHAPE-directed RNA secondary structure predictions

Zsuzsanna Süköső1,2,3, M. Shel Swenson4, Jørgen Kjems1,2 and Christine E. Heitsch4,*

1Interdisciplinary Nanoscience Center, Aarhus University, Ny Munkegade 120, Aarhus C DK-8000, Denmark, 2Department of Molecular Biology and Genetics, Aarhus University, C.F. Møllers Alle 3, Aarhus C DK-8000, Denmark, 3Bioinformatics Research Center, Aarhus University, C.F. Møllers Alle 8, Aarhus C DK-8000, Denmark and 4School of Mathematics, Georgia Institute of Technology, 686 Cherry Street, Atlanta, GA 30332-0160, USA

Received August 20, 2012; Revised November 7, 2012; Accepted November 10, 2012

ABSTRACT

Recent advances in RNA structure determination include using data from high-throughput probing experiments to improve thermodynamic prediction accuracy. We evaluate the extent and nature of improvements in data-directed predictions for a diverse set of 16S/18S ribosomal sequences using a stochastic model of experimental SHAPE data. The average accuracy for 1000 data-directed predictions always improves over the original minimum free energy (MFE) structure. However, the amount of improvement varies with the sequence, exhibiting a correlation with MFE accuracy. Further analysis of this correlation shows that accurate MFE base pairs are typically preserved in a data-directed prediction, whereas inaccurate ones are not. Thus, the positive predictive value of common base pairs is consistently higher than the directed prediction accuracy. Finally, we confirm sequence dependencies in the directability of thermodynamic predictions and investigate the potential for greater accuracy improvements in the worst performing test sequence.

INTRODUCTION

RNA structure predictions have been advancing molecular biology research for decades (1,2), in conjunction with ongoing work (3–5) to improve prediction accuracy. Comparative methods (6,7) are the current gold standard for secondary structure determination, but a suitable set of homologous sequences is too often not available. Hence, thermodynamic optimization remains the most widely used approach to predicting RNA base pairings (8,9).

Thermodynamic optimization methods calculate an optimal structure for an RNA sequence according to the current objective function. Two of the best known prediction programs implementing this approach are UNAfold (10) and RNAfold (11). At their core, each takes an RNA sequence as input, and outputs a secondary structure, which has minimum free energy (MFE), according to the nearest neighbor thermodynamic model (NNTM) (12). Despite the significant utility of these RNA secondary structure predictions to molecular biologists, a crucial caveat has always been the ‘fundamental ill-conditioning of the folding problem’ (13).

One way to understand this issue is through the explosion of suboptimal structures (14); there are an exponential number of distinct structures (15), and even of more abstract ‘shapes’ (16), within a small range of the computed optimum. Equivalently, small perturbations in the NNTM objective function can result in significant changes in the optimal structure (17,18). Either way, this ill-conditioning is one of the reasons why thermodynamic optimization methods are often insufficient on their own (19–22) to predict native base pairings accurately.

Nonetheless, NNTM prediction accuracies can be improved significantly by incorporating auxiliary information (13). For example (22,23), additional criteria can be imposed on the optimization, such as enforcing single-strandedness in regions of high chemical reactivity or prohibiting base pairs between distant nucleotides. In optimization parlance, these are ‘hard’ constraints, as they must necessarily be satisfied in the predicted structure(s).

In contrast, ‘soft’ constraints direct the optimization towards greater accuracy by modifying the reward/penalty structure of the objective function. A notable example of this is the increasing prevalence of SHAPE-directed RNA secondary structure predictions (24–27), including an entire HIV-1 genome (28). These predictions use data from high-throughput chemical probing experiments commonly known as SHAPE (29), for ‘selective’ 2’-hydroxyl acylation analyzed by primer extension’, as soft constraints on the thermodynamic optimization. Although SHAPE-directed prediction
accuracies above 95% have been achieved for large ribosomal RNA sequences (25), such marked improvements have not been universally observed (30).

This discrepancy indicates a need to understand better the extent and nature of improvements in directed prediction accuracy. The problem is that, unlike hard constraints, soft constraints interact with the massive NNTM objective function in ways which are difficult to analyse directly. We address this challenge using a stochastic model of SHAPE data to investigate the accuracy of data-directed predictions for a diverse set of 16S/18S ribosomal sequences.

We find that incorporating this auxiliary data into the thermodynamic optimization as soft constraints consistently improves the directed prediction accuracy over the original MFE structure. However, the extent of the improvement is sequence dependent and roughly correlated with MFE accuracy. Notably, the data-directed predictions for more than 1/3 of our ribosomal test sequences do not achieve the high accuracy of the other 10 sequences—which are at the level of previous experimental studies (25).

Investigating the nature of the correlation between data-directed and MFE prediction accuracies, we find that accurate MFE base pairs are typically preserved in a data-directed prediction, whereas inaccurate ones are not. Thus, if the similarity between the two structures is especially low or high, this provides information about the correlated accuracies of the MFE and data-directed predictions. Furthermore, we show that the positive predictive value (PPV) of the common base pairs is high, even for sequences where the prediction accuracy is not.

Finally, we illustrate clear sequence dependencies in the directability of thermodynamic optimization. Our results show that soft constraints based on SHAPE data are not always sufficient to overcome limitations (19–22) in the NNTM approximation to RNA folding. However, we also give a proof-of-concept demonstration that auxiliary data which more clearly distinguishes between the presence and absence of base pairs can modify the NNTM sufficiently to consistently achieve high prediction accuracy.

**MATERIALS AND METHODS**

**MFE and SHAPE-directed predictions**

For the purposes of thermodynamic optimization, an RNA secondary structure is defined to be a set of base pairs without pseudoknots. Two consecutive base pairs form a stack, and consecutive stacks create a helix. Unpaired nucleotides belong to a single-stranded loop substructure.

Each stack and loop is assigned a thermodynamic value under the NNTM, and the free energy change of the entire secondary structure is approximated by summing over these substructure values. RNA molecules fold to minimize free energy, and an optimal MFE structure according to the NNTM can be computed efficiently using dynamic programming.

As a model of RNA base pairing, the NNTM is known to be more accurate for shorter sequences, including domains within longer ones (19,22). It is also well-known that the quality of the thermodynamic approximation has significant sequence dependencies. In particular, prediction accuracies can vary widely even for sequences that fold into essentially the same secondary structure (20,21).

To improve thermodynamic prediction accuracy, SHAPE data can be incorporated into the optimization as soft constraints. Details of the experimental method are summarized in (31). For our purposes, it suffices to know that SHAPE interrogates conformational flexibility at single nucleotide resolution. Low values are strongly correlated with base pairing—as well as other stabilizing interactions (32,33).

This auxiliary information is used to modify the reward/penalty structure of the thermodynamic objective function. More specifically, the standard practice (25), implemented in the RNAstructure prediction program (34), is to convert the SHAPE value for nucleotide $i$ into a pseudo-free energy term according to the equation:

$$
\Delta G_{\text{SHAPE}}(i) = m \ln(\text{SHAPE}(i+1)) + b
$$

using slope $m = 2.6$ kcal/mol and intercept $b = -0.8$ kcal/mol as parameters. The NNTM is then modified by adding the $\Delta G_{\text{SHAPE}}(i)$ term to the free energy change of each base pairing stack involving nucleotide $i$. Our work assesses the extent and nature of accuracy improvements in such SHAPE-directed predictions using simulated data.

**Modeling SHAPE data**

We give a probabilistic method for simulating SHAPE data for sequences with known secondary structures. Our stochastic model is based on experimental data (Weeks, personal communication) for two *Escherichia coli* ribosomal sequences, 16S rRNA with 1542 nt and 23S rRNA with 2904 nt. Nucleotides without data were removed, leaving a total of 4187 nucleotides in the experimental data set.

We considered three different divisions of this data set for our model. Nucleotides were classified into the categories given below according to the comparative secondary structures (35) for the two *E. coli* sequences. All variation in values within a subdivision was treated as random. We modeled this uncertainty with empirical probability density functions obtained by maximum likelihood fitting using the Statistics Toolbox in MATLAB R2010b. In order of increasing complexity, the models and their subdivision are as follows:

**Unary model**

No division of the SHAPE data set; one probability distribution fit to all values.

**Binary model**

Data set is divided into paired and unpaired nucleotides with different probability functions.

**Ternary model**

Paired nucleotides are further subdivided into stacked or helix-end pairs. The three distinct probability density functions are shown in Figure 1.
The first is our null model. The second reflects that SHAPE chemistry measures conformational flexibility with clear differences between paired and unpaired nucleotides on average. The third distinguishes ‘stacked’ base pairs (which are bracketed by two other base pairs within the same helix) from ‘helix-end’ pairs (which are adjacent to unpaired nucleotides or to a base pair from a different helix). This resembles the approach in (36), and ultimately is the chosen model.

As described further in the Results section, the appropriate granularity of the model was evaluated in two ways. First, we verified with MATLAB that each subdivision generates two new probability functions with statistically significant differences. Second, each model was used to simulate SHAPE data for the E. coli ribosomal 16S sequence by the method described next. The improvement in prediction accuracy for 1000 trials using values generated under the ternary model, but not the other two, was consistent with the experimental data (25).

Simulating SHAPE-directed predictions

Our study is based on a diverse set of 16S/18S ribosomal sequences with secondary structures available through the Comparative RNA Web (CRW) site (35). The 16 test sequences represent a variety of organisms over a wide range of lengths and MFE prediction accuracies. Additional details are provided in Supplementary Table S1.

For simplicity, we refer to the pseudoknot-free comparative structure from the CRW site as the native base pairings. For our purposes, any nucleotides involved in stabilizing interactions, such as pseudoknots or base triples, belonging to the tertiary or quaternary RNA structure were treated as unpaired. However, known non-canonical base pairs were classified in the same manner as the Watson–Crick and wobble base pairs.

Given a sequence and its native structure, each nucleotide was assigned to the appropriate category (distinguishing unpaired from paired, subdivided into stacked or helix-end) for the current model. The corresponding probability density functions were then used to generate a random value from the appropriate distribution for each nucleotide. This produced a single simulated SHAPE data set for the given sequence. Unless otherwise specified, 1000 trials were run for each sequence.

To minimize simulation run times, secondary structures were predicted using GTfold (37), a parallelized multi-core thermodynamic optimization program. Like UNAfold (10) and RNAfold (11), GTfold implements the standard Turner NNTM energy model (22,23). Like RNAstructure (34), GTfold provides integrated support for SHAPE-directed predictions. When comparing prediction results across programs, it is well understood (see for instance (37) or http://rna.urmc.rochester.edu/GUI/html/Introduction.html (26 October 2012, date last accessed) that small implementation differences can result in noticeable differences in predicted optimum structures.

Unless otherwise specified, Equation 1 as implemented in GTfold uses the default parameters ($\rho = 2.6$, $b = -0.5$). Default options were used, and there were no additional constraints on the thermodynamic optimization other than the simulated SHAPE data.

When the thermodynamic optimization is modified by soft constraints from simulated SHAPE data, we refer to the resulting optimal base pairings as the data-directed or simply directed secondary structure. When discussing the average data-directed prediction accuracy for 1000 trials, we will usually refer simply to the directed accuracy.

Prediction accuracy and structure similarity

The accuracy of predicted MFE and data-directed secondary structure predictions was determined against the native base pairings. As seen in Table 1, using the experimental…
SHAPE data as soft constraints improves the prediction significantly.

The numbers are lower than previously reported (25) because we used a simpler method for calculating accuracy. In this work, only a base pair \((i, j)\) occurring in both the native and a predicted structure was counted as a true positive \((TP)\). In particular, 'slipped' base pairs (22,38) are not considered correctly predicted. Base pairs in the predicted but not native structure were classified as false positives \((FP)\), whereas false negative \((FN)\) base pairs occur in the native but not predicted structure.

Predicted structures were scored for both PPV, the fraction of true positives in the predicted structure, and sensitivity, the fraction of true positives in the native structure. When comparing RNA secondary structures, the Matthews correlation coefficient can be approximated by the arithmetic mean of the PPV and sensitivity (6). Hence, the overall prediction accuracy was evaluated as the average of these two values:

\[
\text{accuracy} = \frac{1}{2} \left( \frac{TP}{TP+FP} + \frac{TP}{TP+FN} \right)
\]  

(2)

For comparison purposes, Table 1 also lists the accuracy for the same computation performed with RNAstructure (34). The highest accuracy (96.2% by our measure) reported earlier in (25) required expert curation of the RNAstructure prediction to account for factors such as local refolding. Without manual adjustment, we considered 70% to be a reasonable threshold for high prediction accuracy under our measure.

Finally, the accuracy measurement given above is symmetric in the choice of reference (native) and object (predicted) structure. Hence, we used the same symmetric measure when comparing two predicted structures, and we define the similarity of two secondary structures as the accuracy of one with reference to the other.

### RESULTS AND DISCUSSION

#### Choice of simulation model

The appropriate level of granularity for our stochastic model was determined by two criteria. To begin, we confirmed that each subdivision of the experimental data set is necessary to distinguish nucleotides with different SHAPE behavior. We then verified that the ternary model was sufficient to recapitulate the improvement in prediction accuracy for \(E. coli\) 16S rRNA using experimental SHAPE data.

Each subdivision of the experimental SHAPE data set yields probability distributions with statistically significant differences. The two-sample Kolmogorov–Smirnov test rejected the hypothesis that the paired and unpaired nucleotides had the same distribution \((P = 2.03 \times 10^{-199}, 5\%\) significance). Likewise, the hypothesis that the stacked and helix-end nucleotides had the same distribution was rejected \((P = 1.08 \times 10^{-40}, 5\%\) significance). This justifies distinguishing at least the unpaired, stacked and helix-end nucleotides with different distributions in any stochastic model of SHAPE data.

Next, we computed the directed prediction accuracies for \(E. coli\) 16S rRNA for 1000 simulations under each model. As shown in Table 1, the directed (i.e. the average data-directed prediction) accuracy decreased using the null model. Although the directed accuracy increased under the binary model, only 10% of the predictions were at least as good as the one using experimental data. However, when stacked and helix-end nucleotides are distinguished, the accuracy improvement with experimental data is no longer an outlier. Hence, the ternary model is sufficient to simulate SHAPE data, and all further results were produced using this model.

#### Data-directed predictions vary in accuracy

Using the stacked/helix-end/unpaired model, we investigated the effect of soft constraints on prediction accuracy for 1000 trials for each of our 16 test sequences. In general, the directed accuracy improved over the MFE prediction for each 16S/18S ribosomal RNA sequence; as seen in Figure 2, all boxes lie above the diagonal line. However, our results indicate that the high accuracy and significant improvement seen in the \(E. coli\) data-directed predictions, from 41.1% to 74.35% on average, is not always achieved.

By the 70% accuracy threshold, our test sequences group into three categories. When the MFE accuracy is moderate-to-high (over 50%, e.g. for \(Halofexa volcanii\)), the directed accuracy is consistently above 70%. When the MFE accuracy is particularly low (under 25%, e.g. for \(Encephalitozoon cuniculi\)), the directed accuracy is consistently well below 70%. In between (e.g. for \(E. coli\)), the behavior is variable, with four of the sequences performing well but three significantly less so.

These results indicate a rough correlation between MFE and directed accuracy. However, they also demonstrate that the 'directability' of the NNTM optimization, like MFE predictions, has some critical sequence dependencies. Both points are addressed in more detail in subsequent sections.

In terms of improving thermodynamic predictions through soft constraints, all but one of the middle group of seven sequences exhibited significant gains over their MFE accuracy. In contrast, the six sequences with moderate-to-high MFE accuracy had average

---

**Table 1. Data-directed prediction accuracy for \(E. coli\) 16S rRNA**

<table>
<thead>
<tr>
<th>Prediction details</th>
<th>Accuracy, %</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undirected MFE, GTfold</td>
<td>41.1</td>
<td>–</td>
</tr>
<tr>
<td>Undirected MFE, RNAstructure</td>
<td>34.6</td>
<td>–</td>
</tr>
<tr>
<td>Experimental SHAPE data, GTfold</td>
<td>76.4</td>
<td>–</td>
</tr>
<tr>
<td>Experimental SHAPE data, RNAstructure</td>
<td>72.8</td>
<td>–</td>
</tr>
<tr>
<td>Simulated data, no division</td>
<td>20.94 (mean) 0.00</td>
<td></td>
</tr>
<tr>
<td>Simulated data, paired/unpaired</td>
<td>69.03 (mean) 0.10</td>
<td></td>
</tr>
<tr>
<td>Simulated data, stacked/helix-end/unpaired</td>
<td>74.35 (mean) 0.37</td>
<td></td>
</tr>
</tbody>
</table>

Data-directed predictions were computed with GTfold; accuracy is averaged for 1000 trials for each model. Proportion (above 76.4 threshold) is the fraction of simulated data with prediction accuracy at least as good as the GTfold SHAPE-directed one.
improvements well below ±30%. This suggests that attaining high accuracy requires additional information and/or expert curation, even for sequences with good data-directed predictions.

Finally, all sequences including *E. coli* exhibited a range of accuracies, with the amount of variation broadly but inversely correlated with the average. This raises some questions for further study. On the one hand, SHAPE data have been reported to be 'highly reproducible' across several independent replicants (29), with standard deviations on the order of 0.1 normalized SHAPE unit or less. On the other, MFE predictions can be sensitive even to small perturbations in the thermodynamic parameters (17,18). Hence, further analysis is needed to reconcile these results.

Taken together, our simulation results address the extent of improvements in SHAPE-directed prediction accuracy. Foremost, 3 of 16 test sequences (*E. cuniculi*, *Caenorhabditis elegans*, *Mus musculus*) have a directed accuracy well below 60% and an average improvement over MFE below 15%. From this, we conclude that directed predictions are a complex interplay between RNA sequence, auxiliary data and thermodynamic optimization which do not always result in high accuracy or large improvements. Although this conclusion remains to be confirmed by experiment, it is in full agreement with other results investigating the limitations of SHAPE data (30).

### Data-directed predictions preserve accurate MFE pairs

Having demonstrated variability in the extent of accuracy improvements, we investigate their nature, namely the rough correlation observed in Figure 2. Our hypothesis was that MFE base pairs which are accurately predicted tend to remain optimal in a directed prediction whereas incorrect ones do not.

Indeed, Figure 3 shows a surprisingly strong correlation between the similarity of each directed prediction to the MFE and the MFE accuracy. From this, we conclude that if the former is especially low or high, then the latter is likely to fall outside the typical range (30–60%). In these cases, our previous results indicate that the data-directed prediction accuracy is likely to also be correspondingly low or high.

This is highlighted in Figure 4c, for *E. cuniculi* and *H. volcanii*, the test sequences with the lowest and highest MFE accuracies (below 20% and above 70%), respectively. As expected, the directed prediction accuracy for these sequences is very low/high, as seen by the predominantly blue/red native base pairings annotated with directed frequency in Figure 4c.

Further analysis at the base pair level for these sequences and for *E. coli* confirmed our initial hypothesis, and provides a means of identifying sets of base pairs with high PPV. We found that the majority of MFE base pairs exhibited one of two behaviors: either occurring in nearly all directed predictions or in close to none of them. Moreover, the former were generally correctly predicted, whereas the latter were not.

As seen in Figure 4b, almost all MFE base pairs are colored either orange/red or aqua/blue, according to their frequency in the directed structures. Those occurring...
at high frequency are correctly predicted in the MFE structure [red in column (b) correlates with red in column (a)]. Conversely, MFE pairs occurring at low frequency in directed structures tend to be incorrectly predicted [blue in column (b) correlates with blue in column (a)]. Similar patterns were observed for all 16 test sequences. Hence, the accuracy of directed structures is correlated with the accuracy of the undirected MFE prediction.

This correlation is clearly not perfect, however, as indicated by the high frequency with which *E. coli* native base pairs occur in the directed structures [red in column (c) middle]. However, it does suggest using auxiliary data to identify MFE pairings with high PPV—independent of the data-directed prediction accuracy.

For our test sequences, the PPV of the MFE structure $M$ has a broad range from 0.165 to 0.726 with a median of 0.3785. However, these values improve dramatically for base pairs, which are common to both $M$ and a single data-directed structure $D$.

For each test sequence, we computed the PPV of the subset $M \cap D$, that is the fraction of true positives, for each of the 1000 directed structures. As given in Table 2, these values are high overall, with averages ranging from 0.532 to 0.909 with a median of 0.871. Likewise, the average PPV of the remaining MFE base pairs $M \setminus D$ is low, ranging from 0.245 down to 0.026 with a median of 0.079. Finally, these values are remarkably stable for 1000 trials. Hence, MFE base pairs that are preserved in a directed structure are significantly more likely to be accurately predicted than those that are not preserved. Thus, any base pairs common to both the MFE and a SHAPE-directed structure should have high PPV.

Another method for improving confidence in thermodynamic predictions uses base pair probabilities computed from the partition function (39) or by stochastic sampling (40). It is known (38) that high probability MFE base pairs also have a significantly increased PPV. We confirmed that base pairs in $M \cap D$ are not simply the high probability ones. This was true particularly for sequences with low MFE accuracy, when a SHAPE-directed prediction may also be less accurate (See Supplementary Table S2 for details).

**Directability of NNTM optimization**

The purpose of soft constraints is to direct the optimization towards a more accurate solution. However, like the undirected MFE prediction, the ability of auxiliary data to improve thermodynamic prediction accuracy has sequence dependencies. We illustrate the differences in NNTM directability by parameterizing the slope $m$ and intercept $b$ in Equation 1 against three test sequences: *E. canis*, *E. coli* and *H. volcanii*.

The default parameters ($m = 2.6$, $b = -0.8$) were chosen by identifying a ‘sweet spot’ maximizing both sensitivity and PPV of *E. coli* 23S rRNA using experimental SHAPE data (25). Under the same procedure using a random simulated data set, Figure 5 shows the parameterization space for each of our three 16S sequences.

Our results for *E. coli* 16S recapitulate the 23S experimental ones. Namely, there is a large optimal region with a maximum PPV of 86.2% when $m = 3.2$, $b = -0.4$ and a maximum sensitivity of 78.8% when $m = 1.2$, $b = -0.6$. The average of these values ($m = 2.2$, $b = -0.5$) is close to the default parameters.
The situation with *E. cuniculi* is markedly different. The optimal region is small, and the maximum obtainable PPV is 58% and sensitivity is 47.5%, when \( m = 6.2, b = -1.0 \) and \( m = 4.2, b = -1.0 \), respectively. Repeating the process with other randomly chosen data sets did not qualitatively change the outcome (data not shown). We conclude that no choice of Equation 1 parameters would improve the *E. cuniculi* directed predictions to the level of *E. coli*.

