RNA Energetics And Sequence Design

Dissertation

zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) im

Fachbereich Chemie
Fakultät für Mathematik, Informatik und Naturwissenschaften

der Universität Hamburg

vorgelegt von
Stefan Bienert
geboren am 22.11.1979 in
München, Deutschland

Hamburg, den 27. November 2015

1. Gutachter:
Prof. Dr. Andrew Torda[1]

2. Gutachter:
Prof. Dr. Zoya Ignatova[2]

Termin der Disputation: 29.01.2016

[1]Universität Hamburg
[2]Universität Hamburg

Abbreviations

arb. units	arbitrary units
CCD	Chemical Component Dictionary
CSD	Cambridge Structural Database
DP	Dynamic Programming
Eqn	Equation
FSM	Finite state machines
MFE	Minimum free energy
MFT	Mean Field Theory
NN	Nearest Neighbour
nt	Nucleotides
NUN	Non-unitable nucleotides
PDB	Protein Data Bank
Res	Resolution
SCMF	Self-Consistent Mean Field/ Self-Consistent Mean Field Design
wwPDB	Worldwide Protein Data Bank

Contents

Abbreviations **1**

Zusammenfassung **4**

Abstract **5**

1 Introduction **9**

2 Hydrogen Bond Recognition **11**
2.1 Introduction 11
2.2 State Of the Art 12
2.2.1 Chimera 13
2.2.2 `RNAview` 16
2.3 Finding H-Bonds By Geometrical Features 20
2.3.1 Distances 21
2.3.2 Pseudo Hydrogens 23
2.3.3 Quasi Energy Function 25
2.3.4 Identifying Base Pairs 26
2.4 Implementation 27
2.4.1 PDB Reader 27
2.4.2 Hydrogen Bond Finder 32
2.4.3 Base Pair Finder/ 2D Reader 35
2.5 Results 38
2.5.1 Test Set 39
2.5.2 Evaluation Procedure 42
2.5.3 Pseudo Hydrogens 44
2.5.4 Model Evaluation 53

2.5.5 Base Pair Evaluation 59
2.5.6 Discussion 68

3 Sequence Design **73**
3.1 Introduction 73
3.1.1 The Problem – More Than Inverse Folding 73
3.1.2 The Nearest Neighbour Model 75
3.1.3 State Of the Art 79
3.2 SCMF Design 84
3.2.1 Self-Consistent Mean Field Optimisation 86
3.2.2 Sequence Representation 90
3.2.3 Annealing 92
3.2.4 Energy Terms 93
3.2.5 Implementation 106
3.3 Results 111
3.3.1 Parameter Optimisation 111
3.3.2 Energy Terms 113
3.3.3 Performance 115
3.3.4 Case Studies 149

4 Discussion **159**

Bibliography **163**

A Gefahrstoffe und KMR-Substanzen **175**

B PDB Structures Used For H-Bond Evaluation **177**

C Selbstständigkeitsversicherung **181**

Zusammenfassung

Zusammenfassung

RNA wurde einst lediglich als Träger genetischer Information betrachtet. Jedoch wurde im vergangenen Jahrzehnt eine Reihe von relevanten regulatorischen Funktionen dieses Moleküls entdeckt. Ebenso hat das Design von Molekülen eine lange Tradition in Fachgebieten wie Wirkstoffentwurf, Bio-Engineering und Nanotechnologie. Die Zielsetzung dieser Arbeit ist es diese beiden Themen zu verbinden zur Entwicklung von Methoden zum Automatisierten Design von RNA Sequenzen die sich in eine vorgegebene Form falten. Dies ist ein diskretes Problem, da Design von RNA Sequenzen bedeutet, für jede Position in einer Struktur, eine von vier Basen auszuwählen. Wir hingegen verwenden eine Methode aus der Chemieinformatik (Self-Consistent Mean Field Optimisation), welche es uns erlaubt das Problem als kontinuierliche Optimierung zu behandeln. Wir zeigen, dass es damit möglich ist beliebige Formen von Sekundärstrukturen zu behandeln, im Gegensatz zu anderen Ansätzen.

Eine Prototyp-Implementation um das Prinzip zu zeigen ist die eine Herausforderung, Software zu schreiben die tatsächlich im Laboralltag besteht, hingegen weitaus schwieriger. In dieser Arbeit wird im Rahmen einer Fallstudie die Struktur einer von unserer Software designten Sequenz *in vitro* überprüft. Wenn man sich mit dem Gebiet der computergestützten Strukturanlayse und Design befasst, trifft man immer wieder auf Definitionen die von diskreten Zuständen, also einer starren Welt, ausgehen. Im ersten Teil dieser Arbeit zeigen wir, anhand von Wasserstoffbrücken in RNA Molekülen, das diese starren Definitionen nicht der chemischen Wahrheit entsprechen. Hier ist die Energie von Wasserstoffbrücken eine kontinuierliche geometrische Funktion anstatt einer diskreten Eigenschaft.

Abstract

The polynucleotide, RNA was once thought to be a carrier of genetic information, but in the last decade, a variety of regulatory roles have been discovered. At the same time, molecular design has a long tradition in fields from pharmaceuticals to bio-engineering and nanotechnology. The aim of this work combines these two topics in the development of methods for the automatic design of an RNA sequence that will fold into a desired shape.

This is strictly a discrete problem since design means choosing one of four base types for each position in a structure. We however have taken a procedure from computational chemistry (self-consistent mean field optimisation) which lets us treat it as a continuous optimisation procedure. This means that, unlike other attempts to treat this problem, we show that we can treat arbitrary shapes and structures.

Proof of principle code is one challenge, but producing software which is useful to an experimental group is more difficult. In a case study the software is used to design a sequence which is validated *in vitro* to form the expected structure.

When surveying the field of structure analysis and design computationally, one often finds definitions assuming discrete states, a rigid world. In the first part of this work, we show on hydrogen bonding in RNA molecules, these rigid definitions are a convenience and not the chemical truth. Here, the energy associated with hydrogen bonds is a continuous function of geometry and not a discrete property.

Chapter 1

Introduction

This work tries to cover two related topics: energy concerns and the design of molecules. Here, RNA should obviously be the determinating connection. But not only by the title, more by the fact that one cannot explore any structural space, without an idea of an energy scheme. This strong dependency between the two topics easily defines a natural path for their presentation.

Firstly, we consider the tools. Since Hydrogen bonds vary in their strength and geometry, a tool was necessary that avoided hard thresholds and was tolerant of poor geometry. Furthermore, a major part of the work was a software foundation to serve as the basis for this and other projects. Finally the main affiliation, the design of RNA sequences, is described along with its testing.

Since both parts are rather large, they come with their own additional introduction, more specific on the corresponding topic.

In the central dogma of molecular biology [1], RNA was mainly accepted as mRNA, the information carrier between DNA and protein synthesis, as a start. While its role in protein synthesis is larger, as tRNA carrying amino acids into the ribosome [2]. While the ribosome is a huge functional RNA molecule in itself. In recent years a variety of more functions and regulatory roles of RNA have been discovered [3]. Nowadays, the RNA world hypothesis exists, suggesting RNA as the starting point of evolution and thus the precursor of life [4]. Usually RNA is found as single strand molecule in the cell, being able to form a structure on itself. From HIV it is already known that RNA is also involved in human diseases [5], but in recent years, RNA being pathogenic in connection with its structure has gained some focus [6, 7].

While RNA has a lot of functions in the cell and the first structures were analysed in the seventies of the last century, exploring its conformational features, still lacks behind proteins. Therefore the first part of this work adapts a well established model for hydrogen bond recognition on proteins [8] to RNA.

Being a molecule comprised of only four different building blocks, modifying RNA to achieve new properties seems to be simple. Some effort has been invested to design RNA molecules of certain function or improved properties *in vitro* [9]. This is what the second part focuses on. Understanding and improving designing RNA sequences. This part benefits from the amount of work already done on the computational site for RNA. At least for the two-dimensional space, algorithms are being developed to predict a structure since the eighties of the last century [10]. Those started with simple energy models counting H-bonds until today, when quasi-molecular dynamics methods try to predict three-dimensional folds [11].

Chapter 2

Hydrogen Bond Recognition

Given the early development of methods for predicting 2D structure, computational views of RNA still have a focus on base pairs, rather than 3D coordinates. Therefore, to fit a crystal structure into literature energy models, its base pairs have to be extracted, pointing towards recognising hydrogen bond patterns in the structure. Here, we describe an approach to define hydrogen bonds in a 3D RNA molecule and utilise them to detect base pairs. The method itself is an adaption of an approach applied to proteins by Kabsch & Sander [8].

2.1 Introduction

While the first 3D RNA structures are from the seventies of the last century [12–14], as of today, computational RNA most of the time means working on 2D annotations. The most prominent tools of the field, e. g. Vienna RNA package [15–17] and `mfold` [18–20], still work on lists of base pairs, while new tools exploring 3D space like `MC-Fold/ MC-Sym` [11] receive higher attention.

One reason methods for recognising H-bonds and annotating 3D structures are necessary, is the Nearest Neighbour energy model [21–23], many tools employ. In this energy function, stacks of base pairs and certain structural features have assigned scores which are summed up to define ΔG. In simple words, this gives an estimate of the energy difference between the folded state and the set of unfolded states. But in contrast to the common force field approach, defining H-bonds by electrostatic and Lennard-Jones terms, this model treats base pairs, and thus H-bonds, binary: either two bases are paired, with full H-bonds, or not. Local conditions are completely out of focus, e. g. bases competing with each other for a partner because of special geometry and any other weakening or strengthening effects. Additionally also features like Hoog-

steen interactions and pseudoknots are missing. §3.1.2 gives a larger overview of the Nearest Neighbour model.

Annotating secondary structure features in a 3D structure means detecting H-bonds and recognising certain patterns representing base pairs. But problems already start at the H-bond level: textbooks use distance intervals instead of single thresholds to define them [2]. Thereby interactions with different distances do not have the same strength as suggested by the Nearest Neighbour model. Another problem exists for the patterns defining base pairs. If for a canonical pair all H-bonds are perfect but one just violates distance cut-offs by a tiny amount, should the whole pair be dropped? Even more complicated, most people will decide on the kind of base pair, if it should contain two or three interactions. For such cases, a model which is able to define probabilities, on H-bond and/ or base pairing level, would be better suited. The general question following up is, whether H-bonds are really a binary concept or a concept defining an energetic term. Kabsch & Sander already preferred the term "polar interaction" over "H-bond" in their work, being more descriptive to the nature of the effect [8].

While there are already methods available to detect H-bonds, we want to add our own for several reasons. Generalised approaches, dealing with all possible interactions, are very complex and by this hard to implement in your own software [24]. Our aim is treat only RNA, so the set of chemical groups interacting is limited and there is no need to incorporate everything possible right from the beginning. Focusing on a simple model also lowers the probability of errors. But our approach should not be too stiff to be extended for other molecules, later. Especially in case of RNA, there are already tools which annotate base pairs in 3D structures [25]. But those approaches are too coarse grained for us. We want to be able to produce input for the Nearest Neighbour model as well, but also not lose the ability for a smooth transition towards using 3D structures, later.

2.2 State Of the Art

Whenever one is dealing with 3D structures, models for H-bonds are needed. Protein structures, as well as ligand contacts, are defined mostly by H-bonds and for the simulation of solvent interactions, a huge effort is put into modelling these polar interactions. This short list of examples may be seen as proof that the task of defining H-bonds via a computer must have been addressed before. Two different avenues can be identified by examining recent research and up-to-date software: attempts to approach the problem in a general manner, based solely on donor and acceptor groups, and specialised solu-

tions, applicable to certain groups of molecules. The first type is in theory able to handle everything chemically possible and both, inter- and intramolecular interactions. The cost of this approach is the complexity of implementation and a less fine-tuned parametrisation of H-bond criteria. The alternative is a less general approach, designed and parameterised for a specific class of molecules.

Once we have a list of H-bonds for a nucleic acid, deriving base pairs seems to be easily possible. For Watson-Crick and wobble (`GU`) base pairs this task could be reduced to mere pattern recognition. If the mission is just to produce a list of base pairs, approaches exist which do not need H-bond information. Since those methods only work on geometric data of a structure, they do not provide energy or stability values. This also implies, that H-bonds are a rigid, not a statistical concept.

For the class of general H-bond detectors, we will give a summary of the widely used method of Mills & Dean [24]. For the second class, methods solely acting as a base pair finder, the computational RNA area provides one prominent example. *RNAview* [25] does not find H-bonds in a 3D structure, but reads out base pairs just by investigating geometrical criteria of the whole pairing site rather than single atoms.

2.2.1 Chimera

Chimera, described as “an Extensible Molecular Modelling System” [26], has great visualisation and interactive analysis capabilities for 3D biomolecular data. Needless to say, as a tool to analyse crystal structures, it provides functionality to identify H-bonds using the approach by Mills & Dean [24].

In their study, Mills & Dean describe a set of atoms or chemical groups, determining a set of geometric constraints, distinguishing a polar interaction. For atoms, only the H-bond acceptors and donors are considered. By this, the need of real protons which are rarely found in crystal structures is avoided.

While the study uses the *Cambridge Structural Database* (CSD) [27] as data source, Chimera more often sees data from the *Protein Data Bank* (PDB) [28] as input. Since the CSD stores small organic molecules at high resolution, the error in structure determination for whole proteins as stored in the PDB is expected to be larger by default. This forces `FindHBond` to relax H-bond parameters, as advised by Mills & Dean, while retaining the geometrical criteria. Because the study does not list all possible hydrogen donors and acceptors, some parameters are estimated for real-world application.

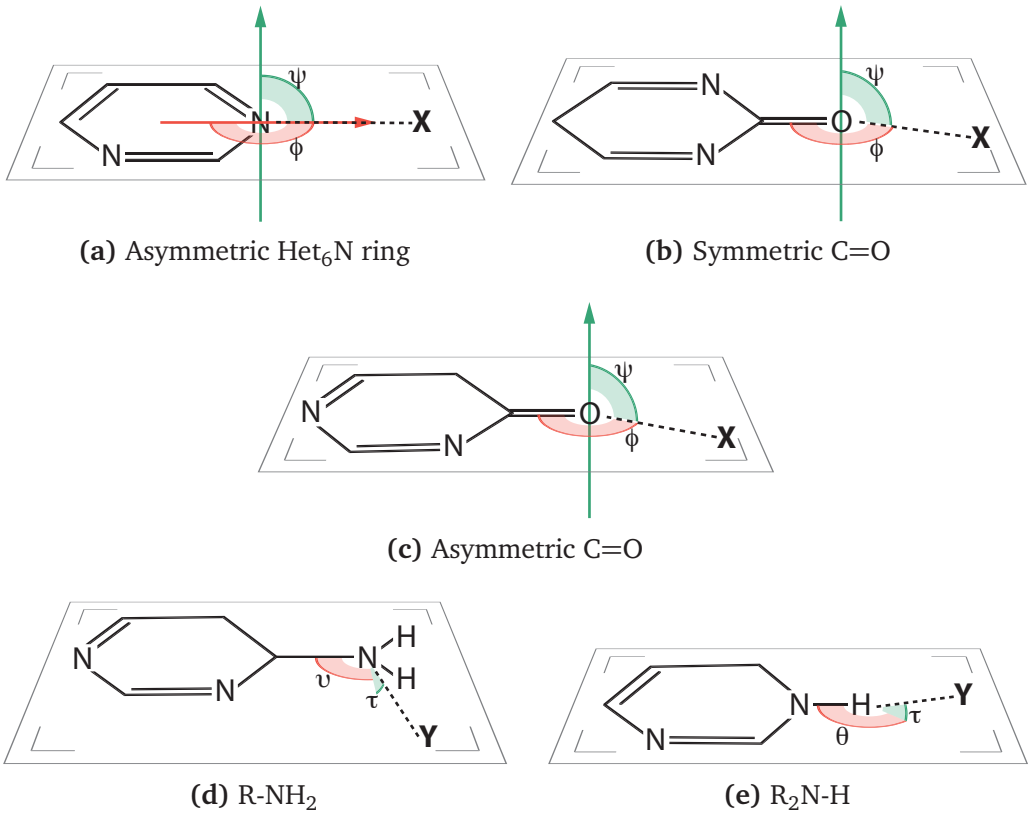
For hydrogen-bonding groups, nitrogen, oxygen and sulfur atoms are considered. To reflect the geometric parameters used to distinguish the interactions formed, considering single atoms are not enough. Instead, they are

divided into two basic classes of chemical groups bearing those bond-forming atoms. Both classes contain a distance threshold between non-hydrogen atoms but differ by angular constraints. The ψ-ϕ groups utilise the ψ-angle, as defined by a plane upon the acceptor atom and the donor atom as well as the ϕ-angle, formed by an axis through the ψ-plane and the donor atom as further parameters. Criteria for the second class, θ-τ groups, take the θ-angle, measured along the donor - hydrogen - acceptor axis, and the τ-angle, describing the torsion around the axis measuring θ. If no explicit hydrogen atom is available or the angle does not subtend it, θ is named υ. Examples for both classes are found in Fig. 2.1, showing ψ and ϕ-angles in Fig. 2.1 (a)–(c), and θ, υ and τ-angles in Fig. 2.1 (d)–(e).

Altogether, both classes provide 76 criteria sets derived from 39 chemical groups. The ψ-ϕ groups only split for *syn* and *anti* addition. But the θ-τ groups are subdivided into hydrogen donors and acceptors, with additional classes formed by the state of hybridisation. Originally, the ψ-ϕ class contains 15 chemical groups, but 22 distance/ angles-tuples. The θ-τ class keeps some of its 24 chemical groups as both, donor and acceptor, and provides 3 different parameter sets for most of the donors, resulting in 54 criteria tuples summed up.

To find H-bonds in a PDB file using these criteria, several steps have to be taken. First, nitrogen, oxygen and sulfur atoms have to be identified, together with adjacent atoms to define the groups they belong to. Next, this pool of chemical groups has to be divided into hydrogen-bond donors and acceptors. To verify possible H-bonds, the members of both categories have to be matched pairwise. For ψ-ϕ groups, parameters can be used without further considerations. For θ-τ donors, we need to detect the state of hybridisation of each possible acceptor. The distances and angles to check the criteria have to be calculated separately for each donor-acceptor-pair, for both classes. Mills & Dean only provide a binary answer, candidates meeting the criteria are accepted as H-bonds, not otherwise. Without another automated method to decide, `FindHBond` returns a list of all positively evaluated candidates, allowing for multiple donations simultaneously. For well-determined hydrogen positions, this physically impossible state may be reduced.

The binary nature of this method immediately reveals one drawback. Since there are no energetic considerations or interaction probabilities, there is no information about the stability of a molecule. For detecting base pairs in RNA crystal structures, this complicates the task. In helical regions, it is common for donors to have more than one acceptor fulfilling the geometric criteria. Without a stability measure, this often leads to H-bonds between one base and multiple possible base-pairing partners. Deciding, which H-bonds will lead to base-pair formation, is therefore the task of a post-processing algorithm.



(a) Asymmetric Het_6N ring

(b) Symmetric C=O

(c) Asymmetric C=O

(d) R-NH_2

(e) R_2N-H

Figure 2.1: *Mills & Dean hydrogen-bonding groups.* Examples of the parameter classes following [24]. X describes a donor, Y an acceptor atom. Original nomenclature attached in parenthesis. ψ-ϕ groups: (a) asymmetric heterocyclic six-membered ring containing a N atom (Asymmetric Het_6N); (b) symmetric carbonyl group (Symmetric C=O); (c) asymmetric carbonyl group (Asymmetric C=O); θ-τ groups: (d) primary amine (R-NH_2), in this case, θ is not subtended by the hydrogen atom and therefore renamed υ; (e) secondary amine (R_2N-H).

2.2.2 RNAview

RNAview [25] is an application from a small collection of RNA analysis tools, focusing on 2D structure annotation. Where "2D" refers only to the representation rather than structural features, since tertiary contacts are also covered along with the usual base-pairing scheme.

More specifically, RNAview finds nucleoside-nucleoside interactions in 3D structures. This includes Watson-Crick base pairs as well as other contacts, e. g. Hoogsteen pairs. Detailed H-bond recognition is not the main focus of the tool. Interaction detection involves considering H-bonds, but annotation is based on interacting sides of the nucleosides rather than individual atoms. Those sides are identified solely by geometric criteria, without consideration of energies. As input, RNAview works on PDB structures and transforms them into a 2D base-pair map with additional interactions highlighted. For the annotation, basically two ingredients are used: a standard coordinate frame applied to each nucleotide in a structure and a nomenclature to describe interactions.

As point of reference for 3D arrangements of nucleic acids, base coordinates as suggested by Olson et al. [29] are used. This standard coordinate frame is derived from ideal Watson-Crick base pairs according to structures of the CSD. Along with the coordinates, the coordinate system is provided, giving each base its own reference point. The geometric criteria for H-bonds can be derived from two frames, when their origins overlap while the y-axes point away from each other. Fig. 2.2 gives a detailed view of the reference frame.

For annotation, the edge-nomenclature from Leontis & Westhof [30] is used. For H-bond mediated interactions between two nucleotides, this concept identifies distinct contact sides. Following geometry, three edges can be identified: the common base-pairing face is denoted Watson-Crick edge. The Hoogsteen edge is roughly on the back of the first edge and the Sugar edge is located below the premier contact side. An illustration of the different edges is given in Fig. 2.3. For a non-ambiguous classification, a base pair needs at least two H-bonds. Additionally, the relative orientation of the bases are labelled *cis* and *trans* according to the position of their glycosidic bond. The orientation is measured along a parallel line between the two H-bonds of contacting edges. If both glycosidic bonds are on the same side of the line, the interaction is called *cis*, *trans* otherwise. For the three edges together with *cis-trans* isomerism, Leontis & Westhof describe twelve possible classes of interactions and also provide a symbolic notation scheme for them.

Before RNAview classifies interactions, the standard coordinate frame is applied to a structure. This means that the real bases, as seen in a crystal structure, are substituted by their analogous reference coordinates. Using the least square fit method, the rigid coordinate frames are translated and rotated onto

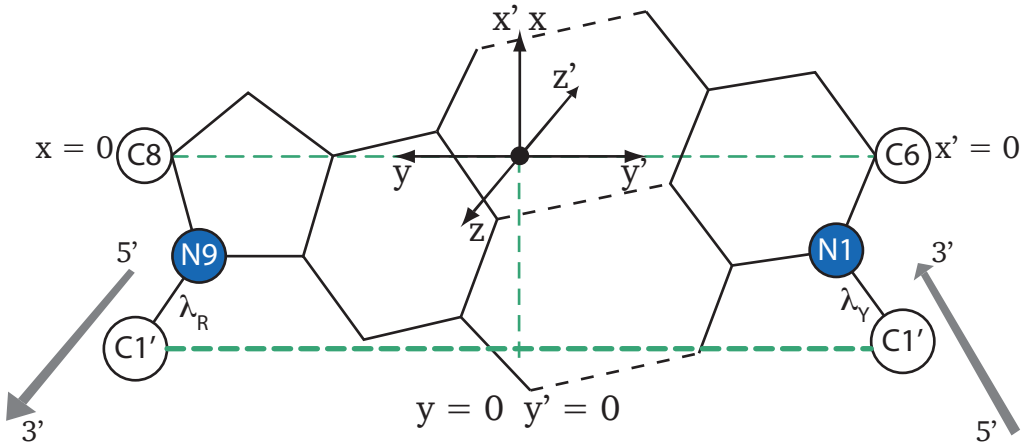
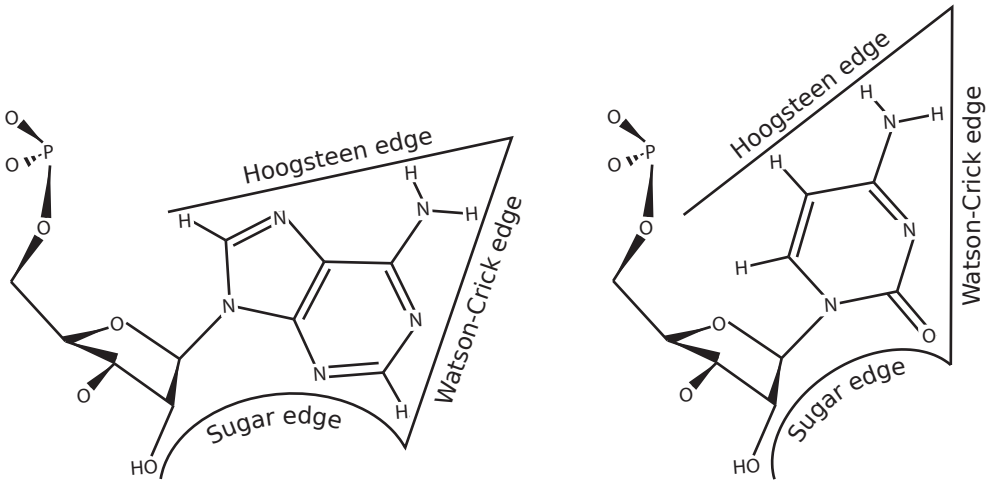


Figure 2.2: *Standard nucleic acid reference frame.* The corresponding parameters are chosen to form an ideal Watson-Crick base pair for pyrimidine (Y) and purine (R) bases. The x-axis points towards the major groove, the y-axis in the direction of the sugar-phosphate backbone and the z-axis in 5'- to 3'-direction along the sequence strand following the right-handed rule. The origins of the coordinate systems are defined by two distances and two angles. $x = 0$ on level with a vector between purine R(C8) and pyrimidine Y(C6). $y = 0$ in the middle between the two sugar-C1' atoms of the nucleotides. The positions of R(C8) and Y(C6) are determined by angles λ_R and λ_Y, formed by the sugar and the base (Taken from [25] and [29] and modified).



(a) Purine edges **(b)** Pyrimidine edges

Figure 2.3: *Nucleotide interaction edges.* Purine and pyrimidine bases with their three edges to form interactions. (a) For purines, the Watson-Crick edge is formed by A(N6)/ G(O6), R(N1), A(C2)/ G(N2). The Hoogsteen edge is comprised of A(N6)/ G(O6), R(N7). The Sugar edge contains A(C2)/ G(N2), R(N3) and the hydroxyl group of the ribose. (b) For pyrimidines, the Watson-Crick edge is formed by U(O4)/ C(N4), Y(N3), Y(O2). The Hoogsteen edge is comprised of U(O4)/ C(N4), Y(C5). The Sugar edge contains Y(O2) and the hydroxyl group of the ribose (Adapted from Leontis & Westhof [30]).

the bases without affecting the phosphate backbone. This approach is also applicable to modified bases. The idea is to use the best fitting standard base as a substitute. Additionally, along with the least square fit, nucleic acid deformation parameters like shear, stretch and propeller are detected. Also the *cis*/*trans* configurations of the glycosidic bonds are determined in this process.

From the set of standard bases, RNAview derives the interactions of a molecule. Since all possible twelve edge combinations are considered as base pairs plus tertiary interactions, there are three different sets of criteria and parameters. Canonical Watson-Crick pairs are described by angular ranges for the standard frame axes, by a distance of the origins of the two coordinate systems and the vertical distance between the base planes. Additionally the glycosidic bonds are only allowed in *cis* mode. All contacts which do not belong to this class are next considered non-Watson-Crick base pairs. According to the definition of Leontis & Westhof, these are all remaining combinations of the 3 edges and position of the glycosidic bond. Everything else is assumed to be a real tertiary interaction. While the vertical distance between the planes is common to all kinds of pairs, the only angle considered for non-Watson-Crick interactions is formed by the two z-axes. The parameters for these geometrical criteria are a bit relaxed compared to canonical base pairs, but the requirements on H-bonds are more strict. For an interaction to belong to the class of non-Watson-Crick pairs, at least two H-bonds are required. One of which has to be established between the bases of interacting nucleotides. Additional distance thresholds differ between the type of H-bond. In contrast to base-base interactions, every other combination gets an atom-dependent range. As a special case, two H-bonds with an identical donor atom are rated, requiring different parameters at the same criteria. All contacts left which are neither canonical base pair nor non-Watson-Crick pair, are assumed tertiary interactions. These need at least one H-bond, meeting an atom dependent distance threshold.

After defining base pairs in a structure, RNAview presents it as a 2D map. But instead of just drawing interactions in an enumerative manner, they are collected into structural features. The most common motif should be helices. They occur with an initialising pair of Watson-Crick base pairs. If two strands of helices are interrupted, e. g. by a bulge loop, they are still shown as one larger pseudo helix. To discriminate base-pair types in the plot, the annotation according to Leontis & Westhof [30] is utilised. Further visual highlights are applied to the stereochemistry of nucleotides, certain stacks and modified bases. However, one problem of 2D RNA structure arrangement remains unresolved by RNAview: overlaps of structural features in the map. They have to be solved manually by editing the plot using RNAMLview.

Similar to the Mills & Dean method, RNAview only generates a binary answer. In contrast to Chimera, there is no H-bond presentation, but edge-to-edge and

tertiary interactions. For comparing or discussing structures, this approach of a holistic, discretised annotation is very convenient. Still, the edge classification remains a simplification. It gathers up interactions while deciding ambiguous scenes which are derived from a modified geometry of a structure. Because everything is solely based on geometrical constraints, no information about the stability of the interactions can be provided.

2.3 Finding Hydrogen Bonds By Geometrical Features

In the last section, two methods were described with different scope but one important similarity. Mills & Dean [24] provide a general approach for all H-bonds, while `RNAview` [25] only goes for interactions in RNA. But both methods only consider pure geometry/ trigonometry. For visualisation of molecules, this is sufficient. However, to any calculations with molecules, e. g. scoring their stability, these methods cannot contribute. Since this is our primary goal, these approaches are not sufficient for us. What we are aiming at as a primary target, is detecting H-bonds & base pairs along with a score to easily assess the stability of a scene.