In contrast, the optimal region for *H. volcanii* is larger than *E. coli* and contains more high sensitivity/PPV combinations of parameters. What is especially striking is the gradual degradation in accuracy as the parameters are varied away from optimal. Hence, the NNTM is a good model for the base pairing of *H. volcanii*, a reasonable one for *E. coli*, as it can be directed to high accuracy predictions using SHAPE data, and a poor one for *E. cuniculi*. In view of this unsatisfactory situation with *E. cuniculi*, we further explored its directability. As a conceptual exercise, we increased the separation between the unpaired and two paired probability distributions in our ternary model. (Recall that all three distributions have a peak at low values.) The original unpaired distribution \( P_{	ext{orig}} \) was modified to be a convex combination with a normal distribution \( P_{\text{norm}} \) of higher mean;

\[
P_w(x) = wP_{\text{norm}}(x) + (1 - w)P_{\text{orig}}(x)
\]

where \( 0 \leq w \leq 1 \) and \( P_w(x) \) denotes the new probability of SHAPE value \( x \). The normal distribution used had a mean of 3.51 and standard deviation of 1.78 obtained by

---

**Figure 4.** Circular arcplots of *E. cuniculi*, *E. coli* and *H. volcanii* 16S secondary structures. Sequence is drawn as a circle, and each arc denotes a base pair. Column (a) shows an overlay of MFE (blue) and native (green) structures with common base pairs in red. Column (b) shows the MFE structure with base pairs annotated by the fraction of data-directed structures also containing that pair, as indicated by the color bar at the bottom. Column (c) shows the native structure likewise annotated.
hypothesizing a 6-fold increase in reagent reactivity for a Gaussian model of SHAPE chemistry kinetics (41). Under this modified model, as the normal component of the unpaired distribution increased to 80%, the directed prediction accuracy for *E. cuniculi* increased above 70% (see Supplementary Figure S1). In fact, when the normal component is 100% of the unpaired distribution, then 15 of 16 of our chosen sequences have a directed prediction accuracy at least this high (see Supplementary Table S3). These ‘proof-of-concept’ results indicate that, even for sequences like *E. cuniculi* whose base pairing are not modeled well by the NNTM, auxiliary information has significant potential for improving prediction accuracy. However, a critical factor in directing the thermodynamic optimization towards the native base pairings may be the strength of the ‘unpaired’ signal.

### Conclusions and future directions

We introduced a stochastic model for experimental SHAPE data, and evaluated data-directed RNA secondary structure prediction accuracy for a diverse set of 16S/18S ribosomal sequences. Using this auxiliary data as soft constraints consistently improved thermodynamic optimization accuracy. However, there was significant variation in the average data-directed prediction accuracy between sequences, correlated with the undirected (MFE) accuracy. Thus, although many of our test sequences achieved the high accuracy reported for experimental SHAPE-directed predictions, this level of accuracy was by no means universally attained.

When accuracy cannot be evaluated by direct comparison with a known structure, our results still yield helpful insights. In particular, the similarity between the undirected and a data-directed prediction is highly correlated with the MFE accuracy, which in turn is roughly correlated with the data-directed prediction accuracy. Hence, if the similarity is particularly low (below 30%), it is likely that the directed prediction accuracy is correspondingly low—as illustrated by the *E. cuniculi* example.

<table>
<thead>
<tr>
<th>Organism</th>
<th>PPV (M)</th>
<th>PPV (M \cap D)</th>
<th>PPV (M \setminus D)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>E. cuniculi</em></td>
<td>0.165</td>
<td>0.532 ± 0.143</td>
<td>0.065 ± 0.026</td>
</tr>
<tr>
<td><em>Vairimorpha necatrix</em></td>
<td>0.177</td>
<td>0.648 ± 0.100</td>
<td>0.026 ± 0.016</td>
</tr>
<tr>
<td><em>C. elegans</em></td>
<td>0.186</td>
<td>0.699 ± 0.152</td>
<td>0.059 ± 0.039</td>
</tr>
<tr>
<td><em>Emferella nidulans</em></td>
<td>0.267</td>
<td>0.770 ± 0.076</td>
<td>0.062 ± 0.026</td>
</tr>
<tr>
<td><em>Nostoc.Tabacum</em></td>
<td>0.303</td>
<td>0.862 ± 0.071</td>
<td>0.063 ± 0.029</td>
</tr>
<tr>
<td><em>Cryptomonas.sp</em></td>
<td>0.333</td>
<td>0.909 ± 0.022</td>
<td>0.044 ± 0.020</td>
</tr>
<tr>
<td><em>M. musculus</em></td>
<td>0.381</td>
<td>0.751 ± 0.080</td>
<td>0.136 ± 0.051</td>
</tr>
<tr>
<td><em>Mycoplasma gallisepticum</em></td>
<td>0.377</td>
<td>0.872 ± 0.037</td>
<td>0.077 ± 0.030</td>
</tr>
<tr>
<td><em>Synchococcus.sp</em></td>
<td>0.380</td>
<td>0.887 ± 0.046</td>
<td>0.081 ± 0.033</td>
</tr>
<tr>
<td><em>E. coli</em></td>
<td>0.399</td>
<td>0.875 ± 0.038</td>
<td>0.053 ± 0.028</td>
</tr>
<tr>
<td><em>Bacillus.subtilis</em></td>
<td>0.500</td>
<td>0.880 ± 0.039</td>
<td>0.102 ± 0.036</td>
</tr>
<tr>
<td><em>Desulfovibrio.desulfuricans</em></td>
<td>0.517</td>
<td>0.885 ± 0.034</td>
<td>0.136 ± 0.050</td>
</tr>
<tr>
<td><em>Chlamydomonas reinhardtii</em></td>
<td>0.526</td>
<td>0.874 ± 0.032</td>
<td>0.133 ± 0.051</td>
</tr>
<tr>
<td><em>Thermotoga.maritima</em></td>
<td>0.541</td>
<td>0.866 ± 0.034</td>
<td>0.125 ± 0.044</td>
</tr>
<tr>
<td><em>Thermoproteus tenax</em></td>
<td>0.589</td>
<td>0.870 ± 0.029</td>
<td>0.150 ± 0.053</td>
</tr>
<tr>
<td><em>H. volcanii</em></td>
<td>0.726</td>
<td>0.903 ± 0.017</td>
<td>0.245 ± 0.064</td>
</tr>
</tbody>
</table>

PPV(M) is the number of native base pairs in the MFE structure M. For each directed prediction D, the PPV of the set of common base pairs (M \cap D) and the set of remaining MFE base pairs (M \setminus D) was computed. Values are the mean for 1000 trials ± standard deviation.

Figure 5. Variation in PPV and sensitivity as a function of the Equation 1 parameters for a random simulated SHAPE data set for (a) *E. cuniculi*, (b) *E. coli* and (c) *H. volcanii* 16S RNA. The color bar indicates percentage measurements for PPV and sensitivity.
Even in these cases, though, the PPV of base pairs common to both the MFE and a directed prediction should be much higher.

The lack of directability for the E. coli predictions suggests several potential directions for further investigation. For instance, no systematic study of more than three sequences has yet been done on the effects of varying the slope and intercept parameters for the current implementation of soft constraints. It would be interesting to analyse more completely this variable aspect of thermodynamic optimization and to characterize its sequence dependencies. Furthermore, although the current method of soft constraints works well in many circumstances, it is not the only one (27). Hence, it is possible that alternative methods of incorporating SHAPE data into secondary structure predictions may address this issue in the future.

Finally, these results demonstrate the importance of statistically reproducible results in SHAPE-directed secondary structure predictions. An alternative method for simulating SHAPE reactivities, particularly when a known secondary structure is not available, would use a structural alignment to map nucleotides back to a related sequence with an experimental data set. In a small, exploratory investigation of this approach, we found that it generated similar results to the stochastic model presented here, and hence should be investigated further in the future.

AVAILABILITY
Further information including the computational tools developed for this study is available online via the website http://users-birc.au.dk/zs/SHAPEsimulations/.

SUPPLEMENTARY DATA
Supplementary Data are available at NAR Online: Supplementary Tables 1–3 and Supplementary Figure 1.

ACKNOWLEDGEMENTS
The authors are indebted to K. Weeks for making available the 16S and 23S ribosomal RNA of E. coli (21). We are grateful to J. Gossett and S. Harvey for their helpful discussions, as well as J. Gossett and S. Harvey for their thoughtful feedback.

FUNDING
The National Institutes of Health [NIGMS R01 GM083621 to C.E.H.]; Danish Council for Strategic Research [06-016856 to J.K.]; Burroughs Wellcome GM083621 to C.E.H.; Danish Council for Strategic Research [09-061856 to J.K.]; Burroughs Wellcome GM083621 to C.E.H.; and the National Institutes of Health [NIGMS R01 GM083621 to C.E.H.].

Conflict of interest statement. None declared.

REFERENCES


Characterizing RNA secondary structure space using information entropy

5.1 Problem statement

Comparative methods for RNA secondary structure prediction use evolutionary information from RNA alignments to increase prediction accuracy. The model is often described in terms of stochastic context-free grammars (SCFGs), which generate a probability distribution over secondary structures. This probability distribution potentially holds a lot of useful information, which is not reported to the end-user.

A useful quantity to describe the “spread” of a probability space is the information entropy, which for a probability distribution \( P \) with a set of events \( \mathcal{X} \) is defined as:

\[
H(P) = - \sum_{x \in \mathcal{X}} P(x) \log_2(P(x)).
\] (5.1)

The information entropy has well-defined lower and upper bounds. The minimum entropy of 0 occurs when there is only one outcome with probability 1. For \( n \) possible outcomes, the maximum entropy is equal to \( \log_2(n) \) and occurs for 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.

In this paper, our purpose was to compute the information entropy of the probability space generated for an alignment by the model in \textit{PPfold}. This could then be used to analyze how the probability distribution changes as a function of the input alignment.
5.2 Results

The computation is based on previous work by Nederhof and Satta [5.1], in which the derivational entropy of an SCFG in the general case is expressed in terms of rule frequencies. Our situation is somewhat different. Firstly, in our case the derivational space is restricted to the set of derivations where the input alignment is emitted. Secondly, a phylo-grammar is a combined model, where the entropy of the combined distribution needs to be calculated, not just the entropy of the SCFG part.

A brief outline of the approach follows. If the combined distribution is expressed as a product of two independent terms in RNA secondary structure grammars in double-emission normal form (Equations 9, 10 and 11), it is possible to split the information entropy for the alignment into three parts (Equation 16). The three parts are then evaluated separately.

The first part is a constant, and uses the total probability of the alignment as evaluated in the inside algorithm. The second part is the information entropy arising due to the SCFG part of the model alone, which can be evaluated in the approach described by Nederhof and Satta [5.1] using the inside-outside algorithm. The third part corresponds to the phylogenetic part of the combined model, and in the paper we show that also this part can be evaluated using the evolutionary likelihoods and the inside-outside variables (Equation 18).

The model was implemented and evaluated in PPfold. In the paper, we also outline several possible applications of this quantity, and explore correlations between entropy, basepair probabilities and prediction accuracy.

5.3 Notes

It is worth taking a moment to appreciate Equations 7 and 8, as first presented by Nederhof and Satta [5.1]. It is indeed quite remarkable that a sum over derivations (of which there are many) can, in fact, be turned into a sum over rules of the grammar (of which there are few). This result lies at the heart of the computation, and allows us to determine the information entropy using the inside-outside algorithm.

Another interesting detail is that the information entropy can actually be computed in $O(n^2)$ time for the KH grammar (where $n$ is the length of the alignment), once the inside-outside variables are known. This is because in the KH grammar, there is a maximum of one bifurcating rule originating from each non-terminal symbol, and the expected frequencies of rules from the same nonterminal symbol sum to the expected frequency of the nonterminal symbol. If the grammar had a nonterminal symbol bifurcating in more than one way, the computation would only be possible in $O(n^3)$ time.

As discussed in the paper, the information entropy delivers complementary information to basepairing probabilities. Basepairing probabilities depend both on the structures and their probabilities, but the entropy is only a function of probabilities, and does not depend on the similarity of the structures to each other. If the basepairing probabilities are low, the entropy can help reveal the underlying reasons.
Chapter 5. Characterizing RNA secondary structure space using information entropy

If there is insufficient structure signal, so there are no structures of high probabilities and the probability distribution is “spread”, entropy will be high. If there are two or more possible (topologically different) structures of high probabilities, so the probability distribution has several “peaks”, entropy will be low.

The concept of information entropy over a set of derivations for an alignment was later further generalized to evaluate the information entropy over an alignment space as well (Appendix A and [5.2]).

An important possible application not discussed in the paper is that the same computational method can be used to evaluate the effect of RNA structure probing data in a probabilistic framework for data-driven secondary structure predictions, such as what is implemented in PPFold 3.0 (Chapter 3).

There are many directions in which this work could be taken further. It would be very interesting to compute other information theoretic quantities to characterize the probability space over secondary structures. The Kullback-Leibner divergence (see Section 4.3 in this thesis), for example, could be computed to measure the difference between the “natural” probability distribution without probing data, and the distribution obtained on the addition of probing data. Furthermore, as a future project it would be interesting to see if the fraction of structures in certain probability ranges could be computed. The algorithms presented by Newberg and Lawrence [5.3] could be useful in this regard.

It would also be possible to implement a probabilistic sampling algorithm also in the case of SCFG-based secondary structure prediction, similar to what has been done by Ding and Lawrence [5.4] in the thermodynamic case. Lastly, it would be very interesting to investigate if errors in model parameters could be propagated throughout the prediction algorithm, and used to provide an even more relevant measure of prediction uncertainty.

5.4 Bibliography


Characterising RNA secondary structure space using information entropy

Zsuzsanna Sükösd1,2,3*, Bjarne Knudsen4, James WJ Anderson5, Ádám Novák5,6, Jørgen Kjems2,3, Christian NS Pedersen1,7

From The Eleventh Asia Pacific Bioinformatics Conference (APBC 2013)
Vancouver, Canada. 21-24 January 2013

Abstract

Comparative methods for RNA secondary structure prediction use evolutionary information from RNA alignments to increase prediction accuracy. The model is often described in terms of stochastic context-free grammars (SCFGs), which generate a probability distribution over secondary structures. It is, however, unclear how this probability distribution changes as a function of the input alignment. As prediction programs typically only return a single secondary structure, better characterisation of the underlying probability space of RNA secondary structures is of great interest. In this work, we show how to efficiently compute the information entropy of the probability distribution over RNA secondary structures produced for RNA alignments by a phylo-SCFG, and implement it for the PPfold model. We also discuss interpretations and applications of this quantity, including how it can clarify reasons for low prediction reliability scores. PPfold and its source code are available from http://birc.au.dk/software/ppfold/.

Background

The function of RNA molecules is known to depend on their three-dimensional structure, which is stabilized by a secondary structure scaffold of basepairing. The secondary structure is defined by hydrogen bonds between nucleotides, which form across the structure for thermodynamic stability and molecular function. Despite its importance, the accurate prediction of RNA secondary structure remains an unsolved challenge in computational biology.

With the advent of next-generation sequencing technologies and new methods in transcriptomics, an explosively growing amount of biological RNA data is available in public databases such as Rfam [1] and RNA STRAND [2]. This makes it possible to acquire a large number of RNA alignments to be used in comparative RNA secondary structure predictions. This is especially significant in the case of long RNAs such as RNA viral genomes and long genomic introns, many of which are known to have functional, conserved secondary structures.

Several methods have been established to predict RNA secondary structures from nucleotide sequences. In this paper, we focus entirely on non-pseudoknotted secondary structure prediction. Thermodynamic optimisation based on minimising free-energy functions has been used to great effect in algorithms such as mfold [3], UNAFold [4] and RNAfold [5]. In a different approach, stochastic context-free grammars (SCFGs) have also been successfully used to model RNA secondary structure. The Pfold program [6,7], for example, combines molecular evolution with a lightweight SCFG model (known as a phylo-grammar) to predict the consensus secondary structure of RNA alignments, and has in the past shown to be highly accurate for structural alignments [8]. PPfold is a recent multi-threaded reimplemention of Pfold [9].

Common to these methods is that they produce a probability distribution over all possible nested secondary structures for the input sequences, but usually only a single, optimal secondary structure is reported to the user. A particularly interesting question is how the underlying distribution changes as a function of input data. Due to the large space of possible secondary structures, however, it is difficult to report useful quantities to describe this. Information entropy is one such measure.

* Correspondence: zs@birc.au.dk
Entropy computations in the context of RNA secondary structure prediction have been considered previously from a thermodynamic perspective, calculating the thermodynamic entropy over both secondary structure space \([10,11]\) and tertiary structure space \([12]\). Positional thermodynamic entropy \([5]\) as well as thermodynamic entropy changes in response to basepair mutations \([13]\) have also been computed. Additionally, SCFG-based methods have recently been utilised for calculating both the information entropy of individual basepairs in a single-sequence context \([14]\), and the thermodynamic entropy changes in response to basepair mutations. However, no form of entropy has been computed previously in the case of comparative RNA secondary structure prediction.

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

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

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. For \(n\) possible outcomes, the maximum entropy is equal to \(\log_2(n)\) and occurs for the uniform distribution. 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.

In the case of RNA secondary structure prediction based on a semantically unambiguous SCFG, the information entropy of the probability distribution over RNA secondary structures can be computed as the derivational entropy of the SCFG that generates the distribution. We restrict ourselves to semantically unambiguous SCFGs, in order to maintain a one-to-one correspondence between SCFG derivations and secondary structures. Thus, throughout this paper we use "information entropy" and "derivational entropy" interchangeably.

**Notation**

Consider RNA alignments of \(k\) sequences \((k \geq 1)\), with the \(i\)th column denoted \(c_i \in \Sigma^k = \{\text{A}, \text{C}, \text{G}, \text{U}, \text{.}\}^k \setminus \{\text{.}\}^k\). A stochastic context-free phylo-grammar (phylo-SCFG) on such alignments is a tuple \(G = (\Sigma^k, \mathcal{N}, \mathcal{S}, \mathcal{R}, P)\), where:

- \(\Sigma^k\) forms the (finite) set of terminal symbols
- \(\mathcal{N}\) is a finite set of nonterminal symbols, such that \(\mathcal{N}^* \cap \mathcal{N} = \emptyset\)
- \(\mathcal{S}\) is the start symbol, \(\mathcal{S} \in \mathcal{N}\)
- \(\mathcal{R}\) is a finite set of production rules, each rule of the form \(A \to \alpha, A \in \mathcal{N}\) and \(\alpha \in (\Sigma^k \cup \mathcal{N})^*\)
- \(P\) is a function from \(\mathcal{R}\) to real numbers in the interval \([0,1]\)

In the case of a phylo-grammar, \(P\) can be interpreted as Bayesian probabilities equal to the product of prior probabilities that only depend on the type of rule being used, and a likelihood factor that is typically derived from a phylogenetic model and is a function of the alignment columns. We will return to this more formally later.

Furthermore, we assume the grammar is proper, that is \(\forall \alpha \in \mathcal{N}, \Sigma^k \cup \mathcal{N} \cup \{\text{.}\}, P(\alpha) = 1\).

Let \(d\) be a complete (left-most) derivation of the grammar. Informally, a complete derivation is a sequence of production rules, such that starting from the start symbol, and sequentially replacing all nonterminals with a production rule emitting from that nonterminal, a string of terminal symbols is obtained. The probability of \(d\) is the product of the probability of all rules occurring in \(d\):

\[
p(d) = \prod_{\alpha \to \alpha} P(A \to \alpha)^{f_d(\alpha \to \alpha)} \tag{2}
\]

where \(f_d(\alpha \to \alpha)\) is the number of times rule \(A \to \alpha\) occurs in derivation \(d\).

The grammar is consistent if \(\Sigma_d p(d) = 1\), where the sum is over all possible derivations of the grammar. In the case of phylo-grammars, consistency implies that the total probability of the grammar emitting all alignments of \(k\) sequences (of all lengths) is 1.

The expected frequency (count) of a rule \(A \to \alpha\) in all derivations of the grammar is:

\[
Ef(A \to \alpha) = \sum_{d} p(d) f_d(A \to \alpha). \tag{3}
\]

The expected frequency of each rule can be computed in practice using a dynamic programming algorithm known as the inside-outside algorithm, as described in \([15]\). Following the approach of \([16]\), we factorise a complete derivation \(d\) at each occurrence of rule \(A \to \alpha\) into an "innermost" sub-derivation \(d_1 : a \xrightarrow{d_1} \xi\) where \(s \in (\Sigma^k)^*\), and two "outermost" sub-derivations \(d_2, d_3 : S \xrightarrow{d_2} \beta, \gamma \xrightarrow{d_3} u\), where \(\beta, \gamma \in (\Sigma^k \cup \mathcal{N})^*\) and \(s, u \in (\Sigma^k)^*\). Then:

\[
Ef(A \to \alpha) = \mathcal{O}(\alpha) h(\alpha) P(A \to \alpha). \tag{4}
\]

with:

\[
h(\alpha) = \sum_{d_1} p(d_1) \tag{5}
\]
The derivational entropy of an SCFG is the information entropy of the probability distribution of all derivations under the SCFG (c.f. equation 1):

\[ H(G) = - \sum \log(p(d)) \tag{7} \]

where the sum is over all possible derivations of the grammar. This quantity can be computed efficiently using expected rule frequencies [16]:

\[ H(G) = - \sum \log(p(A \rightarrow a))E(A \rightarrow a) \tag{8} \]

We assume that the phylo-SCFG describes RNA secondary structure, and while it may be syntactically ambiguous, it is semantically unambiguous. In practice, this means that there may be a number of possible derivations for a particular alignment, but there is a one-to-one correspondence between derivations and consensus secondary structures for the RNA alignment. Furthermore, we express RNA structure SCFGs in double emission normal form [17], allowing only rules of the following types:

Type 1: \( A \rightarrow \epsilon \)

Type 2: \( A \rightarrow \epsilon B \)

Type 3: \( A \rightarrow BC \)

for \( A, B, C \in \Sigma, \epsilon, \eta \in \Sigma^* \). Apart from generating empty strings, all SCFGs modelling nested RNA secondary structures can be written in double emission normal form, so the methods presented here can be adapted to RNA secondary structure grammars of all types. Additionally, they can also be adapted for specific mildly context-sensitive RNA grammars that generate specific types of pseudoknots.

Type 1 rules correspond to the production of a single column of the alignment, and their probability can be expressed as

\[ P(A \rightarrow c) = p_\epsilon(A \rightarrow \epsilon)P_\epsilon(c|\epsilon) \tag{9} \]

where \( P_\epsilon(A \rightarrow \epsilon) \) only depends on \( A \), and \( P_\epsilon(c|\epsilon) \) is the likelihood of observing column \( \epsilon \) under the phylogenetic model, assuming that it is unpaired in the consensus structure (denoted by \( c \)).

Type 2 rules correspond to the production of two basepaired columns of the alignment, and their probability can be expressed as

\[ P(A \rightarrow \epsilon B) = P_\epsilon(A \rightarrow \epsilon)P_\epsilon(B|\epsilon) \tag{10} \]

where \( P_\epsilon(A \rightarrow \epsilon) \) only depends on \( A \) and \( B \), and \( P_\epsilon(B|\epsilon) \) is the likelihood of observing column pair \( \epsilon, \eta \) under the phylogenetic model, assuming that they are paired with each other in the consensus structure (denoted by \( c|\eta \)).

Type 3 rules express bifurcation and correspond to dividing the alignment into two parts. As these rules do not depend on alignment columns, we have:

\[ P(A \rightarrow BC) = P_\epsilon(A \rightarrow B) \tag{11} \]

where \( P_\epsilon(A \rightarrow B) \) only depends on \( A, B \) and \( C \).

It is now clear that the probability of any particular derivation under a phylo-SCFG this structure can be expressed as a product of two probabilities: a probability \( p_\Phi \) that only depends on the types of rules used, and a probability \( p_{\Phi|D} \) that only depends on the emitted alignment columns: \( p(d) = p_{\Phi}(d)p_{\Phi|D}(d) \).

The computation of \( T \) is straightforward using the inside algorithm. The normalized probability of a derivation \( d \) is \( p_\Phi(d) = 1/T(d) = \frac{1}{\sum d p_\Phi(d)p_{\Phi|D}(d)} \). We now define the information entropy of the input alignment under the phylo-SCFG model as:

\[ H_\Phi(G) = - \sum d \Phi(p_\Phi(d)) \log_2(p_\Phi(d)). \tag{15} \]
Note that Equations 4 and 8 still hold when the set of derivations is restricted to a subset, as opposed to the entire space, so we can write the entropy as:

\[ H_\Phi(G) = \sum_{d \in \Phi} p(d) \log_2 \left( \frac{\sum_{t \in T} p_t(d) p_r(t)}{p(d)} \right) \]

\[ = \log_2(T) - \frac{1}{T} \sum_{d \in \Phi} p(d) \log_2(p_t(d)) \]

\[ - \frac{1}{T} \sum_{d \in \Phi} p(d) \log_2(p_r(d)) \]

(16)

We now show how to express the entropy in terms of expected rule frequencies. Note that:

\[ \sum_{s \in \Phi} p(d) \log_2(p_t(d)) = \sum_{\lambda \rightarrow \alpha} \log_2(p_t(\lambda \rightarrow \alpha)) \mathbf{E}(\lambda \rightarrow \alpha) \]

which can be computed using the expected rule frequencies obtained from the inside-outside algorithm (cf. equations 7 and 8). This can be seen by noting that Equations 4 and 7 still hold when the set of derivations is restricted to a subset, as opposed to the entire space.

Furthermore, if \( \mathbf{1}_{d}(i, j) \) denotes the indicator function for whether the column pair \((i,j)\) is emitted from a Type 2 rule (i.e. position \(i\) and \(j\) form a pair), and \( \mathbf{1}'(i) \) denotes the indicator function for whether column \(i\) is emitted from a Type 1 rule (i.e. position \(i\) is unpaired), then:

\[ \sum_{d \in \Phi} p(d) \log_2(p_r(d)) = \]

\[ = \sum_{s \in \Phi} p(d) \left( \sum_{\lambda} \log_2\left( p_r(\lambda) \mathbf{1}_{d}(i, j) \right) \right) \]

\[ + \sum_{\lambda} \log_2\left( p_r(\lambda) \mathbf{1}'(i) \sum_{d \in \Phi} p(d) \right) \]

\[ = \sum_{i,j} \log_2\left( p_r(\lambda, j, i) \right) \sum_{d \in \Phi} p(d) \]

\[ + \sum_{i} \log_2\left( p_r(\lambda, i) \right) \sum_{d \in \Phi} p(d) \]

(18)

We observe that \( \sum_{d \in \Phi} \mathbf{1}'(i) p(d) \) is just the total probability under the model that position \(i\) is unpaired (i.e. it is emitted from rule type 1). The quantity \( \sum_{d \in \Phi} p(d) \log_2(p_r(d)) \) can therefore also be computed using the expected rule frequencies.

Once the values of the inside-outside variables have been calculated for an input string of length \(n\), the expected rule frequencies can be computed in \(O(n^2)\) time for rules of Type 1, \(O(n^3)\) time for rules of Type 2, and \(O(n^4)\) time for rules of Type 3. As the time complexity of the inside-outside algorithm is also \(O(n^4)\), the computation of the entropy over the possible derivations of the input string does not increase the time complexity of RNA secondary structure prediction.

**Interpretation of the derivational entropy**

The derivational entropy provides a measure for the "spread" of the probability distribution on possible secondary structures. For equiprobable events, information entropy increases logarithmically with the number of possible outcomes. It is clear, therefore, that the maximum derivational entropy increases with sequence length. It has been shown [18] that, assuming all nucleotide pairings are possible, the number of secondary structures \(S(l)\) of length \(l\) can be approximated for large \(l\) as:

\[ S_l \approx 1.104 \times l^{3/2} \times 2.618^l \]

(19)

The maximum derivational entropy is therefore expected to increase logarithmically with \(S_l\).

\[ H_{\text{max}}(l) \approx 0.142 \times 3 \log_2(l) + 1.388l \]

(20)

\( H_{\text{max}}(l) \) provides a upper bound on the value of the information entropy for an RNA alignment of length \(l\), and can aid the user in the interpretation of the entropy corresponding to a particular input alignment.

In practice, however, \(H_{\text{max}}\) is rarely attained by nucleotide sequences. To obtain more intuition for the value of the derivational entropy, we generated random nucleotide sequences of varying lengths and nucleotide compositions, and computed the entropy of the probability distribution generated by PPfold for the single-sequence predictions, as a percentage of the theoretical \(H_{\text{max}}(l)\) (Figure 1). Interestingly, we found that the entropies computed for these random sequences are remarkably stable at around 25-35% of \(H_{\text{max}}(l)\) over a wide range of sequence lengths (> 30 nucleotides), with only a slight dependence on nucleotide composition. Particular entropy values can therefore be interpreted in relation to this entropy value of 25-35% of \(H_{\text{max}}(l)\) suggests that the probability distribution for the input data is as "spread" over the structure space as it would be for a single random sequence of that length.

**Using information entropy to interpret low reliability scores**

Derivational entropy is related to various reliability measures already reported by prediction programs. PPfold in
particular computes the probability of a pair between two columns as the sum of the (normalized) expected frequencies of rules of Type 2 emitting that column pair:

\[ P(\hat{i}, \hat{j}) = \sum_{N_1 \rightarrow cN_2c'} O(N_1)P_i(N_1 \rightarrow cN_2c')P_j(i, \hat{j}). \] (21)

The probability of column \( i \) remaining unpaired is

\[ P(i) = 1 - \sum_{i \neq j} P(\hat{i}, \hat{j}). \]

These probabilities function as "reliability scores" for every predicted base-pair or unpaired nucleotide in the structure. The overall reliability score for the secondary structure can be computed as the average of the reliability scores of all positions.

Importantly, while the reliability scores depend both on the structures and their probabilities, the derivational entropy is only a function of probabilities, and does not depend on the similarity of the structures to each other. Derivational entropy therefore provides complementary information to reliability scores. For example, if the reliability scores predict a low accuracy, the entropy can help reveal the underlying reasons. A low reliability score can be observed in different situations, for example (a) if there is insufficient structure signal, so there are no structures of high probabilities and the probability distribution is "spread", or (b) if there are two or more possible (topologically different) structures of high probabilities, so the probability distribution has several "peaks". Entropy will be high in the first case, but low in the second case, and can therefore be used to distinguish the two situations from each other.

To illustrate this with a practical example, a PP-fold prediction of the secondary structure of the random nucleotide sequence:

GACCACGCGACGCAGCGUCACTGTAGGATTTAAA

![Figure 1: The entropy of the structure probability distributions computed by PPfold, for random sequences of various lengths and nucleotide compositions.](image-url)
ACCGAGGGAATGCCGTCAGTAGGGTCGGGTTTAAC reveals that the underlying probability distribution has an entropy of 28.55 bits, with an average reliability score 0.65 for the final predicted structure. In comparison, the following combinatorial sequence of the same length:

GGGGAAACCCCAAAGGGGAAACCCCAAAGGGGAAA
CCCCAAAGGGGAAACCCCAAAGGGGAAACCCCAAA

shows a significantly lower entropy of 18.82 bits, at the same time as a low average reliability score of 0.56. Despite the low reliability scores in both cases, the entropy value correctly reveals that the combinatorial sequence has a relatively small number of very different secondary structures dominating the probability space, whereas the probabilities are more uniformly distributed over a large number of possible structures in the case of the random sequence.

To investigate if similar patterns could also be observed for alignments of longer biological sequences, we applied the same technique on a range of bacterial and eukaryotic 16S/18S rRNA alignments. The generation of the alignments is described in the Methods section. We plotted the entropy against the PPfold reliability scores, and colour-coded the data points after the accuracy of predictions. As Figure 2 shows, entropy can be used in conjunction with basepairing probabilities to evaluate RNA secondary structure prediction results even in the case of long biological sequences. In the cases where reliabilities are high and entropy is low, the probability space is dominated by a clearly defined set of similar structures, which are likely to be well predicted. In the cases where basepairing probabilities are low, a simultaneous high entropy implies that the underlying probability distribution lacks a clear signal (eg. in the case of folding just one sequence). By contrast, low reliabilities coupled to a low entropy suggest a distribution that has several competing structures dominating the probability space.

Relationship to prediction accuracy
An important question with respect to comparative RNA secondary structure prediction is how the accuracy of predictions varies with the quality of input alignment, and to what extent the accuracy of a predicted structure can be predicted. Reliability scores and entropy both measure variation in the secondary structure space, so both are expected to be correlated with prediction accuracy. We note, however, that a natural limitation of both entropy and reliability scores is that they are computed under the model, which effectively assumes that the model itself is an accurate description of the biological folding processes. If this is not the case, high confidence values computed under the model can still correspond to low prediction accuracies in reality.

Nevertheless, we compared how the accuracy of predictions correlates with both the average structure reliability scores and the information entropy, for all alignments in our dataset consisting of bacterial and eukaryotic 16S/18S rRNA alignments. The results are shown in Figure 3. As expected, both the PPfold reliability scores and entropy are correlated with prediction accuracy, for both bacterial and eukaryotic alignments. The correlations are stronger in the case of the bacterial alignments and weaker in the case of the eukaryotic alignments. In the case of the eukaryotic alignments, entropy appears to be slightly better correlated with prediction accuracy than PPfold reliability scores, although the difference is not statistically significant.

We observe that despite the PPfold reliability scores generally suggest somewhat higher prediction accuracies than what was actually observed, they convey absolute

![Figure 2](image_url)
information about the accuracy of predictions. Instead, entropy functions as a relative accuracy measure, when comparing several structure predictions for the same sequence. Entropy can, therefore, complement the currently existing measures for an increased understanding of RNA structure variability.

Lastly, we note that there is large variation in the reliability measures reported by different RNA structure prediction programs. By contrast, information entropy is a method-independent measure: for the same alignment, it will have the same, well-defined interpretation, regardless of the method producing the underlying probability distributions. Additionally, all classic information theoretic results can be applied to it. Information entropy can therefore be used to compare the behaviour of any secondary structure prediction methods that produce a probability distribution over RNA secondary structures, in a way that is not possible with currently existing methods.

Future perspectives include the computation of measures similar to entropy, such as the self-information of particular outcomes, or the Kullback-Leibner divergence of distributions to compare constrained and unconstrained models for RNA folding. We also expect that entropy may provide a number of other possible future applications in RNA secondary structure prediction.

Conclusions
The information entropy of the probability distribution generated by phylo-grammars can be computed efficiently from the inside-outside variables, and has been implemented as part of PPfold. Information entropy is a well-defined characteristic of the underlying probability distribution, which complements the reliability values already reported by algorithms for an increased understanding of RNA structure variability. It is also a method-independent measure of prediction certainty, providing theoretical advantages over existing methods.

Methods
Implementation
We have implemented the algorithm for the Knudsen-Hein (KH99) grammar [6] in PPfold [9]. PPfold has been written in Java 6.0 and is available as a standalone application. The rules of the KH99 grammar are as follows:

$$S \rightarrow L \mid LS$$
$$L \rightarrow c \mid cFc'$$
$$F \rightarrow cFc' \mid LS$$

We note that the KH99 grammar is not in double emission normal form, as it includes the rule $S \Rightarrow L$. Even though the grammar can be expressed in the desired form, adapting the algorithm to include this additional rule is both straightforward and highly efficient; this is therefore what we have implemented.

In the case of the KH99 grammar, there is a maximum of one bifurcating rule originating from each nonterminal symbol. It is also known that the expected frequency of a nonterminal symbol can be computed in $O(n^2)$ time for any SCFG, and the expected frequencies of rules from the same nonterminal symbol sum to the expected frequency of the nonterminal symbol. Hence, in the case of the KH99 grammar, the time complexity of the computation of the derivational entropy (given the inside-outside variables) could further be reduced to $O(n^2)$.

As described in the Results and Discussion section, the value of the entropy depends on the length of the alignments. Hence, small adaptations were made in the algorithm to be able to compare the entropies of alignments that include a particular sequence. The default option in PPfold is to remove columns where fewer than 75% of the sequences have nucleotides. In the case of entropy
computation, this is replaced with removing only the columns where the selected sequence has gaps. This ensures that all alignments that include the selected sequence have an equal prediction length.

### Test sequences and alignments

Random sequences of different nucleotide compositions were generated using the online FaBox tool [19]. For the generation of alignments, among families with an experimentally verified secondary structure in the Rfam database, we chose two families (RF01960 - Eukaryotic small subunit ribosomal RNA, and RF00177 - Bacterial small subunit ribosomal RNA), for which our initial tests indicated that a particularly wide range of structure prediction accuracies may be achieved depending on the choice on the sequences included in the alignment. Starting with a sequence of interest in both families, we constructed our datasets by randomly adding sequences from each family alignment one by one, up to a final size of maximum 15 sequences. For each alignment size (between 1 and 15 sequences in the alignment), the process was repeated 50 times. This way we obtained 1+14 × 50, not necessarily distinct cases per family, with 1-15 sequences each (including the starting sequence as a standalone case). The alignments of the selected sequences and the reference secondary structures thereof were adapted from those in Rfam, by deleting gap-only columns and any base pairs involved with those columns.

### Comparing accuracies

Accuracies are reported in terms of the F-measure, which is the harmonic mean of the sensitivity and the positive predictive value (PPV) of the basepair (bp.) predictions, compared to the comparative (reference) structure. These quantities are defined as:

$$\text{sensitivity} = \frac{TP}{\text{number of bp. in reference}}$$  
$$\text{PPV} = \frac{TP}{\text{number of bp. in prediction}}$$  
$$F - \text{measure} = 2 \times \frac{\text{sensitivity} \times \text{PPV}}{\text{sensitivity} + \text{PPV}}$$

where $TP$ is the number of correctly predicted basepairs.

#### References

14. Wang Y, Marcour A, Shareghi P, Shaw TI, Li YW, Malmberg RL, Cai L: Stable stem enabled Shannon entropies distinguish non-coding RNAs from...
random backgrounds. Computational Advances in Bio and Medical Sciences (ICCABS), 2011 IEEE 1st International Conference on 2011, 184.

doi:10.1186/1471-2105-14-S2-S22
Cite this article as: Sükösd et al.: Characterising RNA secondary structure space using information entropy. BMC Bioinformatics 2013, 14 (Suppl 2):S22.

Submit your next manuscript to BioMed Central and take full advantage of:
• Convenient online submission
• Thorough peer review
• No space constraints or color figure charges
• Immediate publication on acceptance
• Inclusion in PubMed, CAS, Scopus and Google Scholar
• Research which is freely available for redistribution

Submit your manuscript at www.biomedcentral.com/submit
Chapter 6

6.1 Problem statement

The higher-order structure of the \( \sim 10 \text{ kb} \) RNA genome of HIV-1 is currently unknown. A detailed secondary structure model has been proposed for an entire HIV-1 genome, based on nucleotide-level probing data from the SHAPE technique [6.1]. This first complete structural model supports many previously suggested structures, but also features a large number of previously uncharacterized secondary structures, and has raised some interesting questions.

Firstly, one might ask if the low degree of basepairing (60% unpaired nucleotides) in the published structure is an artefact of the prediction method, or if the HIV-1 genome is, in fact, significantly more flexible than known structural RNAs. Secondly, as we have discussed in Chapter 4 of this thesis, SHAPE-directed thermodynamic secondary structure prediction is not a particularly robust method, so locating the structures of highest reliability in the published structure remains of central importance. Indeed, a recent highly sensitive mutational study has not been able to confirm a biological significance for any of 16 selected stems from the proposed structure, and questioned whether HIV-1 has a genome-scale ordered RNA structure (GORS) [6.2].

Importantly, the prediction of long-distance interactions and non-canonical basepairs in the HIV-1 genome remains an open problem, which cannot be addressed by thermodynamic folding alone. Instead, with the increasing number of HIV-1 sequences available, compensating base changes (CBCs) in phylogenetic data can help to improve structure prediction quality.

To address these challenges, in this study we advanced from the previously published structural models. We present an alternative model for the HIV-1 secondary
structure, based on a combined comparative and SHAPE data-driven model using \textit{PPfold} 3.1 (see also Chapter 3). We re-analyse the published SHAPE data to identify general features of the HIV genome, and compare our model with a range of other predictions to identify strong structural signals. We identify several, previously uncharacterized, putative secondary structures, including long-distance interactions supported by nucleotide covariation.

\section*{6.2 Results}

We present a model for the secondary structure of the HIV-1 genome, based on a full-length folding in \textit{PPfold} (see also Chapter 3) that integrates comparative analysis with experimental data in a probabilistic framework (Figures 1, 2, 3). Several novel structural elements were predicted, many of which were supported by independent covariation evidence as well (Figure 6). We also identified a set of long-distance basepairs that organize the genome into three major structural domains (Figures 4 and 5). Additionally, our re-analysis of the SHAPE data from [6.1] suggests that the HIV-1 genome is generally flexible, with limited number of well-defined RNA secondary structure elements.

The concerns about the original normalization of SHAPE data, as outlined in the Supplementary Methods, are also quite significant from a theoretical perspective, and the effects on secondary structure prediction would deserve more attention in a different study. However, new, more rigorous methods of normalizing SHAPE data have emerged since [6.3].

\section*{6.3 Notes}

On the basis of our results, a new view emerges of the HIV-1 genome structure. One could have expected a compact RNA structure stabilized by extensive basepairing. Instead, it appears that the HIV-1 genome can be characterized as a largely flexible RNA with few well-defined and conserved secondary structures, but littered with less stable, transient RNA-RNA interactions that are not generally conserved through evolution. Our results, therefore, support previous speculation that HIV-1 does not actually belong to the group of retroviruses with a GORS. However, the low prediction reliabilities for some helices indicate possibilities for transient interactions, which may be stabilized or destabilized by protein factors, such as the NC (nucleocapsid) protein.

A high degree of flexibility in the HIV-1 genome is not surprising, as most of the genome is protein-coding, even in overlapping reading frames. This puts strong evolutionary pressures on the RNA, which may not be compatible with the development of extensive stable secondary structures. It has also been suggested that before that dsRNA-specific recognition by the cellular host can be evaded by limiting stable RNA structures [6.1]. Curiously, the nucleotide composition of HIV-1 is heavily biased towards A’s [6.4, 5], which are largely unstructured (66% unpaired) even in
the ribosome. This leads to an interesting hypothesis whereby HIV-1 exhibits a preference for A's in order to retain a largely unstructured, flexible genome.

The folding of the HIV-1 genome is also facilitated in vivo by viral proteins, most importantly the nucleocapsid (NC) protein, which possesses ATP-independent RNA chaperone activity, and binds nonspecifically to nucleic acids. A solid body of evidence shows that NC has a destabilizing effect on nucleic acid duplexes, at the same time as enhancing nucleic acid aggregation. Previous studies have shown that RT efficiency is increased by NC, possibly through a destabilization of secondary structures that could cause RT to pause. As NC is found in extremely high concentrations in the virions, it may play an important role in the global destabilization of the HIV-1 RNA structure. On the other hand, it may also facilitate the stabilization of less stable RNA duplexes by destabilizing alternative structures.

An interesting recent observation has been that during viral replication in the cell, RNA helicase A could only be detected to associate with the 5'-UTR and to a lesser extent the RRE, despite effective binding to several other isolated regions in vitro [6.6]. We suggest that a flexible higher-order structure of the genome in vivo may be responsible for this effect, even if isolated portions of the RNA may fold into stable structures in vitro.

6.4 Bibliography


Abstract
The higher-order structure of the ~10 kb RNA genome of HIV-1 is currently unknown. Here, we present a model for the secondary structure of the HIV-1 genome, based on a full-length computational study that integrates comparative analysis of RNA secondary structure with experimental data in a probabilistic framework. We employed enhanced folding algorithms, which allowed the computation of long distance interactions without distance constraints. Besides recovering numerous known structural elements, we predict many novel structural elements that are conserved in HIV-1 evolution. These include a set of long distance interactions that organize the genome into three major structural domains. Interestingly, our results suggest that the HIV-1 genome is generally flexible, with limited number of well-defined RNA secondary structure elements. However, there is evidence of less stable, non-conserved interactions, which may be stabilized by other factors in vivo.

Introduction
The RNA genome of HIV-1 is situated in the viral capsid as two non-covalent linked positive stranded RNAs, each with a total length of ~10 kb. It encodes several layers of information required for the viral replication cycle both for instructing protein synthesis and as functional RNA elements. Each strand of proviral RNA is flanked by long terminal repeats (LTRs), with the “Unique 3’” region missing in the 5’ end, and the “Unique 5’” region missing at the 3’ end. Proteins are encoded in structural (gag, pol, env), regulatory (tat, rev) and accessory (nef, vif, vpr, vpu) genes that have up to three overlapping reading frames in the central part of the genome. HIV-1 is alternatively spliced, giving rise to proteomic diversity at different stages of its replication cycle.

The HIV-1 RNA transcript is 5’-capped, 3’-polyadenylated and contains several well-characterized RNA elements. The 5’ UTR of the transcript contains important structural signals for gene expression (1,2). It begins with the highly conserved transactivator of region (TAR) hairpin which recruits the viral Tat protein (3,4). The polyA-hairpin contains an unused polyadenylation signal, but plays an important role in nuclear export, dimerization and packaging (5). The primer binding site (PBS) primarily binds the host cell tRNA\textsuperscript{Lys3}. The packaging signal Psi (Ψ) plays a role in RNA packaging and includes a major splicing donor site and the dimerization initiation site (DIS) that mediates RNA-RNA interactions between the two RNA strands through the formation of a “kissing loop”.

Further downstream, the gag-pol frameshift signal controls the ratio of Gag and Gag-Pol polyproteins, and a slippery sequence and an RNA stem are believed to be required for ribosomal frameshifting. Towards the 3’ end of the genome, the Rev response element (RRE) (6) is a highly structured, ~350 nucleotide segment in the Env coding region. The RRE recruits the viral Rev protein, and facilitates the nuclear export of the unspliced viral mRNA. A number of other structural studies have elucidated additional functional RNA structures in the HIV-1 genome, some of which have been reported to play roles in all aspects of the viral replication cycle, including modulation of ribosome processivity (7), alternative splicing (8), recombination-mediated gene swapping (9), protein evolution (10), dimerization (11,12) and circularization of the genome (13,14), as well as evading host defence mechanisms (15).
A detailed secondary structure model has been proposed for an entire HIV-1 genome, based on nucleotide-level probing data from the SHAPE technique (selective 2'-hydroxyl acylation analysed by primer extension) and a thermodynamic structure prediction algorithm. In the rest of this paper, we will refer to this as the Watts09 model. This first complete structural model supports many previously suggested structures, but also features a large number of previously uncharacterized secondary structures, and has raised some several questions.

Firstly, the relatively low degree of basepairing (~60% unpaired nucleotides) in the published structure is intriguing, and it is important to ask if it is an artefact of the prediction method, or if there is evidence of a significantly more flexible RNA in HIV-1 than in known structural RNAs. Secondly, the robustness of the proposed model to different folding parameters and algorithms is of central importance in locating the biologically significant structural elements. Indeed, a recent highly sensitive mutational study has not been able to confirm a biological significance for any of 16 selected stems from the proposed structure, and questioned whether HIV-1 has a genome-scale ordered RNA structure (GORS).

The prediction of long-distance interactions and non-canonical basepairs in the HIV-1 genome remains an open problem, which cannot be addressed by thermodynamic folding alone. Instead, with the increasing number of HIV-1 sequences available, compensating base changes (CBCs) in phylogenetic data can help to improve structure prediction quality.

To address these challenges, we advance from the previously published structural models, and present an alternative model for the HIV-1 secondary structure, based on a combined comparative and SHAPE data-driven model using PPfold 3.1. We re-analyse the published SHAPE data to identify general features of the HIV genome, and compare our model with a range other predictions to identify strong structural signals. We identify several, previously uncharacterized, putative secondary structures, including long-distance interactions supported by nucleotide covariation.