The two models described define H-bonds as a rigid concept. While `RNAview` operates in a somewhat relaxed mode, allowing donors with two acceptors, the interactions themselves are binary for both approaches. But a discrete definition is not necessary, probably a hindrance, for determining the stability of a structure. One problem is weak interactions, which are just neglected, but may be significant once summed up. Another one arises by geometrically ambiguous configurations, where H-bonds are equally favourable between single acceptors and multiple donors or vice versa. Thereby a rigid concept forces us to drop all but one interaction. Those two examples should point out, that we always lose energetic contribution, as soon as we decide between distinct interactions.

In the method presented in this work, we try a continuous approach for H-bonds. More precisely, at the start there are no bonds, but polar interactions. These can be described by energies or probabilities, assigning small contributions to weak interactions and penalising certain steric problems. This scheme also allows immediate incorporation of situations with more than one energetic favourable pairing possibility. To get from H-bonds to base pairs with this approach, not much more than a threshold would be needed. This could be used to go back to discrete interactions and easily identify paired bases, with the benefit that beside pure “H-bond counting”, certain energy values

could be added to the criteria.

As often in the RNA field, we can benefit from what was already done for proteins. Kabsch & Sander describe a method to recognise secondary structure features in proteins just by H-bonds and geometrical features [8]. Instead of analysing the whole geometry around donor and acceptor sites, H-bonds are recognised based on a simple model for electrostatic energy.

2.3.1 Distances

The main task of the Kabsch & Sander study is to recognise α-helices and β-sheets in proteins, mainly by H-bond patterns. Single interactions are defined by a threshold energy based on a simple electrostatic model.

Considering all interaction partners, the potential energy has four contributing terms. Distances are measured between all atoms which carry a partial charge in addition to the hydrogen and the acceptor oxygen. These are the donating amino group and the carbonyl-atom, respectively. Fig. 2.4 shows a short piece of β-sheet with all distances highlighted.

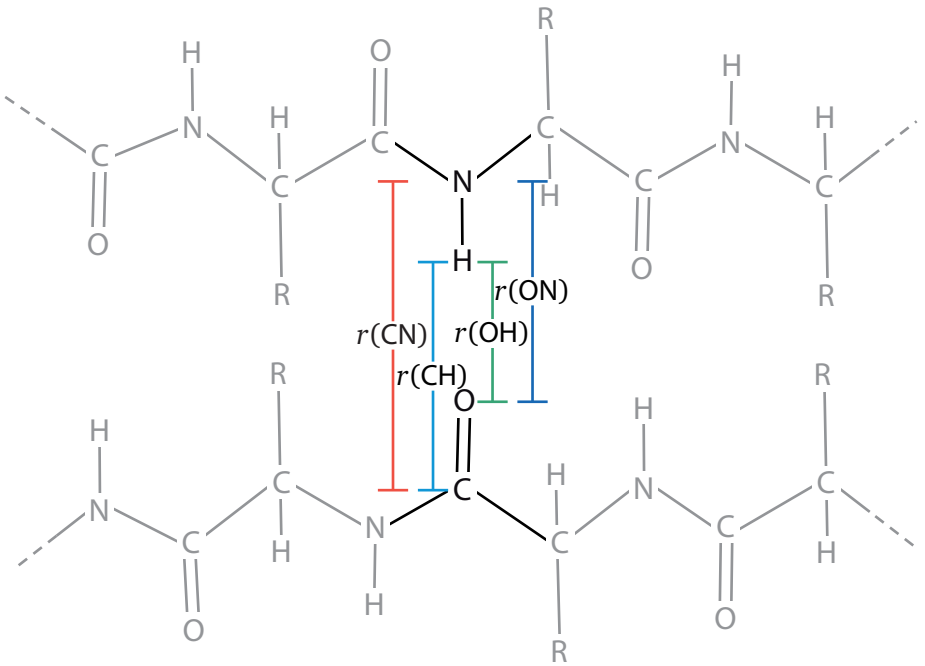
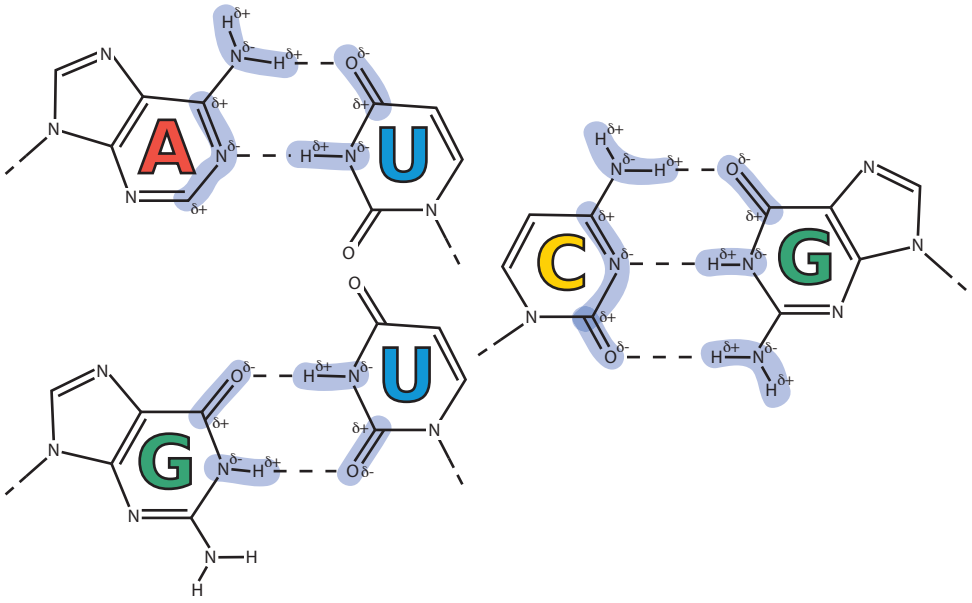
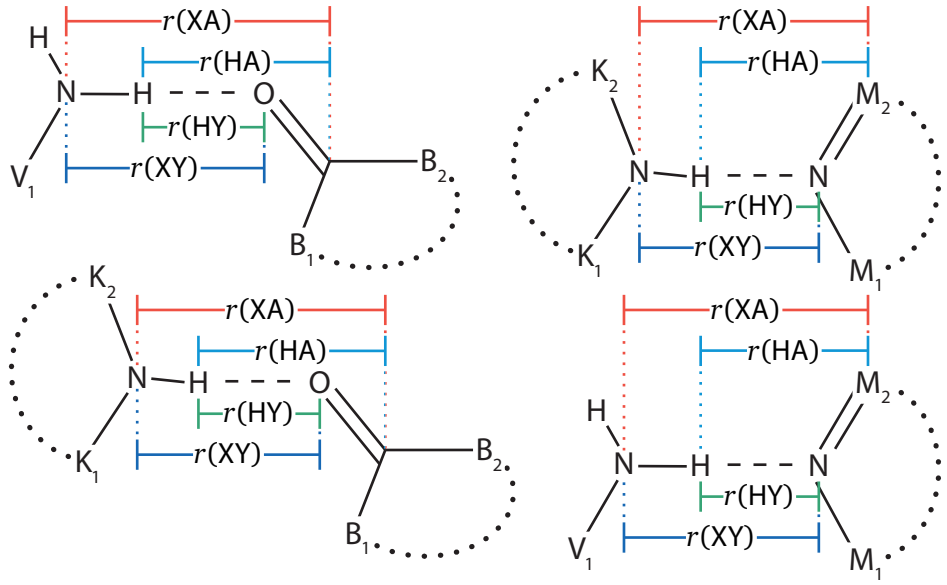


Figure 2.4: *Interaction distances in proteins.* A small excerpt of a β-sheet, with interatomic distances highlighted. These are used by Kabsch & Sander to calculate H-bond energies [8]. r(XY) describes the distance between atom X and Y.



(a) Interacting sites



(b) Distances

Figure 2.5: *Base pair interaction partners and corresponding distances.* (a) H-bond donor and acceptor atoms in canonical and GU pairs. Adjacent atoms involved in forming polar interactions are also highlighted. (b) Distances to calculate interaction energies. If calculating with partial charges, R(N1)/ Y(N3) need both adjacent carbon atoms to be included. V replaces remaining atoms in adenine, cytosine or guanine. B substitutes in cytosine, guanine or uracil. K describes guanine or uracil atoms. M is a placeholder in adenine or cytosine.

In our approach to find H-bonds and RNA base pairs, as proof of principle we only focus on Watson-Crick edges as described in Fig. 2.3. With all polar interactions defined, identifying canonical base pairs and some wobble pairs is left as an enumerative job. Building upon this initial step, an extension towards a holistic annotation as in RNAview (see §2.2.2) is feasible.

Fig. 2.5 shows an overview of the interaction sites and their partners for RNA nucleotides. Here, the donors and acceptors forming AU, GC and GU pairs as well as adjacent atoms are considered. Fig. 2.5(b) lists all distances that need to be evaluated for canonical base pairs and the wobble GU pair. Since the process of finding H-bonds will be performed in an all-against-all manner, the detailed description of the interatomic distances will follow chemical groups rather than whole nucleobases. Altogether, the four canonical bases make up four different groups: three nitrogen- and one keto-groups. As donors, the purine bases and cytosine carry a primary amino group (A(N6), C(N4), G(N2)), guanine and uracil also carry a secondary amine (G(N1), U(N3)). In adenine and cytosine, the acceptor nitrogen (A(N1), C(N3)) is part of an enamine group, formed with the primary amine. Pyrimidines and guanine have keto-groups (C(O2), G(O6), U(O2), U(O4)). While proton donors occur as pairs, acceptors consist of a single atom on first sight. But since the energy of a polar interaction depends on partial charges, adjacent atoms as also used by Kabsch & Sander need to be considered. For the keto-group there is only the connected carbon atom to invoke into distance calculations (C(C2), G(C6), U(C2), U(C4)). The considered ring nitrogen is slightly different to the known scheme since it lives in the neighbourhood of two carbons. In the real world, both carbons contribute partial charges to the interaction. Since we operate in a rather artificial mode, e. g. we are not trying to calculate absolute real world energies, we do not need to consider both atoms in theory. This will simplify the formula to calculate a pseudo-energy. This idea is supported by Kabsch & Sander, since they showed for proteins that four distances/ energies are enough to define an H-bond. In nucleotides, we simply always use the carbon atom counterclockwise to the nitrogen (A(C6), C(C4)).

2.3.2 Pseudo Hydrogens

All methods for finding hydrogen bonds suffer of one general problem. Their common input, PDB files, does not usually have coordinates for protons. There are methods like neutron diffraction which allow to define hydrogen positions. Unfortunately these have only applied to a small fraction of PDB entries.

The approaches presented here use different strategies to tackle this problem. By applying the standard coordinate frame (Fig. 2.2), RNAview avoids the need for explicit protons since the base pair parameters do not incorporate

them. The approach of Mills & Dean (§2.2.1) only requires hydrogen atoms in 18 of the 76 criteria. Where they do not provide a strategy how to deal with an absence for those 18 cases, Chimera does its own proton substitution. Thereby, the placement follows a few rules, specific for certain chemical groups. However, if the input structure carries its own protons, these are used. Kabsch & Sander, the method we are following, give no advice on how to deal with missing protons.

Since our approach needs the positions of hydrogen atoms to calculate distances for the energies, we model artificial atoms into the input structures. As a source of coordinates, the *Chemical Component Dictionary* (CCD) [31] of the *Worldwide Protein Data Bank* (wwPDB) [32] is used. With bond lengths and angles calculated from them, placing the atoms should be a simple task. This is true for the secondary amino group of guanine and uracil, since the proton sits right in front of the nitrogen atom. To get the correct position only a vector has to be calculated, pointing away from the nitrogen. As reference points, the adjacent carbon atoms are used, assuring that the artificial atom lies in a plane with the ring, pointing outwards.

The other donor group, the primary amine in adenine, guanine and cytosine, is more complicated. While in reference coordinates, the two hydrogens always lie in a plane with the ring atoms, they are attached to a rotatable group which makes them hard to be placed, once looking at a larger molecule. Also for determining H-bonds, the question arises, which of the two protons is to be considered. It is even worse, if we assume, that both might have a contribution. As a solution, the idea is to work with a single pseudo hydrogen atom. This will be both, easier to place and simple in choosing interaction partners. From Kabsch & Sander we already know that this should work, since the interacting backbone nitrogen of an amino acid also has only a single proton attached. This leaves the question, where the pseudo hydrogen is to be positioned. To avoid the problems of a rotatable group, the middle between the two real hydrogens seems obvious. The price for this simplification is a slightly distorted geometry at this interaction site. As a substitution, the distance to the pseudo hydrogen will most of the time be shorter than one of the distances to the real protons and longer than the second (see Fig. 2.6). How much this affects the applicability to detect H-bonds, has to be evaluated. Another point to be considered are the contributions to the overall interaction energy. Assuming that both protons of the amino group have an effect on the H-bond, we are loosing one term by introduction of a single pseudo hydrogen. An elegant way to compensate both effects in one go, is to shift the position of the pseudo hydrogen on the bisector between the real protons. §2.5.3 shows the results of the search for a plausible placement of the artificial atom.

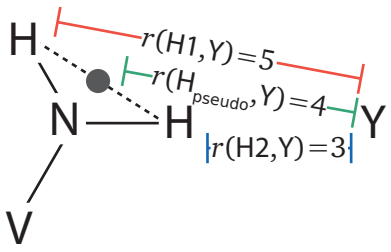


Figure 2.6: *Pseudo hydrogen position.* Schematic view of the positioning problem of a pseudo hydrogen (blue circle). Y is an H-bond acceptor atom, V marks the ring of either adenine, cytosine or guanine. If the artificial atom is placed right between the real protons, distance $r(\mathrm{H}_{\mathrm{pseudo}},\mathrm{Y})$ is smaller than $r(\mathrm{H1},\mathrm{Y})$ and greater than $r(\mathrm{H2},\mathrm{Y})$.

2.3.3 Quasi Energy Function

Now, that all necessary distances are available, we can go on calculating interaction energies using Coulomb's law:

$$E = \frac{q_1 q_2}{Dr} \tag{2.1}$$

where E is the energy, q_1 and q_2 are the charges of the atoms, r is the distance between them and D is the dielectric constant.

For an H-bond, as a set of interacting atoms, all terms are summed up:

$$E = q_1 q_2 \left(\frac{1}{r\,(\mathrm{ON})} + \frac{1}{r\,(\mathrm{CH})} - \frac{1}{r\,(\mathrm{OH})} - \frac{1}{r\,(\mathrm{CN})} \right) \frac{1}{D} \tag{2.2}$$

where q_1 and q_2 are the magnitudes of the partial charges.

The Kabsch & Sander formula looks so simple, because the set of interaction partners defining 2D elements in proteins is very limited. When it comes to nucleic acids, the situation is more diverse due to the two different acceptor and donor groups involved in canonical base pairs (see Fig. 2.5). Instead of defining one expression per possible atom combination, we define a nomenclature on pairing partners to fit everything into a single formula. With X and Y as H-bond donor and acceptor atoms, A marks the atom adjacent to the acceptor and H is always the donated hydrogen. Table 2.1 lists the translation for each atom of a possible interaction.

Because we assume a single magnitude for the partial charges q_1 and q_2, further since we only treat relative energies, our quasi-energy function is given by:

$$E_{\mathrm{quasi}} = \frac{1}{r(XY)} + \frac{1}{r(HA)} - \frac{1}{r(HY)} - \frac{1}{r(XA)} \tag{2.3}$$

Type	Atom in nucleobase	Label
Donor	A(N6), C(N4), G(N1), G(N2), U(N3)	X
Hydrogen	A(N6Hp), C(N4Hp), G(N1H), G(N2Hp), U(N3H)	H
Acceptor	A(N1), C(N3), C(O2), G(O6), U(O2), U(O4)	Y
Adjacent	A(C6), C(C2), C(C4), G(C6), U(C2), U(C4)	A

Table 2.1: *H-bond donor & acceptor mapping.* To simplify the definition of the formula describing H-bond scores between nucleobases, single atoms are grouped together. Classification follows the role atoms play in measuring distances for the Coulomb energies (see Fig. 2.5(b)). For the amino groups, Hp is the pseudo hydrogen artificially added (refer to §2.3.2).

While the results of this function are not correctly scaled, we will speak of *arbitrary units* (arb. units) when referring to them.

2.3.4 Identifying Base Pairs

With H-bonds defined by an energetic term, there is no real need of classifying them into base pairs for our purposes. Therefore the task of identifying canonical base pairs is only fulfilled in a basic, prototype-like manner.

On first thought, one might assume counting for the "right" number of interactions between two bases as the most simple method. But just accepting every two bases forming an H-bond as pair is even simpler. Of course, this is not exactly what is done, here. When searching for polar interactions, there is a high chance for finding bases which interact with more than a single base. Even single atoms are likely to have more than one partner in an energetically favourable state. On discretising the interaction landscape into base pairs, those situations have to be eliminated.

Our approach starts with finding H-bonds in an all-against-all manner, where all donor and acceptor sites of all bases in the whole sequence are evaluated by Eqn (2.3). All putative H-bonds are measured by their energy only. Well chosen cut-off values will avoid most nonsense interactions, like H-bonds over long distances. After this first filtering step, most bases will still show polar interactions with more than one partner in their direct neighbourhood. To bind pairs to single partners, the quasi energies between each two interacting bases are summed up and the H-bonds leading to the lowest energy are kept while all others are deleted. In many cases this should already account for counting H-bonds towards Watson-Crick pairs. Very likely, if two bases form an interaction on one acceptor-donor pair, remaining sites will also interact with the same bases. This leaves us with a list of base pairs, all with

a single partner. Artificially, canonical base pairs can be forced by deleting all non wanted pairs from that list.

2.4 Implementation

For evaluation, the simple, electrostatic-based model was implemented in a tool which could read 3D structures and return a list of H-bonds. This starting point was then used as the basis for a second tool, which could perform base-pair recognition like RNAview [25], but using the more sophisticated definition of H-bonds.

A task which both tools have in common is reading RNA structures from file. Since those are not available in high numbers, we did not want to loose any of them due to common but negligible format violations. Therefore we decided to implement our own PDB file reader, driven towards keeping as many RNA structures as possible.

For faster software engineering, core functions of the *GenomeTools* [33] analysis suite were used.

2.4.1 PDB Reader

Since there are already tools for parsing PDB files, here is our motivation for joining the common sport of implementing a personal version. Amongst the available choices, there was none which entirely met our needs. The top-most important feature is the ability to read data from a high share of RNA carrying PDB files. Such a request immediately forbids the use of available readers, which strictly rely on the PDB format, e. g. the implementation in the WURST framework [34]. The second feature heavily needed, points towards the content of a file, rather than a technical problem. Beside mere atom co-ordinates, PDB files contain much more information. While we are working with nucleotides, we are interested in remarks on modified residues. Those could have been introduced to a structure to solve problems in the process of X-ray crystallography, e. g. phasing [35], or by nature via post-transcriptional modification of a structure, e. g. in tRNA [36]. Just for the whole process of finding H-bonds, modified nucleotides are not a problem. But when it comes to energetic considerations in a 2D space, the literature energy model is only capable of the four canonical bases [21–23]. Hence, nucleotides with changed bases are candidates for substitution with bases manageable by the Nearest Neighbour model (see Table 2.2 for a list of modified residues recognised). With those features in mind, the development process was iterative, starting with the wwPDB format definition [37] and implementing exceptions when

encountering problems in our RNA subset of the PDB. The result is a PDB reader, entirely driven by *finite state machines* (FSM) [38].

	Nucleotide	
CCD id	modified	canonical
1MA	6-Hydro-1-Methyladenosine-5'-Monophosphate	Adenine
CAR	Cytosine Arabinose-5'-Phosphate	Cytosine
OMC	O2'-Methylcytidine-5'-Monophosphate	
5MC	5-Methylcytidine-5'-Monophosphate	
PGP	Guanosine-3',5'-Diphosphate	Guanine
2MG	2N-Methylguanosine-5'-Monophosphate	
M2G	N2-Dimethylguanosine-5'-Monophosphate	
OMG	O2'-Methylguanosine-5'-Monophosphate	
YYG	4-(3-[5-O-Phosphonoribofuranosyl]-4,6-Dimethyl-8-oxo-4,8-Dihydro-3H-1,3,4,5,7A-Pentaaza-S-Indacen-Ylamino-Butyric Acid Methyl Ester	
7MG	7N-Methyl-8-Hydroguanosine-5'-Monophosphate	
PSU	Pseudouridine-5'-Monophosphate	Uracil
BRU	5-Bromo-2'-Deoxyuridine-5'-Monophosphate	
H2U	5,6-Dihydrouridine-5'-Monophosphate	
5MU	5-Methyluridine 5'-Monophosphate	

Table 2.2: *Modified nucleotides.* List of modified nucleotides as found in our PDB subset with the corresponding canonical residues they are cast to. The *CCD id* column gives the identifier as found in a PDB file, *modified Nucleotide* its name and *canonical Nucleotide* the substitutional canonical base.

The reason for setting the reader up as FSM just follows the given structure of PDB files. In top-down direction, the data is organised in different types of records, e. g. REMARK records for extra information and ATOM records for coordinates. Since most of the different record types have their own organisation of information, they have to be parsed by different schemes. This can be easily reflected by modelling one FSM per record type, where the states correspond to the data fields, read by the transitions from each line of a file. Beside the line-wise data, some records may be considered en-block, forming information of higher order. That is, residues are formed by several atoms, sequences by several residues, and so forth. This means, FSMs for parsing records horizontally are not enough, at least one machine is needed gathering additional

information from what is produced by the lower order parsers. This implies a hierarchy of FSMs as shown in Fig. 2.7. Therein the different machines are invoked for transitions between the states of the calling FSM. Of course, the single automatons could have been pulled together into one large FSM, making the scenery unnecessarily confusing. Instead, we organised the machines as separated vertical and horizontal parsers.

In more detail, PDB files are parsed in two larger steps: first the preamble is evaluated followed by the body of the file. As preamble, we assume all lines before the coordinate section. Since the only information read from this part of a file are remarks of type 101, carrying modified nucleic acid residues, and the identifiers of heterogens from HET records, no individual FSMs are provided, here. This data is fetched in a more static way and does not really need own automatons.

The body parser is organised in two FSMs (Fig. 2.9 & 2.10) plus a general data-field reader (Fig. 2.8). This field parser only reads cells of data from a PDB file marked by a certain width or terminal character. Terminators are set on initialisation from the Init to the Read state. Here, a single character is read, exit conditions are checked, and a sequence of characters is stored. Towards the EOC (end of cell) state, data may be validated and cast into an appropriate type.

The field parser is needed to detect many states in the PDB parsing FSM illustrated in Fig. 2.9. This may be regarded as “master parser” for the body section of a file, since it invokes record-based FSMs while running over the file from top to bottom. On the implementation side as a program, it also arranges the information found into a higher degree of organisation. On the conceptual side, the machine starts searching for an ATOM or MODEL record. Following the opening of a new model, coordinates are read by the ATOM state and its transitions. After each set of coordinates read, a chain terminator may occur, followed by more atoms or other coordinate data. For each file processed, the ENDMDL state has to be visited at least once. Even if the atoms read are not enclosed in a model in the file, our implementation stores everything as a model internally. Once a model is closed, more may be read or the parsing process ends.

For defining polar interactions, only the coordinates, element and the position in a residue of an atom is needed. Since we also reconstruct the sequence (primary structure) solely from the coordinate section of a PDB file, only one further FSM is invoked by the PDB parser, the ATOM record parser presented in Fig. 2.10. This is also true when considering HETATM in addition to ATOM records, since they are of equal structure. The model strictly follows the wwPDB format description [37] with one exception. The alternate location indicator is empty for some files in our subset of the PDB, leaving us with nothing to

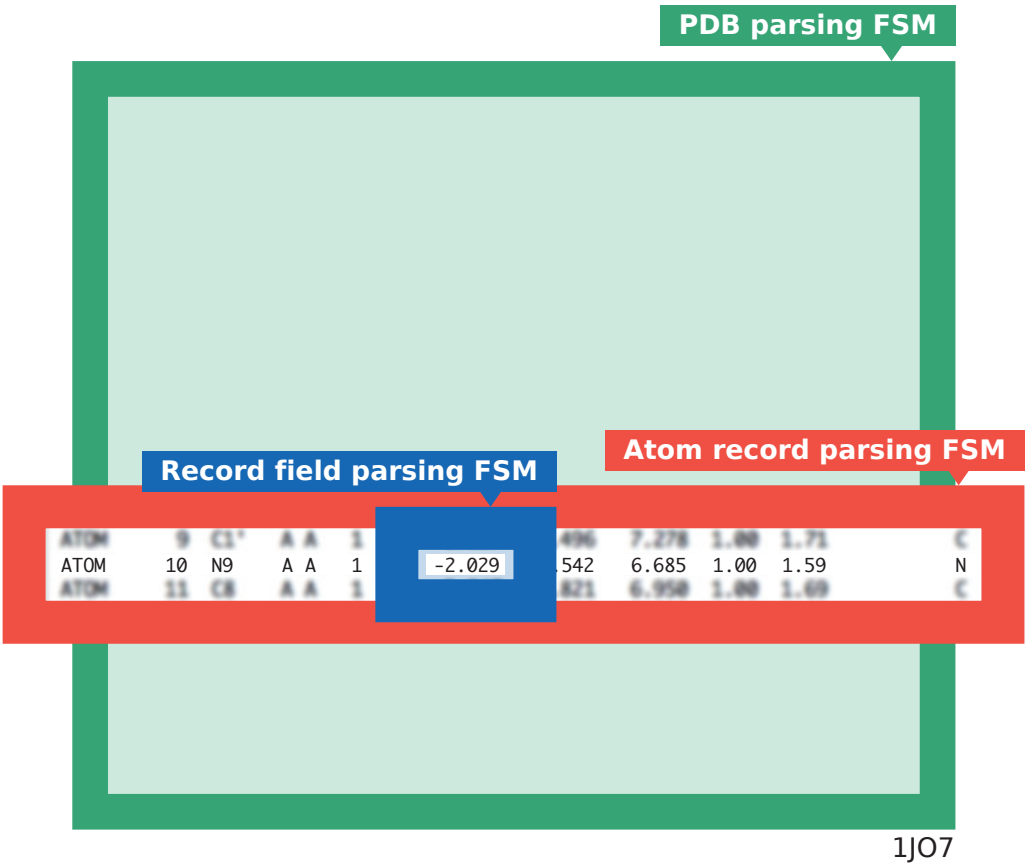


Figure 2.7: *Overview of the PDB file parsing FSMs.* Each FSM is described on its own in separate figures, while this is a sketch how they play together. The PDB parsing FSM (Fig. 2.9) operates on the whole file, detecting different records in a top-down manner and the syntax behind. For example it stores different models of a file by detecting surrounding MODEL and ENDMDL entries. Encountering an ATOM entry, it calls the atom record parsing FSM (Fig. 2.10). This reads the information in horizontal mode, field by field. Thereby the states are modelled following data types of the record, e. g. Cartesian coordinates. As a FSM invoked by both of the former, the field parser (Fig. 2.8) reads data cells of records character-wise.

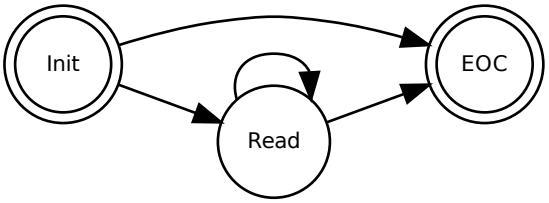


Figure 2.8: *Record field parsing FSM.* Used by the PDB reader to fetch field data of records and feed them to the other FSMs. Data is read character-wise up to a certain field width or until a terminator character occurs.

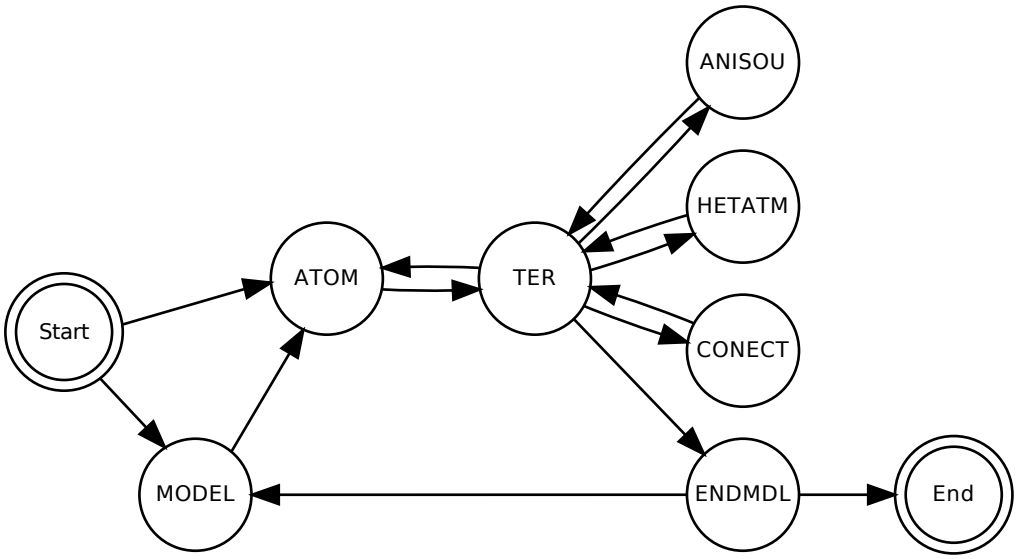


Figure 2.9: *PDB parsing FSM.* Modelling only relevant records, this parser is specialised on our subset of RNA-carrying PDB files. The artificial start state is needed because the data retrieval process skips the preamble of a PDB file, reading only atom information. The end state exists for the case of multiple models in a file and an extra bit of post processing. For the MODEL-ENDMDL tandem, the ENDMDL state is special. It does not require the corresponding record in a file to finalise a set of atoms. While PDB entries exist, not carrying a single model, our implementation of the parser keeps any set of atoms as a model which needs proper closing. The ATOM state is the one producing the highest workload. On the transitions towards it, the atom information is read. The remaining states are mostly there to handle certain records while not storing their information. The names of the states are the record types as found in the wwPDB format guide.

store in a state. This is the reason, why the ALTLOC state may be skipped by an immediate transition from state NAME to RESIDUE.