From our analysis, a new view emerges of the HIV-1 genome structure. One could have expected a compact RNA structure stabilized by extensive basepairing. Instead, it appears that the HIV-1 genome can be characterized as a largely flexible RNA with few well-defined and conserved secondary structures, but littered with less stable, transient RNA-RNA interactions that are not generally conserved through evolution. Our results, therefore, support the hypothesis that HIV-1 does not belong to the group of viruses with a GORS. The transient interactions, however, may be stabilized or destabilized by protein factors, such as the NC (nucleocapsid) protein. Finally, we discuss the biological implications and potential mechanisms of maintaining a largely unstructured genome.

Results
A comparative model for the higher-order structure of the HIV-1 genome

We predicted the consensus secondary structure of an HIV-1 alignment using PPfold 3.1, which is a re-implementation of the pfold algorithm extended with probabilistic support for RNA probing data. As input, we used a manually curated alignment of sequences from HIV-1 strains A-K (excluding strain G; see Methods), and the experimental SHAPE data from without additional processing. In order to maximize useful phylogenetic information and minimize noise, we predicted the HIV-1 structure from 38 representative sequences (Figure 1 and Figure 2). A larger alignment of 1851 HIV-1 sequences was then used as an independent control for evaluating phylogenetic evidence for the predicted structures.

The model implemented in PPfold weighs the evolutionary covariation information derived from the alignment equally with the SHAPE data measured for one B-strain sequence, integrating the available data.
for the best secondary structure model for this particular sequence. Therefore, the consensus structure prediction does not necessarily correspond to the optimal structure of each individual strain, and our result is biased towards the optimal structure for the B-strain sequence with the SHAPE data.

Our prediction setup is different from previous predictions in a number of ways. No restriction is placed on basepairing contact distance, so putative long-distance interactions are predicted. Non-canonical basepairs are also predicted, as the evolutionary model implemented in PPfold supports non-canonical covariance. Finally, the prediction is based on a model that integrates evolutionary and experimental information in a probabilistic framework based on a stochastic context-free grammar, rather than a thermodynamic model. We have previously shown that the performance of the PPfold algorithm on ribosomal datasets is comparable to RNAstructure (22) for single-sequence SHAPE-directed predictions, and superior to RNAstructure in the case of moderately good alignments (18). Our structural model for the HIV-1 genome, therefore, complements the Watts09 model, and the two predictions can be used side-to-side to guide further analysis.

Several structural elements are confirmed in both the Watts09 model and our prediction. These include the 5’ TAR hairpin, the gag-pol frameshift hairpin element and the RRE. As the two prediction models are orthogonal to each other, these consistently predicted structures presumably reflect strong structural signals in the HIV-1 genome. However, it is also immediately apparent that our structure is significantly different from the previously presented secondary structure model based on SHAPE, with only 92 helices (31% of basepairs in our prediction) occurring in both predictions. This indicates a lack of robustness in the prediction models, and an assessment of the disagreeing regions requires additional careful analysis.

In general, the distribution of PPfold reliabilities is biased towards high values. The median reliability of stems was 0.75, whereas the median reliability of unpaired regions was over 0.94. This suggests that in the predicted unpaired regions, SHAPE data and phylogenetic data were in agreement about a lack of basepairing signal. However, in the predicted helical regions with low reliabilities, alternative basepairs were possible with a significant probability mass, which indicates abundant possibilities for less stable, transient interactions. There is a good correspondence between basepairs occurring in both the Watts09 model and the PPfold model, and high-reliability basepairs predicted by PPfold: 74% of the basepairs occurring in both predictions had a PPfold reliability of 0.8 or greater. However, only 50% of all highly reliable basepairs were also predicted in both cases.

A particular challenge in the phylogenetic analysis of HIV-1 is the high degree of sequence conservation in the genome. The median percentage of conservation across all columns of the 1851 genome alignment was 96%. Due to the high degree of conservation, co-variations are sparse, and consistency across prediction methods and prediction reliability values can aid in the evaluation of substructures.

The overall topology of our predicted HIV-1 B-type structure features three “arms”, defined by a number of highly conserved central long-distance interactions. (Figure 2) This topology exposes important structural elements, such as the 5’ leader sequence and the RRE in the distal parts in two of the arms. Interestingly, it appears that a central stretch of ~130 nucleotides (positions ~3045-3175) acts as a structural “sponge” for distant parts of the genome, creating the three-armed topology by forming internal bridges in the HIV-1 genome.

Many basepairs involved in this region show significant covariation support from compensating base changes (Figure 4). Interestingly, RNA secondary structure has previously been proposed in region, which coincides with a recombination hotspot in vivo (23). However, in that study a local secondary structure was suggested on the basis of local foldings of the RNA, rather than the long-distance interaction proposed here.
Additional high-reliability long-distance interactions are predicted in the region 6390-7780, forming the “RRE arm” of the topology. Many of these interactions are also strongly supported by covariation evidence (Figure 5).

SHAPE data indicate a highly flexible HIV genome

An interesting feature of both our prediction model and the one presented in (7) is the relatively low degree of basepairing on comparison to structural RNAs, such as the ribosome. In this next section, we re-analyse the SHAPE data to investigate whether this is indicated by the data itself, or if it is just an artefact of the prediction methods.

A comparison of the distribution of the normalized SHAPE values measured for *E. coli* ribosomal sequences (25) and the HIV-1 genome reveals that in the HIV-1 genome, high SHAPE values are proportionately overrepresented (Supplementary Figure S2). This hints at a higher degree of flexibility in the HIV-1 RNA than in the ribosome. Statistical analysis makes it possible to estimate the percentage of unpaired nucleotides in the HIV-1 RNA more precisely, as the expected distribution of SHAPE values for unpaired and unpaired nucleotides is available from the SHAPE data for *E. coli* ribosomal structures. Briefly, to estimate the proportion of unpaired nucleotides in the HIV-1 genome (denoted by *a*), we assume that the SHAPE reactivities in HIV-1 follow the expected distribution:

\[
P(\text{SHAPE value}) = a \cdot P(\text{SHAPE value|unpaired}) + (1 - a) \cdot P(\text{SHAPE value|paired})
\]  

The “correct” value of *a* is determined by fitting, and once it is found, the SHAPE data can be re-normalized. The procedure is described in more detail in the Supplementary Materials.

We first demonstrated that this statistical technique correctly recovers the proportion of unpaired nucleotides in the ribosome (Supplementary Figure S4). Next, we carried out the same analysis for HIV-1. Interestingly, the null hypothesis of equality of distributions was rejected for all values of *a* at 5% significance in Pearson’s Chi-squared test, indicating that the SHAPE reactivities measured for the HIV sequence are not fully explained by the paired/unpaired model derived from the ribosome. This implies more advanced dependencies in the data, such as correlations between neighbouring values, or tertiary interactions.

However, by visualizing the differences between the two distributions (Supplementary Figure S5), it is clear that the “best match” is obtained for a proportion of over 80% unpaired nucleotides, suggesting that the probed HIV-1 genome is significantly less structured than the ribosome. In this case, if the SHAPE values for HIV-1 were adjusted to the same scale as the ribosomal values (which corresponds to multiplying the published SHAPE values by around 1.8), approximately 30% of values would fall over 1.0.

It is important to note that the originally published SHAPE values can still be used to direct RNA secondary structure prediction methods, as the values still reflect the relative flexibility of nucleotides compared to each other. The scaling factor only affects the proportion unpaired nucleotides in the resulting prediction, and thereby the false positive versus false negative rate of basepair predictions. Thus, it is expected that the previously published HIV-1 secondary structure contains a relatively large number of false positive basepair predictions, but the false negative rate is likely to be very low.

For completeness, we also investigated if the HIV-1 genome appears so unstructured due to a high degree of denaturation, in spite of the gentle purification techniques used during SHAPE probing. This seemed unlikely, as the host tRNA\(^{Lys}\) was reportedly still bound to the purified genome and a pseudoknot in the 5’ untranslated region (UTR) remained stably formed. Indeed, using the above method (data not shown) we could also recover the approximate percentage of unpaired nucleotides both in the RRE (estimated: 30%,...
actual: 36.2%), and the HIV leader sequence (estimated: 20-50%, actual: 42%), confirming that the RNA was not denatured to the extent that these stable RNA secondary structures were removed.

From the above observations, we conclude that the SHAPE data suggest a highly flexible HIV-1 genome, with few stable RNA secondary structures.

**Few strong secondary structure signals in HIV-1**

Despite evidence of a low degree of stable basepairing in the HIV-1 genome, a number of local RNA secondary structures may play important biological roles. To identify the strongest structural signals for short-range interactions, we employed a stringent filtering on the PPfold prediction, consisting of a robustness filter and a phylogenetic filter, as described below.

RNA secondary structure prediction remains an ill-conditioned problem: small changes in prediction parameters have pervasive effects on prediction outcomes, even for relatively short sequences (26). Perturbations in the data used to constrain predictions with otherwise identical prediction parameters can also significantly affect outcomes for some RNAs (16). Consistency across different prediction methods is even less likely, as also evidenced by the large variation between our prediction and the previously proposed model. As a result, the usefulness of a single prediction is limited. However, consistency across different methods, where present, is a good indicator of strong structural signals. Therefore, we predicted the secondary structure of the HIV-1 genome using a wide range of programs and parameter setups (Table 1) and classify basepairs as “consistently predicted” if they are present in 9 or more of the 13 predictions.

A comparison of the resulting predictions revealed that in fact, very few structures are consistently predicted: only 83 basepairs in 15 helices were present in all predictions. Even with less stringent filtering, only 316 basepairs in 66 helices were present in 9 or more predictions. As highly stable structures are expected to be consistently predicted, this observation also supports our hypothesis that only few highly stable structures are actually present in the genome.

To identify those consistently predicted structural elements that are also supported by co-variation evidence, we further filtered the 66 helices according to compensating base changes, as described in the Methods section. After this filtering, 27 helices in 16 substructures remained. The longest version of each helix was selected across all predictions, and these high-confidence substructures are plotted in Figure 6.

Of the known structural elements, the 5’ TAR hairpin (structure 1), the gag-pol frameshift element (structure 6), large parts of the RRE (structure 12) and the 3’ TAR hairpin (structure 16) were all very consistently predicted and supported by extensive covariation evidence, reflecting their strong structural signals. Nevertheless, differences in details between prediction methods can be observed even for these structures, highlighting the need for careful analysis.

For example, the first portion of the 5’ TAR hairpin (structure 1) is missing entirely in several comparative predictions (no. 3,11,12,13), because insufficient sequence data was available for this region from the input alignment, and these methods discard highly gapped alignment columns by default. The remaining methods also predicted several “slipped” basepairs in this region in comparison to each other.

In the proximity of the gag-pol frameshift (structure 6), the hairpin downstream of the slippery sequence is present in all predictions, confirming its high-confidence status and biological role as an enhancer of ribosomal pausing. A hairpin upstream of the slippery sequence is present in 10 predictions. Interestingly, several thermodynamic prediction methods also predicted basepairing in the slippery sequence itself, but this sequence is unpaired in all comparative structures. The SHAPE data in this region is high compared to the neighbouring helices, suggesting that the slippery sequence is in fact unpaired.

The RRE element (structure 11) is very consistently predicted, with several key stems appearing in all 13 of our predictions. Importantly, however, stem-loop IIB in the RRE high affinity site is inconsistently predicted, because of the presence of extensive non-canonical pairings in this region. The comparative,
SHAPE-driven PPfold (prediction no. 4) predicted two C-U (U7354-C7373, C7356-U7370), one G-G (G7355-G7371) and one A-C (A7357-C7369) basepair in the region. All of these non-canonical pairings were predicted by all our comparative setups as well, with or without SHAPE data (predictions 3, 4, 11, 12, 13). However, none of the methods predicted the critical non-canonical basepairs determined by X-ray crystallography (the invariant G7351-A7377 and G7352-G7375 (27)), which are required for Rev recognition. This is possibly a result of 100% sequence conservation in the alignment columns involved in these thermodynamically unusual basepairs.

Only one stem studied in a recent experimental work (17) appears on our list of consistently predicted structures: structure 8 overlaps with “Motif E/POL 1” of (17). With a mutated structure in an ultra-sensitive virus competition assay, no significant replication defects were detected upon extended culturing. This result suggests that this structure may be incidental rather than play a biological role.

Additionally, structure 7 has previously been proposed to be part of a larger secondary structure in the coding region of the pol gene, which may play a role in recombination (28). However, only Stem A passed our stringent filtering. Stem B from the same study was present in all predictions, but was not among the highly covarying stems. Prediction methods were highly inconsistent about the existence of “Stem C” (starting with 2074-2101) proposed in the same study.

The remaining consistently predicted, evolutionarily supported stems have no known function and any biological role for them remains to be confirmed by experiment. Altogether, these results show that already well-characterized elements remain among the strongest structural signals in the HIV-1 genome.

Discussion

Both the SHAPE data and the wide range of secondary structure prediction methods employed in this paper paint a picture of a highly flexible HIV-1 genome structure, with only few stable RNA secondary structures forming. These results strongly support the previous speculation that the HIV-1 retrovirus does not belong to the group of RNA viruses with a global genome-wide RNA conformation (GORS).

A high degree of flexibility in the HIV-1 genome is not surprising, as most of the genome is protein-coding, even in overlapping reading frames. This puts strong evolutionary pressures on the RNA, which may not be compatible with the development of extensive stable secondary structures. It has also been suggested that before that dsRNA-specific recognition by the cellular host can be evaded by limiting stable RNA structures (7). Curiously, the nucleotide composition of HIV-1 is heavily biased towards A’s (29,30), which are largely unstructured (66% unpaired) even in the ribosome. This leads to an interesting hypothesis whereby HIV-1 exhibits a preference for A’s in order to retain a largely unstructured, flexible genome.

The folding of the HIV-1 genome is also facilitated in vivo by viral proteins, most importantly the nucleocapsid (NC) protein, which possesses ATP-independent RNA chaperone activity, and binds nonspecifically to nucleic acids. A solid body of evidence shows that NC has a destabilizing effect on nucleic acid duplexes, at the same time as enhancing nucleic acid aggregation. Previous studies have shown that RT efficiency is increased by NC, possibly through a destabilization of secondary structures that could cause RT to pause. As NC is found in extremely high concentrations in the virions, it may play an important role in the global destabilization of the HIV-1 RNA structure. On the other hand, it may also facilitate the stabilization of less stable RNA duplexes by destabilizing alternative structures.

An interesting recent observation has been that during viral replication in the cell, RNA helicase A could only be detected to associate with the 5’-UTR and to a lesser extent the RRE, despite effective binding to several other isolated regions in vitro (31). We suggest that a flexible higher-order structure of the genome
in vivo may be responsible for this effect, even if isolated portions of the RNA may fold into stable structures in vitro.

The lack of extensive stable secondary structure does not indicate that transient basepairing interactions may not form. Both the SHAPE data and the predictions suggest rich possibilities for forming less stable basepair stacks.

Methods

Alignments

Alignments were downloaded from the Los Alamos HIV Sequence Database, http://www.hiv.lanl.gov/. For structure predictions, the 2010 HIV-1 subtype reference alignments were used, for HIV-1 strains A-K (group M, no recombinants). Strain G sequences are suspected recombinants and were removed. To be able to compare our results with currently existing models, the alignment was extended with three strain B sequences: the sequence used in (7), the sequence used in (8) (acc. nr. AF324493), and the GenBank reference sequence (acc. nr. NC_001802) with annotations. The alignment was lightly edited to correct any obvious misalignments. The resulting alignment consists of 38 sequences. For evaluating phylogenetic support for proposed structures, the complete 2011 HIV-1/SIVcpz web alignment (all sequences, including recombinants) was used. This alignment contains 1850 sequences. To aid structural comparison, the sequence from (7) was aligned to it by hand. Subtype alignments were extracted from the two full genomic alignments directly, without manual curation. Alignments were edited in CLC Main Workbench 6 and MEGA 5.0 (32).

SHAPE data

The SHAPE data for a complete HIV-1 genome used in this study had been previously published in (7). SHAPE data for *E. coli* 16S and 23S rRNA sequences, used in (25) was obtained from K. Weeks (personal communication).

Structure predictions

Structures were predicted using several state-of-the-art programs for RNA secondary structure prediction. For thermodynamic predictions based on free energy minimization, RNAstructure (22), UNAfold (33) and GTfold (34) were used. Each of them implements the nearest-neighbour thermodynamic model for RNA folding, but due to slight implementation differences the prediction results can differ in less stable regions. RNAstructure and GTfold have integrated support for SHAPE data. Comparative predictions were done using RNAalifold (35,36), PPfold and PETfold (37,38). RNAalifold implements the nearest-neighbour thermodynamic model coupled to a model for converting covariation to pseudo-free energies. PPfold is based on a lightweight stochastic context-free grammar (SCFG), and is a re-implementation of Pfold with added support for probing data, such as SHAPE. PETfold integrates SCFG-based and thermodynamic predictions, but currently has no support for SHAPE data.

Reliability values

The reliability values reported by PPfold provide insights into the nature of the folding landscape under the model. The reliability value for a nucleotide is the probability of the predicted structure under the model. Given the alignment and SHAPE data, PPfold obtains the probability of all possible basepairs using the inside-outside algorithm, which is similar to the partition function algorithm used in thermodynamic
approaches. A highly reliable unpaired nucleotide implies that any possible basepairs have very low probabilities under the model. A highly reliable basepair implies that the probability space is highly dominated by structures in which that particular basepair occurs. A single-stranded nucleotide with a reliability value of 0 always arises from a lack of evolutionary information and experimental data. A low but non-zero reliability value for a nucleotide implies competing basepairs in that position, but no single basepair is more probable than the final prediction.

**Phylogenetic filtering of structures**

Taking the SHAPE-directed Pfold prediction (prediction nr. 4), covariations were computed based on an independent, 1851-sequence alignment.

For each predicted basepair, one-sided and two-sided covariations were determined according to the corresponding alignment columns in the following steps:

1. Sequences that are gapped in either column are discarded (in the consideration of this basepair only).
2. Of the remaining sequences in the alignment, the most frequently occurring basepair is identified.
3. Sequences with the most frequent basepair are discarded.
4. Canonical basepairs are counted among the remaining sequences, and classified into one-sided or two-sided covariations with respect to the most frequent basepair.
5. One-sided and two-sided covariations are expressed as a % of the number of possible covariations.

Prediction results were then filtered by removing basepairs that fulfill at least one of the following conditions:

1. The number of possible covariations is less than 5% of the number of sequences. This was to make sure a sufficient number of sequences was available for appropriately measuring covariation.
2. One-sided covariation is less than 75% AND two-sided covariation is less than 25%.

**Data processing**

Data processing and analysis was done in MATLAB R2010b. Covariation was evaluated by counting compensating base changes in alignments.

**Structure drawings**

Structure drawings were made in XRNa 1.1.12 and JViz.RNA.

**Data access**

The prediction results are included in the Supplementary Materials.

**Acknowledgments**

We would like to thank K. Weeks for providing the SHAPE data used in this analysis, B. Shabash and K. C. Wiese for making available a more efficient, beta-version of JViz.RNA, and N. Larsen, J. S. Pedersen and S. Harvey for useful discussions.
References

<table>
<thead>
<tr>
<th>Prediction number</th>
<th>Software</th>
<th>Method</th>
<th>Alignment</th>
<th>SHAPE data</th>
<th>Max. contact distance = 600 nt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RNAstructure</td>
<td>Thermod.</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>RNAstructure</td>
<td>Thermod.</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>3</td>
<td>PPfold 3.1</td>
<td>SCFG</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>4</td>
<td>PPfold 3.1</td>
<td>SCFG</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>5</td>
<td>GTfold (Aug 2011)</td>
<td>Thermod.</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>6</td>
<td>GTfold (Aug 2011)</td>
<td>Thermod.</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>7</td>
<td>GTfold (Aug 2011)</td>
<td>Thermod.</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>8</td>
<td>GTfold (Aug 2011)</td>
<td>Thermod.</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>9</td>
<td>UNAfold</td>
<td>Thermod.</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>10</td>
<td>UNAfold</td>
<td>Thermod.</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>11</td>
<td>PETfold</td>
<td>Combin.</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>12</td>
<td>RNAalifold</td>
<td>Thermod.</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>13</td>
<td>PPfold 3.1</td>
<td>SCFG</td>
<td>YES</td>
<td>YES (re-normalized)</td>
<td>NO</td>
</tr>
</tbody>
</table>

*Table 1*: Summary of secondary structure predictions. Prediction 1 is identical to the structure published in (7). In SHAPE-driven predictions, unless otherwise indicated, the SHAPE data published in (7) was used without re-normalization. In phylogenetic predictions, the manually curated HIV-1 alignment described in Methods was used. The full pairings are provided in the *Supplementary Materials*.
Figure 1: A comparative model for the secondary structure of the HIV-1 genome, predicted with PPfold 3.1, integrating covariation information from a 38-way manually curated alignment and structure probing data from the SHAPE method. Each nucleotide is coloured according to the probability of the predicted structure in that position. Long-distance interactions (further than 600 nt apart) are indicated as rectangular boxes with letter codes. A higher resolution version of the figure with nucleotide identities is available in the Supplementary Materials.
Figure 2: Topology of the predicted consensus secondary structure of the HIV-1 genome, predicted with PPfold 3.1. The structure was drawn in jViz.Rna (24) and annotated manually after visual inspection.
Figure 3: Phylogenetic analysis of the proposed structural model. The proportion of CBPs (two-sided covariation) is plotted. This is the proportion of basepairs consistent with the structural prediction (A-U, G-C, G-U), when removing gaps and the most frequent canonical basepair.
Figure 4: Putative long-distance interactions in the central domain of the HIV-1 genome, coordinating the formation of 3 arms. The letters correspond to those in Figure 1. Basepairs with significant CBCs are indicated, in 5’-3’ direction.

Figure 5: Putative long-distance interactions in the “RRE domain” of the HIV-1 genome. The letters correspond to those in Figure 1. Basepairs with significant CBCs are indicated, in 5’-3’ direction.
Figure 6: Short-distance interactions supported both by covariation evidence and consistency of prediction. CBCs are indicated as boxes for qualifying basepairs.
Supplementary Methods

Obtaining distributions for SHAPE values

The pseudoknotted comparative structures were downloaded from CRW (13). Nucleotides with invalid SHAPE data (-999) were removed. The remaining nucleotides were grouped according to structural category into “paired” and “unpaired”. Non-canonical basepairs and pseudoknotted nucleotides were considered pairing, but nucleotides participating in other tertiary interactions were considered unpaired. The distribution of SHAPE values was obtained for each category by counting frequencies at a resolution of 0.1 SHAPE units. The frequency distributions were normalized by dividing by the total number of observations.

Recovering the proportion of unpaired nucleotides from SHAPE data

The published SHAPE counts had originally been normalized to a scale such that 1.0 represents the nucleotides with “highest reactivity”. The percentage of nucleotides with normalized SHAPE reactivities over 1.0 was 9% in the case of the ribosomal dataset, and 5% in the case of HIV-1. However, it is clear from the distributions for unpaired versus paired nucleotides in the ribosome that the percentage of values over 1.0 is lower in the case of paired nucleotides, and higher in the case of unpaired nucleotides. (Supplementary Figure S3) This implies that for a highly flexible sequence, normalizing to the highest reactive nucleotides will bias the normalized SHAPE reactivities towards low values.

To estimate the proportion of unpaired nucleotides in the HIV-1 genome (denoted by $a$), we assume that the SHAPE reactivities in HIV-1 follow the expected distribution:

$$P(\text{SHAPE value}) = a \cdot P(\text{SHAPE value|unpaired}) + (1 - a) \cdot P(\text{SHAPE value|paired}) \quad [1]$$

When comparing the observed distribution to this expected distribution, the observed SHAPE data must be scaled, such that the threshold for the top 5% of values matches that of [1]. This criterion ensures consistent normalization across the ribosome-based models and the HIV data, and the scaling factor differs for each value of $a$. The distributions $P(\text{SHAPE value|paired})$ and $P(\text{SHAPE value|unpaired})$ were determined using SHAPE data measured on the *E. coli* 16S and 23S sequences, and correspond to the plots shown in Figure S2. We expect that that for the “correct” value of $a$, the distribution of observed SHAPE values is statistically indistinguishable from [1].
Supplementary Figures

Supplementary Figure S1
High-resolution nucleotide pairings, see PDF

Figure S1: High-resolution drawing of the predicted secondary structure of HIV-1 RNA. The colour scale indicates prediction reliabilities.
Supplementary Figure S2

Figure S2: Distributions of SHAPE data for ribosomal RNA vs. HIV-1 RNA.
Supplementary Figure S3: Distributions of SHAPE data for E.coli 16S and 23S ribosomal rRNA, for paired and unpaired nucleotides.

**Figure S3:** Distributions of SHAPE data for E.coli 16S and 23S ribosomal rRNA, for paired and unpaired nucleotides.
Supplementary Figure S4:

**Figure S4**: Estimating the proportion of unpaired nucleotides in the ribosome. The proportion of unpaired nucleotides in the whole sequence determines the theoretical shape of the normalized frequency distributions for SHAPE values. The actual distribution of SHAPE values is subtracted from the theoretical distributions, and the probability differences are plotted. The “best-fit” case is when the differences are least prominent, which in this case is clearly obtained around 40% unpaired nucleotides. The true percentage of unpaired nucleotides in the E.coli ribosomal 16S and 23S rRNA taken together is 39.5%.
**Supplementary Figure S5:**

Figure S5: Estimating the proportion of unpaired nucleotides in the ribosome. The proportion of unpaired nucleotides in the whole sequence determines the theoretical shape of the normalized frequency distributions for SHAPE values. The actual distribution of SHAPE values is subtracted from the theoretical distributions, and the probability differences are plotted. The “best-fit” case is when the differences are least prominent, which in this case is above 80% unpaired nucleotides.
Appendix

A

Quantifying variances in comparative RNA secondary structure prediction

A.1 Summary

This paper was produced in collaboration with Jotun Hein’s group at Oxford University, with James Anderson as the main author.

The purpose of the study was to explore possible factors that can contribute to poor quality in comparative RNA secondary structure prediction. The main factors we investigated are alignment quality, the number of sequences chosen, and the evolutionary relationships between the sequences.

On the basis of reference alignments from a curated dataset based on the Rfam database, the StatAlign program was used to generate random alignment samples according to their probabilities. \textit{PPfold} and RNAalifold were then run on the dataset, and the resulting structures evaluated.

Generally, increasing the number of sequences in an alignment increased accuracy in \textit{PPfold}, but not RNAalifold. We hypothesized that this was due to the increasing number of non-canonical basepairs in the alignment, which the thermodynamic method could not predict. Nevertheless, we concluded that in the case of our datasets, 5 sequences were sufficient for approaching maximal predictive accuracy. The required number of sequences might be larger for longer RNA sequences.

As expected, there was also a clear relationship between alignment quality and loss of accuracy (Figure 2). However, we found no significant correlation between evolutionary distance in sequences of the alignment and predictive accuracy, suggesting that evolutionary distance is not an underlying factor for variation in RNA secondary structure prediction.

We also defined a novel “reliability score”, which considers the estimated distance of the alignment from the reference alignment (Figure 3) as well as basepairing prob-
Appendix A. Quantifying variances in comparative RNA secondary structure prediction

abilities (Figure 5). This predictor improved on the PPfold reliability score, which is based on basepairing probabilities only. An extension of information entropy (Chapter 5) was also developed to consider variation over alignments.

We noted that even after accounting for uncertainties in the input alignment, there was still variability in the prediction quality, which we therefore suggest comes from the RNA secondary structure predictive model itself.
Quantifying variances in comparative RNA secondary structure prediction

James WJ Anderson¹, Adám Novák¹, Zsuzsanna Sükösd², Michael Golden³, Preeti Arunapuram⁴, Ingolfur Edvardsson⁵ and Jotun Hein¹

Abstract

Background: With the advancement of next-generation sequencing and transcriptomics technologies, regulatory effects involving RNA, in particular RNA structural changes, are being detected. These results often rely on RNA secondary structure predictions. However, current approaches to RNA secondary structure modelling produce predictions with a high variance in predictive accuracy, and we have little quantifiable knowledge about the reasons for these variances.

Results: In this paper we explore a number of factors which can contribute to poor RNA secondary structure prediction quality. We establish a quantified relationship between alignment quality and loss of accuracy. Furthermore, we define two new measures to quantify uncertainty in alignment-based structure predictions. One of the measures improves on the "reliability score" reported by PPfold, and considers alignment uncertainty as well as base-pair probabilities. The other measure considers the information entropy for SCFGs over a space of input alignments.

Conclusions: Our predictive accuracy improves on the PPfold reliability score. We can successfully characterize many of the underlying reasons for and variances in poor prediction. However, there is still variability unaccounted for, which we therefore suggest comes from the RNA secondary structure predictive model itself.

Background

RNA secondary structure prediction is still an important problem in computational biology. With the advent of next generation sequencing and RNA-seq technologies, many RNA structural changes are being found to play important roles in regulating gene expression [1,2]. Gene regulation studies can now be done on a genome-wide scale. In some cases RNA secondary structures can be experimentally determined on a genome-wide level [3], but these methods require RNA isolation and may not preserve in vivo structures. RNA secondary structure prediction programs are still often used to predict structures across the genome [4]. The predicted secondary structures, and predicted structural changes, are being used to find relationships and suggest mechanisms in gene regulatory networks.

Some methods for RNA secondary structure prediction only consider a single sequence as input. However, prediction quality can be improved by using multiple sequences, assuming that RNA secondary structure is conserved through evolution. Even without a complex evolutionary model, these additional structural constraints provide valuable information on folding. Comparative methods for RNA secondary structure prediction are based on this observation, and use evolutionary information from multiple alignments to improve prediction quality.

Methods for RNA secondary structure prediction generally fall into two categories. Thermodynamic models make use of free-energy functions, which take experimentally determined energy parameters for individual structural elements. Dynamic programming is then used to find the secondary structure with the minimum free energy, which is reported as the predicted structure. This has been successfully implemented in programs such as RNAfold [5] and UNAfold [6]. Thermodynamic methods typically deal with the single-sequence prediction problem, but extensions such as RNAalifold [7] and PETfold [8] allow for comparative prediction.

Stochastic context-free grammars (SCFGs), on the other hand, define a probability distribution over the
space of RNA secondary structures. Posterior decoding techniques are typically used to determine, for example, the maximum expected accuracy structure [9]. SCFGs have been used for the single-sequence prediction problem [10], but their advantage comes through coupling with a molecular evolution model. The first comparative SCFG-based approach was developed in Pfold [11,12], where alignment column probabilities were determined through single and paired column evolution models, calculated via the Felsenstein pruning algorithm [13]. For a more complete review on RNA secondary structure prediction, see [14].

In genome-wide predictions of RNA secondary structure, the accuracy of the secondary structure prediction benchmark is rarely factored into analysis. Typically only the mean accuracy is reported in RNA secondary structure prediction benchmarks [14], with the variance in the accuracy given little thought. Variance in accuracy is particularly problematic in the case of single-sequence prediction. Figure 1 shows the cumulative density function of predictive accuracy for two single-sequence applications of RNA secondary structure prediction, RNAfold and PPfold (a recent implementation of Pfold, [15]), on 443 sequences taken from the RNASTRAND database [16]. Additionally, a uniform (0,1) cumulative density function is shown for comparison. The figure illustrates that for sequences in this data set, the predictive accuracy of RNAfold and PPfold is not very much different from a random number between 0 and 1. When genome-wide RNA secondary structure prediction is done on a large number of single sequences, many predictions will be poor ones.

Consequently, it is important to understand more about the variability of RNA secondary structure prediction programs. In comparative prediction, there are many sources of variability: alignment quality, the number of sequences chosen and which alignment samples have been taken, the evolutionary relationships between sequences, as well as the ill-conditioned nature of the folding model itself. Understanding and quantifying these variances is key for biological applications that rely on these folding programs. Additionally, other bioinformatics software that utilize these folding programs— for example inverse RNA folding algorithms [17]— may often experience a fundamental limitation in performance due to variance in structure prediction quality.

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 [18,19]. Although this methodology is successful when the alignment is well resolved, it has been shown in the context of downstream analyses that the end result can be highly sensitive to the choice of alignment [20-22]. RNA secondary structure prediction methods take a variety of approaches with respect to possible errors in RNA alignments. Some methods (e.g. [23]) invoke a fold-and-align approach directly, where alignment is done simultaneously with structure prediction. Pfold, instead, takes a fixed alignment as input, but allocates a finite probability to a nucleotide being any other nucleotide; this makes the model more robust to poor alignment quality. Most modern methods (e.g. [24]) that use constrained scoring matrices incorporate the variability of the alignment.

In comparative prediction, the accuracy of the secondary structure prediction benchmark is rarely factored into analysis. Typically only the mean accuracy is reported in RNA secondary structure prediction benchmarks [14], with the variance in the accuracy given little thought. Variance in accuracy is particularly problematic in the case of single-sequence prediction. Figure 1 shows the cumulative density function of predictive accuracy for two single-sequence applications of RNA secondary structure prediction, RNAfold and PPfold (a recent implementation of Pfold, [15]), on 443 sequences taken from the RNASTRAND database [16]. Additionally, a uniform (0,1) cumulative density function is shown for comparison. The figure illustrates that for sequences in this data set, the predictive accuracy of RNAfold and PPfold is not very much different from a random number between 0 and 1. When genome-wide RNA secondary structure prediction is done on a large number of single sequences, many predictions will be poor ones.

Consequently, it is important to understand more about the variability of RNA secondary structure prediction programs. In comparative prediction, there are many sources of variability: alignment quality, the number of sequences chosen and which alignment samples have been taken, the evolutionary relationships between sequences, as well as the ill-conditioned nature of the folding model itself. Understanding and quantifying these variances is key for biological applications that rely on these folding programs. Additionally, other bioinformatics software that utilize these folding programs— for example inverse RNA folding algorithms [17]— may often experience a fundamental limitation in performance due to variance in structure prediction quality.

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 [18,19]. Although this methodology is successful when the alignment is well resolved, it has been shown in the context of downstream analyses that the end result can be highly sensitive to the choice of alignment [20-22]. RNA secondary structure prediction methods take a variety of approaches with respect to possible errors in RNA alignments. Some methods (e.g. [23]) invoke a fold-and-align approach directly, where alignment is done simultaneously with structure prediction. Pfold, instead, takes a fixed alignment as input, but allocates a finite probability to a nucleotide being any other nucleotide; this makes the model more robust to poor alignment quality. Most modern methods (e.g. [24]) that use constrained scoring matrices incorporate the variability of the alignment.
[24]) still consider prediction from a single, fixed alignment. Recently, alignment-free methods have also been proposed [25]. 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 variable. 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 sequences, particularly [26], which is based on evolutionary models and structural constraints. This is implemented in [27], which shows strong results in RNA secondary structure prediction.

In this paper we consider the problem of variances in comparative RNA secondary structure prediction. We present statistical analyses of different variances including the relationship between sequence prediction quality and chosen alignment distance from the reference alignment, and a predictive algorithm for accuracy is provided. Factors like the number of sequences in the alignment and the evolutionary distance of the sequences are considered. Finally, a novel method is presented which extends information entropy for stochastic context-free grammars [28] to consider variation co-sampling other entities such as the evolutionary model parameters. This high-dimensional joint distribution can be found by selecting useful sequences.

Alignment and RNA secondary structure accuracy metrics

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 [36], which is generalised to an alignment method in [37].

Let \( a_{i \rightarrow j} \) be an alignment of a sequence \( s_1 \) of length \( n \) to a sequence \( s_2 \) of length \( m \). Each column of \( a_{i \rightarrow j} \) can be expressed as pairs of the form \( (s_{1 \rightarrow j}^1, s_{1 \rightarrow j}^2) \), \((s_{1 \rightarrow j}^1, -)\), and \((- s_{1 \rightarrow j}^2)\). We define

\[
H(a_{i \rightarrow j}) = \{ (i, j) \mid (s_{1 \rightarrow j}^1, s_{1 \rightarrow j}^2) \in a_{i \rightarrow j} \}
\]

\[
I(a_{i \rightarrow j}) = \{ (i, j) \mid (-, s_{1 \rightarrow j}^2) \in a_{i \rightarrow j} \}, \quad \text{and}
\]

\[
D(a_{i \rightarrow j}) = \{ (i, j) \mid (s_{1 \rightarrow j}^1, -) \in a_{i \rightarrow j} \}.
\]

sets which represent ‘homology’, ‘insertion’ and ‘deletion’ respectively. Given these sets, we define the distance between two alignments \( a_{i \rightarrow j}^1 \) and \( a_{i \rightarrow j}^2 \) of two sequences \( s_1 \) and \( s_2 \) to be

\[
d(a_{i \rightarrow j}^1, a_{i \rightarrow j}^2) = n + m - 2H(a_{i \rightarrow j}^1) - H(a_{i \rightarrow j}^2) - 2|D(a_{i \rightarrow j}^1) \cap D(a_{i \rightarrow j}^2)| - |I(a_{i \rightarrow j}^1) \cap I(a_{i \rightarrow j}^2)|.
\]

For example, consider the case \( a_{i \rightarrow j}^1 = a_{i \rightarrow j}^2 \). Then we have \( n = m \cdot 2H(a_{i \rightarrow j}) \) as there are no deletions, and \( I(a_{i \rightarrow j}) \cap I(a_{i \rightarrow j}) = 0 \) as there are no insertions. This gives the distance between the alignments as zero, as would be expected.

Equation 1 can be generalised to sequence alignments with more than two sequences. Assuming now that \( a^s \)

\[
\text{Equation 1 can be generalised to sequence alignments with more than two sequences. Assuming now that } a^s
\]

\[
\text{the simple statistics of marginalised single dimensions of sequence alignments, evolutionary trees and model parameters. This high-dimensional joint distribution can be analysed in several ways, ranging from the simple statistics of marginalised single dimensions (e.g. the posterior distribution of a single rate parameter) to the application of other tools to the alignment samples.}
\]

\[
\text{Alignment and RNA secondary structure accuracy metrics}
\]

\[
\text{To analyse variances of RNA secondary structure as alignment quality varies, we calculate a similarity score that measures how close a sample alignment is to the reference alignment. We use an alignment metric, taken from [36], which is generalised to an alignment method in [37].}
\]

\[
\text{Let } a_{i \rightarrow j} \text{ be an alignment of a sequence } s_1 \text{ of length } n \text{ to a sequence } s_2 \text{ of length } m. \text{ Each column of } a_{i \rightarrow j} \text{ can be expressed as pairs of the form } (s_{1 \rightarrow j}^1, s_{1 \rightarrow j}^2), (s_{1 \rightarrow j}^1, -), \text{ and } (-, s_{1 \rightarrow j}^2). \text{ We define}
\]

\[
H(a_{i \rightarrow j}) = \{ (i, j) \mid (s_{1 \rightarrow j}^1, s_{1 \rightarrow j}^2) \in a_{i \rightarrow j} \},
\]

\[
I(a_{i \rightarrow j}) = \{ (i, j) \mid (-, s_{1 \rightarrow j}^2) \in a_{i \rightarrow j} \}, \quad \text{and}
\]

\[
D(a_{i \rightarrow j}) = \{ (i, j) \mid (s_{1 \rightarrow j}^1, -) \in a_{i \rightarrow j} \}.
\]

sets which represent ‘homology’, ‘insertion’ and ‘deletion’ respectively. Given these sets, we define the distance between two alignments \( a_{i \rightarrow j}^1 \) and \( a_{i \rightarrow j}^2 \) of two sequences \( s_1 \) and \( s_2 \) to be

\[
d(a_{i \rightarrow j}^1, a_{i \rightarrow j}^2) = n + m - 2H(a_{i \rightarrow j}^1) - H(a_{i \rightarrow j}^2) - 2|D(a_{i \rightarrow j}^1) \cap D(a_{i \rightarrow j}^2)| - |I(a_{i \rightarrow j}^1) \cap I(a_{i \rightarrow j}^2)|.
\]

For example, consider the case \( a_{i \rightarrow j}^1 = a_{i \rightarrow j}^2 \). Then we have \( n = m \cdot 2H(a_{i \rightarrow j}) \) as there are no deletions, and \( I(a_{i \rightarrow j}) \cap I(a_{i \rightarrow j}) = 0 \) as there are no insertions. This gives the distance between the alignments as zero, as would be expected.

Equation 1 can be generalised to sequence alignments with more than two sequences. Assuming now that \( a^s \)
and $a^i$ are alignments of $n$ sequences $s_1, ..., s_n$ of lengths $n_1, ..., n_m$, we have

$$d(a^i, a^j) = \sum_{k=1}^{m} \sum_{q=p-1}^{n_m} d(a^i_k, a^j_k)$$

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

$$SS(a^i, a^j) = 1 - \frac{d(a^i, a^j)}{(m-1) \sum_{l=1}^{n_l}}$$

The denominator of the fraction, $(m-1) \sum_{l=1}^{n_l}$, is the normalizing constant, the maximum that the alignment distance $d(a^i, a^j)$ can be. The similarity score is bounded by 0 and 1, with 1 indicating that the sample alignment is identical to the reference alignment.

**RNA secondary structure metrics**

There are a wealth of available metrics on RNA secondary structure prediction [14, 38]. Here we use sensitivity, positive predictive 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

- **Sensitivity**
  $$\text{Sensitivity} = \frac{TP}{TP + FN}$$
- **PPV**
  $$\text{PPV} = \frac{TP}{TP + FP}$$
- **F-score**
  $$\text{F-score} = \frac{2 \times TP}{2 \times TP + FN + FP}$$

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.

**Information entropy**

As we later develop calculations for information entropy for a set of alignments, here we outline the computation of information entropy for a single alignment. The information entropy $H$ of a probability distribution $P$ with a set of events $X$ is defined as:

$$H(P) = -\sum_{x \in X} P(x) \log_2(P(x)).$$

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 [28]. This is outlined here.

Let the set of all derivations for the input alignment be $\Phi$. Since the probability of a derivation $d$ can be written as the product of the SCFG production rule probabilities and the phylogenetic probabilities, we can write the total probability $T$ of the grammar producing the input string as

$$T = \sum_{d \in \Phi} P[d] = \sum_{d \in \Phi} P_{\text{SCFG}}[d]P_{\text{phylo}}[d],$$

where $P_{\text{SCFG}}[d]$ denotes the prior probabilities obtained from the SCFG part of the model, and $P_{\text{phylo}}[d]$ are the likelihood factors obtained from the phylogenetic model. Conditioning on producing the input string, the normalized probability of a derivation $d$ is $P[d] = \frac{T}{T}P_{\text{SCFG}}[d]P_{\text{phylo}}[d]$. Consequently, we have that the information entropy of the input alignment under the phylo-SCFG model is

$$H_{\text{phylo}}(G) = -\sum_{d \in \Phi} P_{\text{SCFG}}[d] \log_2(P_{\text{phylo}}[d]),$$

which can be written using Equation 5 as

$$H_{\text{phylo}}(G) = \log_2(T) - \sum_{d \in \Phi} P[d] \log_2(P[d]) - \sum_{d \in \Phi} P_{\text{SCFG}}[d] \log_2(P_{\text{phylo}}[d]),$$

that is, separating out the SCFG contribution and the phylogenetic contribution. To calculate the entropy in practice, firstly we use a simplified form of the SCFG
contribution. For a SCFG with set of production rules \( R \),
we can write the SCFG contribution in terms of the
expected production rule frequency,
\[
\sum_{d \in D} P[d] \log_2 (P_r[d]) = \sum_{d \in D} \log_2 (P_r[r]) \mathbb{E}[\text{uses of } r].
\]  

(8)

Secondly, we can simplify the phylogenetic contribution.
Let \( r_a \in R \) be a SCFG rule which produces base pairs,
and \( r_b \in R \) a SCFG rule which produces unpaired bases.
Define \( I^d(i, j) \), the indicator function for whether
the column pair \((i, j)\) is emitted from a rule \( r_a \) (i.e. position
\( i \) and \( j \) form a pair), and \( I^c(i, j) \), the indicator function
for whether column \( i \) is emitted from a rule \( r_b \) (i.e. position
\( i \) is unpaired). Finally, define \( f^d(r_a) \) as the frequency
that rule \( r_a \) is used in derivation \( d \). Then
\[
\sum_{d \in D} P[d] \log_2 (P_I[i,j]) = \sum_{d \in D} P[d] \left( \sum_{r_a} \log_2 (P_r[i,j]) \mathbb{E}[\text{uses of } r_a] \right)
+ \sum_{d \in D} P[d] \left( \sum_{r_b} \log_2 (P_r[i,j]) \mathbb{E}[\text{uses of } r_b] \right)
+ \sum_{d \in D} P[d] \left( \sum_{r_c} \log_2 (P_r[i,j]) \mathbb{E}[\text{uses of } r_c] \right)
\]

As \( \sum_{d \in D} I^d(i, j) P[d] \) is the total probability under the
model that positions \( i \) and \( j \) are emitted from a rule \( r_a \),
and \( \sum_{d \in D} I^c(i, j) P[d] \) is just the total probability under the
model that position \( i \) is emitted from a rule \( r_b \), the quantity
\( \sum_{d \in D} P[d] \log_2 (P_I[i,j]) \) can be computed using the
expected rule frequencies obtained through the inside-outside algorithm [39].

Methods

Data

StatAlign Dataset
To test factors relating to alignment quality and secondary structure prediction quality, a large number of alignment samples from trusted reference alignments with known secondary structures are needed. We have created a curated RNA dataset based on the Rfam database [40] 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 (see Additional file 1) in the section “StatAlign Dataset”. From each family, in a pre-filtering step we removed divergent sequences with long indels, as follows. 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-penalize long indels, we applied the super linear 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 50 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 sequences in alignments. From these samples of known reference alignment, we could produce many different alignment samples using StatAlign [32]. For each RNA alignment, 200 alignment samples were taken, and the reference alignments were also kept to for comparison. We refer to this dataset throughout as the StatAlign dataset.

Random alignment data

We also wanted to measure the effect of alignment accuracy on secondary structure independently of alignment method. Therefore 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. 2, we created a Metropolis-coupled MCMC framework that runs several parallel MCMC chains to take alignment samples from the target distribution

\[
\pi(a) = \exp \left( \frac{2|d(a, a') - d|}{t} \right)
\]  

(9)

where \( a' \) is the reference alignment, \( d \) 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. 9) 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 alignments 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 \( A \rightarrow B \) and \( B \rightarrow A \) 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,
Appendix A. Quantifying variances in comparative RNA secondary structure prediction

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. 3), 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.