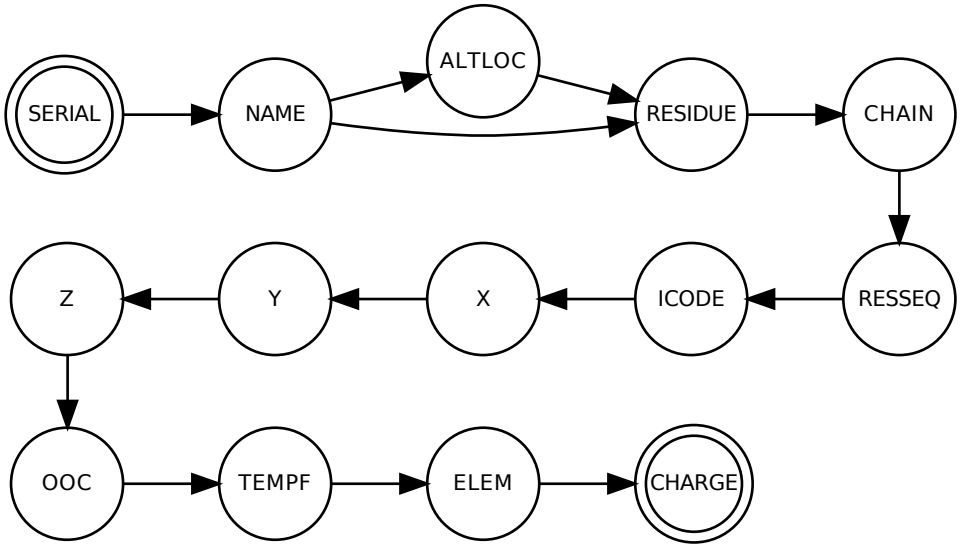


Figure 2.10: *Atom record parsing FSM.* Called by the PDB parsing FSM (Fig. 2.9) for the transitions towards its ATOM states, this parser is modelled along the fields of an ATOM record. Beside varying field widths, the only more dramatic disturbance of the strict format are some missing fields for alternate location indicators (ALTLOC) in our subset of the PDB used for evaluation. A list of states and their corresponding record fields may be found in Table 2.3.

2.4.2 Hydrogen Bond Finder

In this section, the basics of the implementation of the evaluation of polar interactions for nucleic acids are presented. More precisely, the program described here, is not a true H-bond finder, but a tool to assess the (quasi) energy of putative interactions. Its main usage is to proof that our model works in principle, and to define its working parameters.

In contrast to the natural habit of a software developer, running time is not a primary objective at this stage, but a proper implementation not using heuristics, e. g. distance cut-offs. Also the memory usage is negligible with a good assistance for the evaluation process in mind as one major goal. In other words, the idea is to drive an all-against-all computation of possible interaction partners while extracting as much information of the system as possible.

The core of the procedure is described in algorithm 2.1. It does not start by reading a PDB file, but assumes the already processed structure as input.

Input : PDB structure S
Output : List of interaction partners with quasi energy

1 $interactions \leftarrow \emptyset$, $acceptors \leftarrow \emptyset$, $donors \leftarrow \emptyset$
2 $i \leftarrow 0$
3 **foreach** $b \in S$ **do**
4 **if** `is_adenine`$(b) = True$ **then**
5 $acceptors \cup \{\langle b, \texttt{xyz}(b, \text{N1}), \texttt{xyz}(b, \text{C6})\rangle\}$
6 $donors \cup \{\langle b, \texttt{xyz}(b, \text{N6}), \texttt{pseudo\_h}(b, \text{N6})\rangle\}$
7 **else if** `is_cytosine`$(b) = True$ **then**
8 $acceptors \cup \{\langle b, \texttt{xyz}(b, \text{N3}), \texttt{xyz}(b, \text{C4})\rangle\}$
9 $acceptors \cup \{\langle b, \texttt{xyz}(b, \text{O2}), \texttt{xyz}(b, \text{C2})\rangle\}$
10 $donors \cup \{\langle b, \texttt{xyz}(b, \text{N4}), \texttt{pseudo\_h}(b, \text{N4})\rangle\}$
11 **else if** `is_guanine`$(b) = True$ **then**
12 $acceptors \cup \{\langle b, \texttt{xyz}(b, \text{O6}), \texttt{xyz}(b, \text{C6})\rangle\}$
13 $donors \cup \{\langle b, \texttt{xyz}(b, \text{N1}), \texttt{calc\_h}(b, \text{N1})\rangle\}$
14 $donors \cup \{\langle b, \texttt{xyz}(b, \text{N2}), \texttt{pseudo\_h}(b, \text{N2})\rangle\}$
15 **else if** `is_uracil`$(b) = True$ **then**
16 $acceptors \cup \{\langle b, \texttt{xyz}(b, \text{O2}), \texttt{xyz}(b, \text{C2})\rangle\}$
17 $acceptors \cup \{\langle b, \texttt{xyz}(b, \text{O4}), \texttt{xyz}(b, \text{C4})\rangle\}$
18 $donors \cup \{\langle b, \texttt{xyz}(b, \text{N3}), \texttt{calc\_h}(b, \text{N3})\rangle\}$
19 **end**
20 **end**
21 **foreach** $\langle a, Y, A\rangle \in acceptors$ **do**
22 **foreach** $\langle d, X, H\rangle \in donors$ **do**
23 **if** $a \neq d$ **then**
24 $e \leftarrow \frac{1}{\texttt{r}(XY)} + \frac{1}{\texttt{r}(HA)} - \frac{1}{\texttt{r}(HY)} - \frac{1}{\texttt{r}(XA)}$
25 $interactions \cup \{\langle a, d, e, i\rangle\}$
26 $i \leftarrow i + 1$
27 **end**
28 **end**
29 **end**
30 **return** *interactions*

Algorithm 2.1: *Calculating interaction energies.* The `is_`"`nucleotide`" functions are used to identify (modified) nucleotides (Table 2.2). Coordinates of atoms (Table 2.1) are collected from PDB residues by `xyz()`. Hydrogen coordinates are generated by `pseudo_h()` and `calc_h()` (§2.3.2). Distances are calculated by `r()`. i is used to keep interactions of equal donors & acceptors and quasi energy.

State	Field
SERIAL	serial
NAME	name
ALTLOC	altLoc
RESIDUE	resName
CHAIN	chainID
RESSEQ	resSeq
ICODE	iCode
X	x
Y	y
Z	z
OOC	occupancy
TEMPF	tempFactor
ELEM	element
CHARGE	charge

Table 2.3: *Atom parsing FSM states.* List of the states modelled in the parser for PDB ATOM records (Fig. 2.10) and the corresponding data fields of the wwPDB contents guide, "Coordinate Section", subsection "ATOM" [37].

However, file reading is described in §2.4.1. As output, a list of quasi energies together with the interacting residues is provided. In the output of the actual implementation, information provided may vary. Before energies are calculated, in lines 3 to 20 all residues of the input structure are iterated, and acceptor and donor atoms are stored in two different lists. To distinguish between nucleotides and atoms to be stored, possible candidates are verified by conditionals in lines 4, 7, 11 and 15. Thereby the `is_"nucleotide"()` functions check for the name-giving residues and modifications thereof, as described in Table 2.2. What is actually stored for each acceptor and each donor in the lists, are tuples of the current residue and coordinates. According to the branch taken in the conditional block, atoms following Table 2.1 are chosen and their coordinates extracted from the residue by function `xyz()`. A special case are donor atoms with their protons artificially placed. For them, functions `calc_h()` and `pseudo_h()` generate coordinates as discussed in §2.3.2. An example for an union of an acceptor list and a new coordinates/ residue set can be found in line 5. Here, the residue *r* we are considering together with the coordinates of the N1 and C6 atoms of an adenine are stored in the acceptor list.

After the input structure was entirely evaluated, calculation of our quasi energy term takes place in lines 21 to 29. The first loop runs over all stored

acceptors and the second over all stored donors. Thereby Y is loaded with the acceptor atom and A with its adjacent atom. For donors, X takes the coordinates for the donor atom and H for the artificial hydrogen. Following a test for operating on different residues in line 23, the energy is calculated as given by Eqn (2.3). The result is stored in a list to be returned by the algorithm, together with evaluated residues a and d.

The running time of algorithm 2.1 is dominated by the two nested loops at the end. The first loop only enumerates all residues, while the second iterates all donors for each acceptor in the list. Since both are always stored together, the size of the lists results out of the n residues of a structure. This gives the algorithm an asymptotic complexity of $O(n^2)$. The memory usage is asymptotically also $O(n^2)$, because for each acceptor values for each donor are stored.

The actual implementation of the algorithm also includes thymine as a source for acceptor and donor atoms but is omitted here for compactness.

2.4.3 Base Pair Finder/ 2D Reader

A simple variant of a tool to define a nucleic acid's 2D structure is implemented, based upon the interaction search tool of the last section. The basic idea is first to define H-bonds in a structure using a threshold, and just count interactions between bases towards Watson-Crick pairs. When thinking of competing interactions between more than two bases, it gets a bit more complicated.

The general idea is again not to produce a cutting edge tool, but a proof of principle, that our H-bond model is comparable to established ones in matters of RNA. With this claim in mind, we can easily reuse the algorithm (implementation) of the H-bond finder for measuring interactions. Also, problems which might occur, e. g. in the treatment of incomplete pairs, are just solved, without bringing the solution to an optimised state. Because of using the algorithm for finding interactions here, the running time cannot get better than quadratic concerning the number of residues of the input structure.

Algorithm 2.5 summarises an intuitive approach to read base pairs out of a 3D structure. The filtering of H-bonds to be considered, is done by procedure 2.2. This small routine picks all allowed interactions as reported by algorithm 2.1 and applies a threshold onto each of them. To test only for Watson-Crick pairings, `pair_is_allowed()` in line 3 is defined by corresponding rules. The result is still a list of interactions between atoms, most probably smaller than the input.

In the next step, these pairs of atoms are gathered into pairs of residues, by procedure 2.3. The list of interactions is searched for matching residues, summing up energies for a hit. Since residues may have both, acceptor and

Input : List I of interactions as delivered by algorithm 2.1, energy threshold e_{th}
Output : List of H-bonds

```
1 hbonds ← ∅
2 foreach ⟨a, d, e, i⟩ ∈ I do
3 |  if pair_is_allowed(a, d) = True then
4 |  |  if e ≤ e_th then
5 |  |  |  hbonds ∪ {⟨a, d, e, i⟩}
6 |  |  end
7 |  end
8 end
9 return hbonds
```

Procedure 2.2: *filter_interactions*. Create a list of H-bonds between nucleotides being allowed to form base pairs. The `pair_is_allowed` function returns *True*, if an acceptor a and a donor d might form a valid pair. i is used to keep tuples which are equal in the first three members.

Input : List H of H-bonds as delivered by procedure 2.2
Output : List of base pairs

```
1  basepairs ← ∅
2  foreach ⟨a, d, e, i⟩ ∈ H do
3  |  foreach ⟨a_t, d_t, e_t, i_t⟩ ∈ H \ {⟨a, d, e, i⟩} do
4  |  |  if ⟨a_t, d_t⟩ = ⟨a, d⟩ or ⟨d_t, a_t⟩ = ⟨a, d⟩ then
5  |  |  |  e ← e + e_t
6  |  |  |  H ← H \ {⟨a_t, d_t, e_t, i_t⟩}
7  |  |  end
8  |  end
9  |  basepairs ∪ {⟨a, d, e⟩}
10 end
11 return basepairs
```

Procedure 2.3: *pairwise_sum_energies*. Add up single interaction energies for pairs sharing acceptor- and donor residues. Residues may occur more than once in the returned list.

```
Input  : List B of base pairs as delivered by procedure 2.3
Output : List of base pairs with each residue only occurring once
1  foreach ⟨a, d, e⟩ ∈ B do
2      foreach ⟨a_t, d_t, e_t⟩ ∈ B \ {⟨a, d, e⟩} do
3          if a_t = a or a_t = d or d_t = a or d_t = d then
4              if e < e_t then
5                  B ← B \ {⟨a_t, d_t, e_t⟩}
6              else
7                  B ← B \ {⟨a, d, e⟩}
8                  break
9              end
10         end
11     end
12 end
13 return B
```

Procedure 2.4: *unify_base_pairs*. Clear a list of base pairs from considering a base in more than one pair. Function `break` forces the algorithm to step out of a loop immediately.

```
Input  : List I of interactions as delivered by algorithm 2.1, energy
         threshold e_th
Output : List of base pairs
   /* Create list of H-bonds                                          */
1  hbonds ← filter_interactions(I, e_th)
   /* Gather H-bonds to pairs                                         */
2  basepairs ← pairwise_sum_energies(hbonds)
   /* Solve pairs competing for the same residue                      */
3  basepairs ← unify_base_pairs(basepairs)
4  return basepairs
```

Algorithm 2.5: *Base pair detection*. Procedures 2.2, 2.3 and 2.4 put together to create a list of base pairs starting with a set of polar interactions. The list returned contains tuples of residues involved and an energy value.

donor atoms, line 4 examines all possible combinations of a current tuple of pairs. The newly assembled list is at maximum half the size of the H-bond list.

The list of base pairs returned by the last step may contain some residues in more than one pair. Therefore, procedure 2.4 deletes redundant entries. The comparison runs over all pairs, searching for shared residues. For a match, higher energies are immediately excluded from the list, avoiding considering the corresponding residue combination more than once. The result is our final list of base pairs.

Algorithm 2.5 queues these three procedures up to form our base pair finder. Since procedure 2.2 has only a single loop on the input, it has a linear running time. Procedures 2.3 and 2.4 can both be implemented to run with $\binom{n}{2}$ tests, skipping redundant comparisons. Asymptotically this is of order $O(n^2)$. For an average running time, we have to keep in mind, that usually the lists decrease from one step to the next. Since we do not store any data dependencies, the memory usage grows linearly with the number of interactions.

One important thing to note about the algorithm is in procedure 2.4. The way, elements are removed in direct comparison, creates a local minimizer. The global pendant would be to challenge all possible combinations of shared residues and then pick the one of lowest energy. An efficient algorithm for this problem is described by Nussinov [10] and needs $O(n^3)$ time and $O(n^2)$ space. Since in our case we do not need to fold an RNA molecule but occupy a given structure with short range interactions, the local variant should be sufficient. In practice, the problem of shared residues occurs nearby only between neighbouring residues, making an expensive global search obsolete. Additionally, moving away from a binary view on H-bonds, shared interactions are easily allowed in the model.

The actual implementation of algorithm 2.5 operates on a list of bases, all initialised with an unpaired state. Thereby, we do not need to meddle with non-paired sites because the algorithm updates only paired bases. The result of the pair detection process is then an annotated sequence of paired and unpaired bases.

2.5 Results

As mentioned before, the evaluation of our model is based on PDB structures. Since we are only interested in RNA, all entries not carrying it are first filtered. Evaluation is then performed in an all-against-all manner per structure. All nucleotides of an entry are matched with all other nucleotides in the same PDB file. If the model is capable to differentiate between interacting and non-

interacting sites, the calculated quasi energies should divide into at least two distributions.

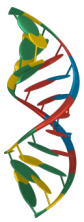
Before describing the procedure and the expectations of the evaluation, our data foundation is presented. After a short explanation about the filters used to create a reduction of the PDB, this contains a few statistics about the test set itself. A few structures will be reserved for testing a possible parametrisation of our model, that might come out of the process.

2.5.1 Test Set

A complete list of PDB entries used for testing may be found in Appendix B.

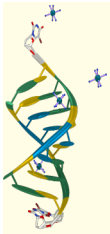
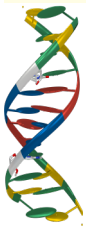
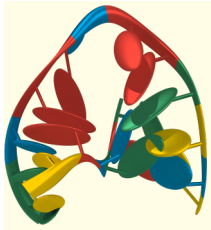
Assembling our list of RNA structures can be done immediately via the PDB web interface and its search tool [28]. In the *Advanced Search Interface*, the filters for *Structure Features* are used to set restrictions on the *Macromolecule Type* of our query. For retrieving a decent sample size, we need to include all entries carrying RNA, not only structures solely made of it. This means, all fields of the filter are set to *ignore*, while *Contains RNA* is enabled with *Yes*. Applying these settings leaves us with a list of 1070 entries, 314 of which only carry nucleic acids, and a total of 404 404 nucleotides.

Table 2.4 shows a list of 15 PDB entries, excluded from large-scale evaluation for a deeper investigation of the model parameters. To preserve the possibility to manually explore H-bonds and base pairs, the chosen structures are kept rather small. As an outlier, the structure of a ribosome [39] with 2904 nucleotides is stored in the small set. Beside this, the set contains non-Watson-Crick base pairs [40–43], base triples [44–46], pseudoknots [46, 47] and modified nucleotides [48, 49]. For comparisons, the original structures are either copied from corresponding publications or determined using Chimera.

Illustration	PDB Id	Size [nt]	Res [Å]	Method	Description
	157D	24	1.80	X-ray	RNA duplex containing two G(*anti*)·A(*anti*) base pairs

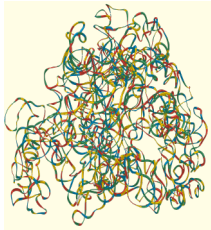
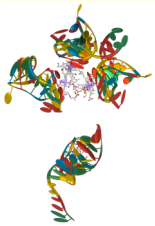
Continued on next page

Continued from last page

Illustration	PDB Id	Size [nt]	Res [Å]	Method	Description
	165D	18	1.55	X-ray	Mispaired RNA double helix
	170D	24	–	NMR/ MD	DNA dodecamer containing arabinosylcytosine
	17RA	21	–	NMR	Yeast binding site for phage GA coat proteins
	1A51	41	–	NMR	Loop D/ loop E arm of *E. coli* 5 S rRNA
	1AJT	19	–	NMR	Five-nucleotide bulge loop from *T. thermophila* group I intron
	1AKX	30	–	NMR	HIV-2 *trans*-activating region with argininamide

Continued on next page

Continued from last page

Illustration	PDB Id	Size [nt]	Res [Å]	Method	Description
	1B36	38	–	NMR	Hairpin ribozyme loop B domain
	1BZT	17	–	NMR	$tRNA^{Lys,3}$ anticodon domain with an $A^{+}C$ base pair
	1C2W	2904	7.50	Cryo EM	3D arrangement of the 23 S and 5 S rRNA subunits of *E. coli*
	1DDY	35	3.00	X-ray	Molecular recognition by the vitamin B_{12} RNA aptamer
	1EHZ	76	1.93	X-ray	Yeast $tRNA^{Phe}$
	1ESH	13	–	NMR	Stem loop C 5'AUA3' triloop of brome mosaic virus

Continued on next page

Continued from last page

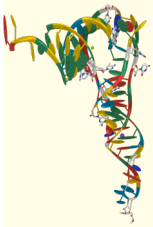
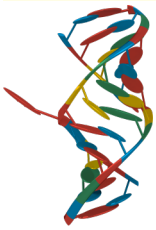
Illustration	PDB Id	Size [nt]	Res [Å]	Method	Description
	1FIR	76	3.30	X-ray	HIV-1 reverse transcription primer $tRNA^{Lys,3}$
	1I9X	26	2.18	X-ray	BPS-U2 snRNA duplex

Table 2.4: *Model evaluation set.* List of structures excluded from the model evaluation for parameter testing.

2.5.2 Evaluation Procedure

To evaluate our model, we calculate quasi energies using Eqn (2.3) and algorithms 2.1/ 2.5 on our test sets. For a perfect model, this should give us intervals for avoiding false positive and false negative interactions.

The values obtained during evaluation will be presented by histograms, showing the occurrences of quasi energies. To get a fine grained distribution, each energy value is its own bin. Since all values are brought to the same precision, some columns may still be merged.

Fig. 2.11 shows an example histogram that resembles the perfect result of our computer experiments. In an ideal world, the model would produce exactly two clearly separated distributions. The smaller and narrow distribution would represent the true positive interactions, while the larger heap would show all true negatives. We simply can expect a larger amount of non-interaction energies, because of our approach to collect the data. For each interaction, the corresponding partners are also computed in all other possible combinations for each structure considered. The standard deviation of the true positive distribution would ideally reflect the common distance ranges for H-bonds. Acceptable distances should produce quasi energies within the interval of the distribution, while different ones should significantly drop out.

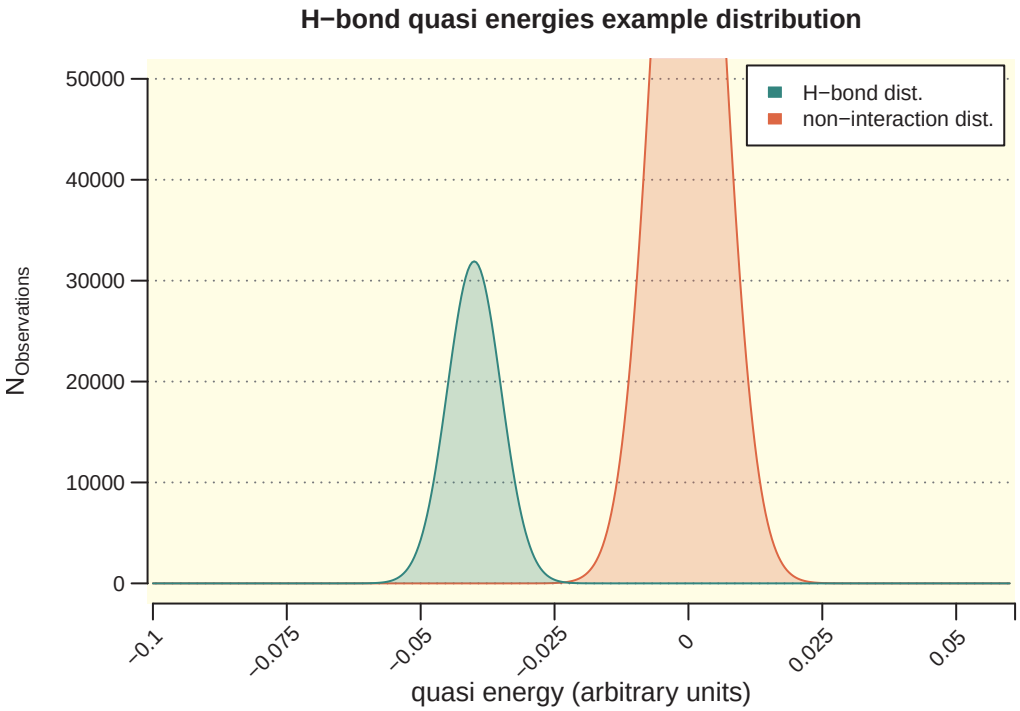


Figure 2.11: *Ideal quasi energy distribution.* This plot is completely made of artificial data. It was actively designed to look like close to the best possible outcome of our experiments. That is, the H-bond and the non-interaction distributions are entirely separated, giving us sharp intervals to define interactions. A Gaussian-like shape would allow to estimate the probability of an H-bond being present and remove the need to work with physical energies.

Once the model is evaluated in a large-scale manner, we will verify the results by investigating the real world structures from Table 2.4. For predicting base pairs, algorithm 2.5 will be used. Since we also get a list of putative interactions with this, pseudoknots and important non-base-pair patterns from the literature may also be covered.

2.5.3 Pseudo Hydrogens

Prior to the evaluation of the model at large scale, the problem described in §2.3.2 has to be solved. That is, we have to find an adequate position for the pseudo hydrogens of NH_2 groups. Three placements will be tested: Using the centre between the two hydrogens placed following standard geometry, using a centred position further away, and a placement closer to the nitrogen.

The procedure is similar to the general evaluation, while at this point only NH_2 groups are considered as hydrogen donors. For all three approaches, quasi energies are calculated and plotted as histograms. If this leads to the afore described shapes more than once, the method which leads to the smaller overlap of distributions will be used.

Centred Pseudo Hydrogen

This approach resembles the start situation described by Fig. 2.6. The pseudo atom is located right between the two hydrogens without considering the position of the nitrogen. If the hydrogens are missing in a target structure, they are placed using coordinates of the CCD.

Fig. 2.12 shows the energy histogram, where only the NH_2 group carrying the centred pseudo hydrogen is used as a donor. According to the idea presented in Fig. 2.11 we tried to divide the histogram by various density functions, but were only successful for a distribution of putative interactions. The function used is a Gaussian with an additional scaling factor a, as described by:

$$f(x;\mu,\sigma,a) = a\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.4}$$

Beside the missing distribution for the non-interaction energies, the low state of separation has the higher impact: the normal distribution has an integral of 231 292 values, but 95 357 of these values are obscured by overlap with the bulk of the data points. This leaves us with a rather small interval to accept interactions in a real world use case.

Looking at the histogram including all Watson-Crick hydrogen acceptors and donors presented in Fig. 2.13, the problems of this distribution get more prominent. Here, the fitted Gaussian of possible interactions does not overlap

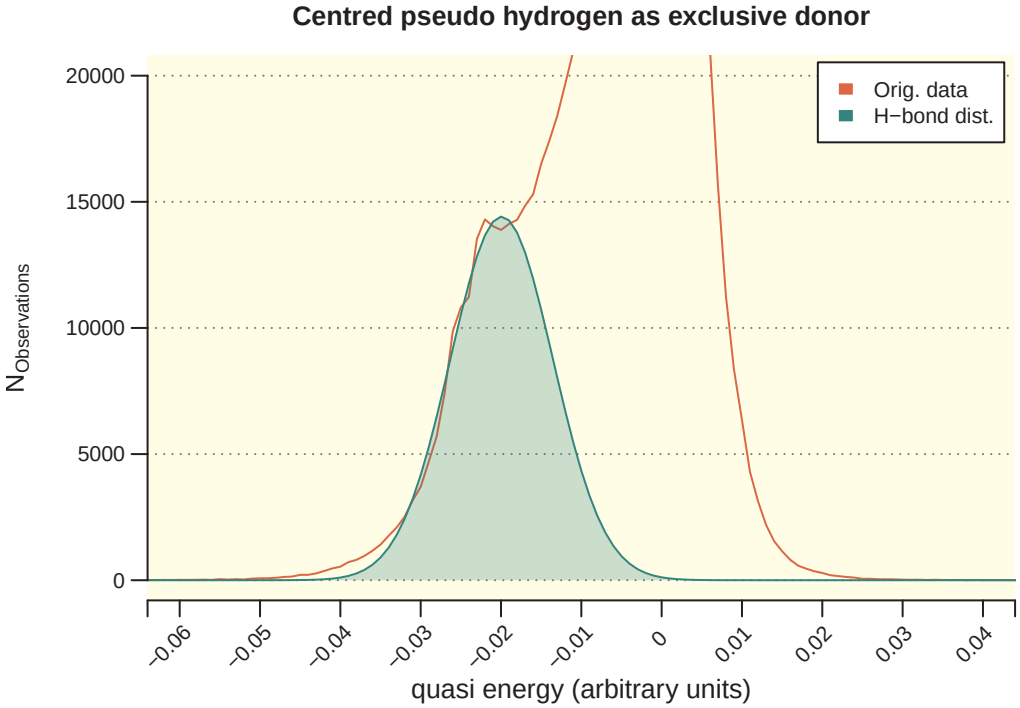


Figure 2.12: *Quasi-energy distribution, exclusive centred pseudo hydrogen.* The histogram (red) shows all energy values for NH_2 groups matched against all canonical acceptors. With a maximum of 867 102 928 hits with an energy of -0.001, the plot was cropped for a better representation of the modelled distribution (green). This Gaussian ($\mu = -0.02$, $\sigma = 0.01$, $a = 231.30$) marks putative true interactions. Within the interval of $(-0.047, 0.009)$, the curve covers 231 292 H-bonds, 95 357 (41 %) of which cannot be distinguished from the histogram as a whole.

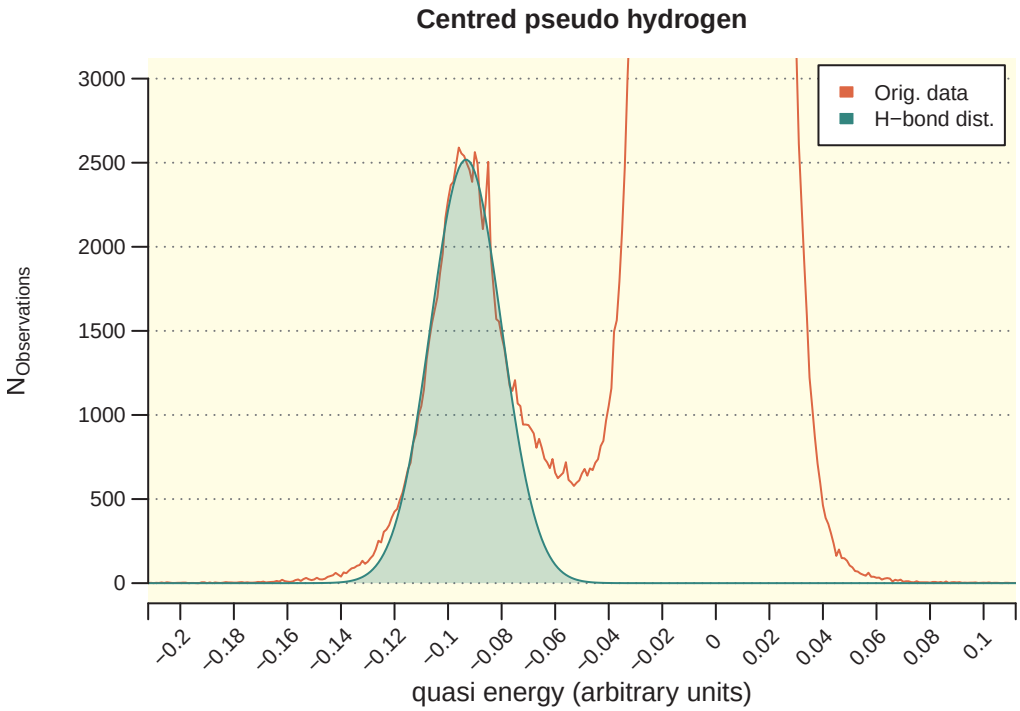


Figure 2.13: *Quasi-energy distribution, centred pseudo hydrogen.* The histogram (red) shows energy values calculated between all canonical hydrogen donors and acceptors. The protons of NH_2 groups were substituted by a single pseudo hydrogen, placed on the bisector between them. With a maximum of 1 394 649 197 values with an energy of 0, the plot was cropped for a better representation of the fitted distribution (green). This Gaussian ($\mu = -0.09$, $\sigma = 0.01$, $a = 83.98$) marks putative true interactions. Within the interval of $(-0.145, 0.040)$, the curve covers 83 971 H-bonds, 17 757 (21 %) of which are covered by the histogram.

with the interval defined for NH_2 groups in Fig. 2.12. Instead, the interactions issued by our pseudo hydrogen are indistinguishable from the distribution for dumped interactions. On the other hand, considering the remaining donor/ acceptor pairings, the model seems to work as intended. With a centred pseudo hydrogen, this would mean that parameters have to be split according to the donor in our model.

Pseudo Hydrogen 1 Å Away From Nitrogen

With this strategy, we stay on the axis between the nitrogen of an amino group and its adjacent carbon atom. For adenine these are C6/ N6, for cytosine C4/ N4, and for guanine C2/ N2. Instead of placing the pseudo hydrogen in the centre between the real ones, it is pushed to a distance of 1 Å away from the nitrogen. As an advantage, we do not need to care about missing hydrogens in a structure with this approach. Only the carbon and nitrogen atoms are needed to calculate the directional vector.

Results for this method are shown in Fig. 2.14 with NH_2 groups as sole donors. As for the centred pseudo hydrogens, we were only able to fit a single Gaussian into the histogram. The larger area, supposedly showing non-interaction values, is of a shape which was not covered by any tested density function. Compared to Fig. 2.12, this normal distribution is slightly shifted more outwards of the main area by its smaller mean. But by the larger standard deviation, the number of interactions being not clearly separated is more than 10 % higher.

In Fig. 2.15, a composition including all canonical donors is shown. The distributions show a small overlap, but still not enough to detect hydrogen bonds by a single set of parameters. In comparison to the last approach, the relative portion of putative interactions merged into the non-interactions is higher by 7 % while the integral is larger in total.

Pseudo Hydrogen 0.25 Å Away From Nitrogen

Similar to the last approach, again we place the pseudo hydrogen in one straight line with the carbon and the nitrogen atoms. With a distance of 0.25 Å, we are closer to the nitrogen than the original protons, this time.

The results presented in Fig. 2.16 show a similar shape as for the other two approaches. For an exclusive NH_2 donor, the normal distribution of putative interactions is not completely detached from the bigger part of the histogram. With the largest mean value of the testing field, one could expect that the majority of the distribution is merged with the histogram. But due to the small standard deviation, it provides the smallest overlap in relative numbers.

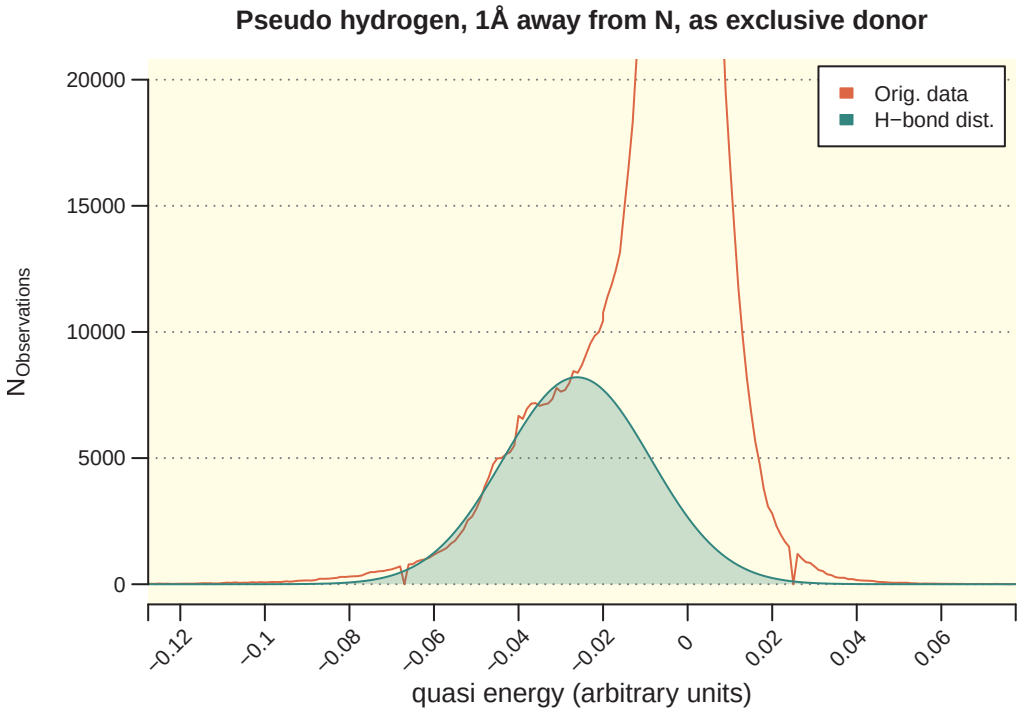


Figure 2.14: *Quasi-energy distribution, exclusive pseudo hydrogen at 1 Å.* The histogram (red) shows energy values for the all-against-all computation, using NH_2 groups as donors. Their protons were substituted by a single pseudo hydrogen, placed in front of the nitrogen. Because of the maximum of the histogram at $(-0.001, 865\,654\,113)$, the plot was cropped for a better representation of the fitted distribution (green). The Gaussian ($\mu = -0.03$, $\sigma = 0.02$, $a = 358.48$) covers 366 182 interactions within the interval of $(-0.100, 0.048)$. Thereof 197 801 interactions (54 %) overlap with the larger body of the histogram.

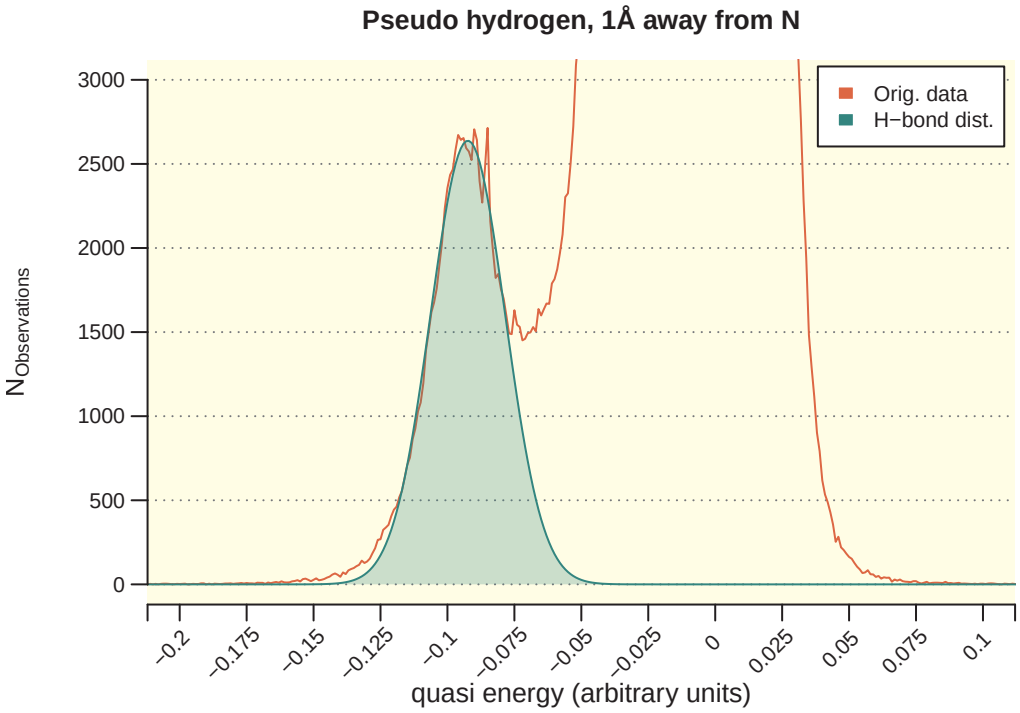


Figure 2.15: *Quasi-energy distribution, pseudo hydrogen at 1 Å.* The histogram (red) shows energy values for all possible acceptor-donor combinations. Because of the maximum of the histogram at $(0, 1\,393\,200\,238)$, the plot was cropped for a better representation of the fitted distribution (green). The Gaussian ($\mu = -0.09$, $\sigma = 0.01$, $a = 92.41$) covers 92 403 interactions within the interval of $(-0.147, -0.036)$. Thereof 26 309 interactions (28 %) are indistinguishable from the bulk of the histogram.

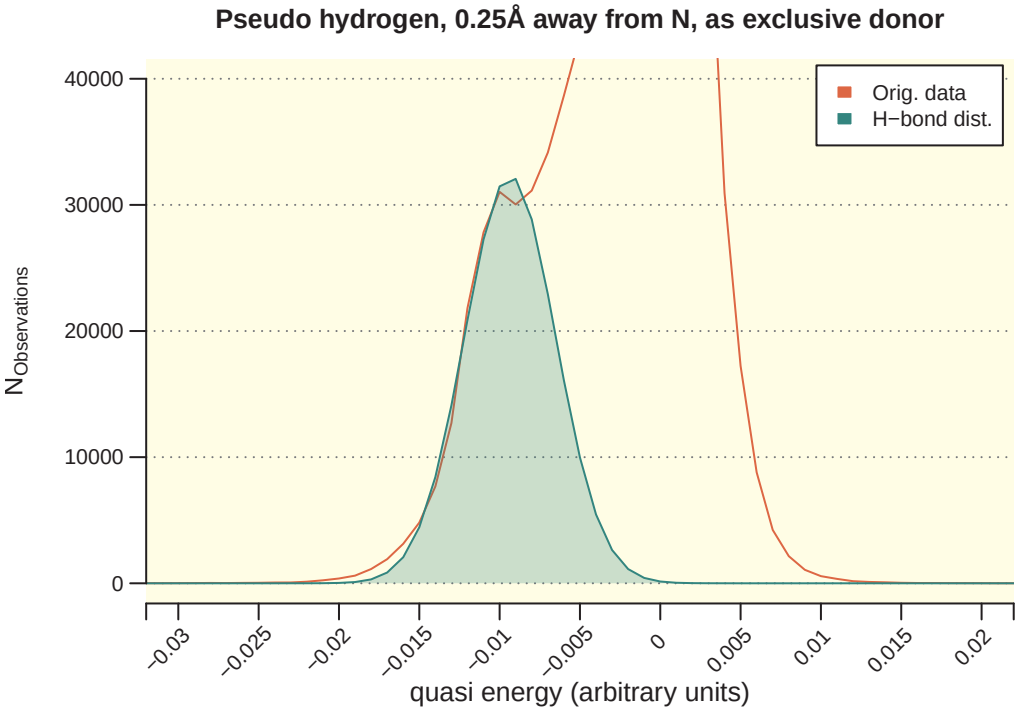


Figure 2.16: *Quasi-energy distribution, exclusive pseudo hydrogen at 0.25 Å.* The histogram (red) shows energy values for the all-against-all computation, using NH_2 groups as donors. Their protons were substituted by a single pseudo hydrogen, placed in front of the nitrogen. With a maximum at $(0, 868\,151\,553)$, the plot had to be cropped for a better representation of the fitted distribution (green). The Gaussian ($\mu = -0.01$, $\sigma = 0.00$, $a = 229.72$) covers 229 721 interactions within the interval of $(-0.022, 0.004)$. Thereof 87 691 interactions (38 %) overlap with the larger body of the histogram.

Thereby the total number of covered interactions is the lowest, but not dramatically far off the number for the centred placing.

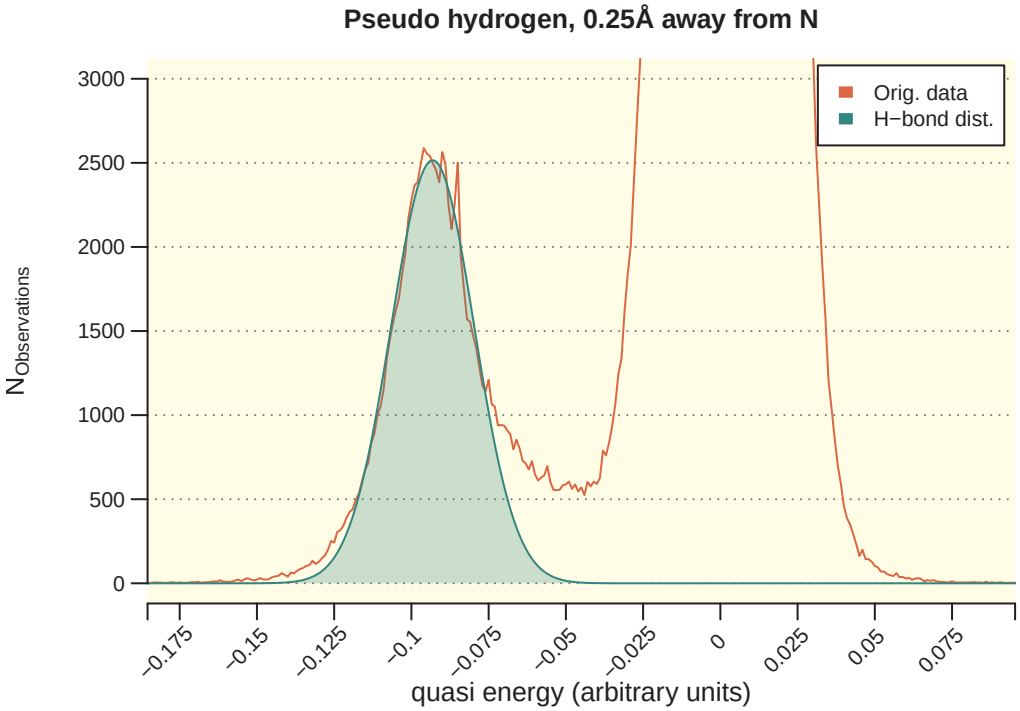


Figure 2.17: *Quasi-energy distribution, pseudo hydrogen at 0.25 Å.* The histogram (red) shows energy values for all possible acceptor-donor combinations. Because of the maximum of the histogram at $(0, 1\,395\,681\,600)$, the plot was cropped for a better representation of the fitted distribution (green). The Gaussian ($\mu = -0.09$, $\sigma = 0.01$, $a = 84.93$) covers 84 926 interactions within the interval of $(-0.146, -0.039)$. Thereof 18 364 interactions (22 %) are indistinguishable from the bulk of the histogram.

Also the plot for the complete all-against-all evaluation shown in Fig. 2.17 has similar features seen before. Considering the distribution of interactions with NH_2 groups, there is no overlap with the interval of the Gaussian here. The area merged into the histogram is lower than for the 1 Å shift but larger than for a centred pseudo hydrogen. The total number of interactions covered is slightly smaller than for the centred approach.

Conclusion

The first question is, does the system work at all as one would expect over the three approaches? Looking at the all-against-all plots, this seems to be the

case: the maxima are at the same positions and of similar value and the Gaussians covering putative H-bonds also have similar parameters. All together it seems like interactions not involving NH_2 groups are identical throughout the histograms.

Since the system seems to work in principle, the follow up question targets the treatment of missing hydrogens in the structures. Using average coordinates for single protons in NH groups leads to good results, like mentioned before. For our approach of a pseudo hydrogen substituting a rotatable pair, the results do not look that promising. We were able to fit a small Gaussian distribution, covering putative H-bonds, for each of the three positions. But none of them looked completely satisfying. Comparing the two plots for each placing, its obvious that they miss our first goal: an overlap of the majority of the Gaussians for accepted interactions. But even this solution using two distributions in our model, comes not without problems. None of the resulting intervals is completely free from the major part of the histogram, covering rejected interactions. This leaves only a small portion of the Gaussians to accept or reject a weak interaction, pointing towards a rather rigid H-bond dogma.

For the remaining evaluation, we still need to chose one of the placement strategies. For the all-against-all plots, the centred pseudo hydrogen and placing it 0.25 Å away from the nitrogen, look almost identical. Both histograms have a maximum at a quasi energy of 0 with about $1.39 \cdot 10^9$ members. The Gaussians are at ($\mu = -0.09$, $\sigma = 0.01$, $a = 84.93$) and ($\mu = -0.09$, $\sigma = 0.01$, $a = 83.98$) covering 84 926 and 83 971 interactions while 22 % and 21 % of them fall in the unclassified area. This is also remarkably similar to the histogram we get ignoring NH_2 groups as donors, shown in Fig. 2.18. As biggest difference, here the maximum is at $(-0.001, 527\,530\,047)$. The normal distribution is again very similar with ($\mu = -0.09$, $\sigma = 0.01$, $a = 84.29$) and 84 280 interactions. Since the pile of rejected interactions is missing contributions of the pseudo hydrogens, only 16 % of the Gaussian overlap. If the two placements are compared directly, without any other donor than NH_2 groups, the differences get a bit larger. While the maxima are still similar, at $(0, 868\,151\,553)$ and $(-0.001, 867\,102\,928)$, the Gaussians are relatively well separated with ($\mu = -0.01$, $\sigma = 0.00$, $a = 229.72$) and ($\mu = -0.02$, $\sigma = 0.01$, $a = 231.30$). For centred pseudo hydrogens, 231 292 interactions are covered, 41 % of which fall into the area of rejected interactions, and for pseudo hydrogens at a distance of 0.25 Å, 229 721 are enclosed. The latter distribution only overlaps at 38 % with members to be rejected, which is the smallest number of the whole test.

For the all-against-all comparison, placing pseudo hydrogens at a distance of 1 Å, the plot is still similar to the other ones. The fitted normal distribution has only slightly different parameters and the maximum is at the same pos-

ition with a value similar to that seen before. The differences start with the number of interactions covered by the Gaussian. Here, roughly 10 000 samples more are fetched, while 28 % are indistinguishable from the rest of the histogram. Also the bridge, connecting the distribution and the major part of the histogram, is around three times bigger than for the other plots. There, the connecting bit levels around the mark of 500, while here it lies around 1500. This shows, that the 1 Å placement really has an effect to the all-against-all plot, which we do not see for the other two strategies.

If we explore the plot only using NH_2 groups as donors, a major drawback of this placement surfaces. Since around 54 % of the fitted Gaussian mix with the non-fitted data, we can not give an interval for accepting interactions. This is because for normal distributions threshold intervals usually mirror around μ because events on both sides are equally likely. With this forbidding the use of this position for our pseudo hydrogens, we will go on using the distance of 0.25 Å simply because the overlap with the unclassified data is smaller than for the centred variant. Additionally we will do the evaluation on 3D structures using parameters split for donors in §2.5.5.

2.5.4 Model Evaluation

After deciding on a placement strategy for pseudo hydrogens, we explore the model more deeply, here. This means, the histograms are split for the groups making the interactions and compared how they assemble to the larger picture. Since we already have to split for the donors, the separation will go down to a one-on-one level, measuring each of the two acceptor- against the two donor groups.

The goal of this procedure is to verify that distributions of accepted interactions overlap for the individual pairings. Within the overlap we can then choose a threshold/ interval to gather H-bonds from.

The histogram showing only NH as donor with all canonical acceptors in Fig. 2.18 was already explained in the last section. Basically, this should present the sum of Fig. 2.19 and Fig. 2.20. Fig. 2.19 holds energy values, calculated between NH groups and nitrogens as corresponding atoms, while Fig. 2.20 uses oxygens.

On first sight, the numbers of the Gaussians seem not to add up. The integrals of the distributions for nitrogen- and oxygen acceptors together, exceed the number compared to the combined plot. The reason could be, that the distributions in the split plots cut across the original data line several times. This is an artefact from fitting the Gaussians separately without knowledge of each other. If we look at the data itself, e. g. the split maxima of 210 434 504

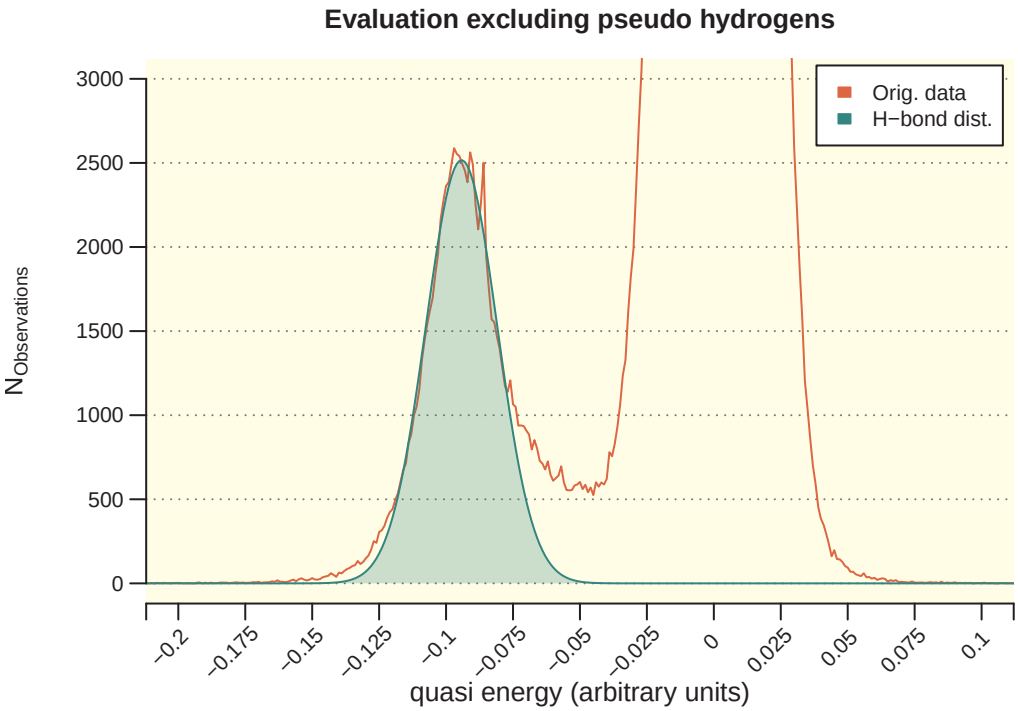


Figure 2.18: *Quasi-energy distribution, no NH_2 groups.* Values of the histogram (red) were calculated using NH groups as donor only. The maximum is at $(-0.001, 527\,530\,047)$, but was cropped for a better representation of the fitted distribution (green). The Gaussian ($\mu = -0.09$, $\sigma = 0.01$, $a = 84.29$) covers 84 280 values in the interval of $(-0.147, -0.041)$. Thereof 13 133 interactions (16 %) are indistinguishable from the histogram.

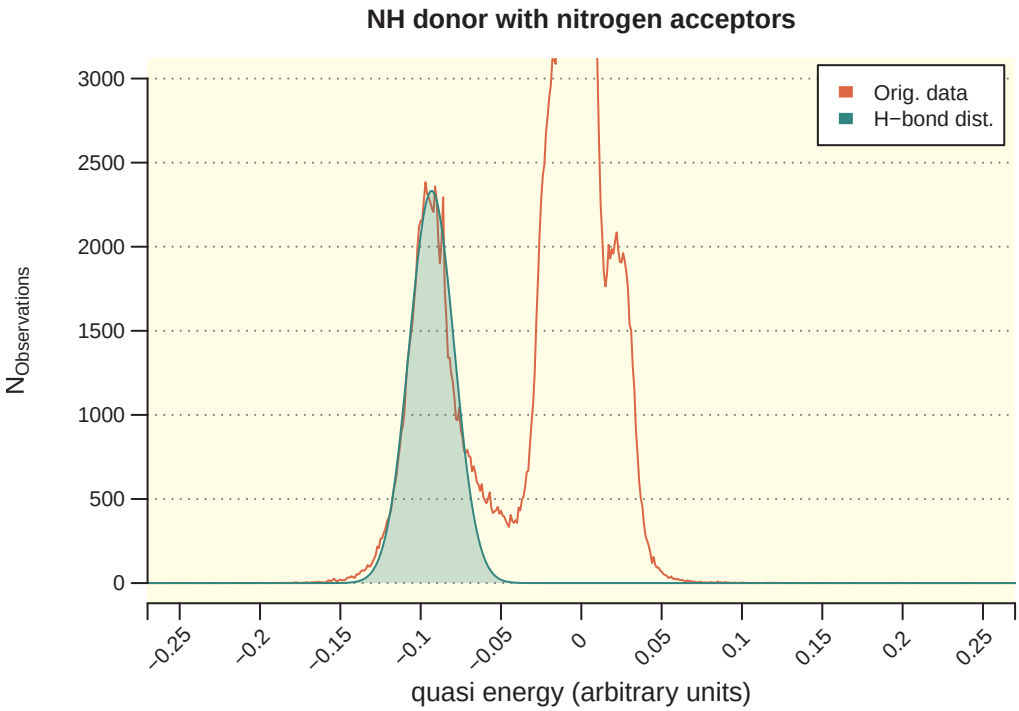


Figure 2.19: *Quasi-energy distribution of NH donors with nitrogen acceptors.* This plot shows the histogram (red) of energies calculated between canonical NH donor groups with nitrogen atoms as acceptors. With a maximum at $(-0.001, 210\,434\,504)$, the plot was cropped for a better representation of the fitted distribution (green). This Gaussian ($\mu = -0.09$, $\sigma = 0.01$, $a = 81.66$) covers 81 656 interactions within an interval of $(-0.148, -0.038)$. The interval may seem larger than the distribution, but counts full interactions. This extends the considered part to values greater or equal to 1. The number of interactions not distinguishable from the major part of the histogram is 5675 (7 %)

and 317 095 543 add up to the joined one of 527 530 047. This holds for all data points. As a sum, Fig. 2.19 and Fig. 2.20 perfectly reassemble Fig. 2.18.

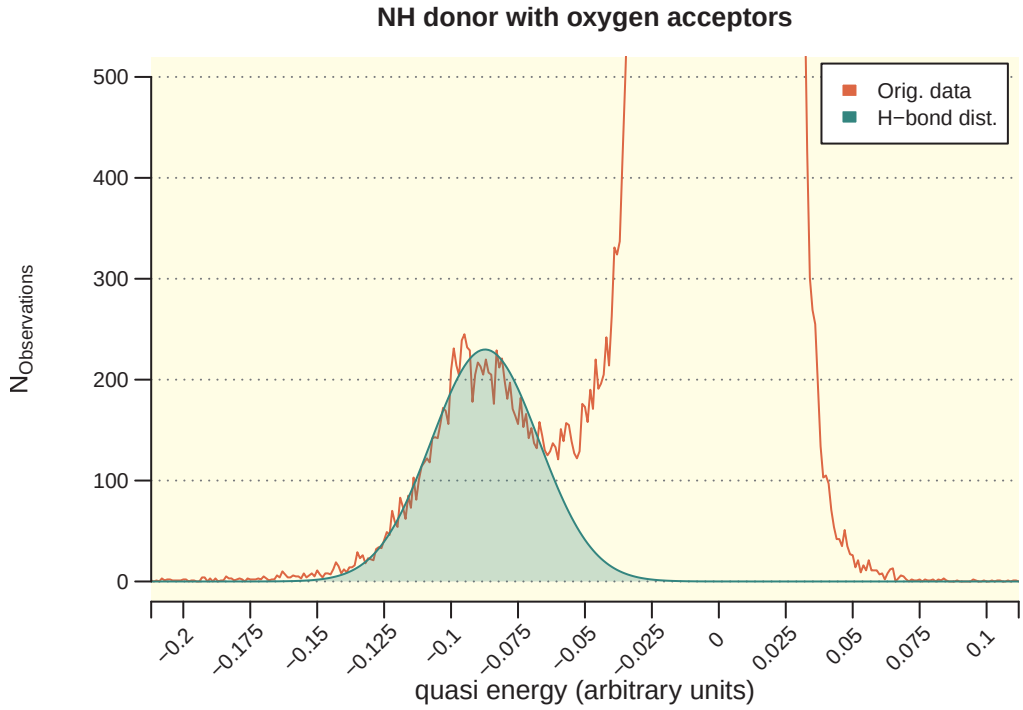


Figure 2.20: *Quasi-energy distribution of NH donors with oxygen acceptors.* The histogram (red) shows energies calculated for canonical NH donor groups with oxygen atoms as acceptors. Because of a maximum at $(-0.001, 317\,095\,543)$, the plot was cropped for a better representation of the fitted distribution (green). The Gaussian ($\mu = -0.09$, $\sigma = 0.02$, $a = 11.59$) covers 11 577 interactions within an interval of $(-0.153, -0.020)$. 2326 (20 %) of them are indistinguishable from the major part of the histogram.

Since the Gaussians for each acceptor group show a large overlap, here no further splitting of a derived parameter set seems to be needed. Especially when considering that the interval to accept interactions from will be cropped to avoid the low probability areas of the normal function. The last bit pulling this result away from being completely satisfying is an additional segmentation of the nitrogen bearing histogram. Around an energy value of 0.04, there seems to be another local extremum which could fit a distribution. What exactly it is, we could not determine. In the same region of the oxygen acceptor histogram, this area is totally covered by rejected interactions.