Extending information entropy to alignment space

The information entropy defined for a single alignment contains the length of the alignment as a parameter. Attempting to extend the measure to the probability mass over RNA secondary structure space, variable alignment length is a concern. For example, if we have two alignments and corresponding secondary structures

\[(\begin{array}{clllll} C & C & C & C & C & C \\ A & G & T & T & T & T \\ A & G & T & T & T & T \\ A & G & T & T & T & T \\ A & G & T & T & T & T \\ A & G & T & T & T & T \\ \end{array}) \quad (\begin{array}{clllll} C & C & C & C & C & C \\ A & G & G & G & G & G \\ C & C & A & A & A & G \\ G & G & C & C & A & G \\ G & G & C & C & A & G \\ G & G & C & C & A & G \\ \end{array}) \]

we would not want to suggest that these alignments give two different secondary structures. Consequently, we use a projection method to give alignment column pairing probability matrices the same dimension, so that the matrices can be averaged.

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 corresponding to gap positions in the reference sequence, then end up with an n × n matrix as required.

To calculate information entropy over alignments, we need to be able to calculate the probability of each alignment. However, we cannot calculate the information entropy explicitly, since the probability of a given secondary structure is

\[ P[A] = \sum_{s} P[s|A]P[A], \]

and there is no known efficient way to recurse over all possible alignments. Instead, we create an information entropy measure based on samples from the alignment space, 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, the probability of being unpaired

\[ P[c \text{ unpaired}] = \sum_{i=1}^{n} P[c \text{ unpaired in alignment } A_i | P[a_i]]. \]

with an analogous result holding for paired columns. We now define a sample phylogenetic probability \( P \) as the average of the sample phylogenetic probabilities:

\[ P[c \text{ unpaired}] = \frac{1}{n} \sum_{i=1}^{n} P[c \text{ unpaired in alignment } A_i | P[a_i]]. \]

To show this sample probability converges to the true probability as sample size tends to infinity, we first note that rearranging Equation 12:

\[ P[c \text{ unpaired}] = \sum_{A \in \text{Alignments}} P[c \text{ unpaired in alignment } A] \left( \sum_{i=1}^{n} \frac{1}{n} |A_i| \right). \]

Taking the limit \( n \to \infty \) by the weak law of large numbers

\[ P[c \text{ unpaired}] \to \sum_{A \in \text{Alignments}} P[c \text{ unpaired in alignment } A | P[a_i] \text{ as } n \to \infty] \]

\[ = P[c \text{ unpaired}] \]

as required.

Now, we have from above that the entropy \( H_0(P) \) of grammar derivations \( \Phi \) of a grammar \( G \) is

\[ H_0(P) = \log_2(T) \sum_{d \in \Phi} P_d |d| P_Y(d) \log_2(P_Y(d)). \]

Since tree probabilities are the product of unpaired and paired column probabilities in the derivation, the tree probabilities can be recalculated from the sample of alignments. These can then be substituted into the above equation to get an approximation to the information entropy over the space of sampled alignments as well. We refer to this entropy of more than one alignment as the alignment consensus entropy.

Results and discussion

Alignment quality and predictive accuracy

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 [11], 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 was increased. 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 normalizing 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.

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 2A), we observe a weaker correlation between alignment distance and accuracy than for the Random Alignment dataset.
Appendix A. Quantifying variances in comparative RNA secondary structure prediction

Figure 2 Log-scale heatmaps for mean accuracy, and for percentage of accuracy retained, when alignment quality varies. 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 2B). 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 evolutionary model, which the random alignments do not consider, and so StatAlign’s alignments could be considered more realistic. This confirms the expectation that StatAlign’s alignments are more useful for RNA secondary structure prediction than random alignments.

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, 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 2B), 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 functions better than RNAalifold when given a low-quality alignment, likely due to its more complete model for molecular evolution.

**Evolutionary distance**

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 normalized 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.

We then looked at the correlation between relative evolutionary distance and predictive accuracy. All methods 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 those other factors, such as those seen in [26], play a more important role in determining predictive accuracy.

**Alignment distances and maximum posterior decoding**

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 [36]. First, the sequences were aligned with ClustalW [18]. The sequences were then re-aligned using 4 other programs (Align-m [41], MUSCLE [42], Prob-Cons [43], and T-Coffee [44]) 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 [36] 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 [36] 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 0.6224.

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. Figure 3 shows an example of the correlation between predicted similarity and true similarity, here giving 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.
To test the information entropy extension developed above, we calculated the alignment consensus entropy for samples of alignments from the StatAlign dataset. Figure 4 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. See Methods section for details on entropy calculation.
rightmost figure where the alignment samples were very different, and the middle figure closer to the median value of alignment sample similarity. For family 3 (where sample alignments had low diversity), we see that the alignment consensus entropy is comparative to the mean entropy of the individual samples. This is expected, as it indicates there is little uncertainty in the alignment. On the other hand, the high-diversity family has much higher alignment consensus entropy than for each individual sample. This is again expected, as the difference in entropies indicates a high uncertainty in alignment. In this way, we can incorporate alignment uncertainty into our understanding of comparative RNA secondary structure prediction.

**Predicting secondary structure accuracy**

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 is more reliable than the one currently reported by the PPfold. 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 F-scores. 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 probabilities). 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. Figure 5 shows the predicted F-score against the true F-score, for 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. These results seem to indicate that while alignment quality does affect structure prediction quality, the actual structure prediction model still plays a great role in the overall prediction accuracy. Consequently, improving these models, possibly by incorporating other kinds of information (such as experimental probing data), is an area where new research efforts are still needed in RNA secondary structure prediction.

**Conclusions**

In this paper we have explored a number of factors which can contribute to poor RNA secondary structure prediction quality. We established a relationship between alignment quality and expected loss of accuracy.

![Figure 5 Predicted RNA secondary structure accuracy](http://www.biomedcentral.com/1471-2105/14/149)

**Figure 5** Predicted RNA secondary structure accuracy. Predicted F-score of PPfold on MPD alignment against actual F-score of PPfold on MPD alignment. The adjusted R-squared value for a linear fit is 0.496.
Appendix A. Quantifying variances in comparative RNA secondary structure prediction

Furthermore, we provided a method to predict alignment quality based only on statistical alignment scores. While our predictor of accuracy improves on the PfPfold reliability score, there is still a large amount of variability unaccounted for, which we therefore suggest comes from the predictive model itself. To consider this further, we extended the information entropy measure for SCFGs to consider uncertainty in alignments.

The fact that our accuracy predictor did not account for all the variances associated with RNA secondary structure prediction, despite good predictors being found for alignment quality and a strong correlation between alignment quality and predictive accuracy, suggests that whilst alignment quality is an important factor, the predictive model itself determines a large part in the quality of prediction. Given what is shown in Figure 1 for single sequence predictions, that the accuracy of PfPfold and RNAfold is very variable, it is unsurprising that variances remain. Clearly then, further efforts should be put into creating stronger single-sequence models, and then the advantages of evolutionary modelling and additional structural constraints will benefit further. The use of experimental data from new probing experiments as well as more biologically realistic constraints, such as kinetic or co-transcriptional folding, may improve the results of RNA secondary structure prediction.

Additional file

Additional file 1: Information on the 50 RNA families randomly selected from Irfam.

Competing interests

The authors declare they have no competing interests.

Authors’ contributions

JWA, AN, and ZS formulated the ideas presented here, carried out some of the analyses, and wrote the manuscript. AN generated the Salign and Random-Alignment datasets, and MJ, PA, and EC generated some of the jobs for the reliability scores, alignment distances, and predictive algorithm. JWA and ZS developed the theoretical framework for extending information entropy. All authors read and approved the final manuscript.

Acknowledgements

In part, this work was carried out as part of the Oxford Summer School in Computational Biology, 2012, in conjunction with the Department of Plant Sciences and the Department of Zoology. Funding was provided by the EU通过OCISB. JWJA would like to thank the EPSRC for funding.

Published: 1 May 2013

Received: 30 November 2012 Accepted: 21 March 2013

Page 12 of 13

References


doi:10.1186/1471-2105-14-149

Cite this article as: Anderson et al.: Quantifying variances in comparative RNA secondary structure prediction. BMC Bioinformatics 2013 14:149

Submit your next manuscript to BioMed Central and take full advantage of:

• Convenient online submission
• Thorough peer review
• No space constraints or color figure charges
• Immediate publication on acceptance
• Inclusion in PubMed, CAS, Scopus and Google Scholar
• Research which is freely available for redistribution

Submit your manuscript at www.biomedcentral.com/submit
B.1 Summary

This work was done in collaboration with IT University of Copenhagen and CLC Bio. Neelanarayanan Venkataraman and I have contributed equally.

The purpose of this study was to deploy the PPfold algorithm on a mini-grid of personal computers. This could have advantages over supercomputer clusters and specialized hardware, which require high technical knowledge and dedicated hardware and software resources to install, configure and maintain.

The Mini-Grid Framework is a runtime infrastructure and programming API enabling the creation of peer-to-peer and ad-hoc “mini-grids” in a local network environment. The main benefit is that it is ready to use for the end-user without any configuration or management overhead; a user submitting a job simply exploits the devices visible nearby at the moment. In addition, it is a symmetric infrastructure, and any participating user can submit jobs to the neighboring machines. This is in contrast to other projects e.g. “Folding@home”, which feature a “master-slave” architecture.

We experienced many challenges arising from taking a complex real-world bioinformatics algorithm and trying to deploy it on top of a non-shared memory, distributed, ad-hoc grid infrastructure. In this report, we describe how both the PPfold algorithm and the Mini-Grid Framework were tailored to each other to enable deployment.

The main problem was that the jobs generated by PPfold are extremely data-intensive, requiring extensive data transfer between different machines. To address this issue, we describe how we utilized the context-awareness feature of the Mini-
Summary

Grid framework to allow the algorithm to coordinate its task complexity with the Mini-Grid, and thereby allow for more optimal task distribution.

We investigated the real-time execution characteristics of this setup, and found that in some cases, we were able to achieve a speed-up when the algorithm was deployed on the Mini-Grid. In any case, despite the data-intensive nature of the algorithm, it did not perform worse on the Mini-Grid than when run locally. The experiments also demonstrated that even though grid infrastructures are normally designed to host the execution of the so-called Bag-of-Tasks (BoT) which are independent of each other, the context-awareness feature of the Mini-Grid allowed for complex task coordination between the algorithm and the grid.
Technical Report
RNA secondary structure prediction on an ad-hoc peer-to-peer network infrastructure
The Collaborative Mini-Grid Project
Zsuzsanna Sükösd∗, Neelanarayanan Venkataraman∗, Jakob E. Bardram, Tinus Abell, Morten Værum, Ebbe S. Andersen, Bjarne Knudsen, Jørgen Kjems
November 2011

Abstract
The computational complexity of bioinformatics algorithms is a significant obstacle in the analysis of the growing amounts of available biological data. This problem is increasingly addressed through multithreading technologies, but supercomputers and clusters are both expensive and require expert setup and maintenance. Ad-hoc grids, such as the Mini-Grid, provide an attractive alternative for the typical biology laboratory. Here we discuss the challenges of the deployment of a complex RNA secondary structure prediction algorithm on such a framework, and demonstrate that the performance of the algorithm on the Mini-Grid can be significantly improved by utilizing the context awareness feature for optimal communication between the algorithm and the Mini-Grid.

Keywords
ad-hoc peer-to-peer grid; context-awareness; bioinformatics; RNA secondary structure prediction; SCFG model; inside-outside algorithm

1 Introduction
In recent years, the amount of biological data in public databases has grown explosively, and this trend is expected to continue with the advent of the newest deep-sequencing technologies in biology. There is consequently an increasing need for fast bioinformatic methods to analyze biological data. In general, however, bioinformatics algorithms are computationally complex, and heuristic methods to speed them up are often inappropriate due to a reduction in the accuracy of analysis. Multithreading paradigms in computer science offer a convenient solution to this problem, and there have been several attempts to parallelize existing, computationally complex algorithms in bioinformatics, including algorithms for RNA secondary structure prediction. [9, 12, 10]

RNA secondary structure describes the pattern of “pairing” (hydrogen-bonding) between the bases of RNA, and is believed to be of utmost importance to the function of the RNA molecule in a biological setting. It is difficult to study experimentally, and its computational prediction from a single RNA sequence or a set (alignment) of sequences is a particularly challenging problem in bioinformatics. It is typically addressed through dynamic programming algorithms, based on a thermodynamic or a stochastic context-free grammar (SCFG) model for RNA folding. Both situations exhibit complicated dependencies between data points.

∗Authors ZS and NV contributed equally.
Even in the simplest case of non-pseudoknotted structure prediction, the execution time of state-of-the-art algorithms scale as \(O(n^3)\) or worse with the length of the sequence, \(n\). In this study, we focus on pfold ([5, 6]), an SCFG-based RNA secondary structure prediction algorithm that we have recently parallelized. (PPfold; [11]). This algorithm is computationally interesting as well as biologically useful, as it is based on the inside-outside algorithm for the parsing of a stochastic context-free grammar - this approach also finds applications in widely different contexts, including protein structure prediction and natural language processing.

Many efforts to parallelize RNA structure prediction algorithms have been directed towards supercomputer clusters [9, 10] or even specialized multithreaded hardware [12]. However, installing, configuring and customizing such solutions requires high technical knowledge and dedicated hardware and software resources. For these reasons, the deployment and operational costs of such systems are substantial, which prevents their adoption and direct use by biological researchers. Another option has been provided by volunteer computing systems, which allow the formation of parallel computing networks by enabling ordinary internet users to share their computer’s idle processing power [8, 1]. One example is the Folding@Home project [4]. However, such volunteer computing systems require a centralized control system responsible for managing the contributing clients, who in turn periodically request work from a central server. As such, volunteer computing is highly asymmetric; it is a ‘master-slave’ architecture in which volunteers supply computing resource but do not submit any work to be done. Public outreach and incentive structures (like high-score competitions) play a significant role in attracting volunteers.

In contrast, the Mini-Grid Framework [2] is a runtime infrastructure and programming API enabling the creation of peer-to-peer and ad-hoc “mini-grids” in a local network environment. Our current work creates support for bioinformatics analysis on the available resource infrastructure, such as the desktop and laptop PCs in a biology laboratory. The main benefit is that the “Mini-Grid” is ready to use for the end-user without any configuration or management overhead; a user submitting a job simply exploits the devices visible nearby at the moment [2]. In addition, the “Mini-Grid” is a symmetric infrastructure, and any participating user can submit jobs to the neighboring machines.

In this paper, we describe how the PPfold algorithm was tailored to and deployed on the Mini-Grid Framework. We discuss some of the technical challenges that arise from taking a complex real-world bioinformatics algorithm and trying to deploy it on top of a non-shared memory, distributed, ad-hoc grid infrastructure. We describe also how we were able to utilize the context-awareness feature of the Mini-Grid framework to allow the algorithm to coordinate its task complexity with the Mini-Grid, and thereby allow for more optimal task distribution. We report a detailed experiment that investigated the real-time execution characteristics of this setup, and compare it to the theoretically best achievable results.

This experiment shows that despite the complex nature of the PPfold algorithm, we were able to achieve a speed-up improvement when deployed on the Mini-Grid; in any case, the algorithm did not perform worse than when run locally, and in parts of the algorithm the distribution and parallelization provided by the Mini-Grid speeded up the calculations. These results may sound conservative, but the experiments demonstrated that even though grid infrastructures are normally designed to host the execution of the so-called Bag-of-Tasks (BoT) which are independent of each other, the context-awareness feature of the Mini-Grid allowed for complex task coordination between the algorithm and the grid.

Subsection 2 provides an overview of the PPfold algorithm and its parallelization; subsection 3 describes the main characteristics of the Mini-Grid Framework; and in subsection 4, we detail some of the challenges that arise when trying to combine the PPfold algorithm with the Mini-Grid, and our solutions to them. Subsection 5 describes our experiment and the obtained results, which is discussed in subsection 6. Subsection 7 concludes the paper.

2 The PPfold algorithm

The PPfold algorithm is a recently developed, parallelized version of pfold. Here we provide a short summary of the model and the algorithm. For more details, readers are directed to references [5, 6, 11]. Pfold is a single-threaded algorithm for predicting RNA secondary structure. It couples a phylogenetic model to a stochastic context-free grammar (SCFG), and obtains high-quality predictions for the consensus structure of alignments of RNA sequences in \(O(n^3)\) time and \(O(n^2)\) space, where \(n\) is the length of the alignment. The
Table 1: The general characteristics of the jobs generated by the PPfold algorithm. There are large variations even within the same group. The simultaneous job generation pattern means that all jobs of that type are generated at the same time, and thus all are available from the start of execution. The asynchronous job generation pattern means that the jobs are generated continuously at different times, and the number of jobs ready for execution can decrease (inside/EM algorithms) or increase (outside algorithm) with time. EM algorithm: Expectation value maximization algorithm.

<table>
<thead>
<tr>
<th>Job type</th>
<th>Time complexity</th>
<th>Dependencies per job</th>
<th>Data required</th>
<th>Computational intensiveness</th>
<th>Generation pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phylogenetic</td>
<td>(O(n^2))</td>
<td>none</td>
<td>small</td>
<td>high</td>
<td>simultaneous</td>
</tr>
<tr>
<td>SCFG: Inside alg.</td>
<td>(O(n^3))</td>
<td>complex</td>
<td>small/ moderate</td>
<td>low/ moderate</td>
<td>asynchronous</td>
</tr>
<tr>
<td>SCFG: Outside alg.</td>
<td>(O(n^3))</td>
<td>complex</td>
<td>moderate/ large</td>
<td>moderate/ high</td>
<td>asynchronous</td>
</tr>
<tr>
<td>SCFG: EM alg.</td>
<td>(O(n^3))</td>
<td>complex</td>
<td>small/ moderate</td>
<td>low</td>
<td>asynchronous</td>
</tr>
</tbody>
</table>

algorithm consists of two time-consuming parts: the phylogenetic calculations, which are column-based and involve post-order traversal through an evolutionary tree, and the inside-outside algorithm, which is a dynamic programming algorithm first described in the context of natural language modelling using SCFGs[7]. In pfold, this is followed by an “expectation value calculation” algorithm that is very similar to the inside algorithm. We will be referring to this as the “EM algorithm” for brevity.

The time complexity of the phylogenetic calculations is \(O(n^2)\) and their parallelization is straightforward, as the columns of the input alignment are considered to be independent of each other. The inside-outside and EM algorithms, on the other hand, exhibit heavy dependencies and have a time complexity of \(O(n^3)\). In PPfold, we have parallelized both the phylogenetic and the inside-outside/EM algorithms. The details of the implementation are described in [11], but here we reproduce the geometry of dependencies of jobs on each other in Figure 1 for reference. The generation of jobs happens in an asynchronous wavefront; a new job is generated and queued for immediate execution as soon as its dependencies are completed, which precludes extra waiting time.

Figure 1: The dependencies of jobs on each other in the inside-outside algorithm, as implemented in PPfold. Each parallelogram represents a job, which includes a number of points to be calculated. The number of divisions in the first row is a parameter of the algorithm. The size of each job depends on the number of divisions. The dependencies in the EM algorithm are analogous to the inside algorithm. (Figure from [11])

For the purposes of this study, we consider PPfold to be an example of a general algorithm that generates “jobs” of different characteristics at uneven intervals (see Table 1 for a summary of PPfold job characteristics).

Each job is sent to an executor interface, which may be local execution in one thread, local execution in

Appendix B. RNA secondary structure prediction on an ad-hoc peer-to-peer network infrastructure
several threads, or the Mini-Grid environment. (Figure 2)

In [11], we describe that the algorithm scales very well with additional cores on a single multi-core machine when using the multithreaded local executor. In this paper, we discuss performance considerations when the executor interface is the Mini-Grid framework.

3 The Mini-Grid Framework

The Mini-Grid framework [2] is a runtime infrastructure and programming framework for the formation of ad-hoc ‘mini-grids’ in a local area network (LAN). This is illustrated in Figure 3. The Mini-Grid follows the general vision of a desktop volunteer grid computing system, aiming at turning a set of regular personal computers into a runtime environment for job distribution and execution. The Mini-Grid aggregates resources such as servers and desktop PCs connected in a local area network. In contrast to other desktop grids, however, the Mini-Grid is designed to also support laptop computers connected through the wireless LAN (WLAN) and the Mini-Grid framework hence supports the creation of a volunteer grid infrastructure using more unstable devices that leave and re-enter the grid on an ad-hoc basis. Moreover, in contrast to existing volunteer computing infrastructures that use a master-slave architecture, the Mini-Grid is symmetric by design. By using a peer-to-peer architecture, the Mini-Grid allows all participating devices to both consume resources by submitting jobs to the grid, as well as donate resources to the grid by executing jobs.

A conceptual illustration of the Mini-Grid Framework is shown in Figure 4. The Mini-Grid Framework consists of four logical components: Submitter, Executor, TaskBus and MessengerComponent. As a distributed computing platform, the framework allows Mini-Grid applications to submit jobs from the computer acting as Submitter, to be executed on computers acting as Executors. Note that a physical device (like a laptop computer) can act as both a Submitter and an Executor, in this way enabling symmetry of the grid.

The distribution of jobs in the Mini-Grid works as follows. First, a Submitter accepts a set of tasks from a local Mini-Grid application. Then the Submitter distributes these tasks to the Executor computers according to a scheduling policy, which in the current version of the Mini-Grid uses an auctioning process. Then, the tasks are sent to the Executors who were selected to execute them. Once the Executor has finished the execution of its task(s), it notifies the Submitter, which then in turn contacts the Executor to collect the result.

The framework uses the auctioning mechanism for resource discovery (identifying the Executors participating in the Mini-Grid), as well as for finding the best Executor for the execution of a particular task. The latter is based on matching the requirements stated by the application submitting the task and the capabilities of the Executor. For doing this, the Mini-Grid is “context-aware” [3] in the sense that each Executor is aware of its execution context, and the Submitter uses this information in the auctioning process. Context information is used to express both the resource requirements of an application and the resource characteristics of an Executor. For example, the application can state that it expect a certain amount of memory for the execution of a job, and when bidding for jobs an Executor will report its amount of (free) memory. Based on this auctioning process, the submitting machine can determine which Executor machines are suitable candidates for executing a job.
for receiving the jobs. The information required for the execution of a job is advertised using the Messenger Component. Only the Executors that have the requested context can participate in the auctioning process. The Executors calculate their bid value using a utility function based on the advertised bidding strategy, such as fastest completion of the task. The Submitter selects the Executor that has submitted the bid of highest utility value.

### 4 Distributing PPfold on the Mini-Grid

The performance of an algorithm in a Mini-Grid-like environment depends on a number of factors, even if the algorithm scales well on a supercomputer.

In our initial deployment of PPfold on the Mini-Grid, we did not utilize context-awareness, so all jobs generated by the algorithm were distributed directly on the Mini-Grid as soon as they were generated. We observed that the performance of the algorithm on the Mini-Grid was very poor: in many cases, the time taken to obtain a result on the Mini-Grid was orders of magnitude longer than the time taken to produce the same result when executing the algorithm locally in one thread on the Submitter. In addition, in several tests, many Executor computers were rendered useless because they lacked the essential features (notably memory) required to execute the jobs they received. We concluded that the poor performance was due to the following factors:

- **Memory requirements**: Some jobs required the storage of very large amounts of data, which overloaded Executors without sufficient memory. The memory resources of each Executor were not considered in the job distribution process.
- **Computational intensiveness**: Some jobs required very large amounts of data to be transferred to the Executor computers, without sufficient computational time spent on the Executor. Sending a job to an Executor is only advantageous if the job is highly computationally intensive, i.e. if the time needed to transfer data between the Executor and the Submitter is small compared to the time needed to execute the job.
- **Theoretical speedups**: An analysis of the algorithm revealed that even in theory, the distributed algorithm does not scale well with additional machines, despite scaling on a multicore shared-memory infrastructure.
architecture is near-perfect. However, this can be improved significantly by an appropriate choice of the parameters of the algorithm (number of divisions, see Figure 1).

In the following, we describe each of these issues in more detail and show how we have addressed them by utilizing the context-awareness of the Mini-Grid, in order to produce runtimes on the Mini-Grid that are always at least as fast as local execution, despite the complexity of the distributed algorithm.

4.1 Memory Requirements

PPfold, like many bioinformatics algorithms, is extremely memory-intensive. For the folding of a large RNA, such as the HIV genome of 10000 bases, 5-6 GB of memory are required. This places extreme requirements on the capabilities of the Submitter node in the Mini-Grid, as this machine will have to store most sub-results until the end of the calculations, because of the dependencies of the inside-outside algorithm. As this is an inescapable feature of the algorithm (whether it is run locally or distributed), in our deployment we assumed that the Submitter node was in all cases capable of storing all the data.

On the other hand, the memory requirements on Executor machines are not so high, as each Executor only needs to store the data that is necessary for the calculation of one job. As described in [11], it is desirable to choose a high number of divisions and thus generate many jobs, in part because this reduces the job size and the amount of data that needs to be transported and stored on an Executor. However, in our implementation the number of divisions is a parameter of the algorithm (and thus the Mini-Grid is not able to influence it), and at a small number of divisions, even a reduced memory requirement can place demands that an Executor cannot meet.

To avoid overloading Executors, we utilized the context-awareness feature of the Mini-Grid framework. For each task submitted to the Mini-Grid framework, the algorithm specified how much memory is expected to be used in order to execute the calculations. The Submitter announced the memory requirement as context information attached to the task. In this way, only resources with sufficient memory could participate in the bidding process.
4.2 Computational Intensiveness

A significant characteristic of the Mini-Grid environment is its inherent dependence on data transport through the network. This effect is absent in the case of supercomputers and it poses additional challenges for deploying complex algorithms such as PPfold on the Mini-Grid.

As noted in subsection 4.1, it is desirable to generate many small jobs rather than few large jobs. However, if the amount of time taken to transfer data for the execution of a job is comparable to the amount of time taken to execute that job, the performance of the algorithm will be poor; in fact, it will in many cases run slower on the distributed infrastructure than it would on a single machine. Moreover, the time it takes to execute each job in the algorithm can vary significantly; the first jobs in the EM algorithm might take several orders of magnitude less time than the last jobs in the outside algorithm. The size of the data to be transferred as part of a job also differs significantly from job to job.

To mitigate these effects, we designed a sophisticated data distribution mechanism based on the context-awareness feature of the Mini-Grid. When submitting jobs, the algorithm specified a rough estimate of the time required to execute it, together with an estimate of the amount of data transport needed for its remote execution. The Mini-Grid framework then calculated the ‘computational intensiveness index’ of the job, which is the ratio between the execution time of the job and the time take for data transport:

\[ I_{\text{intensiveness}} = \frac{T_{\text{execution}}}{T_{\text{transfer}}} \]

If the computational intensiveness index of a job falls below a threshold value, the framework scheduled the task locally. Otherwise, it is distributed to the Mini-Grid environment. This is illustrated in Figure 5.

![Flowchart illustrating the deployment of PPfold (as an example of a general algorithm) on the Mini-Grid, including support for job context information (cf. Figure 2, which does not include context information). The Mini-Grid rejects jobs with unsuitable computational intensiveness (calculated from the job’s context information), thus redirecting them for local execution. In addition, the Mini-Grid ensures that only Executors with sufficient memory receive any jobs with large memory requirements.](image)

Based on performance tests during our trial run, a threshold value for the computational intensiveness index of 3 turned out to be the best value for our technical setup. At the moment, this computational intensiveness index has to be calibrated for a specific technical setup. In the future, however, it should be possible to have the framework adjust this index dynamically. For our experiments, we used the utilization function based on the fastest completion of the job. We modeled the bid using the clock frequency of the executing host and its current workload (measured by the time taken by the host machine to respond).

4.3 Theoretical Speedup

Finally, it is worth noting that even in theory, the distributed algorithm might not scale well with an additional number of machines, despite scaling on a multicore shared-memory architecture is near-perfect [11]. We illustrate this phenomena in Figure 6, using the inside algorithm in PPfold as an example. Figure 6 shows that each job is modeled to take a certain number of (relative) “units of time” for execution, which depends...
linearly on its height in the job-triangle. Jobs further up in the triangle take a longer time to execute. Jobs in each row are assumed to be executed simultaneously if additional cores are available. The theoretical speedup for \( n_G \) cores in the Mini-Grid compared to \( n_S \) cores on the Submitter is expressed as:

\[
A_n = \frac{\text{total units of time on } n_G \text{ cores in Mini-Grid}}{\text{total units of time on } n_S \text{ cores on Submitter}}
\]

The theoretical speedup can be calculated in a similar manner for any number of divisions, for the inside as well as the outside and EM algorithms. Figure 7 illustrates the theoretical maximum speedups for execution on a 2-core Submitter machine versus distributed on a Min-Grid of several more cores, as a function of divisions for the different parts of the algorithm. This model assumes that all jobs are distributed all the time with zero data transfer, as in a shared-memory environment. This model is not fully descriptive of the Mini-Grid, which is a non-deterministic execution environment with significant time spent in data transfer. Real-world performance when using the Mini-Grid is therefore expected to be worse than what is theoretically possible.

Figure 6: The dynamic programming algorithm in PPfold can not be fully parallelized. The smaller the number of divisions, the greater the number of jobs that cannot be executed in parallel. In our implementation, the unparallelized jobs are also the slowest in the inside algorithm and the expectation value calculations. Here we illustrate the effect of this on the speedups that can theoretically be obtained at all on a varying number of cores, for the inside algorithm only, using 6 divisions (corresponding to the 6 jobs in the bottom row). With a single additional core, the algorithm is expected to complete in 62.5% of the time taken when executed on one core. With 4 additional cores, the algorithm is expected to complete in 42.6% of the time taken when executed on one core.

5 Experiments – Distributing PPfold on the Mini-Grid

In order to evaluate the performance of the PPfold algorithm as implemented on top of the Mini-Grid, we made a series of experiments in a laboratory setup. Individual performance evaluations of the PPfold algorithm [11] and the Mini-Grid framework [2] have been done separately. Therefore, the focus of this study was to see, on the one hand, how the PPfold parallelization would behave when using a non-shared memory and distributed execution environment, and on the other, how the Mini-Grid environment would perform when non-trivial, real-world bioinformatics tasks were submitted to it. In particular, the focus was to evaluate how the mechanism of providing task context information from the PPfold algorithm to the Mini-Grid was improving the speed of distribution and execution of jobs.
Figure 7: The theoretical optimal speedups vary wildly with the chosen number of divisions ($N$) and the available number of cores in the Submitter ($n_S$) and all of the Mini-Grid ($n_G$), for both the inside/EM and the outside algorithms. The periodicity in the figures is related to the divisibility of the number of cores with the number of chosen divisions. In the limit where $N \to \infty$, the factor speedup approaches $n_G n_S$. The values in this figure were calculated using the method shown in Figure 6. a) Assuming perfect parallelization, the comparison of the theoretically obtainable speedup when moving from a 2-core machine executing jobs locally to a 6-core Mini-Grid setup (3 machines each with 2 cores), as a function of the number divisions. b) Assuming perfect parallelization, the comparison of the theoretically obtainable speedup when moving from a 2-core machine executing jobs locally to a 12-core Mini-Grid setup (6 machines each with 2 cores), as a function of the number divisions.

5.1 Experimental setup

In our study, the Mini-Grid consisted of a set of six heterogeneous desktop PCs having clock frequencies ranging from 1.1GHz to 2.1GHz, and physical memory ranging from 1GB to 2GB. These desktop PCs were connected through a wired gigabit ethernet and were connected to the same physical switch. For simplicity reasons, we configured one device to act as a Submitter and the rest as Executors, even though every device can act as both an Executor and a Submitter. Dedicated access to desktop PCs was obtained for the duration of the experiments.

It is not possible to exhaustively test the performance of PPfold on the Mini-Grid under all circumstances. We have therefore restricted ourselves to a single biological problem, namely the folding of an alignment of 3,000 nucleotides in length. This alignment represents a sequence that is somewhat longer than the large ribosomal subunit, and thus presents a realistic scenario of a problem that a biologist could solve on their own laptop, but would probably prefer to speed up with the aid of the Mini-Grid. Significantly smaller alignments can be solved sufficiently fast on the local machine, and distribution in the mini-grid would not be attractive in this case. Significantly larger alignments would require so much memory on the submitting host that it would be prohibitive for a normal PC to act as a Submitter.

5.2 Experiments

Table 2 shows the parameters used in our controlled experiments. These parameters are chosen at the algorithm level, and the Mini-Grid infrastructure has no influence on them. Note that the number of divisions in the SCFG-based parts (inside, outside and EM algorithms) is the same in each test. However, the phylogenetic part of the algorithm is independent of the SCFG-based parts, has a different job distribution parameter, and its runtime can also be considered separately from the runtime of the SCFG-based parts in all tests.
Table 2: Experimental details – In total 6 experiments were executed, varying the number of jobs in the phylogenetic calculations ($n_P$), the number of divisions in the SCFG-based parts of the algorithm ($n_S$), and the number of cores ($N$) used in the Mini-Grid environment. The Submitter machine always had 2 cores.

<table>
<thead>
<tr>
<th>#</th>
<th>$n_P$</th>
<th>$n_S$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>14</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
<td>24</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 3: Execution time in seconds on a single machine with 2 cores for the six experiments. The table shows the execution time for the phylogenetic ($T_{phylo}$), the inside ($T_{inside}$), the outside ($T_{outside}$), and the expectation ($T_{exp}$) part of the algorithm.

<table>
<thead>
<tr>
<th>#</th>
<th>$T_{phylo}$</th>
<th>$T_{inside}$</th>
<th>$T_{outside}$</th>
<th>$T_{exp}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>386.5</td>
<td>668.4</td>
<td>1406.3</td>
<td>188.9</td>
</tr>
<tr>
<td>2</td>
<td>408.7</td>
<td>623.2</td>
<td>1296.7</td>
<td>175.3</td>
</tr>
<tr>
<td>3</td>
<td>468.2</td>
<td>678.2</td>
<td>1482.3</td>
<td>201.5</td>
</tr>
<tr>
<td>4</td>
<td>415.2</td>
<td>592.1</td>
<td>1253.7</td>
<td>169.0</td>
</tr>
<tr>
<td>5</td>
<td>415.2</td>
<td>592.1</td>
<td>1253.7</td>
<td>169.0</td>
</tr>
<tr>
<td>6</td>
<td>375.2</td>
<td>591.3</td>
<td>1231.1</td>
<td>164.3</td>
</tr>
</tbody>
</table>

Table 4: Execution time in seconds on the Mini-Grid for the six experiments. As in Table 3, this table shows the execution time for the phylogenetic ($T_{phylo}$), the inside ($T_{inside}$), the outside ($T_{outside}$), and the expectation ($T_{exp}$) part of the algorithm.

<table>
<thead>
<tr>
<th>#</th>
<th>$T_{phylo}$</th>
<th>$T_{inside}$</th>
<th>$T_{outside}$</th>
<th>$T_{exp}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>160.0</td>
<td>413.0</td>
<td>612.2</td>
<td>176.4</td>
</tr>
<tr>
<td>2</td>
<td>197.1</td>
<td>371.0</td>
<td>545.9</td>
<td>148.2</td>
</tr>
<tr>
<td>3</td>
<td>263.7</td>
<td>522.9</td>
<td>766.3</td>
<td>178.5</td>
</tr>
<tr>
<td>4</td>
<td>240.0</td>
<td>358.4</td>
<td>550.7</td>
<td>155.0</td>
</tr>
<tr>
<td>5</td>
<td>97.2</td>
<td>349.7</td>
<td>415.6</td>
<td>145.9</td>
</tr>
<tr>
<td>6</td>
<td>85.7</td>
<td>332.4</td>
<td>354.3</td>
<td>168.7</td>
</tr>
</tbody>
</table>

5.3 Results

For each of these experiments, we first ran the algorithm on a single machine with a dual-core CPU. These results are provided in Table 3. We then ran the algorithm on the Mini-Grid, with the specified number of cores. These results are provided in Table 4. The ratio of the corresponding numbers in these two tables is the experimentally observed speed-up when using the Mini-Grid instead of local execution in multiple threads on the Submitter.

In Figure 8, we compare the speedups achieved in the Mini-Grid environment with the theoretical maximal speedup we could have obtained if the Mini-Grid had been a shared-memory architecture (without need for data transfer). The theoretical speedups were calculated as shown in Figures 6 and 7.

The first thing to note from our results is that the speed-up factor is always greater than 1 (with one exception, when it is not significantly less than 1). This means that the algorithm always ran as fast as, or faster, on the Mini-Grid compared to running locally on the dual core CPU of the submitting computer. This is a significant improvement from the non-context-aware version of the deployment, where each of these tests ran slower on the Mini-Grid than on the Submitter.

The second thing to note from the results is that the experimentally observed speed-ups are lower than the
Figure 8: The experimental results of the 6 tests (see Table 2) compared to the theoretical maximum speedups. Factor speedup is measured as the ratio of the amount of time spent when executing locally on the Submitter to the amount of time spent when executing on the Mini-Grid. The phylogenetic part is independent of the SCFG-part (inside, outside and EM algorithms). P: phylogenetic algorithm, I: inside algorithm, O: outside algorithm, E: EM algorithm.

theoretically possible ones. This is because the theoretical calculations underestimate the time needed by the Mini-Grid, as they do not consider any overhead in the Mini-Grid due to bidding mechanisms, data transfer or redirection of jobs with low computational intensiveness. The latter factor is especially important in the case of the EM algorithm, where the amount of required data transfer is so high in relation to the execution time of each job that the majority of the jobs were returned for local execution in all our tests. This is confirmed experimentally since this part of the algorithm shows the worst speed-up factors. The phylogenetic part, in contrast, scales very well with the availability of additional cores, given that enough jobs are produced to utilize all available cores (test 5 and 6). This is because the jobs produced in this part of the algorithm are highly computationally intensive, “Bag-of-Tasks” jobs. The Mini-Grid framework performs better when the individual tasks of the application have higher computational complexity, because this balances the time needed for executing the calculation of a task with the time needed to auction, distribute, and recollect the task.

In summary, our data show that all individual parts of the algorithm run at least as fast on the Mini-Grid as when executing on a single machine. This is achieved by the context-awareness exchanged between the Mini-Grid environment and the algorithm. The parts of the algorithm that exhibit a high computational intensiveness scale very well with additional cores in the Mini-Grid, and the parts with a low computational intensiveness do not run slower than when executing locally on the Submitter machine. Nonetheless, the overall performance of PPfold does not scale as well on the Mini-Grid as it would on a shared-memory infrastructure, because of the complex dependency relationships and the need for extensive data transfer in the algorithm. This was, however, expected since a non-shared and distributed environment like the Mini-Grid would never achieve this.

6 Discussion

The Mini-Grid has been designed as a highly “programmer-friendly” environment: from the algorithm developer’s perspective, a non-shared memory design is enough to deploy any distributed algorithm to the Mini-Grid. This means that the algorithm developer does not need to consider the number of nodes or distribution mechanisms when deploying an algorithm to the Mini-Grid. Due to its context-aware task distribution mechanism, the Mini-Grid can make intelligent distribution decisions based on comparing requirements of each job with the capabilities of the executing host devices. However, this loose coupling and high degree of flexibility comes with the disadvantage that the algorithm developer can only optimize the algorithm to
a certain degree, which again means that the performance of the algorithm may not be optimal when executed on the Mini-Grid. The deployment of PPfold on the Mini-Grid shows this very clearly; the application developer who chooses the number of divisions is not aware of the number of available nodes or their performance characteristics. This can easily result in a situation where even the theoretical optimal speedup is rather low. Currently, the Mini-Grid framework do not have any real-time awareness of the size of the grid and its capacity. But if such a feature was added to the framework, this could be taken into account by the application developer. However, we expect the realistic Mini-Grid environments to be highly dynamic, which would make it hard to predict how many machines will be available at any given point in time.

From the algorithm side, we remark that dynamic programming algorithms such as the inside-outside algorithm are particularly difficult to parallelize, even for shared memory environments. And for a non-shared memory environment, a number of additional challenges arise due to the complex dependencies and need for large amounts of data transfer. However, we expect that increasing the computational complexity of each job for the same amount of data transfer would result in better performance characteristics on the Mini-Grid. In practice, this could for example occur if one added extra nonterminals to the stochastic context-free grammar model, and thus increased the time complexity of every job in the inside-outside algorithm. This is significant, because a more complex model for RNA structure folding involving more nonterminal symbols would potentially yield more accurate results.

7 Conclusions

In this paper, we have described the challenges and solutions of deploying a complex bioinformatics algorithm on top of a desktop grid infrastructure. The PPfold algorithm is a bioinformatics application for the prediction of RNA secondary structure which has complex requirements regarding parallelization, memory usage, and computational intensiveness. The Mini-Grid is an ad-hoc, symmetric desktop grid infrastructure, mainly designed to support the execution of independent, so-called Bag-of-Tasks jobs, but this fits poorly with the requirements of the PPfold algorithm. The performance of the algorithm on the Mini-Grid was initially unacceptable, despite excellent performance characteristics and scaling of both the algorithm on a shared-memory environment and the grid infrastructure with highly computationally intensive jobs.

We identified several factors that contributed to this, and solved them using an intelligent context-awareness system through which the algorithm and the infrastructure communicate, enabling the intelligent distribution of the appropriate jobs to the appropriate Executors. As a result of this new feature, we found that the algorithm never ran slower on the Mini-Grid than when executing the same jobs locally, and in some cases significant speed-ups were obtained. Our results demonstrate more generally that context-awareness in ad-hoc network infrastructures enables an improved performance of complex algorithms.

Acknowledgments

This research has been funded by the Danish Strategic Research Council under the project “PC Mini-Grids for Prediction of Viral RNA Structure and Evolution”, #09-061856.

References

Appendix B. RNA secondary structure prediction on an ad-hoc peer-to-peer network infrastructure


Appendix

C

An $n$-free-passes CYK algorithm for error-correction and the prediction of non-canonical base-pairs in RNA secondary structure

C.1 Summary

This work was done in collaboration with James Anderson at Oxford University, and we contributed equally.

The prediction of non-canonical basepairs in RNA secondary structure prediction has become increasingly important with the advent of next-generation sequencing technologies, where sequencing errors can introduce artificial non-canonical basepairs in RNA secondary structure. These basepairs are not appropriately accounted for by the currently existing models.

In this study, we focused on SCFG-based RNA secondary structure prediction, and introduced an “$n$-free-passes” variant of the CYK algorithm, which allows a fixed maximum number of basepairs to be predicted with a probability from a different distribution than the original grammatical model. The algorithm introduces an additional coordinate in the dynamic programming tables, resulting in an increased time complexity of $O(n \times k^3)$ and space complexity of $O(n \times k^2)$, where $k$ is the length of the input sequence, and $n$ is the number of free passes allowed. The algorithm was implemented and tested in an unpublished version of PPfold. Our results showed that this algorithm improved the prediction of artificially introduced non-canonical basepairs in RNA sequence, but it did not improve the prediction of naturally occurring non-canonical basepairs. We also identified other shortcomings in the algorithm.

Firstly, when allowing free-passes, the “probability” of a given secondary structure derivation will be greater than or equal to the regular CYK case, and so the
probability distribution over secondary structures will not be normalised, and the "probabilities" of derivations are in fact pseudo-probabilities.

Secondly, it is not possible to implement an "n-free-passes" version of the inside-outside algorithm in practice: for a single free-pass, the inside-outside algorithm would require summing probability contributions from multiple non-canonical base-pairs. If only a single nucleotide or basepair were of interest, the inside-outside algorithm could be used to sum over all possible predictions that can happen outside of that nucleotide or basepair. But with an entire structure prediction, the sum would not be over the correct distribution.

Lastly, an extension to alignments is not obvious. In the case of alignments, nucleotide likelihoods are replaced with alignment likelihoods, which are derived from an evolutionary model. One would not expect a whole alignment column to be erroneous, but only a single entry in that column. As a result, the Felsenstein algorithm would also have to be modified in addition to the CYK algorithm to allow a fixed number of free-passes in this framework.
Technical Report: An \textit{n}-free-passes CYK algorithm for error-correction and the prediction of non-canonical base-pairs in RNA secondary structure

James W. J. Anderson, Zsuzsanna S"uk"osd, Christian N. S. Pedersen, and Jotun Hein

April 5, 2013

Abstract

\textbf{Background:} The prediction of non-canonical base-pairs in RNA secondary structure prediction has become increasingly important with the advent of next-generation sequencing technologies, where sequencing errors can introduce artificial non-canonical base-pairs in RNA secondary structure. These base-pairs are not appropriately accounted for by the currently existing models.

\textbf{Results:} Here we focus on SCFG-based RNA secondary structure prediction, and introduce an \textit{n}-free-passes CYK algorithm, which allows a fixed maximum number of base-pairs to be predicted with a probability from a different distribution than the original grammatical model. Our results show that the \textit{n}-free-passes algorithm improves the prediction of artificial non-canonical base-pairs in RNA sequence, even though it does not improve the prediction of naturally occurring non-canonical base-pairs.

\textbf{Conclusions:} The \textit{n}-free-passes CYK algorithm is a novel approach to address the problem of predicting non-canonical base-pairs that occur artificially due to sequencing errors. The implementation in PPfold and its source code are available from the authors on request.

\textbf{Keywords:} RNA secondary structure, SCFG, non-canonical base-pairs
An $\mathit{n}$-free-passes CYK algorithm

1 Background

Accurate RNA secondary structure remains a challenging problem in computational biology. The most commonly used programs use either a thermodynamic model (Markham et al. 2008, Hofacker et al. 1994) or stochastic context-free grammars (SCFGs) (Knudsen & Hein 2003), and implement a dynamic programming algorithm. A review of RNA secondary structure prediction can be found in (Gardner & Giegerich 2004).

With the increasing availability of next-generation sequencing technologies, it is desirable to look for increasingly robust algorithms in bioinformatics. Sequencing errors occur at a relatively high rate: Roche 454 pyrosequencing has, for example, been reported to have error rates of up to 10%. Even though error sequencing correction methods do exist, they do not typically consider the structural conservation of RNA and random sequencing errors can "mutate" canonical base-pairs into non-canonical ones in an unpredictable fashion. Similarly, RNA secondary structure prediction methods usually do not appropriately address sequencing errors or non-canonical base-pairs. Thermodynamic models do not predict non-canonical base-pairs at all because of lack of thermodynamic data, and stochastic context-free grammar based models only predict non-canonical base-pairs with very low probabilities. Because of global dependencies in RNA secondary structure, a single error can create large-scale changes in the underlying model probability distributions.

A recent study (Reinharz et al. 2013) has shown a remarkable ability to predict sequencing errors if the secondary structure is known. By considering a modified Boltzmann distribution (McCaskill 1990) and summing over all possible sequence mutations, the sites of most likely sequencing errors could be identified. Unfortunately, homologous structures are not always available (Engelen & Tahi 2007), and neither the position nor the number of sequencing errors is typically known in advance. It is therefore interesting to investigate the behaviour of the method when we allow a fixed maximum number of "free" base-pairs, without knowing their exact location, to possibly correct for errors.