The combined histogram for canonical acceptors and NH_2 groups with their pseudo hydrogens is illustrated in Fig. 2.16. The split for nitrogen and

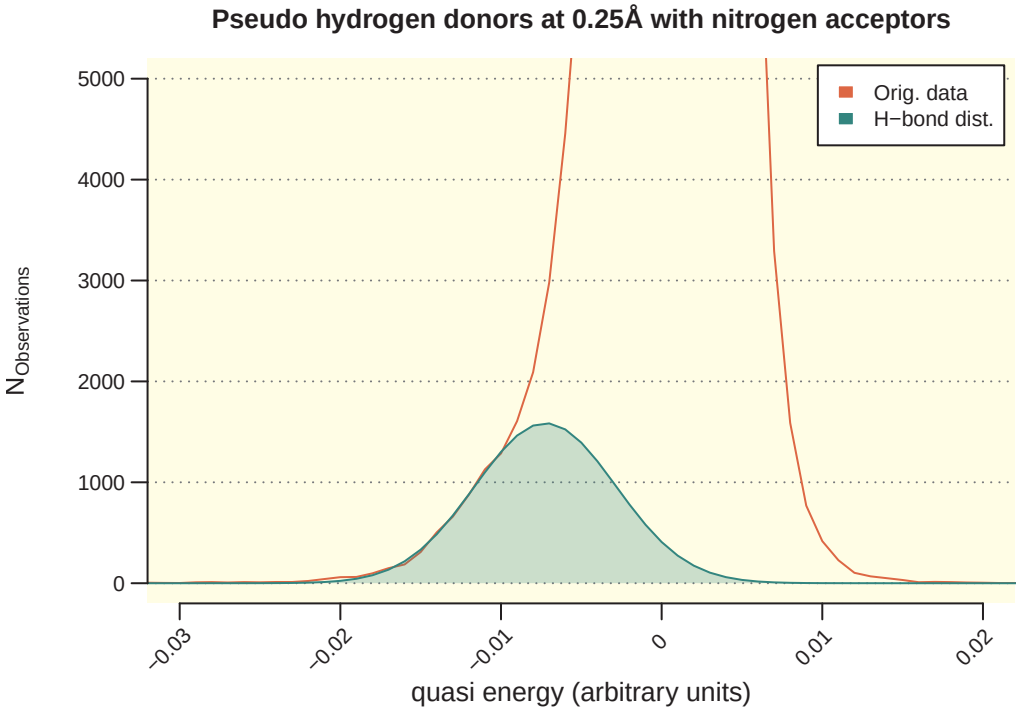


Figure 2.21: *Quasi-energy distribution of NH_2 donors with nitrogen acceptors.* The original data (red) shows the histogram of energy values for NH_2 groups matched with canonical nitrogen atoms as acceptor. The pseudo hydrogens are placed 0.25 Å away of the central nitrogen. With the maximum at $(0, 346\,254\,042)$, the plot had to be cropped for an adequate representation of the fitted distribution (green). This Gaussian ($\mu = -0.01$, $\sigma = 0.00$, $a = 17.46$) contains 17 462 interactions within an interval of $(-0.024, 0.010)$. 12 174 (70 %) of them fall into the body of the rejected part of the histogram.

oxygen follows in Fig. 2.21 and Fig. 2.22, respectively. As for the discussion of NH groups, adding those two plots up, leads to the combined plot. For technical aspects, this means the evaluation also works here. Looking at the integrals of the Gaussians, their sum fits even better together than in the latter case. The difference for NH_2 groups is 1625 compared to 8953 in the case of NH. The Gaussians themselves, with a magnificent, almost perfect overlap, show that for this donor also no more splitting of a possible parameter set is needed.

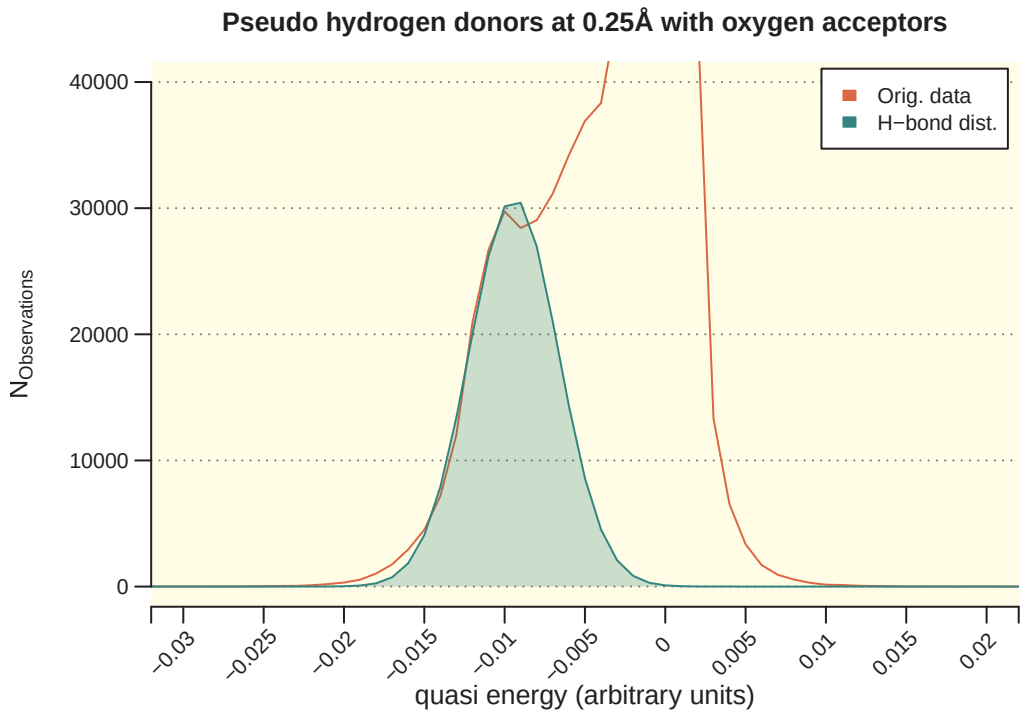


Figure 2.22: *Quasi-energy distribution of NH_2 donors with oxygen acceptors.* The histogram (red) shows energy values calculated between NH_2 donor groups with oxygen atoms as exclusive acceptors. The pseudo hydrogen of the amino group is placed at a distance of 0.25 Å away from the nitrogen. The maximum at $(0, 521\,897\,511)$ made it necessary to crop the plot, so the fitted distribution (green) becomes visible. The Gaussian ($\mu = -0.01$, $\sigma = 0.00$, $a = 213.89$) contains 213 884 interactions in an interval of $(-0.022, 0.004)$. From this interval, 132 002 values (62 %) cannot be distinguished from the major part of the histogram.

Problems start, when inspecting the acceptor plots on their own. For a nitrogen acceptor, the normal function is not separated from the rest of the histogram. The only sign that this is not a coincidental fit is the remarkable

agreement in the interval and distribution parameters with the other acceptor. But using oxygen atoms as acceptor is only marginally better, not really satisfying. Here, data and fitted distribution show a far better match. The left edge is clearly set out of the histogram, while the right is again completely absorbed into it. This forces us to only accept interactions from a small interval in the tryouts to define base pairs from this model in the next section.

2.5.5 Base Pair Evaluation

This part of the evaluation is looking for qualitative comparisons, finding an answer to the question if our model is helpful in defining secondary structure of RNA. That is, for the crystal structures listed in Table 2.4 interaction energies will be calculated and then gathered towards base pairs following algorithm 2.5. Unlike the original algorithm, two thresholds will be used in procedure 2.2 according to the donor group. While giving each example a bit of individual treatment, we can also have a closer look at challenging structural features like non-Watson-Crick pairs and pseudoknots.

To get the 2D reference structures, all targets are evaluated manually. Considering reliability, annotations from corresponding publications are accepted first order. If no complete secondary structure is available, Chimera is used to detect interactions. The H-bond patterns are then manually transferred to base pair lists, in some cases assisted by publications highlighting special structural features.

The perfect outcome of this test would of course be the correct detection of all base pairs. But this is just unrealistic. It will be almost impossible to get all the experimental details right since we are dealing with real world structures, here. At this moment our model completely ignores some factors influencing structure formation, e. g. ligands and solvent.

As thresholds, an interval of $(-0.113, -0.075)$ will be used for NH groups and $(-0.010, -0.008)$ for NH_2 groups.

Table 2.5 presents a short summary of the results, followed by a target-wise inspection.

157D

Structure 157D is a helix formed by two identical sequences CGCGAAUUAGCG. The peculiar thing are two GA base pairs, formed by G_4A_{21}, shown in Fig. 2.23, and $G_{16}A_9$. The secondary structure is correctly annotated by our model, once we stop forbidding non-canonical pairs.

This example is important because it shows the power of our approach: there is no need to actively test for special structural features, the mechanism

	Base pairs	
PDB Id	missing	wrong
157D	0	0
165D	2	0
170D	0	0
17RA	0	0
1A51	2	0
1AJT	1	0
1AKX	0	0
1B36	0	0
1BZT	2	0
1C2W	297	226
1DDY	0	0
1EHZ	0	0
1ESH	0	0
1FIR	0	0
1I9X	0	0

Table 2.5: *Summary of base pair detection evaluation.* "missing" counts the number of pairs annotated for the original structure but missed by our model, "wrong" are pairs that do not exist in the original structure.

reports any interacting bases until the pairing is explicitly not allowed.

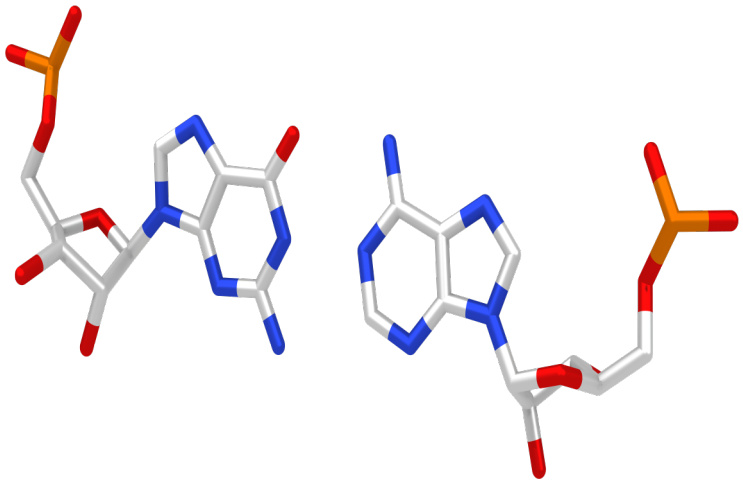


Figure 2.23: *GA base pair.* Nucleotides G_4 and A_{21} of structure 157D forming a G(*anti*)·A(*anti*) base pair [40]. The A(N6) interacts with G(O6) and G(N1) with A(N1). An additional interaction between G(N2) and A(N1) is supposed to stabilise the pair. As in the crystal structure, hydrogen positions are not available.

165D

This is another structure forming a helix by two identical sequences. The most interesting part of PDB entry 165D are two CU base pairs. Fig. 2.24 renders the pair C_5U_{13}, the other is supposedly formed by $C_{14}U_4$. All the expected H-bonds and base pairs are confirmed. The CU pairs referred to by Cruse et al. could not be recognised. But these are water- or cation-mediated interactions and rely on atom pairs that are 1.5 Å longer than any conventional H-bond [41].

170D

The first NMR structure in our list is again a helix by two dodecamer DNA sequences CGCGAATTara-CGCG [48]. While only showing canonical base pairs, the interesting bit is a modified residue ara-C_9. Cytarabine is a pyrimidine analogue by a cytosine base and an arabinose group used in chemotherapy.

With an unchanged interface, our model should still be able to detect Gara-C base pairs ignoring slightly changed geometry or interaction pattern by the sugar. For this example, the complete secondary structure is detected.

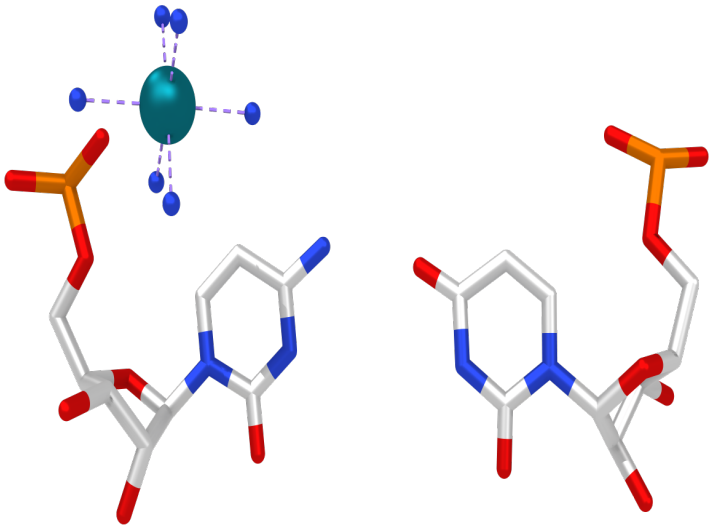


Figure 2.24: *CU base pair.* Assumed base pair of C_5 and U_{13} in structure 165D [41]. A direct H-bond should only exist between C(N4) and U(O4), while water (not shown) mediates between C(N3) and U(N3). The C(O2) interacts with a rhodium hexamine cation. As in the crystal structure, hydrogen positions are not available.

17RA

As first test for loop detection, a 21 nt sequence is used. In case of problems, this is small enough for in-depth manual inspection. Nevertheless, our model assigns all base pairs, the hairpin region and an asymmetric interior loop correctly.

1A51

This larger hairpin loop is a mixture of the usual and of non-canonical base pairs by Dallas & Moore [42]. Around the middle is a stack of three very special pairs. First, a non-wobble $G_{102}U_{74}$ pair where the interactions are not completely known. At least in our model, a proposed G(N1), U(O4) interaction seems crucial for the pair. A base pair which we miss is $G_{75}A_{101}$, with none of the calculated energies matching the threshold. This is somehow consistent with Dallas & Moore, since they also note problems with the traditional pairing interface. The last pair, $G_{76}G_{100}$, is again positively detected. The whole scene is illustrated by Fig. 2.25. Another pair missed by the model is $G_{81}U_{95}$. An explanation could be, that G_{81} is stacked closely with G_{96}. Dallas & Moore report, that an interaction of the six-membered rings is involved in the formation of the GU base pair. Our model is not capable of such triangles.

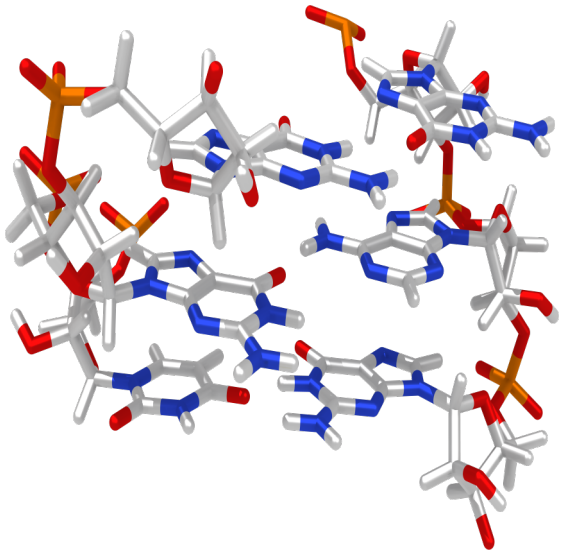


Figure 2.25: *GU, GA, GG base pairs.* Stack of $G_{102}U_{74}$, $G_{75}A_{101}$ and $G_{76}G_{100}$ base pairs in the loop E region of 5 S rRNA of *E. coli*. The GU pair is not the usual wobble base pair. The exact H-bond pattern is not absolute certain.

1AJT

This bulge loop is part of a *T. thermophila* group I intron with an interesting $A_{22}U_4$ pair as shown in Fig. 2.26. The unusual placement of A_{22} is result of stack formation with A_5 as claimed by Luebke et al. [50]. For our model, looking only for standard Watson-Crick interaction partners, this conformation is not accepted as base pair. All other pairs and unpaired bases are correctly annotated.

1AKX

Another example where our model assigns all structural features correctly is a *trans*-activating region of HIV-2, coupled with an argininamide [45]. The shape is a hairpin loop housing a small bulge in its stem. The interesting bit is the backbone distortion around the bulge, introduced by an $A_{27}U_{38}U_{23}$ base-triple. $A_{27}U_{38}$ form a standard base pair, U_{23} interacts via its Watson-Crick interface with the Hoogsteen edge of A_{27} and is therefore not detected. Despite the presence of the ligand, all base pairs around the interface are found with the argininamide itself being ignored by our model.

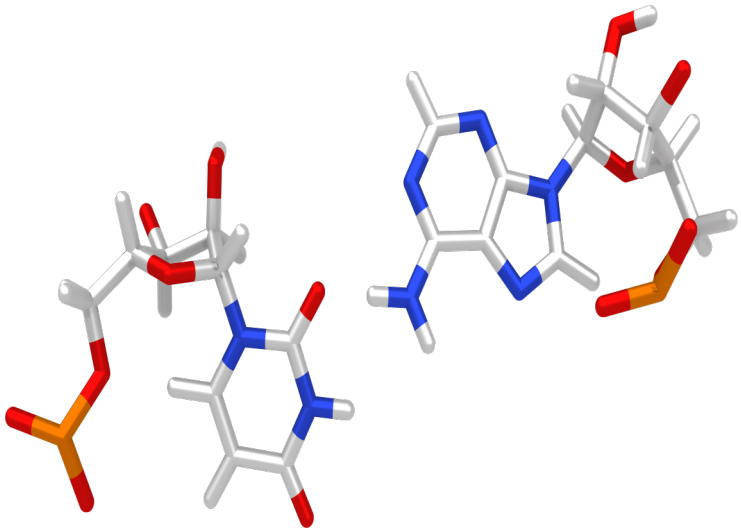


Figure 2.26: *Non-Watson-Crick AU base pair.* The $A_{22}U_4$ pair of a five-nucleotide RNA bulge loop, which is not interacting via the complete Watson-Crick edge [50].

1B36

As a usual hairpin with an asymmetric internal loop, the secondary structure is correctly annotated by our model.

1BZT

In this $tRNA^{Lys,3}$ anticodon loop we find two non-standard situations. First, it contains one pseudouridine (ψ) and then there is an adenine with an extra proton (A^+) at the N1, paired to a cytosine shown in Fig. 2.27. A base pair between the pseudouridine and A_{31} we miss, while it violates common distant constraints. Stryer gives a distance range of 2.4–3.5 Å for the acceptor and the heavy atom of the donor group [2]. The distance of ψ(N3), A(N1) is 2.5 Å but for A(N6), ψ(O4) it gets to 4.5 Å. The $A^+_{38}C_{32}$ is also not detected with a A^+(N6), C(N3) distance of 3.7 Å and a proposed C(O2), A^+(N1) interaction not being modelled. All remaining base pairs and the loop are positively identified.

1C2W

Entry 1C2W is a ribosome, but with only 7.5 Å resolution. Mueller et al. propose 807 base pairs [39] of which 510 were detected by our model. Our calculations suggest 226 base pairs which were not noted by the authors. The

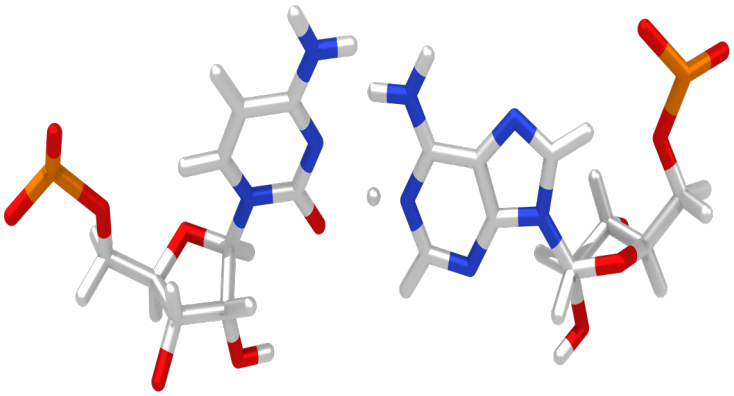


Figure 2.27: *AC base pair.* Putative $A^{+}_{38}C_{32}$ pair in the anticodon stem-loop of $tRNA^{Lys,3}$ [49]. H-bonds should be formed between A^{+}(N6), C(N3) and A^{+}(N1), C(O2).

structure is interesting because of its size, but at this low resolution does not merit further interpretation.

1DDY

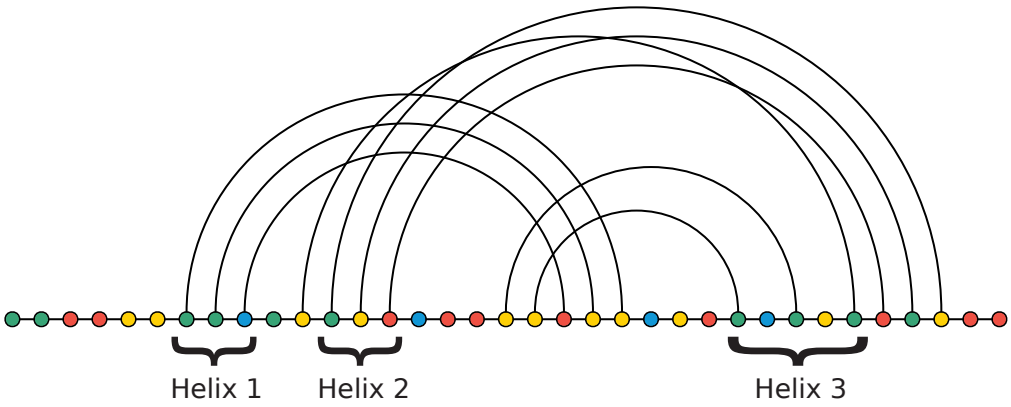


Figure 2.28: *Dome plot of the pseudoknots in the vitamin B_{12} aptamer.* From the loop of helix 1, helix 2 forms a stem with the 3' end of the sequence. Helix 3 starts with one base pair before helix 2 but closes its loop earlier.

A very interesting topology, annotated by Sussman et al., has two pseudoknots, four base-triples and one non-canonical AA pair [44]. The adenine-only base pair is formed by A_{14}, also interacting with the ligand vitamin B_{12}, and A_{31}, perpendicular stacking on C_{15}. Interactions are formed by Watson-Crick acceptors and donors. All three helices of the vitamin B_{12} aptamer are involved in

pseudoknot formation, which do not form simple kissing hairpins but spawn from each other as shown in Fig. 2.28. While we annotate all Watson-Crick pairs including the AA pair correctly, and by this also the pseudoknots, none of the base-triples are found.

The first triple is formed by G_7, C_{22} and U_{23}, with a canonical base pair G_7C_{22} and a single weak interaction between G_7(N2) and U_{23}(O4). With a distance of 5.8 Å of the interaction partners, the calculated energy does not fit in our interval to accept H-bonds. For the second triple, base pair G_8C_{21} and A_{25}, all additional interactions are established via the Sugar edge, which is not considered in our model. For the third triple, again the energies of Watson-Crick interactions are filtered out, while one further interaction makes use of the Hoogsteen edge. Here, G_{28} and C_{18} form a base pair, while G_{10} is linked to both of them at short distance. The last triple does not contain a complete base pair. U_{15} and A_{17} are both linked to C_{29} by only a single H-bond, which are to weak to be counted for pairing.

1EHZ

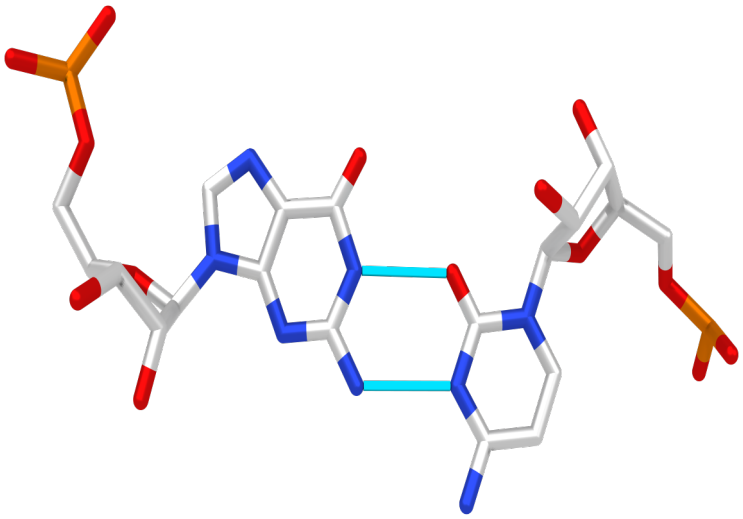


Figure 2.29: *Incomplete GC base pair.* Found in PDB entry 1EHZ, G_{15} and C_{48} share two H-bonds while not being annotated as base pair. The interactions (cyan) where detected using Chimera.

For this phenylalanine tRNA, Shi & Moore only deliver the traditional cloverleaf annotation [47], which is in agreement with our model. But we get three additional pairs: $G_{18}\psi_{55}$, $G_{19}C_{56}$ (Fig. 2.30) and $G_{15}C_{48}$ (Fig. 2.29). G_{18}

and G_{19} sit together in the D loop, connecting it to the TψC loop, stabilising the L-shaped tertiary form. G_{15} of the D loop, points to the beginning of the variable loop, also forming a known tertiary interaction. As these additional interactions help forming the accepted 3D structure of tRNA, and since they were annotated in other studies, we assume them valid. Especially the $G_{15}C_{48}$ couple even has its own name "Levitt pair" [51].

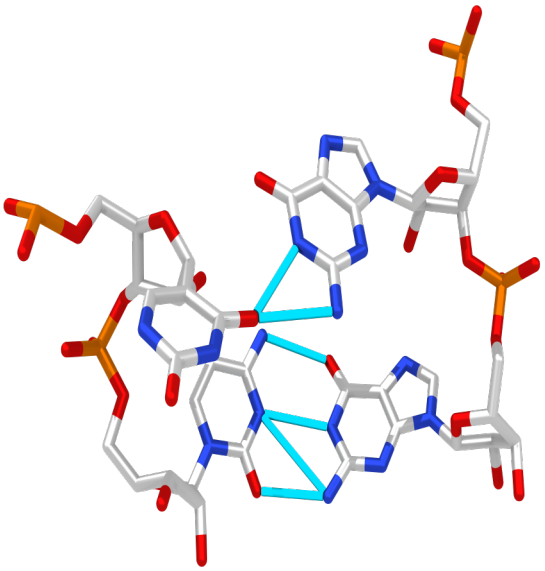


Figure 2.30: *Additional base pairs in 1EHZ.* Found by our model but not annotated by Shi & Moore [47]. The first interaction partners are G_{18} and ψ_{55}, connected by one H-bond with two possible end points (cyan) on the guanine side. G_{19} and C_{56} form the second pair, with all necessary interactions detected by our model as well as Chimera (cyan).

1ESH

This is a small hairpin with 13 nt and a loop of size 3. All pairs are annotated correctly.

1FIR

Structure 1FIR is published with a complete textbook annotation by Bénas et al. [46]. They present the usual cloverleaf as secondary structure and list all known tertiary interactions as found in their crystal. Ignoring Hoogsteen interactions and triples, the list of traceable tertiary pairs contains: a *cis* $G_{18}\psi_{55}$ pair, the *trans* Levitt pair $G_{15}C_{48}$, interactions between $A_{26}G_{44}$ and a somewhat

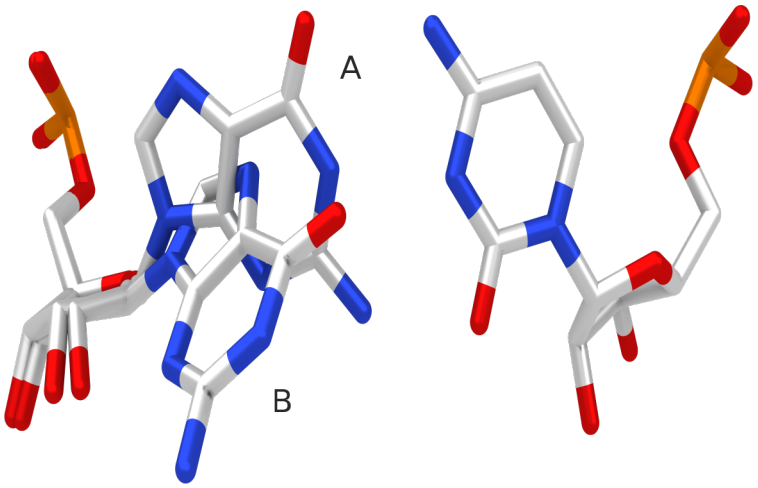


Figure 2.31: *A and B form of a tRNA GC base pair.* The $G_{19}C_{56}$ pair of $tRNA^{Lys,3}$ is claimed to exist in a closed (A) and open (B) form [46]. This scene was found as is in PDB file 1FIR. The atoms of G_{19} are duplicated with new coordinates and labelled with a "B".

special $G_{19}C_{56}$ pair. The latter one gave our model a bit of trouble. Because Bénas et al. assume a special function of G_{19}, being able to flip out of its pairing to interact with other molecules, they put coordinates for both forms in the PDB file. The result can be seen in Fig. 2.31, as drawn by Chimera using the original database entry. With this input, the implementation of the model finds a wrong annotation for several sites. Once the B form coordinates are removed, all interactions based on the Watson-Crick edge are positively detected.

1I9X

This is another small RNA duplex with a bulge loop of size 1. All base pairs are detected.

2.5.6 Discussion

For closing remarks on this chapter, the model itself and its use in detecting base pairs will be discussed.

The idea of a simple way to describe H-bonds, weak interactions defining spatial structure, in RNA, seems to work almost as intended. What we wanted is a definition skipping an elaborate survey of geometry and only a single value to decide upon. That is basically what the approach by Kabsch & Sander does: no consideration of any angles like Mills & Dean, only measuring distances and combining them into a Coulomb energy. The essential difference from

protein structures covered by Kabsch & Sander to RNA is that there are four possible interaction sites to be included rather than only one. This could lead to four different energy thresholds to be maintained while we would prefer a single one in terms of simplicity.

Using Coulomb energies as H-bond descriptor definitively works for RNA. What we did not achieve so far, is fitting the energy range for acceptance into a single interval. But at least the number of thresholds needed can be bound on the donor level and has not to be based on a per-base or chemical group policy. Also the energy distributions of §2.5.3 and §2.5.4 show, that simple distances are enough to discriminate interactions.

One chance to force the true positive distributions of the two donor groups into an overlap is the placing of pseudo hydrogens. From the results of their evaluation we already know that the current placement strategy is suboptimal. The effect of shifting the proton along the bisector of the two real hydrogens seems not to have the effect we are looking for. It is mostly affecting σ, allowing for a narrower distribution, which would make the interval to accept interactions from more prominent. But what we want is shifting μ to create a large overlap between NH_2 and NH distributions. Placing hydrogens in crystal structures is an old topic in the literature. Approaches range from squeezing the last bit of information out of electron density maps to inventing atomic networks upon coordinates [52]. For us, the idea is not to keep up with the way of sophisticated, complex methods but to follow a simple route. If modifying the distance between the nitrogen and its pseudo hydrogen does not improve the distributions scenery, probably adapting the angle does. But this would lead back to the problem of orientation for this rotatable pair of protons. An idea, on how to avoid this, may be observed in the standardised base pair in Fig. 2.2. There, the hydrogen is in line with its adjacent nitrogen and the acceptor atom. Looking at NH groups also, the arrangement is almost identical.

Borrowed from the standard coordinate frame, the next approach could be always to point the pseudo hydrogen into acceptor direction. Getting coordinates is simply done by calculating the vector pointing off the donor to the acceptor atom, multiplied by some constant and added to the donor coordinate. Thereby the factor setting the proton-donor distance would have to be determined empirically. A disadvantage of this approach is the increase in computational costs, since each single acceptor atom would need an individual position. But this already leads to a slightly related idea. Instead of calculating individual coordinates, an averaged position could be determined over all nucleic acid structures stored in the PDB.

Even more drastic, with atoms in line, it seems tempting to neglect the hydrogen contribution with terms $1/r(HA)$ and $-1/r(HY)$ in Eqn (2.3), but this was not even tested. One would lose the geometric considerations that are implicit

in the distance terms. It would, however, be appealing, since crystallographic coordinates rarely have hydrogen coordinates.

While thinking about improving the model, there is also room for extensions. What is missing at the moment are Hoogsteen interactions and any H-bonds along the Sugar edge. That both types are relevant, is already shown by our test set, e. g. 1DDY and 1FIR contain such interactions which are essential to their tertiary structure. The good thing is, none of the remaining edges have hydrogens on rotatable groups. Also an extensive annotation and classification scheme is already provided by Leontis & Westhof [53]. Their work provides information on all possible pairings far beyond canonical Watson-Crick interactions. The evaluation of the extended model could easily follow the same procedure as the basis model. One caveat in the extension of interaction sites may be the growth of data consumption. While asymptotically everything stays quadratic, with each acceptor added, we have to store n more elements. In practice this should not really be an issue for a single-molecule analysis. But probably the evaluation queue needs a bit of improvement concerning real disk usage.

The use of our model for detecting base pairs has been proven on a small collection of samples. For a large scale test, there was no database available, providing 3D coordinates together with secondary structure annotation in an appropriate format.

If we look at annotated base pairs our model missed, there are only somewhat special situations. For 165D, two CU pairs lie beyond our thresholds. While the implementation of the model is capable of the paring in principle, the bases seem to be to far apart for valid interactions. Also in 1BZT the missing Aψ pair violates common distance constraints. Example 1A51 provides two cases we missed. A GA pair which is also described as problematic by the authors and a GU pair which seems to need a third base to mediate the interactions. Such interaction triangles are not objective to our model.

While all other base pairs are positively detected, there is one outstanding result. For tRNA 1EHZ, only the cloverleaf is documented, ignoring the original L-shape. Assuming a published annotation as gold standard our model would be too sensitive, detecting three extra pairs. But in the process of identifying a reason why we find interactions where Shi & Moore do not list them, denial became harder in every step. The energies out of the model are not sitting at the edges of the tested interval, hinting a good probability for the existence of H-bonds. Also distances are within textbook range. Since the model parameters did not point to any problem, the structure was inspected using Chimera. But this only revealed, that the Mills & Dean method behind FindHBond detects the same H-bonds as we do, showing pairs which are in good Watson-Crick shape. As a last source of information, literature on tRNA structures was con-

sulted. In agreement this easily identified our extra pairs as both, valid and crucial for the native L-shaped conformation. Therefore we considered the finding of these pairs not as failure, while Shi & Moore probably just did not mention the obvious.

With the results of our little survey on real world structures, the model seems to be good enough for such tasks as basic feature extraction. But already this small test set shows, that complex structures are more than just a sum of base pairs. Especially base triples seem to occur more often than one would expect, as in structures 1AKX, 1DDY and 1FIR. But it also looks like the model could be extended towards complete automated annotation of RNA structures.

The last point that remains uncovered in this discussion are the algorithms for using the H-bond model. Here, the relevant bit is resource consumption. For single-molecule use cases, a quadratic dependency on the input data should not be a problem. But as soon as the model is integrated in some sort of database/ index based application, e. g. structure searching, at least the memory consumption should be lowered.

During base pair selection, memory usage only is quadratic, because we first compute the complete list of interactions and then select. Merging algorithms 2.1 and 2.5 to an online algorithm, immediately checking the use of an interaction for base pairing or drop it, would reduce the memory footprint. Such an algorithm would not necessarily be asymptotically slower, but harder to maintain and extend because of higher complexity.

A more common approach is the use of a cut-off distance, reducing the number of potential interaction partners. The complexity of the algorithms would be equal in the worst case, since all acceptor donor pairs would still be visited and at least one distance calculated. Based on this distance a decision could be made for further consideration or not. This would save memory and lower the constant calculation time per test. But one could even go further with a slightly modified approach. If two interaction partners of a potential base pair are too far apart, it seems very likely, that the same holds for all partners in this specific couple. Therefore one check could be enough to drop any more calculations between two bases. As an approximation which could gain a bit more accuracy in some cases, one could also just compare the centre of the atom cloud building a base. While the whole idea does not slow down the algorithms, it would dramatically reduce memory usage for collections of molecules.

Chapter 3

Sequence Design

Nowadays, molecular design is a well established topic in fields from pharmaceuticals to bioengineering and nanotechnology, which also includes RNA with its variety of regulatory roles. In the special case of RNA, designing means finding a sequence which will fold into a desired shape. While this is a strictly discrete problem, for every position in a structure one of four base types is chosen, we use a procedure from computational chemistry (self-consistent mean field minimisation) to treat it as a continuous optimisation problem. Unlike other attempts, this enables us to tackle arbitrary shapes and structures.

3.1 Introduction

Molecular design is a keystone in modern drug development, with many significant achievements in the past [5, 54]. But also in the fields of bioengineering and nanotechnology [55, 56], it makes the difference between just searching or driving a system to exhibit desired properties [57]. Especially for RNA, aside from the typical industrial use, there is also scientific interest in sequence design, e. g. for function determination [9].

3.1.1 The Problem – More Than Inverse Folding

Independent of a specific molecule, molecular design means identifying desired features and trying to construct a system which preserves those properties. This does not necessarily imply creating completely new molecules but most of the time to work along a scaffold, modifying certain attributes. Often an active compound should gain higher throughput under changed environmental conditions, or substrate specificity is to be tweaked. Designing

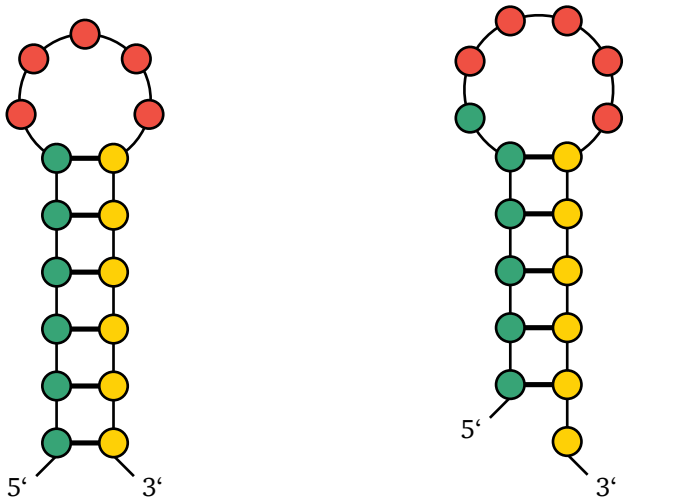
RNA seems to be simple in the first place, because most of the time it is seen as being assembled from a limited alphabet of building blocks. The connecting backbone is always the same while the four canonical bases vary in sequence. Therefore designing RNA molecules is often called sequence design or sequence prediction.

Usually, energy is modelled as a function of the secondary structure and its optimisation the classic structure prediction or folding problem. In this work, we note that energy is also dependent on the RNA sequence. Now, we regard the structure as fixed and treat the sequence as the variable to be optimised. This is often referred to as the *inverse folding problem* [15, 17, 58, 59]. The design task itself may be described as picking a sequence, providing certain features, out of a search space of size 4^n, with n as the sequence length. In a numerical description of the problem, for example Andronescu et al. optimised a distance metric on structures [60]. This metric reflects the difference between the desired fold and a predicted structure for some sequence. The goal is then to find a sequence leading to a distance of zero.

But minimising the distance between the target structure and the predicted fold of a designed sequence is only a solution and not the description of the optimisation problem. Getting the right fold is an obvious part of our objective. However, just because a sequence folds into a requested shape, it does not mean that this structure is also *stable*.

Predicting the secondary structure of an RNA sequence by finding its *minimum free energy* (MFE) will lead to an optimal configuration. But it is not unlikely that similar conformations exist for the same sequence with close energies. As an example, a small hairpin like in Fig. 3.1 is already enough. Fig. 3.1(a) shows the sequence with 6 base pairs, a loop of 5 bases and no dangling ends. In Fig. 3.1(b), one base pair is removed, thus the loop gets larger by one and the 3'-terminus has one unpaired base. Using the Nearest Neighbour model (NN), a literature standard scoring function, the energy difference is just $1\,\mathrm{kcal\,mol^{-1}}$. Wuchty et al. describe the relation between Fig. 3.1(a) and (b) as *suboptimal folding* and list a variety of scenarios why this may occur [61]. First of all, they note that the energy parameters of any model may be inaccurate. This means, if a sequence has several structures with similar energy, any of them may be the most stable one. Additionally, unknown biological constraints or certain physiological conditions may force a sequence into another state than the calculated MFE structure.

Especially when modelling sequences for two-dimensional structures, there will be another problem. Naïvely, simple sequences such as in Fig. 3.1 are a valid solution to the energy minimisation problem. In three dimensions, this type of sequence is unlikely to fold to a unique structure. As soon as there are symmetries or repetitions, alternative conformations become possible with



(a) $\Delta G = -12\,\mathrm{kcal\,mol^{-1}}$ **(b)** $\Delta G = -11\,\mathrm{kcal\,mol^{-1}}$

Figure 3.1: *Suboptimal helices.* The same RNA sequence in two different conformations. While (b) has one base pair less than (a) and a free base at the 3' end, the energy difference is only $1\,\mathrm{kcal\,mol^{-1}}$.

similar energy. These alternatives will also be significantly populated. Furthermore, this type of homogeneous sequence is rarely seen in nature. This suggests that one should add some variation to sequences, hopefully without disturbing their energetic properties.

This leads to three major tasks in RNA sequence design: positive design to allow intended interactions, negative design to discourage folding to the wrong form and artificial variation to give the sequence a more natural look. For being useful in real world scenarios, the method should also allow fixing of substrings in the sequence. For example, when reinventing a tRNA, the anticodon loop is not allowed to change.

Dirks et al. also have a similar list of criteria for nucleic acid sequence design, describing them in more detail but with slightly different focus [58].

3.1.2 The Nearest Neighbour Model

Probably the first idea of a dedicated energy estimation for nucleic acids, neglecting the need of coordinates, is by Bolton et al. in 1962 [62]. The main use of it, until today, is calculating melting temperatures of PCR primers [63]. Although this seems to be sufficient for DNA primer design, it is known not to be that accurate [64]. In the 1980's early versions of the Nearest Neighbour model started to evolve, considering the immediate environment

around a base pair for energetic contributions [65, 66]. Albeit the basic idea is unchanged, having an additive scheme, the focus of the model has broadened with additional studies. The parameter sets for helical regions are now extended for the various kinds of loops in RNA secondary structure and non-Watson-Crick pairs [21, 22, 67, 68]. The level of detail even goes so far as to label certain loops of special sequence with individual parameters [69, 70]. The development of computational methods evolved, starting with H-bond counting [10] and exploring the complete parameter set today [15, 17–20, 22, 23, 61, 71]. Nowadays, the NN model is the most popular scoring function in the field of computational RNA.

The whole system is driven by Gibbs energy ΔG. If temperature effects are to be included, contributions in Eqn (3.1) can be considered as estimated by the NN model.

$$\Delta G = \Delta H - T\Delta S \tag{3.1}$$

In which ΔH captures the potential energy contributions, ΔS the entropic contributions and T has its normal meaning of temperature.

The NN model is favoured by computational scientists, because it does not define a force field but assigns parameters to the sequence and the state of nucleotides. Thereby, only the Watson-Crick edge is considered to be paired or not. For many loops, the model is even more simple, having parameters only depending on their length. To calculate the energy of a secondary structure, it may be decomposed into certain structural features, evaluated separately and summed up. Fig. 3.2 illustrates this idea for a single hairpin, showing the overlapping fragments to be scored.

Fig. 3.3 lists six fragment classes recognised by the NN model. Further divisions are formed by special geometries within the classes. An important example are tetraloops. They denote a certain kind of hairpin but with an additional sequence-dependent energy term, increasing stability [70]. Usually unpaired bases in loops are just measured by length up to 30 nt after which the score is estimated using a logarithmic scale. In general, contributions are defined by multiple terms per loop type, originating from the many exceptions in the model as indicated by Fig. 3.2(c). With this, the model has more than 16 000 parameters.

One major advantage of this scoring function is its low computational cost. Scoring a structure works in linear time depending on its size, and more important, local changes only require recalculation on fragments involved. That is, we change a base in the sequence and only update by parameters of the structural features affected. Another benefit is the simplicity of the model. Since the parameters do not heavily depend on each other, they can easily be modified. This is important for introducing artificial noise to a system. Look-

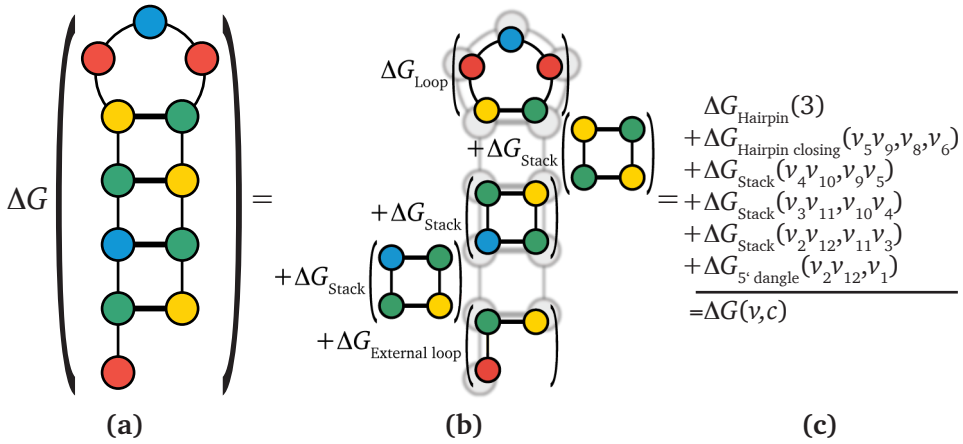
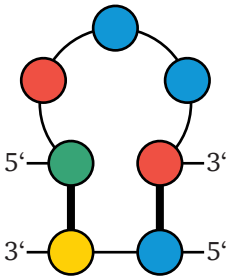


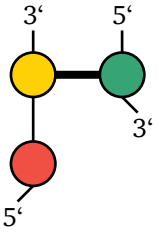
Figure 3.2: *Nearest Neighbour decomposition.* Calculation of ΔG of a hairpin loop q from sequence v using the NN model. (a) ΔG of a nucleic acid structure is defined by the sum of energies of overlapping structural features. In (b), fragment classes include dangling ends (*external loop*), doublets of base pairs (*stack*) and the unpaired region (*loop*). (c) sums up the energy terms referring to positions in v. Thereby the external loop has a contribution for the unpaired 5' end and the hairpin loop for the closing base pair with its free adjacent bases and the loop length. Thick black lines indicate base pairs.

ing at real numbers, it is fairly simple to estimate random values which enable the exploration of a suboptimal energy landscape without immediately driving the system into extreme states.

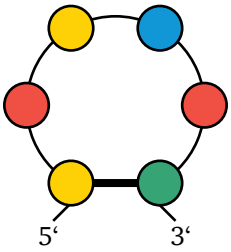
But its advantages also make the NN model a rather crude approximation of the real world. A major drawback is the lack of long range interactions. In terms of RNA, this usually refers to pseudoknots, which are base pairs spawning from a loop region. Since they may consist of Watson-Crick complements and can form stem-like loops, they could in principle be added to the model. In practice, the problem is the parameters, which are commonly determined experimentally by optical melting curves or calorimetry [21, 65–70, 72]. Unfortunately the corresponding measurements are extremely difficult to make on structures with pseudoknots [73]. Nowadays, most attempts to estimate energies associated with pseudoknots are rarely based on direct experimental measurements [73, 74]. For different interaction sites than the Watson-Crick edge, parameter determination seems to be problematic in a similar way. However, beside knots, other interactions seem to exist as described by He et al. [72]. They show an influence on stability of stacked base pairs by their context. Also by its experimental foundation, the model is limited in its applicability when believing in its results to be physical. Since all parameters have to



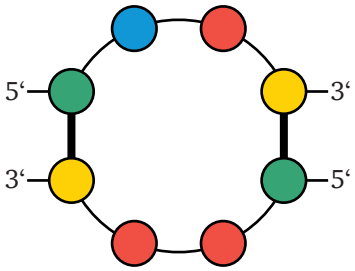
(a) Bulge loop



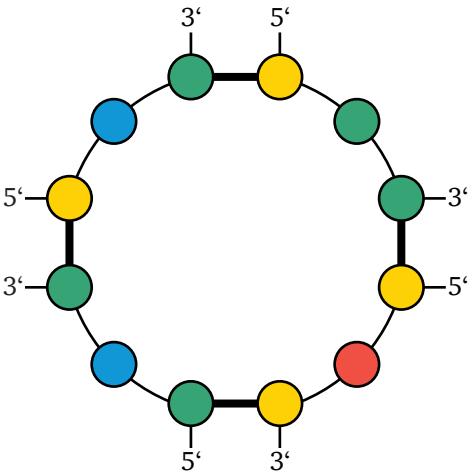
(b) External loop



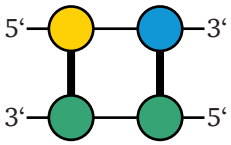
(c) Hairpin loop



(d) Internal loop



(e) Multiloop



(f) Stack loop

Figure 3.3: *RNA secondary structure loop types.* An overview of the structural features recognised by the NN model. For each class, only relevant nucleotides are shown. Thick black lines indicate base pairs.

be measured under equal conditions, concentrations, salts and temperature, all values produced are bound to them. Only the temperature may be varied since ΔS and ΔH values are also available.

The parameters we use are the same as of the Vienna RNA package [15].

3.1.3 State Of the Art

With the Nearest Neighbour model as additive scoring scheme and a folding algorithm based on *Dynamic Programming* (DP) [10, 15, 18, 75–77], approaching the inverse folding route seems not to be too complicated. The basic idea of current methods is, to modify a sequence, predict its structure and check if the changes disturb the target structure.

DP helps to keep the running time low in structure prediction, where a huge search space would have to be explored otherwise. The basic principle is to solve sub-problems of a task, store the solution and finally, to combine them to the global answer. DP especially gains effectiveness for problems which share sub-problems that only need to be computed once and reused. This is the case for RNA structures rated by the NN model which may be seen as sum of smaller structures.

However, with the running time of current structure prediction tools, exploring the exponential sequence space of an RNA structure is not possible. The algorithms of Nussinov et al. [10, 78], Zuker et al. [18–20, 71] and the refinement by Hofacker et al. [15–17, 61] all scale cubically ($O(n^3)$) with sequence length, even without considering pseudoknots. Including pseudoknots means a running time of at least $O(n^4)$, with a poor prediction quality for the most general cases [79, 80].

This makes sequence design not the straightforward approach it seems to be. Most notably, there are three tools attempting to conquer the task: `RNAinverse` by Hofacker et al. [15], RNA-SSD by Andronescu et al. [60] and INFO-RNA by Busch & Backofen [59]. Comparing the strategies they use to handle the huge search space, the three approaches look like an evolutionary line. The oldest one, `RNAinverse`, introduces the basic ideas. Next, RNA-SSD picks up those principles and significantly improves them. The most recent tool, INFO-RNA extends the improvements by RNA-SSD and tries to optimise solutions in the design process.

To avoid visiting to much of the vast search space, current tools put much effort in identifying the best starting point of sequence design. In practice, this means that the initial sequence is compatible with the target structure concerning base pairing. From there, possible violations of the structural constraints are detected and treated by mutations of the sequence. The process of changing bases is stopped, once all constraints are met or after a maximum

number of steps. One obstacle in this strategy is that changing bases may lead to new violations, which require more mutations. Because this could result in oscillations, the three tools discussed have different strategies to avoid this.

For estimating the conformation a sequence will adopt, `RNAfold` by Hofacker et al. [15, 17] is used, with a cubic running time and no support for pseudoknots. It has to be run each time the constraints are checked, making it the rate-limiting factor for the sequence space explored. To reduce the time spent on folding, `RNAinverse`, RNA-SSD and INFO-RNA split an input structure into substructures, solve and combine them to a complete solution. The idea is that shorter sequences fold faster. In sum, folding non-overlapping subsequences requires considerably less time than including all bases of a sequence in one run. For the sequence-splitting approach, it is assumed that sequences designed for structural fragments should most likely fold into the overall shape once combined. In effect, the complete sequence would need to undergo structure prediction fewer times, only as a control step at the end of a design cycle.

The main difference of the three tools to be discussed, are the exact strategies on how to initialise, split and alter a sequence. As a summary, their common approach is illustrated in Fig. 3.4.

`RNAinverse`

`RNAinverse` is part of the Vienna RNA package [15, 17] but has only seen minor modifications during its life. The initial sequence gets filled by random bases or base pairs, where the input structure requires them. The sequence is optimised along a cost function, minimising the distance of its MFE structure and the constraints. Following immediately out of the NN model, as splitting strategy, only hairpins are considered to be elongated towards larger structural features. Basically this means, that first loops of low complexity are designed and then joined together as multiloop branches. To solve constraint-violations, mutations of the sequence are introduced randomly and only accepted if they immediately improve the cost function. Thereby bases to be unpaired are changed as single while for pairs both bases are modified. If no answer is found after a predefined number of steps, the whole process is restarted with a new initial sequence.

A serious drawback of `RNAinverse` is its long running time. Therefore, to considerably reduce the search space, the algorithm can be forced to only alter positions which are wrongly paired once folded. At the same time, this lowers chances of finding an acceptable sequence for the input structure.

With Hofacker et al. already pointing out the problem of suboptimal folding described in §3.1.1 (Fig. 3.1), `RNAinverse` also offers a partition-function

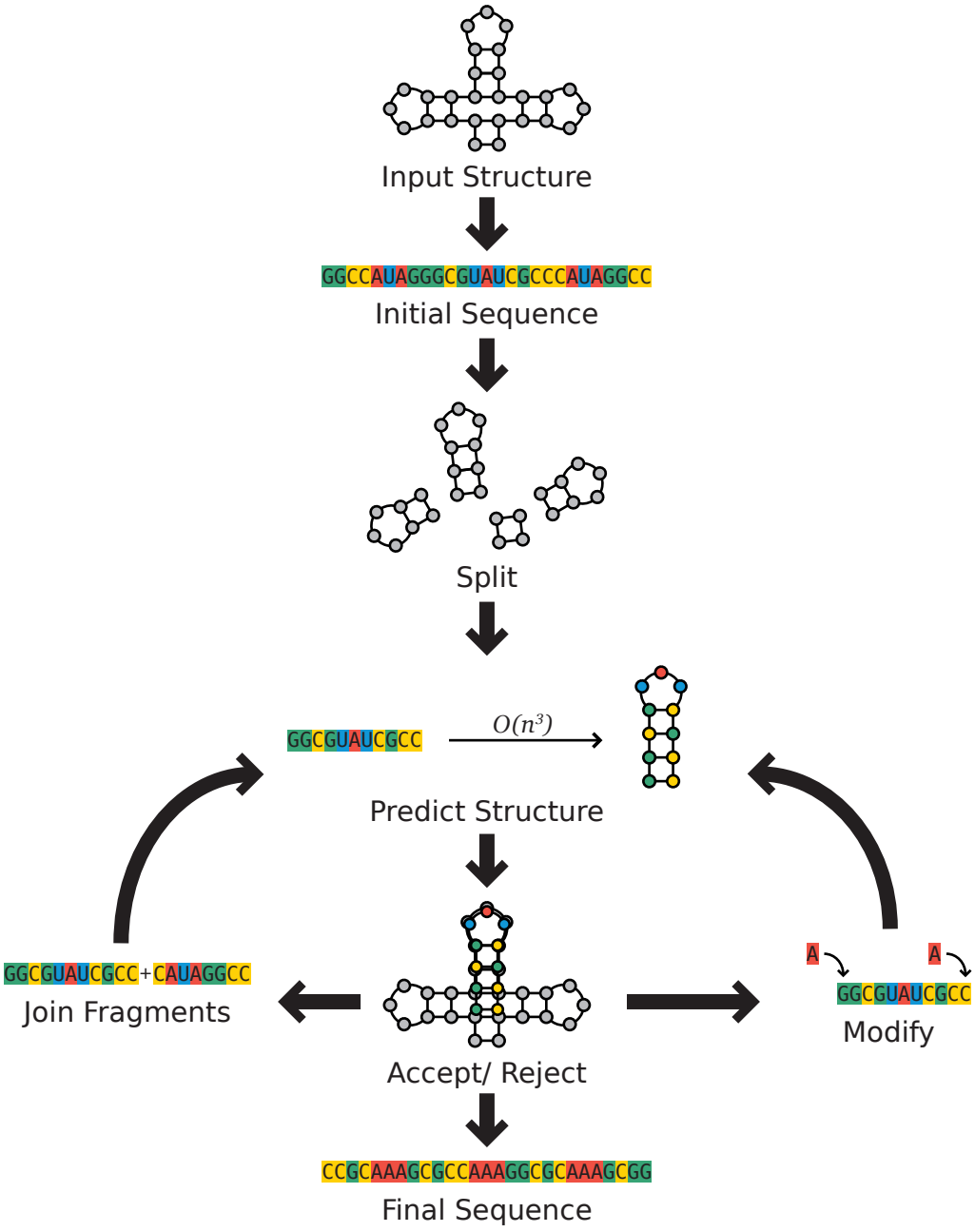


Figure 3.4: *Sequence design cycle.* The basic approach as implemented by `RNAinverse` [15], RNA-SSD [60] and INFO-RNA [59]. As starting point, a sequence compatible with the input structure is created. To speed up the folding step, the sequence is divided along structural features of the input. Structure prediction takes place every time sequences are joined or modified. Once a full sequence meets all structural constraints, the process ends.

mode. With this statistical description of all possible structures of the sequence by the McCaskill algorithm [81], optimisation for folding-probability becomes available. Since this requires recalculation of the predicted optimal structure in each step, this mode is impractical for structures of relevant size.

RNA-SSD

RNA-SSD closely follows the route of `RNAinverse` but adds two significant improvements. First, instead of an *adaptive walk* method, RNA-SSD employs a *stochastic local search* algorithm for sequence modifications. By accepting also bad moves at low probability, this should help avoiding local optima. Second, the splitting strategy of elongated hairpins is replaced by a sophisticated decomposition scheme, focused on the optimisation of local substructures. An essential step in this approach is the addition of small "cap" hairpins at split points to mimic an embedded structure in the folding step.

For the initial sequence, differences are not so drastic. Coupled positions are again populated by complementary bases, this time fed from a distribution built from biological sequences. To avoid too many non-intended interactions right from the beginning, the base distribution may be modified, e. g. to prevent canonical pairings after the end of a helical region.

As splitting scheme, RNA-SSD performs a hierarchical decomposition, leading to a balanced tree. Actual structural fragments are stored in the leafs, merged together to form larger fragments on the way towards the root, representing the full input structure. To create the hierarchy, fragments are split into two pieces aiming at equal size in each step. Leaf nodes are created for subsequences within a certain range of lengths. While only transitions between paired and unpaired regions are used as split points, some fragments may consist of two separated strands. Since `RNAfold` only predicts structures for single sequences, the ends of such fragments have to be artificially connected. This is done by adding a small hairpin of variable sequence to one end of the helix. For producing the cap sequence, the same approach as for the initial sequence is used. Rejoining fragments follows the hierarchy of the tree, with an evaluation of the MFE at each node. On violations of the structural constraints, a new solution is searched at leaf-node level. This means, that mutations only take place for the smallest substructures, trying to minimise the number of cubic run time MFE evaluations of larger subsequences.

The central routine for modifying sequences, a stochastic local search, is limited to the smallest substructures. Similar to `RNAinverse`, there is only one modification at a time: either a single unpaired base or a complete pair is exchanged. Instead of choosing randomly, an arbitrary or a conflicting position is altered chosen with a fixed probability. By the settings of RNA-SSD, attacking

constraints-violations directly, is much more likely. The mutation itself then follows the same rules as sequence initialisation. Once a sequence is changed, its structure is predicted and compared to the input. If the mutation leads to new conflicts it is stored to prevent doing the same step again in further iterations. If conflicts are encountered after merging substructures, the one showing more problems is improved. As a characteristic of the stochastic local search, it is also possible to accept a bad move by small chance. This, and choosing the position to be altered stochastically, is meant to prevent getting stuck in a local optimum and reoccurring conflicts.