In this work, we investigate the effect of introducing an $\mathit{n}$-free-passes algorithm in a stochastic context-free grammar-based method. The method is implemented for single sequence prediction in PPfold (Sukosd et al. 2012). A free-pass allows the production of single base-pair with a higher probability than would otherwise be indicated from the training of the SCFG for the actual nucleotides occurring in the sequence. This allows the model to freely choose a low-probability base-pair, and replace it with a high-probability one, in order to maximize the probability of the overall structure prediction. If the correct non-canonical base-pair is chosen, we can expect that the accuracy of the overall prediction will increase. The $\mathit{n}$-free-passes algorithm can be interpreted in the light of sequencing error correction. Rather than changing the probabilities used in the SCFG, the algorithm allows $\mathit{n}$ coupled changes in the sequence, such that two bases that could otherwise form a low-probability base-pair are replaced with "neutral" bases that can form a base-pair with a higher probability.

2 Methods

Stochastic Context-Free Grammars

Context-free grammars are part of the Chomsky hierarchy (Chomsky 1956) and can be defined as a four-tuple $(\mathit{V}, \Sigma, P, \mathit{S})$, containing $\mathit{V}$, a set of non-terminal variables, $\Sigma$, a set of terminal variables (usually the alphabet to generate sequences over, e.g. \{A, C, G, U\}), $P$, a set of production rules, and $\mathit{S} \in \mathit{V}$, a unique start symbol. Starting from the start symbol, one applies the production rules until left only with a string of terminal variables. This sequence of production rules is known as a derivation of a grammar. The stochastic context-free grammar can be obtained from a context-free grammar by associating probabilities with each production rule. In this case, the probability of a derivation is simply the product of the probabilities of the rules used in it.

To work with algorithms designed for context-free grammars, it is useful to restrict the statement of the grammar to a normal form. Often Chomsky Normal Form is used, but for RNA secondary structure prediction, it is convenient to use double-emission normal form (Anderson et al. 2012).
An n-free-passes CYK algorithm

This normal form allows production rules of the type \( T \to UV, T \to \ldots \), and \( T \to (U) \), where \( T, U, V \) are non-terminal symbols and \( (,), \in \{A,C,G,U\} \) representing paired and unpaired nucleotides. The advantage of double-emission normal form in RNA secondary structure modelling is that it allows for the formation of unpaired and paired bases, whereas Chomsky Normal Form can only output a single terminal variable. It is worth noting that all SCFGs producing valid RNA secondary structures can be written in double-emission normal form. Consequently, algorithms here will be presented for SCFGs in double-emission normal form, however, they can be easily adapted for SCFGs not in this form.

### The CYK algorithm

Given an RNA sequence, one approach for identifying a single “best” secondary structure is the CYK algorithm (Younger 1967), which predicts the structure with maximum probability. The CYK algorithm is also a dynamic programming algorithm, which is based on the following recursion relations. Given non-terminal \( U \), and sequence indices \( i \) and \( j \), we compute

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

where, for a sequence \( s \) with entries \( s[1], \ldots, s[k] \), \( \text{P}_u[s[i]] \) is the probability of \( s[i] \) being unpaired, and \( \text{P}_a[s[i], s[j]] \) the probability of \( s[i] \) and \( s[j] \) being paired. \( C(S, 1, k) \) then gives the highest probability of a derivation, and by backtracking through the CYK table the derivation itself, and hence the structure prediction, can be found.

The CYK algorithm is less suitable for use when the grammar is semantically ambiguous (Reeder et al. 2005) (i.e. if more than one derivation is associated with a given secondary structure), because the derivation with highest probability might not correspond to the secondary structure with highest probability, which might be derived in more than one way.

### The n-free-passes CYK Algorithm

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.

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 one free-pass, the CYK algorithm is allowed ‘(((.....)))’. Note the non-canonical base-pair G-A in positions 2 and 10. With one free-pass, the CYK algorithm is allowed

\[
\text{P}_a'[\text{GA}] \geq \text{P}_a[\text{GA}] \text{ and the correct structure '(((.....)))' can become the most likely structure.}
\]

Two important issues are also illustrated by this example. The first is the choice of distribution for \( \text{P}_a'[\text{GA}] \). If we choose \( \text{P}_a'[\text{GA}] \) to be greater than the probability of canonical base-pairs, as the algorithm will always choose to use a free-pass. Equally, if \( \text{P}_a'[\text{GA}] \) is lower than the standard probability of non-canonical base-pairs, then it is likely that no alternative structures will be predicted at all. A natural choice for \( \text{P}_a'[\text{GA}] \) might be

\[
\max_{\text{non-canonical base-pairs}} \{\text{P}_a[\text{base-pair}]\} < \text{P}_a'[\text{any base pair}] < \min_{\text{canonical base-pairs}} \{\text{P}_a[\text{base-pair}]\} 
\]

\[
(2)
\]
Secondly, by allowing free-passes as proposed, we are not dealing with probabilities anymore, but pseudo-probabilities. When allowing free-passes, the “probability” of a given secondary structure derivation will be greater than or equal to the regular CYK case, and so the probability distribution will not be normalised. Furthermore, the normalising constant will be different for different sequences. However, since the derivation probabilities can be normalised for every sequence, for brevity we will continue to refer to the pseudo-probabilities as probabilities in this paper.

Pseudocode for the \( n \)-free-passes CYK algorithm for SCFGs in double-emission normal form is given in Algorithm 1. 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 1 CYK Algorithm with at most \( n \) free-passes

\[
\begin{align*}
\text{for } m = 0 \text{ to } n & \text{ do} \\
& \text{for } i = 0 \text{ to } |s| \text{ do} \\
& \quad C(m, U, i, j) = 0 \\
& \quad \text{if } i = j \text{ then} \\
& \quad \quad C(m, U, i, i) = P[U \rightarrow .] P_u[s[i]] \\
& \quad \text{if } i < j \text{ then} \\
& \quad \quad C(m, U, i, j) = \max_{l_1 \leq m < l_2 \leq j} \max_{u = \phi^{(U \rightarrow VW)}} \left[ \max_{V \rightarrow (W)} P[U \rightarrow VW] C(l, V, i, k) C(m - l, W, k + 1, j) \right] \\
& \quad \quad \quad \quad \quad \quad \quad \text{max}_{V \rightarrow (W)} P[U \rightarrow (V)] P_d[s[i] \mid s[j]] C(m, V, i + 1, j - 1) \\
& \quad \quad \quad \quad \quad \quad \quad \text{max}_{V \rightarrow (W)} P[U \rightarrow (V)] P_d[s[i] \mid s[j]] C(m - 1, V, i + 1, j - 1) \mathbb{I}_{m > 0}
\end{align*}
\]

The complexity of the original CYK algorithm is \( O(k^3) \) in time and \( O(k^2) \) in space, where \( k \) is the length of the RNA sequence. With \( n \) free-passes, the CYK table must be completed for \( n \) additional “layers”, and so the complexity becomes \( O(n \times k^3) \) in time and \( O(n \times k^2) \) in space. If only a small number of errors or non-canonical base-pairs is expected, this is only a constant time multiplier, but if the error rate were proportional to the length of the sequence, this algorithm would become \( O(k^4) \).

### 3 Implementation and Testing

PPfold and the CYK algorithm

PPfold is a recent multithreaded reimplementation of the Pfold algorithm (Knudsen & Hein 1999, Knudsen & Hein 2003), and the two programs use the same lightweight stochastic context-free grammar, which we will refer to as KH99. The KH99 grammar is:

\[
S \rightarrow L \mid LS \\
L \rightarrow . \mid (F) \\
F \rightarrow (F) \mid LS
\]

where . indicates a single unpaired nucleotide, and ( and ) indicate two nucleotides forming a base-pair. KH99 is not presented in double-emission normal form but the adaption of the algorithm is trivial (Anderson et al. 2012).

Typically, a SCFG is parameterised with a separate nucleotide distribution. The probability of a SCFG generating a particular unpaired nucleotide, say an unpaired \( A \), is the product of the SCFG probability of producing any unpaired nucleotide multiplied with the likelihood that the unpaired nucleotide is an \( A \), that is: \( P[L \rightarrow A] = P[L \rightarrow s] P_u[A] \). In this way, the probability of the derivation corresponding to an RNA secondary structure string can also be written as a product of the SCFG rule probabilities and the nucleotide probabilities. The nucleotide probabilities (likelihoods) are typically obtained using frequency counts from RNA sequences with known secondary structures.
An n-free-passes CYK algorithm

4 RESULTS AND DISCUSSION

In Pfold and PPfold, the dynamic programming algorithm known as the inside-outside algorithm (Lari & Young 1990) is implemented, to compute posterior basepairing and single-stranded probabilities. Using these probabilities, the predicted secondary structure is determined by maximizing the expected number of correctly predicted secondary structure elements. For the Pfold base-pair probabilities, the maximum probability non-canonical base-pair is UU with probability 0.0027, and the minimum probability canonical base-pair is GU with probability 0.0490. Throughout, then, we will use a free-pass probability of 0.025.

Testing data
To test the algorithm, we took a data set of 443 RNA sequences with known secondary structures from RNASTRAND (Andronescu et al. 2008). The data set was filtered to remove too similar sequences (greater than 80% base pair similarity) or sequences with ambiguous base-pairs. The dataset contained no pseudoknotted structures, as these cannot be predicted by standard SCFG methods.

From the set of 443 structures, three data sets were created. Firstly, the 443 structures were partitioned into two sets, those structures which contained natural non-canonical base-pairs, and those structures which did not. There were 206 structures containing non-canonical base-pairs, which will be hereafter called the NC dataset. The NC dataset contained an average of 3.74 non-canonical base-pairs per secondary structure, with 43 of the secondary structures only containing a single non-canonical base-pair. The 237 structures which did not contain non-canonical base-pairs will be denoted the Non-NC dataset.

Secondly, we created a simulated dataset by taking all the structures in the Non-NC dataset, choosing a base-pair at random, and changing it to a random non-canonical base-pair. Errors simulated in base-pairs might simulate a sequencing or processing error, and allows for examining the performance of the n-free-passes CYK to evade the effects of this error. We note that errors were not simulated in unpaired regions. For a given secondary structure, the probability of two sequences which only differ by an unpaired base will differ only slightly– by the ratio of the unpaired nucleotide probabilities– and so the n-free-passes CYK algorithm would simply choose the nucleotide with higher probability. Consequently, this error simulation would not allow for useful evaluation of the n-free-passes CYK algorithm. This dataset will be known throughout as the Sim-NC dataset.

Measuring accuracy
To evaluate prediction accuracies, we used the F-score, which is the harmonic mean of the sensitivity and the positive predictive value (PPV) of the base-pair (bp.) predictions, compared to the reference structure. These quantities are defined as:

\[ \text{sensitivity} = \frac{TP}{\text{number of bp. in reference}} \]  \hspace{1cm} (4)

\[ \text{PPV} = \frac{TP}{\text{number of bp. in prediction}} \]  \hspace{1cm} (5)

\[ \text{F-score} = \frac{2 \times \text{sensitivity} \times \text{PPV}}{\text{sensitivity} + \text{PPV}} \]  \hspace{1cm} (6)

where \( TP \) is the number of correctly predicted base-pairs.

4 Results and discussion

Probability of a free-pass
To investigate the behaviour of the method as a function of the free-pass probability, we plotted the accuracy against the fractional free-pass probability for all 3 datasets, shown in Figure 1. We expected that the best performance would be obtained if the probability of the free-pass was between
An $n$-free-passes CYK algorithm

4 RESULTS AND DISCUSSION

Figure 1: The relative F-score, $F_{score \text{ with } 1 \ free \ pass} / F_{score \text{ without free passes}}$, is plotted as a function of free pass probability, for all 3 datasets. A free pass probability of 0.049 corresponds to the probability of a G-U wobble base pair. Probabilities below this value are smaller than the probabilities of all canonical pairs. Probabilities larger than 0.049 are greater than the probabilities of all non-canonical pairs.

the highest probability for non-canonical base-pairs (0.0027) and the lowest probability for canonical base-pairs (0.0490). Against our expectations, any nonzero probability decreased the average accuracy in the case of the NC dataset, and only gave rise to positive changes in the case of the simulated dataset. These results indicate that the $n$-free-passes approach does not generally increase the accuracy of RNA secondary structure prediction when the sequences do not include sequencing errors. However, it appears that the algorithm does improve prediction quality in the case where “non-canonical” base-pairs are not of natural origin but rather are introduced artificially due to errors in the sequence.

We suggest the observed pattern can be explained as follows. 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. In these cases, therefore, it is possible to improve prediction quality.

Case studies

Next, we wanted to investigate the patterns observed in Figure 1 in more detail. On a closer inspection of the resulting structure predictions, we identified that sequences generally fell into
Figure 2: Non-canonical base-pairs are highlighted in red in both the sequence and graphical representation of the secondary structure. 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. See text for further analysis.
An $n$-free-passes CYK algorithm

RESULTS AND DISCUSSION

three categories:

(A) Prediction quality is significantly worsened, $\Delta$ F-score $\leq -0.05$

(B) Prediction quality is significantly improved, $\Delta$ F-score $\geq 0.05$

(C) No significant change in the F-score, $|\Delta$ F-score$| < 0.05$

The majority of the structures fell in category (C). In Figure 2, we have chosen 3 structures which illustrate the above categories. For varying number of free-passes, all with probability 0.025, secondary structure predictions were produced to demonstrate the algorithm.

In the case of (A), we have a structure with a single non-canonical base-pair, a GA on the inside of a stem. With zero free-passes (that is, the CYK prediction) the prediction quality is reasonably good, with additional CG base-pairs produced on the outsides of each of the stems. The one free-pass prediction is identical to the zero free-passes prediction. When KH99 produces unpaired bases in helices, it must use a specific sequence of production rules to change from producing paired nucleotides to producing paired nucleotides, which does not depend on how many unpaired bases KH99 goes on to produce. In this way, the probability of producing a GA base-pair on the inside of the stem is not higher than leaving the two bases unpaired. When two free-passes are used, though, the predicted structure changes entirely to a single stem. The sequence is almost a base-pairing palindrome— the first base pairing with the last, and so on— but there are two exceptions to this. With two (or more) free-passes, a structure with one stem occurs with higher probability than a structure with two stems, and so the structure prediction is significantly worse.

Structure (B) is an RNA hairpin containing three non-canonical base-pairs. The prediction with zero free-passes avoids all of these non-canonical base-pairs, predicting only a small stem in the middle of the sequence. With a single free-pass, however, two non-canonical base-pairs are predicted for a much better structure prediction. Once a single non-canonical base-pair is allowed at a slightly higher probability, predicting a second non-canonical base-pair, despite the low probability, allows the SCFG to produce a single stem structure. Two (or more) free-passes produce the same structure, but by allowing a second free-pass, the structure occurs at a higher probability. As with structure (A), even with a free-pass the algorithm does not produce the non-canonical base-pair on the inside of a stem.

Finally, structure (C) is a more complex structure, with five non-canonical base-pairs. However, many of the non-canonical base-pairs are on the inside of stems, which we have seen will not be predicted with free-passes. With zero free-passes, the structure prediction is reasonably accurate. With a single free-pass added, the prediction is identical. Two or three free-passes, though, allows KH99 to significantly extend a stem, creating a higher probability structure. In particular, the true non-canonical base-pairs are not predicted, the model instead choosing to extend a stem— a choice which allows more efficient use of production rules by KH99, and hence a higher probability structure. Until larger numbers of free-passes are allowed, though, the structure stays approximately the same.

Statistics

Results of the implementation of the $n$-free-passes CYK for all three datasets can be found in Table 1. For clarity of analysis, we show only the results for a single free-pass. We see that, in all three data sets, the majority of secondary structure predictions have little or no change in F-score. This is particularly pronounced in the Non-NC dataset, where there are no non-canonical base-pairs, so we might expect fewer changes.

Structures which did produce positive or negative changes in F-score tended, on average, to be shorter. This is partially due to the dependence of F-score on sequence length, as is also the case for many other RNA secondary structure metric (Freyhult et al. 2005). Predicting five more correct base-pairs, as in (B) of Figure 2, will create a much higher change in F-score than predicting five more correct base-pairs for a significantly longer sequence. However, the opposite is true in the case of the Non-NC dataset, where the average length of the positively improved predictions is notably

Appendix C. An $n$-free-passes CYK algorithm for error-correction and the prediction of non-canonical base-pairs in RNA secondary structure
An n-free-passes CYK algorithm

4 RESULTS AND DISCUSSION

<table>
<thead>
<tr>
<th></th>
<th>#structures</th>
<th>Avg. length</th>
<th>% increase in C bps</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Δ F-score ≤ -0.05</td>
<td>43</td>
<td>128</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>ΔF-score</td>
<td>&lt; 0.05</td>
<td>139</td>
</tr>
<tr>
<td>Δ F-score ≥ 0.05</td>
<td>24</td>
<td>119</td>
<td>16</td>
</tr>
<tr>
<td>Sim-NC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Δ F-score ≤ -0.05</td>
<td>28</td>
<td>133</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>ΔF-score</td>
<td>&lt; 0.05</td>
<td>178</td>
</tr>
<tr>
<td>Δ F-score ≥ 0.05</td>
<td>31</td>
<td>109</td>
<td>11</td>
</tr>
<tr>
<td>Non-NC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Δ F-score ≤ -0.05</td>
<td>24</td>
<td>130</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>ΔF-score</td>
<td>&lt; 0.05</td>
<td>197</td>
</tr>
<tr>
<td>Δ F-score ≥ 0.05</td>
<td>16</td>
<td>172</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 1: Probability = 0.025, 1 free pass compared to 0 free passes.

higher. This is due to several outliers with length over 300 in a small data set, skewing the sample average.

Also notable is the percent increase in the number of canonical base-pairs for each catagory. As might be expected, when the F-score changes very little, the number of canonical base-pairs is almost the same as it was previously. To make a significant change in F-score, for better or worse, the prediction needs to predict more than just the non-canonical base-pairs. As in (B) of Figure 2, where prediction of one non-canonical base-pair allows for prediction of other, canonical, base-pairs, we would expect larger F-score change to require larger change in canonical base-pairs.

Discussion

As the n-free-passes approach changes the KH99 model for RNA secondary structure in fundamental ways, parameterisation is of concern. All grammar probabilities were taken from (Knudsen & Hein 1999), so that testing was independent of parameterisation. In that work, the SCFG probabilities were obtained through counting in derivations of known structures, and similarly the nucleotide probabilities were determined from frequency counts. These parameters maximise the probability of ordinary SCFG prediction, not n-free-passes SCFG prediction. Since KH99 is semantically unambiguous, the production rules used to generate the known secondary structures would be identical for the n-free-passes algorithm, so the probabilities for the production rules will remain constant. The only probabilities that might change in our model are therefore the nucleotide probabilities. If we allowed a free-pass every time there was a non-canonical base-pair in the training structure, we would have zero probabilities for all non-canonical base-pairs, as a free-pass would be preferable each time. Allowing free-passes, in addition to the usual small non-canonical base-pair probabilities, will permit occasional prediction of non-canonical base-pairs not predicted by free-passes, as we see in (B) of Figure 2.

Additionally, one might consider an alternative approach of allowing non-canonical base pairs only in the backtracking part of the CYK algorithm. However, this does not alter the underlying probability distribution created by the SCFG, and we would not predict more than a single additional base-pair. Part of the design of the n-free-passes algorithm was to allow a complete change of structure if it is preferred by the algorithm, and allowing non-canonical base-pairs in the backtracking would not achieve this.

Another possibility would have been to implement the n-free-passes algorithm for the inside-outside algorithm. The advantage of this would be that we could sum over the entire probability distribution generated by the SCFG, rather than simply taking the maximum. Unfortunately, this is not possible to do in practice. For a single free-pass, it would be necessary to sum over all possible subsequences and substructures that can be generated; unlike in the CYK algorithm, the inside-
An \( n \)-free-passes CYK algorithm

outside algorithm would require summing probability contributions from multiple non-canonical base-pairs. If only a single nucleotide or base-pair were of interest, the inside-outside algorithm could be used to sum over all possible predictions that can happen outside of that nucleotide or base-pair. But with an entire structure prediction, the sum would not be over the correct distribution.

The algorithm can be used to answer different questions with just a few small changes. For example, changing the initialization condition to only include a non-zero probability on the top “layer” \((m = n, i = j)\), as well as prohibiting the “copying” of values from a lower layer,

\[
\left( \max_{U \rightarrow (V)} P[U \rightarrow (V)] P_d'[s[i]s[j]] C(m - 1, V, i + 1, j - 1) \mathbb{1}_{m > 0} \right)
\]

, would cause the algorithm to return a structure with exactly \( n \)-free-passes, in contrast to the current implementation, which returns maximum \( n \)-free-passes.

However, the extension of the \( n \)-free-passes algorithm to alignments is less simple. In the case of alignments, nucleotide likelihoods are replaced with alignment likelihoods, which are derived from an evolutionary model, and will vary considerably depending on the sequence content and evolutionary tree. Similarly, one would not expect a whole column to be erroneous, but only a single entry in that column. To implement this, the choice of distribution \( P' \) is therefore not trivial, and the Felsenstein pruning algorithm (Felsenstein 1981) would also have to be modified in addition to the CYK algorithm to allow a fixed number of free-passes in this framework.

The general approach presented in this paper could be extended in a number of ways. By considering SCFG rules which simulate base-pair stacking as in (Dowell & Eddy 2004), it might be possible to gain a more realistic biological signal from sites of possible sequencing errors. Similarly, base-pair stacking will encourage the extension of helices in predictions. The \( n \)-free-passes approach could also be considered in the case of the thermodynamic method, and algorithms that can predict pseudoknots.

5 Conclusions

In this paper, we have developed an \( n \)-free-passes CYK algorithm for the prediction of non-canonical base-pairs in RNA secondary structure, which allows the prediction of maximum \( n \) base-pairs with probabilities that are different from the original SCFG model. If this probability is set appropriately, \( n \) low-probability non-canonical base-pairs can be replaced with higher-probability ones. We implemented the algorithm for the PPfold SCFG, and tested it on 443 sequences. Our results show that the \( n \)-free-passes approach does not generally increase the accuracy of an RNA secondary structure prediction in the case where sequences do not include sequencing errors, but does increase accuracy when “artificial” non-canonical base-pairs are introduced due to sequencing errors.

References


An n-free-passes CYK algorithm

REFERENCES


Index

ambiguity, semantic, 16
ambiguity, syntactic, 16

backward equation, 22
basepair stack, 3
basepair, canonical, 3
basepair, non-canonical, 3
basepair, Watson-Crick, 3
bifurcation, 9
bulge, 3

Chomsky-hierarchy, 14
compensating base changes, 26
context-independence, 14
covariation, 25
CYK algorithm, 16
derivation, 14
dynamic programming, 9, 11
ensemble, NpT, 11
expectation maximization algorithm, 19

Felsenstein algorithm, 43
Felsenstein vector, 22
forward equation, 22
free energy, Gibbs, 11

GTR model, 21

helix, 3
inside algorithm, 17
inside variable, 17
KH grammar, 15
loop, external, 3
loop, hairpin, 3
loop, internal, 3
loop, multi, 3

Markov chain, 21
maximum likelihood, 21
minimum free energy, 11
mutual information, 26

nearest-neighbour approximation, 11, 14
nested secondary structure, 3
normal form, Chomsky, 15
normal form, double emission, 15
normal form, double-emission, 75
outside variable, 17

partition function, 13
pfoi, 14
phylo-SCFG, 27
phylogenetic tree, 21
posterior probability, 29
pseudoknot, 3, 10, 14
purines, 2
Index

pyrimidine, 2
rate matrix, 22
stacking, 3
stationary distribution, 22
stochastic context-free grammar, 14
tetraloop, 3
time-reversible, 21
transition matrix, 22