The design process stops, when a sequence is found with no conflicts or after a maximal number of steps taken. Similar to `RNAinverse`, if no answer occurs after a certain time, the process is restarted.

Compared to `RNAinverse`, RNA-SSD is considerably faster and able to find valid answers for more structures. Moreover, Andronescu et al. claim, that sequences they design predict to a more stable fold then real biological sequences concerning both, MFE and folding probability [60].

INFO-RNA

INFO-RNA tries to improve sequence initialisation and uses a modified search method similar to RNA-SSD. For splitting, the same strategy as by `RNAinverse` is employed.

As starting point, the input structure is populated with bases, minimising energy according to the NN model. That is, amongst all possible sequences compatible with the structure, the initial sequence has the lowest energy. This is not to be mistaken with the actual structure this sequence will fold into. Conformations may exist which have a lower energy than the input structure. This feature of the sequence is achieved by a DP algorithm, running in linear time. The optimisation step only considers paired positions and exclusively Watson-Crick pairs. Concerning run time, this has to be optimal, since filling an initial sequence always requires visiting each base at least once.

Modifying the sequence works very similar to `RNAinverse` and RNA-SSD. In each step, the objective is to lower the number of wrong base pairs, starting with smaller substructures like `RNAinverse`. If the target structure is never met, the procedure stops after a maximum number of steps. The actual routine to find adequate mutations introduces a new idea. Where RNA-SSD keeps track of states already visited, INFO-RNA tries to look ahead, what is probably a good route to take. Instead of just changing an arbitrary position, possible new sequence candidates are evaluated using the NN model assuming the target structure and compared to the current sequence. The next mutation is then chosen as the largest improvement concerning score among all candid-

ates. Changes to be evaluated are only introduced at wrong paired positions or in their immediate neighbourhood. While for prioritisation only the change in energy from the current to a candidate sequence has to be calculated, evaluating a mutation needs structure prediction by `RNAfold`. For accepting or rejecting a move, a stochastic local search strategy is employed to avoid local optima. While the search must not necessarily end up in the best sequence, the best candidate visited is stored during the process and returned as result.

Aside from optimising for MFE, INFO-RNA also offers a mode targeting high folding probability.

Comparing the three tools, Busch & Backofen report far better results for INFO-RNA than for `RNAinverse` and claim a slight improvement over RNA-SSD [59]. For their competition, only MFE structures are considered. These are based on artificial and biological data. The biological test set is divided into sequences with only predicted structures, and sequences with known structures. Concerning running time, INFO-RNA was always faster than the other two. In finding a correct answer, reliability is only marginally higher than for RNA-SSD. Compared to biological structures, the designed sequences calculate to a higher folding probability than original sequences. As a drawback of the optimised initial sequence, low performance at sampling different sequences for a target structure is reported. Using always the same starting points just leads to the same result fairly often.

3.2 Self-Consistent Mean Field Design

All three tools described in the last section focus on MFE structures. The reason seems simply to be that those are efficient to calculate. Thereby the approaches put much effort into circumventing the caveats of the extended Nussinov algorithm [10, 18] instead of the task of sequence design.

This means, that what should be just the evaluation method, commands what can be designed. Another philosophy would be to allow no restriction on the design while not having covered everything possible by the evaluation. From here, one could imagine using various testing approaches for different design patterns. But instead, current tools only cover limited structural features.

For the limitations in adding new structural features, pseudoknots serve as a good example. Fitting them into the standard DP prediction method has often been tried, but up to now only with poor success [74, 79, 80]. What makes them complicated for current design methods, beside the higher running time for structure prediction, is the nature of pseudoknots being long range interactions. Splitting such an input structure is not possible by the approaches

presented. That is, designing pseudoknots means facing higher computational costs, while the strategy to prevent them is invalid.

Beside single molecules, structural biology nowadays demands inclusion of ligands and oligomers. The latter one could be somehow supported by tools utilising `RNAfold`, because the Vienna package also provides `RNAduplex` [15, 17], `RNAcofold` [82] and `RNAup` [83] for folding duplexes and exploring RNA-RNA interactions. Adapting should be easy because tools of the Vienna RNA suite are designed to keep a similar interface down to the code level. But with extending just by functions related to `RNAfold`, limitations are imminent. The oligomeric state is limited to only two strands, sometimes only to homomers, `RNAduplex` neglects intramolecular base pairs and of course, only 2D structures are supported.

Incorporation of ligands is only possible, if they are parameterised in the NN model and adapted by the `RNAfold` family. Due to the nature of the scoring function, this would mean to experimentally determine individual parameters per ligand, various interaction patterns and bases affected like demonstrated by Vieregg et al. [84]. They measured the effect of cations on hairpin stability and provide a set of NN parameters for a small number of sequences forming loops. But instead of details on the interaction, the system is concentration dependent. Because of the limited set of measurements and the lack of physical background, one cannot extrapolate to different environmental conditions and a larger number of sequence patterns.

After all, the suggested extensions are just of minor applicability, if taking place only for 2D representations of molecules. But porting the described approaches to work on coordinates is hardly possible at the moment. With their trial and error approach, this requires a function which is able to fold arbitrary RNA sequences into 3D molecules. That does not exist, yet. There is a 3D pipeline available, `MC-Fold` and `MC-Sym` [11], working well on short sequences but using this as a substitute for `RNAfold` would again render all optimisation tricks void. Practically this means that only the sequence initialisation ideas would survive switching to 3D tasks.

Obviously, we want to introduce a method, which does not incorporate the full evaluation in its inner core, immediately denying the idea of enumerating sequences and testing. Instead it would be elegant to use gradient information within a classical optimisation scheme. This, of course, brings with it the dangers of local minima, but these should not be worse than in a discrete approach.

But while computational chemistry usually deals with optimising continuous properties like positions of atoms, we operate on a discrete set of choices, i. e. four nucleobases. When looking at an RNA molecule, nature follows a rather rigid regime concerning its composition. Each position is populated by

exactly one base and of course, the result of our method will be a discrete sequence. But nothing forces us to stay in a physical world for what happens on the way from input to output as long as it is sound. To use a textbook optimisation method, we will not heavily adapt an algorithm to our system, but the system to the algorithm for simulation time. More precisely, we have to define a chain of states or probabilities as working representation, which will be translated into a realistic sequence in the end.

As with all methods, one needs an energy function. We have used the NN model as it is the most popular in the literature. Even without discrete bases, using the NN model should not be a problem as long as the scores are weighted by the local composition of states. Since we do not test folding during the process, we have to define our own term to push the system away from adapting an energetically favourable but unstable configuration during optimisation. Additional terms will be needed, e. g. to avoid high GC-contents.

For the transition of the continuous representation to a discrete sequence we will need to apply a cooling method to our system. While the effect of different annealing algorithms is unknown for this novel approach, there are expectations concerning its behaviour [85, 86]. The rule of thumb would be, the smoother transitions are, the less likely it is to get stuck in a local optimum. In a graphical representation, a plot energy vs. temperature, this means that a step curve is less favourable.

To combine all parts together into a simulation, self-consistent mean field minimisation (SCMF) will be used [87–92]. This method is used by quantum chemists to find consistent wave functions [92]. Koehl & Delarue used it, to optimise side-chain selection in protein sequences [90]. For nucleotide design it has never been used before. In SCMF, all states of the system start with equal probabilities, which are subject to changes until the system converges. New populations are calculated from the mean field of interactions weighted by their probabilities.

A last component is missing in our new sequence design method. Since the system will be based on an invented, artificial space, nothing is known about its parametrisation. The only thing fixed, are the values of the NN model. But scaling factors and other constants are unknown, and for the complete model, a large number of parameters may be expected. Therefore we will search parameters for our optimisation method, utilising another classical optimisation method, the simplex algorithm [93].

3.2.1 Self-Consistent Mean Field Optimisation

At its very heart, SCMF operates on a system of sites, with each site being represented by multiple states. The task is then, to find a composition for the

states, which minimises some force field of the system [89, 94]. A very prominent example comes from homology modelling [95, 96], with several methods utilising SCMF for side-chain placement. There, sites easily correspond with the sequence of amino acids while states are fed by libraries of known side-chain coordinates. As objective, one rotamer per site has to be chosen leading to a low energy of the whole molecule [87, 88]. Here the subject is RNA, again with one site per sequence position but with copies of all four nucleotides attached as states. Delarue & Koehl call such a system *the chimeric molecule* [90].

To find an optimal composition of states, SCMF minimises the *effective energy* of the system given by *mean field theory* (MFT) [87–91]. By definition, all possible compositions of the system are considered in a single term, weighted by probabilities:

$$E_{\text{eff}}(M) = \sum_{j=1}^{n} \sum_{\alpha=1}^{4} m_{\alpha j} E(\alpha, j) \tag{3.2}$$

with M as the matrix representing our system, $m$an element of M, n its width (length of the sequence) and $E(\alpha, j)$ as the energy of all interactions of state (nucleobase type) α at position j, weighted by the probability of interaction partners. Thereby states of the same site never interact with each other, only with other cells of M. A more detailed definition of E will follow in §3.2.4. The sum over all states is limited to $\alpha = 1, \ldots, 4$, since this is the size of our RNA alphabet.

For the mean field optimisation, the probability for each of the multiple bases in every position needs to be refined. Populating M with new values, considering the interactions of a site, means transforming energies into probabilities, using the *Boltzmann relation* [97]. Following Delarue & Koehl, this step is calculated by [88–90]:

$$m_{\alpha j} = \frac{\exp\left(-\frac{E_{\text{local}}(\alpha, j)}{RT}\right)}{\sum_{\beta=1}^{4} \exp\left(-\frac{E_{\text{local}}(\beta, j)}{RT}\right)} \tag{3.3}$$

where R is the *gas constant*, T the temperature and $E_{\text{local}}(\alpha, j) = 2E(\alpha, j)$ the quasi local mean field of base type α at position j in the sequence.

On the way from the initial distribution of probabilities to a discrete solution, the system is in turns evaluated by E_{local} and updated by Eqn (3.3). Once another step of repopulating the sites leads to the same distribution of states as before, the system is said to be self-consistent and the process stops. The scheme described, divides SCMF into two parts: applying MFT for evaluation and the simulation, calculating changes and monitoring stop criteria [89, 91].

For the flexibility of the method, this is important, enabling systems of varying complexity to be optimised using the same tool. As an example, one could think of designing sequences for oligomers. Literature energy models supporting MFT exist (refer to §3.1.2) and the optimisation routine just sees a larger sequence matrix M. Also going for real coordinates would be possible with an adequate force field, since interactions could be cached and updating the matrix scales linearly with the sequence length [87, 91]. Additionally, the separation of sequence representation by a matrix and its evaluation, no restrictions concerning base pair ordering exist. This means, modelling pseudoknots is only a matter of an appropriate energy function. The rest of this section describes the simulation part of SCMF.

Eqn (3.3) utilises R and T, by its original meaning to incorporate temperature dependency of a system, but sometimes treated differently by SCMF variants. Delarue & Koehl just use RT as normalisation factor, set to the value which fits their method best [88, 90]. In other studies, R is the gas constant and the temperature T is used for a simulated annealing approach [89, 91]. We also use R with its physical value, since it shows adequate performance with decreasing temperature compared to other values. In such a system, T is the annealing parameter and the choice of a constant factor is not as important.

One notable feature of SCMF is its robustness concerning the starting point [90, 91]. Supposedly, the result of a simulation does not change from a different input distribution of states. A high initial temperature will assure this behaviour, forcing the system into an equilibration phase in the beginning. Beside helping the sequence matrix to converge faster [91], T plays its own important role as one of the stop criteria of the annealing process. When RT gets smaller than any interesting energy barrier, the system is trapped and there is no point in cooling further. Also for technical reasons there has to be a final temperature, not only because we can not divide by zero, but simply because the simulation will run on a computer which has a limited number range. A more detailed view on the annealing step will follow in §3.2.3.

A common strategy to avoid local minima in simulated annealing is slow cooling [85, 86], adding a *memory* to the simulation should help to avoid oscillations. That is, new values are only slowly accepted in the matrix while being mixed with the last step [87, 90, 91]:

$$m_{\alpha j} = \lambda m_{\alpha j}^{\mathrm{cur}} + (1-\lambda) m_{\alpha j}^{\mathrm{old}} \tag{3.4}$$

where λ defines the ratio between the values in M^{cur}, evaluated on the current system, and the values of the last step stored in M^{old}. As a further advantage, not entirely relying on the currently evaluated force field, is known to improve convergence of the system.

As stop criterion, a final temperature was already mentioned and obviously, there should be a maximal number of steps taken, before the system seems unlikely to converge. But those are criteria for the case when the simulation does not find a solution. The true goal in SCMF is still to drive M into a self-consistent state. A naïve approach to test this, would be to compare an old matrix with the current one. While operating with two matrices is already necessary, one to be evaluated and another one to store updated probabilities, considering oscillations gets more complex. As an example, this may be a base pair, which is alternating between becoming `GC` and `CG` for energy parameters, symmetric concerning the 5' to 3' direction. This would mean keeping track of enough steps to match the frequency and comparing them to recognise a cyclic relation between individual positions in the matrix. A more elegant solution would already incorporate our objective of pulling a distinct sequence out of M, instead of just technically observing convergence. Assessing how well a sequence is defined in a chain of states, could be done by estimating how many possible solutions are left from undecided states, if they have to add up to 1 in each position. This cumulates to a single-value measurement of sequence variability and is defined as a quasi-entropy-like property of the sequence matrix, similar to the conformational entropy by Delarue & Koehl [87]. The idea is, to add no contribution for columns already decided, while open columns should be considered according to their current distribution. Since this measure should be used to control the simulation, it should be designed to be easily interpreted. A value of 0 obviously means low entropy, while an upper bound is introduced through normalisation by sequence length n, here:

$$s(M) = -\tfrac{1}{n} \sum_{j=1}^{n} \sum_{\alpha=1}^{4} m_{\alpha j} \ln m_{\alpha j} \tag{3.5}$$

This adds up to a *sequence entropy* of 1.39 for the initial matrix, with all equal probabilities [90, 91] and the highest grade of variability. Also since the measure is calculated on an artificial, non-chemical system, the Boltzmann constant is omitted as a factor. As a stop criterion, a threshold would then suffice, chosen low enough that variability is not likely to drop any further. Following oscillations based on a single value, rather than the whole matrix, is already easier, but with an entropy-like term, we can exploit an important feature of the annealing approach: as long as the system is not heated up and forced into another equilibration phase after the start, variability does not increase. It is either reduced or not changed at all. For constant intervals, one might simply count how many steps they last and react to it. In our approach, we observe the change of quasi-entropy and let the step size of cooling depend upon it.

As already mentioned by the stop criteria, another attribute of SCMF is,

that it comes without a guarantee to find an optimal solution and sometimes does not find any at all [91]. Assessing the confidence of a solution in simulated annealing, is sometimes done by sampling more results [86]. The idea is, if several runs of a simulation lead to the same or very similar answers, it should be near optimum. The problem for SCMF is, that it is known to resemble a deterministic behaviour: the same input should always give the same result [91]. A simple solution to increase the sampling rate of the method, is to add artificial noise to the energy function. That is, adding random contributions to the parameters of the NN model, small enough to not decrease performance of the final sequences, but with high enough impact to change the route of the simulation.

As requested in the beginning, SCMF easily allows for predefining sequence positions. In M, one state will be set to a probability of 1 and everything else to 0. Those columns may then be skipped from updates, but are still usable by interaction partners.

3.2.2 Sequence Representation

Introduced in the last section, a central part of the SCMF method is the concept of a chimeric molecule [90]: during simulation time, the polymer to be designed has copies of all four RNA bases attached to each backbone position. Since the result needs to be a one-dimensional sequence, only singular bases allowed, all copies come with a probability to emit the final base. These probabilities can be represented by a $4 \times n$ matrix M (*sequence matrix*), with n columns as positions of the sequence and 4 rows for the RNA alphabet. Each column represents a site in the molecule. The sum in any column (the total probability) is always 1:

$$\sum_{\alpha=1}^{4} m_{\alpha j} = 1 \text{ for all } j = 1, \ldots, n \tag{3.6}$$

With this constraint satisfied, $m_{\alpha j}$ is the probability, that position j of the sequence will be assigned base α [87, 88, 90]. Fig. 3.5 illustrates a *chimeric hairpin* structure and its corresponding sequence matrix.

After the simulation stops, the task remains to read an unambiguous RNA sequence out of the matrix. For columns which assign all probability to a single state, the solution is obvious. Undecided positions are a bit more complicated to handle. The probability distribution of a site results from the network of structural feature it is involved in. At least one other column exists with probabilities tailored to the current one, rendering a strategy like uneducated majority voting impractical. As an example, looking at a base pair, corresponding

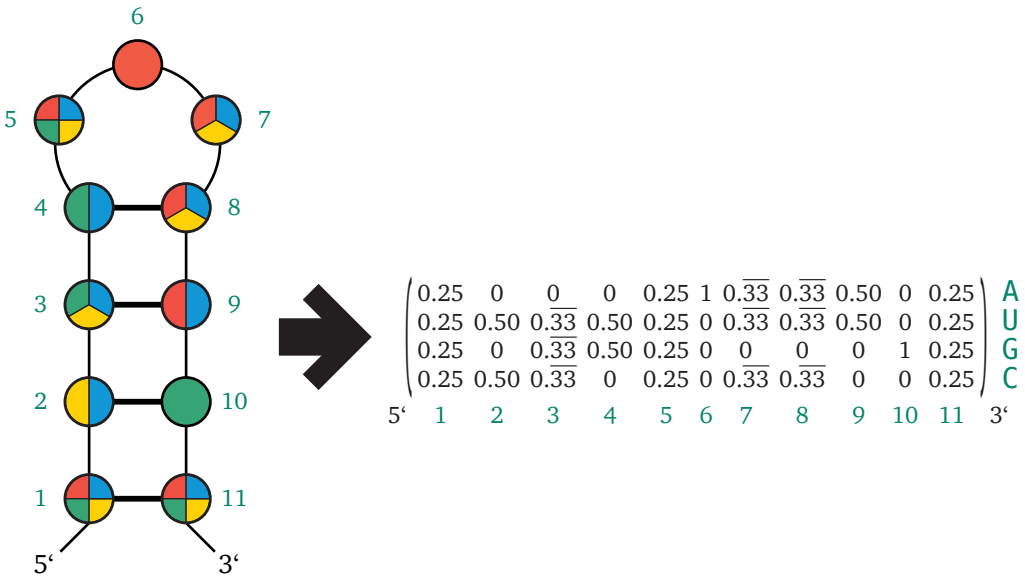


Figure 3.5: *Chimeric hairpin & sequence matrix.* Beads representing nucleotides are divided into weights of base type by NDB colours. The probabilities are stored in the sequence matrix with columns corresponding to sequence positions (green numbers). In the matrix, rows represent base type, as annotated by the RNA alphabet (green letters).

sites may both end up with a probability of 0.5 to become G or C. If the decision is made without any knowledge about the pairing, both positions will be fixed to the same base. Designing a strategy to pay attention to the structural environment solved, would mean to add some bias to the method. Additionally, this is also complex to implement, since there are more complicated features than base pairs. A strategy to retain the generality of the method, while adding only a very small bias at the same time, would be rerunning the simulation. To assure that at least one more position has a decision afterwards, the column containing the highest probability of the whole matrix is fixed before. Then the SCMF protocol is restarted, excluding already finished positions from being redesigned. Also the sequence entropy threshold is set to a value of 0, since for an almost converged matrix it may be already too low from the beginning. This is then repeated, until the whole sequence could be retrieved. Concerning the running time of the whole process, there should be no major increase. When repeating the simulation, large parts of the matrix usually can be excluded and each new run should fix several sites.

3.2.3 Annealing

As mentioned before, the SCMF procedure presented here comes with a simulated annealing component [85, 86]. In particular this means, an appropriate cooling scheme has to be defined. What has to be avoided, are temperature steps too big for a smooth transition, baring the danger to get stuck in a local optimum. But with a small step size, running time of the simulation may be unnecessarily prolonged. Those are the criteria to observe when choosing a scheme especially tailored for our system. Strategies tested include exponential, linear and adaptive cooling.

Exponentially decreasing temperature turned out to be too fast to give good answers. For most structures, more complex than a small hairpin, designed sequences are not likely to fold as intended. Improved performance is achieved by linear cooling at a considerably increased simulation time. As a disadvantage, parameters for this strategy do depend on the input system. With the size and composition of structural features, a varying amount of time may be necessary to converge. But when the temperature is decreased by a constant, the maximal number of steps is already determined by the cooling constant, the initial and the final temperature. To assess adequate parameters, one has to run the simulation several times for different setups. Another solution would be to operate at very slow cooling to increase the number of steps, while also increasing running time.

The final annealing scheme should minimise the simulation time and avoid the need to tune parameters for specific systems:

$$T(t) = T(t-1) \cdot c(t) \tag{3.7}$$

for temperature T at time t ($T(0)$ is the initial temperature) and the current cooling rate c. In adaptive cooling, temperature steps are increased, if the system undergoes only small changes, and decreased if it moves to fast:

$$c(t) = \begin{cases} \sqrt{c(t-1)} & \text{if } \frac{s_{\text{short}}(M)}{s_{\text{long}}(M)} < s_c, \\ \mathrm{c}(t-1)^2 & \text{if } \frac{s_{\text{short}}(M)}{s_{\text{long}}(M)} \geq s_c \text{ and } c(t-1) > c_{\min}, \\ \mathrm{c}(t-1) & \text{otherwise} \end{cases} \tag{3.8}$$

with $c(0)$ set to a value smaller than 1, $c_{\min}$ a lower border for the cooling rate and the decision to speed up or slow down controlled by sequence entropy. Thereby s_{long} keeps the trace of old instances of M:

$$s_{\text{long}}(M) = \beta_{\text{long}} s_{\text{long}}(M) + (1 - \beta_{\text{long}}) s(M) \tag{3.9}$$

with β_{long} determining the period tracked. To avoid following every single turn the sequence entropy takes, a little bit of delay is also added for the

comparison value from the current M by introducing s_{short}:

$$s_{\text{short}}(M) = \beta_{\text{short}} s_{\text{short}}(M) + (1 - \beta_{\text{short}}) s(M) \tag{3.10}$$

with β_{short} being considerably smaller than β_{long}. The ratio between s_{short} and s_{long} is then compared to a constant threshold s_c.

The calculation starts by equilibrating the system at a high temperature [91]. By using a cooling rate very close to 1, the system is kept excited for some time and can evolve to an appropriate speed. For the values depending on sequence entropy, it is also important to retain the initial temperature until they adapt to the system. Since s_{short} is initialised with the sequence entropy of the initial matrix, it needs a few steps for the delay to establish. Given s_c, s_{long} needs to be started at a value which prevents immediate cooling speed-up. For us, using two times the initial sequence entropy provides a long enough grace period until actual cooling starts. From there, the system will run self-controlled into the final temperature.

3.2.4 Energy Terms

The score of the system closely follows the estimates of Gibbs energies given by the NN model and usually labelled as ΔG values. In principle, this refers to a change in Gibbs energy upon melting or folding [97]. Our formulation contains additional terms, so it is labelled as score E, since it is simply an energy-like value which we wish to optimise. The score is most easily broken into four contributions described below. E_{neg} (negative design), E_{het} (heterogeneity), E_{NUN} (non-canonical base pairs) and E_{NN} (conventional NN contribution). E_{NN} is the most complicated term and is in turn divided into contributions depending on the structural motifs, as described in the following section.

During development, two simpler scoring functions were used. The initial scheme is inspired by Nussinov [10, 18] and simple enough to be followed by pen & paper as it just counts H-bonds between base pairs. After verifying that the simulation works on a technical level, a crude approximation of the NN model was set up. Only measuring overlapping stacked base pairs, and pairs with adjacent mismatches, already gave an idea of how the final system will look like while retaining the possibility of manual inspection for small input structures.

The following definitions in this section usually contain several sums iterating all states of M involved. To save the space occupied by this recurring scheme, we define a function

$$b(f) = \sum_{\beta \in \mathcal{A}} \sum_{\gamma \in \mathcal{A}} \sum_{\omega \in \mathcal{A}} f(\beta, \gamma, \omega) \tag{3.11}$$

for substitution, summing up f for all configurations of states from the RNA alphabet $\mathcal{A}$.

Nearest Neighbour Contribution

Since the structure is fixed in our problem, it can be decomposed into different contributions as shown in Fig. 3.2. Thereby it is important to note, that structural motifs overlap and a base may get more than one contribution. Taking into account all motifs in Fig. 3.3, there are stacked base pairs, hairpins, bulge loops, internal loops, external loops and multiloops.

The first term explained covers pairs of adjacent base pairs, usually called stacks. For each of the four bases involved, the score is calculated by

$$E_{\text{Hlx}}(\alpha, j, l) = \begin{cases} \frac{1}{4} b(m_{\beta k} m_{\gamma l} m_{\omega k-1} E_{\text{Stack}}(\alpha, \beta, \omega, \gamma)) & \text{if } (j,k),\ (l,k-1) \text{ are paired,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma k+1} m_{\omega l} E_{\text{Stack}}(\beta, \alpha, \omega, \gamma)) & \text{if } (k,j),\ (k+1,l) \text{ are paired,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma l} m_{\omega k+1} E_{\text{Stack}}(\gamma, \omega, \beta, \alpha)) & \text{if } (j,k),\ (l,k+1) \text{ are paired,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma k-1} m_{\omega l} E_{\text{Stack}}(\gamma, \omega, \alpha, \beta)) & \text{if } (k,j),\ (k-1,l) \text{ are paired,} \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

where $E_{\text{Stack}}(\alpha, \beta, \omega, \gamma)$ is a tabulated value of the NN model for base types α, β, ω, γ at sites shown in Fig. 3.6. Each of the contributions is weighted by the corresponding probability in M. With four bases taking part in a stack loop, the contribution is divided amongst them. The extra parameter l is necessary for nested bases in a helix, where a base is involved in two stacks. Since the second base pair has to include an immediate neighbour of position j, we calculate contributions for $j-1$ and $j+1$ as

$$E_{\text{SL}}(\alpha, j) = E_{\text{Hlx}}(\alpha, j, j+1) + E_{\text{Hlx}}(\alpha, j, j-1) \tag{3.13}$$

Another simple motif are hairpin loops, a region of unpaired bases closed by one base pair. In the default case, the NN model provides values for the closing base pair, the first two unpaired bases and a length-dependent score. Since the latter one does not carry any sequence information, it is ignored by

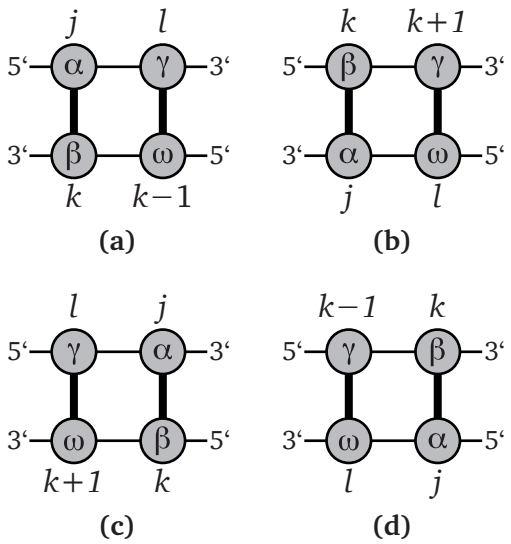


Figure 3.6: *Base pair arrangements for stack loop contributions.* The annotation corresponds to variables in Eqn (3.12). Specimen of bases are given by α, β, ω, γ. Positions of bases in the sequence are given by j, k, l. (a) Configuration when calculating the contribution for the 5' base of the first base pair, (b) the 3' base of the first pair, (c) the 3' base of the second pair and (d) the 5' base of the second pair. Thick black lines indicate base pairs.

the contribution we calculate:

$$E_{\text{Hairpin}}(\alpha, j) = \begin{cases} \frac{1}{4} b(m_{\beta k} m_{\gamma j+1} m_{\omega k-1} E_{\text{Hpin}}(\alpha, \beta, \omega, \gamma)) & \text{if } (j,k) \text{ closes a hairpin,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma k+1} m_{\omega j-1} E_{\text{Hpin}}(\beta, \alpha, \omega, \gamma)) & \text{if } (k,j) \text{ closes a hairpin,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma j-1} m_{\omega k+1} E_{\text{Hpin}}(\gamma, \omega, \beta, \alpha)) & \text{if } (j-1,k+1) \text{ closes a hairpin,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma k-1} m_{\omega j+1} E_{\text{Hpin}}(\gamma, \omega, \alpha, \beta)) & \text{if } (k-1,j+1) \text{ closes a hairpin,} \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

where $E_{\text{Hpin}}(\alpha, \beta, \gamma, \omega)$ is a tabulated value of the NN model for base pair (α, β) in 5' to 3' direction and adjacent free bases γ, ω as shown in Fig. 3.7.

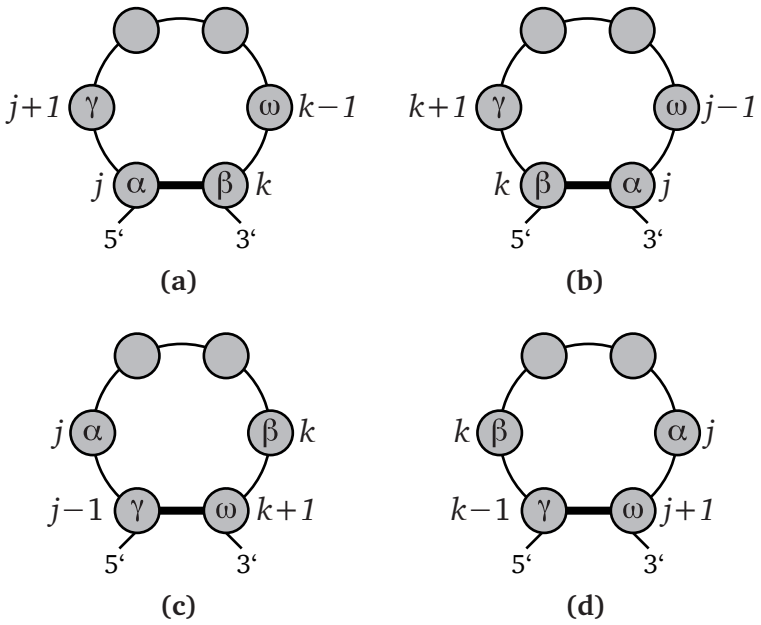


Figure 3.7: *Base arrangements for hairpin loop contributions.* The annotation corresponds to variables in Eqn (3.14). Base types are labelled α, β, ω, γ. Positions of bases in the sequence are given by j and k. (a) Configuration when calculating the contribution for the 5' base of the closing base pair, (b) the 3' base of the base pair, (c) and (d) the unpaired bases. Thick black lines indicate base pairs.

A special case of hairpins are *tetraloops*, with only four unpaired bases. The NN model lists certain combinations of bases to have an additional stabilising effect on loops [69, 70]. Since this is a value depending on sequences of size

six (closing base pair plus loop of size four), we define this contribution using V as the list of tetraloops in the NN model, where v_i is the i^{th} base of sequence v:

$$E_{\text{Tetraloop}}(\alpha, j) = \begin{cases} \frac{1}{6} \sum_{\substack{v \in V \\ v_{j-l}=\alpha}} E_{\text{Tetra}}(v) \prod_{\substack{k=0 \\ l+k \neq j}}^{5} m_{v_k l+k} & \text{if } (l, l+5) \text{ closes a loop and } l \leq j \leq l+5, \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

where $E_{\text{Tetra}}(v)$ is the tetraloop score for sequence v, with base type α at position $j-l$, which is weighted by all probabilities from the loop sites and k, l are positions in the sequence. With $E_{\text{Tetraloop}}$, the full contribution for a position in a hairpin loop is

$$E_{\text{HL}}(\alpha, j) = E_{\text{Hairpin}}(\alpha, j) + E_{\text{Tetraloop}}(\alpha, j) \tag{3.16}$$

The next loop motif is the bulge loop, formed by a single-sided region of unpaired bases between two base pairs. Again the loop region itself only scores by length so only the interacting bases are considered here. This fits the calculation into the known scheme, collecting probabilities to weight the NN score and normalise by bases involved. But to get the sequence positions right for the probabilities, we need to know on which side of the structure the loop region is, totalling in eight cases rather than four:

$$E_{\text{BL}}(\alpha, j) = \begin{cases} \frac{1}{4} b(m_{\beta k} m_{\gamma l} m_{\omega k-1} E_{\text{Bulge}}(\alpha, \beta, \omega, \gamma)) & \text{if } (j,k), (l,k-1) \text{ are pairs, loop between } j, l, \\ \frac{1}{4} b(m_{\beta k} m_{\gamma l} m_{\omega j-1} E_{\text{Bulge}}(\beta, \alpha, \omega, \gamma)) & \text{if } (k,j), (l,j-1,) \text{ are pairs, loop between } k, l, \\ \frac{1}{4} b(m_{\gamma l} m_{\omega k+1} m_{\beta k} E_{\text{Bulge}}(\gamma, \omega, \beta, \alpha)) & \text{if } (l,k+1), (j,k) \text{ are pairs, loop between } l, j, \\ \frac{1}{4} b(m_{\gamma l} m_{\omega j+1} m_{\beta k} E_{\text{Bulge}}(\gamma, \omega, \alpha, \beta)) & \text{if } (l,j+1), (k,j) \text{ are pairs, loop between } l, k, \\ \frac{1}{4} b(m_{\beta k} m_{\gamma j+1} m_{\omega l} E_{\text{Bulge}}(\alpha, \beta, \omega, \gamma)) & \text{if } (j,k), (j+1,l) \text{ are pairs, loop between } l, k, \\ \frac{1}{4} b(m_{\beta k} m_{\gamma k+1} m_{\omega l} E_{\text{Bulge}}(\beta, \alpha, \omega, \gamma)) & \text{if } (k,j), (k+1,l) \text{ are pairs, loop between } l, j, \\ \frac{1}{4} b(m_{\gamma j-1} m_{\omega l} m_{\beta k} E_{\text{Bulge}}(\gamma, \omega, \beta, \alpha)) & \text{if } (j-1,l), (j,k) \text{ are pairs, loop between } k, l, \\ \frac{1}{4} b(m_{\gamma k-1} m_{\omega l} m_{\beta k} E_{\text{Bulge}}(\gamma, \omega, \alpha, \beta)) & \text{if } (k-1,l), (k,j) \text{ are pairs, loop between } j, l, \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

where $E_{\text{Bulge}}(\alpha, \beta, \omega, \gamma)$ is the tabulated score for a bulge loop enclosed by base pairs (α, β), (γ, ω) at sequence positions as shown in Fig. 3.8.

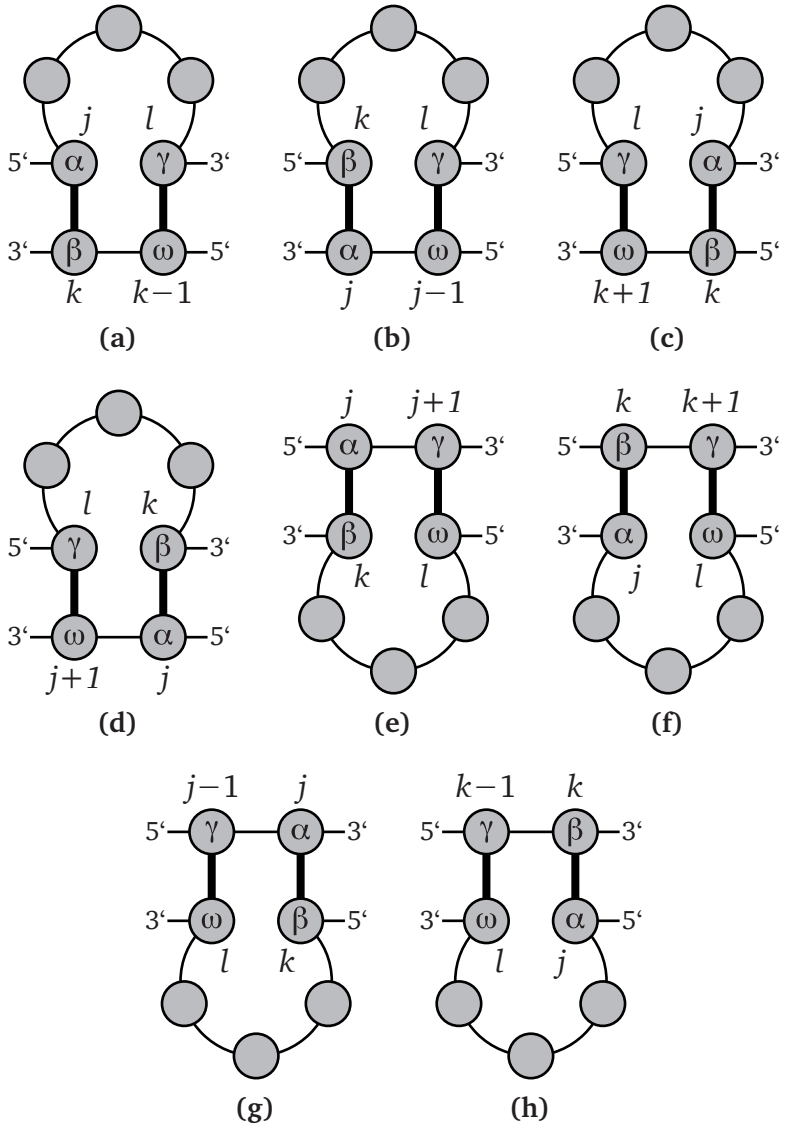


Figure 3.8: *Base arrangements for bulge loop contributions.* The annotation corresponds to variables in Eqn (3.17). Base types are labelled α, β, ω, γ. Positions of bases in the sequence are given by j, k and l. (a) – (d) show configurations with the loop region between the 5' base of the first and the 3' base of the second base pair, (e) – (h) cover the four remaining cases. In each setup, $E_{\mathrm{BL}}(\alpha, j)$ is calculated. Thick black lines indicate base pairs.

One of the more complex loop types are internal loops. These are formed by two base pairs separated by two unpaired regions. While a general scheme exists in the NN model for loops larger than two bases, everything below is treated as special cases. For internal loops with only two (1×1), three (2×1, 1×2) and four (2×2) unpaired bases, sequence dependent scores exist like for tetraloops. The smallest loop, with two loop regions of size one, is calculated for a list of sequences $I_{1\times1}$, all of length six, by

$$E_{\mathrm{IL}\,1\times1}(\alpha, j) = \begin{cases} \frac{1}{6}\sum_{\substack{v\in I_{1\times1}\\ v_{j-k}=\alpha}} E_{\mathrm{Int}\,1\times1}(v)\prod_{\substack{i=0\\ k+i\neq j}}^{2} m_{v_i k+i}\prod_{i=3}^{5} m_{v_i l+i-3} & \text{if } k \le j \le k+2, \\ \frac{1}{6}\sum_{\substack{v\in I_{1\times1}\\ v_{j-l}=\alpha}} E_{\mathrm{Int}\,1\times1}(v)\prod_{i=0}^{2} m_{v_i k+i}\prod_{\substack{i=3\\ l+i-3\neq j}}^{5} m_{v_i l+i-3} & \text{if } l \le j \le l+2, \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

where we define the conditions as true, if positions $(k, l+2)$, $(k+2, l)$ are paired to form a 1×1 internal loop and $k \le j \le k+2$ or $l \le j \le l+2$. $E_{\mathrm{Int}\,1\times1}(v)$ is the score for sequence v from the NN table, while the sum only considers sequences with base type α at position $j-k$ or $j-l$. The two products fetch the base type from v and look up the probability in M at positions corresponding to the internal loop. The biggest of the loops covered as special case are 2×2 loops with two unpaired bases on each side. The calculation of the contribution is similar to the last one with indexes extended for larger loops:

$$E_{\mathrm{IL}\,2\times2}(\alpha, j) = \begin{cases} \frac{1}{8}\sum_{\substack{v\in I_{2\times2}\\ v_{j-k}=\alpha}} E_{\mathrm{Int}\,2\times2}(v)\prod_{\substack{i=0\\ k+i\neq j}}^{3} m_{v_i k+i}\prod_{i=4}^{7} m_{v_i l+i-4} & \text{if } k \le j \le k+3, \\ \frac{1}{8}\sum_{\substack{v\in I_{2\times2}\\ v_{j-l}=\alpha}} E_{\mathrm{Int}\,2\times2}(v)\prod_{i=0}^{3} m_{v_i k+i}\prod_{\substack{i=4\\ l+i-4\neq j}}^{7} m_{v_i l+i-4} & \text{if } l \le j \le l+3, \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

where the conditions are true, if positions $(k, l+3)$, $(k+3, l)$ are base pairs forming a 2×2 internal loop and $k \le j \le k+3$ or $l \le j \le l+3$. Here, $I_{2\times2}$ is the set of sequences and $E_{\mathrm{Int}\,2\times2}(v)$ the score for sequence v. The normalisation factor is given as 8, due to the higher number of participating bases in this motif. The last internal loop type with dedicated parameters is asymmetric, with a single unpaired base on one side and two on the other. To populate unpaired and paired positions correctly in the NN model parameter table, this contribution is split up into 1×2 (loop size 1 at downstream position in the

strand) and 2×1 (loop size 1 at a upstream location) loops. With $I_{1\times2}$ holding the sequences for the first part, we calculate the contribution as

$$E_{\mathrm{IL}\,1\times2}(\alpha, j) = \begin{cases} \frac{1}{7}\sum_{\substack{v\in I_{1\times2}\\ v_{j-k}=\alpha}} E_{\mathrm{Int}\,1\times2}(v)\prod_{\substack{i=0\\ k+i\neq j}}^{2} m_{v_i k+i}\prod_{i=3}^{6} m_{v_i l+i-3} & \text{if } k \le j \le k+2, \\ \frac{1}{7}\sum_{\substack{v\in I_{1\times2}\\ v_{j-l}=\alpha}} E_{\mathrm{Int}\,1\times2}(v)\prod_{i=0}^{2} m_{v_i k+i}\prod_{\substack{i=3\\ l+i-3\neq j}}^{6} m_{v_i l+i-3} & \text{if } l \le j \le l+3, \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

where the conditions are defined to be true, if positions $(k, l+3)$, $(k+2, l)$ are base pairs forming a 1×2 internal loop and $k \le j \le k+2$ or $l \le j \le l+3$. $E_{\mathrm{Int}\,1\times2}(v)$ is the tabulated score for sequence v resembling a 1×2 internal loop. The first product covers positions k to $k+2$, the shorter unpaired region, while the second one operates between l and $l+3$, the larger loop. These intervals are switched for 2×1 loops:

$$E_{\mathrm{IL}\,2\times1}(\alpha, j) = \begin{cases} \frac{1}{7}\sum_{\substack{v\in I_{2\times1}\\ v_{j-k}=\alpha}} E_{\mathrm{Int}\,2\times1}(v)\prod_{\substack{i=0\\ k+i\neq j}}^{3} m_{v_i k+i}\prod_{i=4}^{6} m_{v_i l+i-4} & \text{if } k \le j \le k+3, \\ \frac{1}{7}\sum_{\substack{v\in I_{2\times1}\\ v_{j-l}=\alpha}} E_{\mathrm{Int}\,2\times1}(v)\prod_{i=0}^{3} m_{v_i k+i}\prod_{\substack{i=4\\ l+i-4\neq j}}^{6} m_{v_i l+i-4} & \text{if } l \le j \le l+2, \\ 0 & \text{otherwise} \end{cases} \tag{3.21}$$

where the conditions are defined to be true, if positions $(k, l+2)$, $(k+3, l)$ form pairs enclosing a 2×1 internal loop and $k \le j \le k+3$ or $l \le j \le l+2$. The scores are listed in $E_{\mathrm{Int}\,2\times1}(v)$ for sequence v in $I_{2\times1}$.

Internal loops with at least one unpaired region larger than 2 are treated differently. Similar to hairpins, a closing base pair and the first loose bases are scored. Since an internal loop comes with two ends, we have four bases per

case and eight cases all together:

$$E_{\text{IL}>2}(\alpha, j) = \begin{cases} \frac{1}{4} b(m_{\beta k} m_{\gamma j+1} m_{\omega k-1} E_{\text{Int}}(\alpha, \beta, \omega, \gamma)) & \text{if } (j,k) \text{ is the left pair of a loop,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma k+1} m_{\omega j-1} E_{\text{Int}}(\beta, \alpha, \omega, \gamma)) & \text{if } (k,j) \text{ is the left pair of a loop,} \\ \frac{1}{4} b(m_{\gamma j-1} m_{\omega k+1} m_{\beta k} E_{\text{Int}}(\gamma, \omega, \beta, \alpha)) & \text{if } (j-1,k+1) \text{ is the left pair of a loop,} \\ \frac{1}{4} b(m_{\gamma k-1} m_{\omega j+1} m_{\beta k} E_{\text{Int}}(\gamma, \omega, \alpha, \beta)) & \text{if } (k-1,j+1) \text{ is the left pair of a loop,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma j-1} m_{\omega k+1} E_{\text{Int}}(\beta, \alpha, \gamma, \omega)) & \text{if } (j,k) \text{ is the right pair of a loop,} \\ \frac{1}{4} b(m_{\beta k} m_{\gamma k-1} m_{\omega j+1} E_{\text{Int}}(\alpha, \beta, \gamma, \omega)) & \text{if } (k,j) \text{ is the right pair of a loop,} \\ \frac{1}{4} b(m_{\gamma j+1} m_{\omega k-1} m_{\beta k} E_{\text{Int}}(\omega, \gamma, \alpha, \beta)) & \text{if } (j+1,k-1) \text{ is the right pair of a loop,} \\ \frac{1}{4} b(m_{\gamma k+1} m_{\omega j-1} m_{\beta k} E_{\text{Int}}(\omega, \gamma, \beta, \alpha)) & \text{if } (k+1,j-1) \text{ is the right pair of a loop,} \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

where all conditions are only valid for internal loops with at least one unpaired region of a sizer larger than 2. $E_{\text{Int}}(\alpha, \beta, \omega, \gamma)$ is the tabulated score of a base pair (α, β) at positions j, k and two adjacent unpaired bases. Combining all terms for internal loops, the contribution to the NN design term adds up to

$$\begin{aligned} &E_{\text{IL}}(\alpha, j) = \\ &\quad E_{\text{IL}\,1\times1}(\alpha, j) + E_{\text{IL}\,2\times2}(\alpha, j) + E_{\text{IL}\,1\times2}(\alpha, j) + E_{\text{IL}\,2\times1}(\alpha, j) + E_{\text{IL}>2}(\alpha, j) \end{aligned} \tag{3.23}$$

The last two structural motifs, external and multiloops, share a penalty for initialising stem regions with base pairs not built from bases `G` and `C`. As a value also depending on base types, it needs to be weighted by corresponding probabilities to be considered:

$$E_{\text{non-GC}}(\alpha, j) = \begin{cases} \frac{1}{2} \sum_{\beta \in \mathcal{A}} m_{\beta k} E_{\text{pnon-GC}}(\alpha, \beta) & \text{if } (j,k) \text{ is a base pair,} \\ \frac{1}{2} \sum_{\beta \in \mathcal{A}} m_{\beta k} E_{\text{pnon-GC}}(\beta, \alpha) & \text{if } (k,j) \text{ is a base pair,} \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

where β is iterated for the whole RNA alphabet $\mathcal{A}$ and $E_{\text{pnon-GC}}(\alpha, \beta)$ is the penalty tabulated for base types α, β. Obviously, for `GC` pairs, $E_{\text{pnon-GC}}$ is 0.

The external loop, is a motif at the very end of a structure. It includes the last base pair at the end of a strand and immediate unpaired neighbours, if there are dangling ends. Without, there is no contribution by the NN model, while it is unaffected if the free end is longer than one base. In case there are unpaired bases at the 5' and the 3' end of a structure, both are evaluated

separately and added to the full score. This means, we define dedicated contributions for both ends and combine them together with the non-GC penalty to the full score for external loops. For both sides, three cases have to be evaluated, both paired bases and the free base. An external loop with the unpaired end at the 5' side is calculated by

$$E_{\mathrm{EL\,5'}}(\alpha, j) = \begin{cases} \frac{1}{3}\sum_{\beta\in\mathcal{A}}\sum_{\gamma\in\mathcal{A}} m_{\gamma j-1} m_{\beta k} E_{\mathrm{5'\,dgl}}(\alpha,\beta,\gamma) & , \\ \frac{1}{3}\sum_{\beta\in\mathcal{A}}\sum_{\gamma\in\mathcal{A}} m_{\gamma k-1} m_{\beta k} E_{\mathrm{5'\,dgl}}(\beta,\alpha,\gamma) & , \\ \frac{1}{3}\sum_{\gamma\in\mathcal{A}}\sum_{\omega\in\mathcal{A}} m_{\gamma j+1} m_{\omega l} E_{\mathrm{5'\,dgl}}(\gamma,\omega,\alpha) & , \\ 0 & \text{otherwise} \end{cases} \tag{3.25}$$

where the first case is defined to be true, if base types α, β form the base pair of an external loop at positions j, k and γ starts the 5' dangling end in position $j-1$. The second case is true for a pair β, α in positions k, j of an external loop with a dangling base γ in position $k-1$. The third case is true, if evaluating the 5' unpaired base α at position j, while γ, ω are the base pair of an external loop at positions $j+1$, l. $E_{\mathrm{5'\,dgl}}(\alpha,\beta,\gamma)$ is the score of the loop with base types α, β paired and an adjacent 5' dangling base of type γ from a NN table. The 3' end follows the same scheme:

$$E_{\mathrm{EL\,3'}}(\alpha, j) = \begin{cases} \frac{1}{3}\sum_{\beta\in\mathcal{A}}\sum_{\omega\in\mathcal{A}} m_{\beta k} m_{\omega k+1} E_{\mathrm{3'\,dgl}}(\alpha,\beta,\omega) & , \\ \frac{1}{3}\sum_{\beta\in\mathcal{A}}\sum_{\omega\in\mathcal{A}} m_{\beta k} m_{\omega j+1} E_{\mathrm{3'\,dgl}}(\beta,\alpha,\omega) & , \\ \frac{1}{3}\sum_{\gamma\in\mathcal{A}}\sum_{\omega\in\mathcal{A}} m_{\gamma l} m_{\omega j-1} E_{\mathrm{3'\,dgl}}(\gamma,\omega,\alpha) & , \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

where the first case is true for an external loop with a base pair of types α, β at positions j, k and a free base γ at the 3' end at position $k+1$. The second case is defined to be true for a base pair β, α in positions k, j of an external loop, with ω as the unpaired base at the 3' end in position $j+1$. The third case is true for an external loop comprised of a base pair γ, ω at positions l, $j-1$, while evaluating base α in the dangling end at position j. $E_{\mathrm{3'\,dgl}}(\alpha,\beta,\gamma)$ is the tabulated score of the loop with base types α, β paired and an adjacent 3' dangling base of type γ. With the non-GC penalty, the full contribution is calculated as

$$E_{\mathrm{EL}}(\alpha, j) = \begin{cases} E_{\text{non-GC}}(\alpha, j) + E_{\mathrm{EL\,5'}}(\alpha, j) + E_{\mathrm{EL\,3'}}(\alpha, j) & \text{if } E_{\mathrm{EL\,5'}}(\alpha, j) + E_{\mathrm{EL\,3'}}(\alpha, j) \neq 0, \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

where the first condition checks if there is any external loop contribution and if not, skips the penalty.

Multiloops, as shown in Fig. 3.3(e), are comprised by base pairs, which enclose at least two more stem loops. In *Vienna notation* ("(" opens and ")" closes a base pair and "." is an unpaired position [15]), the smallest such loop would be `(()())`, which has not been observed in nature, presumably for steric reasons. For larger structures there is in theory no upper limit for unpaired positions and helices spawning off the enclosing base pair. In the NN model the different branches of a multiloop are treated like external loops. Each base at the start of a new stem and the adjacent 5' and 3' bases get a contribution by $E_{5'\,\mathrm{dgl}}$ and $E_{3'\,\mathrm{dgl}}$. As important difference to external loops, the score is not restricted to unpaired neighbouring bases, but is also applied for paired bases adjacent to the current base pair. Subsequently, there is no splitting of contributions necessary, according to the 5' and 3' end like for external loops. When looking at paired sites, $E_{5'\,\mathrm{dgl}}$ and $E_{3'\,\mathrm{dgl}}$ have to be considered, while for the immediate neighbours only one score needs to be incorporated:

$$E_{\mathrm{ML}}(\alpha, j) = \begin{cases} \frac{2}{3}\sum_{\beta\in\mathcal{A}} m_{\beta k}\left(\sum_{\gamma\in\mathcal{A}} m_{\gamma k-1} E_{5'\,\mathrm{dgl}}(\alpha,\beta,\gamma) + \sum_{\omega\in\mathcal{A}} m_{\omega j+1} E_{3'\,\mathrm{dgl}}(\alpha,\beta,\omega)\right) & , \\ \frac{2}{3}\sum_{\beta\in\mathcal{A}} m_{\beta k}\left(\sum_{\gamma\in\mathcal{A}} m_{\gamma j-1} E_{5'\,\mathrm{dgl}}(\beta,\alpha,\gamma) + \sum_{\omega\in\mathcal{A}} m_{\omega k+1} E_{3'\,\mathrm{dgl}}(\beta,\alpha,\omega)\right) & , \\ \frac{1}{3}\sum_{\gamma\in\mathcal{A}}\sum_{\omega\in\mathcal{A}} m_{\gamma l} m_{\omega j+1} E_{5'\,\mathrm{dgl}}(\gamma,\omega,\alpha) & , \\ \frac{1}{3}\sum_{\gamma\in\mathcal{A}}\sum_{\omega\in\mathcal{A}} m_{\gamma j-1} m_{\omega l} E_{3'\,\mathrm{dgl}}(\gamma,\omega,\alpha) & , \\ 0 & \text{otherwise} \end{cases} \tag{3.28}$$

where the first condition is said to be true, if (α, β) at positions j, k are a base pair in a multiloop and position $k-1$ is the 5' neighbour of k and $j+1$ the 3' neighbour for j. The second case is defined true, if base types (β, α) at positions k, j are a pair of a multiloop and $j-1$, $k+1$ the 5' and 3' neighbours. The third condition is true for a multiloop base pair by (γ, ω) in positions l, $j+1$ and an adjacent 5' base in position j. The fourth condition is true for a base pair by (γ, ω) in positions $j-1$, l and a 3' neighbour at position j. The first two cases contribute by $\frac{2}{3}$, since they get $\frac{1}{3}$ of the 5' and 3' score, each.

Combining the terms for all structural motifs, the NN contributions for base type α at position j in the sequence adds up to

$$E_{NN}(\alpha, j) = E_{\mathrm{SL}}(\alpha, j) + E_{\mathrm{HL}}(\alpha, j) + E_{\mathrm{BL}}(\alpha, j) + E_{\mathrm{IL}}(\alpha, j) + E_{\mathrm{EL}}(\alpha, j) + E_{\mathrm{ML}}(\alpha, j) \tag{3.29}$$

Negative Design Term

In MFT, adding constraints to a system usually means extending the mean field by additional terms [90]. The negative design term deals with base pairs in wrong positions, which the tools discussed in §3.1.3 identify by folding prediction and comparing to the input structure. We do not see the sequence folded in any step of our approach, thus we add a quasi local mean field of unwanted interactions to E. For positions within general loop regions, E_{neg} is especially important, since they do not get contributions by the NN model. The idea is to assume stack loops between the current and all other positions, add the weighted contributions of the interactions, normalise by sequence length and change the sign to create an antagonist to E_{NN}:

$$\begin{aligned} E_{\text{neg}}(\alpha, j) = -\frac{1}{n} d \Bigg(& \sum_{k=j+1}^{n} b(m_{\beta k} m_{\gamma j+1} m_{\omega k-1} E_{\text{Stack}}(\alpha, \beta, \omega, \gamma)) \\ & + \sum_{k=1}^{j-1} b(m_{\beta k} m_{\gamma k+1} m_{\omega j-1} E_{\text{Stack}}(\omega, \gamma, \beta, \alpha)) \Bigg) \end{aligned} \tag{3.30}$$

where k iterates M downstream and upstream of position j, skipping the case when (j, k) form an intended interaction, n is the sequence length and d is a scaling factor of the term.

Heterogeneity Term

The negative design term already introduces some sequence variation, avoiding patterns likely to lead to suboptimal folding as shown in Fig. 3.1. But still the NN model favours a high GC content for pairs and adenine for loop regions. To reduce uniform patterns in the designed sequence, we introduce a completely artificial contribution to the system, the heterogeneity term. An ideal solution would force the system into more diverse states, without affecting the stability of the requested structure. For a current position in M, the strategy is to push it away from adapting likely states of its neighbourhood. Considering only the same row as the state we are looking at, probabilities of its environment are used as a score. In a first attempt to define the heterogeneity contribution, all positions in M were used with an exponential decay of influence on increasing distance in downstream and upstream direction. Because results were not completely satisfying, the final version was developed to use a window around the position of interest, giving each site within full

contribution:

$$E_{\text{het}}(\alpha, j) = \begin{cases} \frac{1}{2w} h \sum_{k=j-w}^{j-1} m_{\alpha k} + \sum_{k=j+1}^{j+w} m_{\alpha k} & j > w, j+w \leq n, \\ \frac{1}{j-1+w} h \sum_{k=1}^{j-1} m_{\alpha k} + \sum_{k=j+1}^{j+w} m_{\alpha k} & j < w, j+w \leq n, \\ \frac{1}{w+n-j-1} h \sum_{k=j-w}^{j-1} m_{\alpha k} + \sum_{k=j+1}^{n} m_{\alpha k} & j > w, j+w > n, \\ \frac{1}{n-1} h \sum_{k=1}^{j-1} m_{\alpha k} + \sum_{k=j+1}^{n} m_{\alpha k} & j < w, j+w > n \end{cases} \tag{3.31}$$

with w as half the size of the window, h a scaling factor and n the sequence length. The term is divided into four cases to cover positions near to the beginning or end of the sequence and sequences smaller than the window size.

NUN Contribution

This terms helps to prevent paired positions in the sequence to be populated with bases which are not supposed to form a pair. With the NN model, only canonical Watson-Crick complements and the wobble pair are covered. Since each position to be designed can be part of several structural motifs, contributions may add up to probabilities which favour *non-unitable nucleotides* (NUN) on both sides of a base pair. This makes E_{NUN} a penalty calculated as

$$E_{\text{NUN}}(\alpha, j) = \begin{cases} \sum_{\beta \in \mathcal{A}} m_{\beta k} E_{\text{NUNp}}(\alpha, \beta) & (j,k) \text{ are paired positions,} \\ 0 & \text{otherwise} \end{cases} \tag{3.32}$$

where $E_{\text{NUNp}}(\alpha, \beta)$ is the tabulated penalty for base types α, β inhabiting paired positions but not being included as pair in the NN model. For base tuples allowed to pair, E_{NUNp} is 0.

With all contributions defined, E becomes

$$E(\alpha, j) = E_{NN}(\alpha, j) + E_{\text{neg}}(\alpha, j) + E_{\text{het}}(\alpha, j) + E_{\text{NUN}}(\alpha, j) \tag{3.33}$$

for base type α at position j of M. All contributions are only defined for structures with at least two bases. Designing shorter sequences is not considered.

Thermal Noise

As mentioned before, SCMF design is deterministic, which means a low performance at sampling different sequences for the same input structure. This determinism can be avoided by adding thermal noise to the energy function used. For the NN score, this means adding small random numbers to all parameters in the tables, every time a different sequence is requested. As long as

the fluctuations are small enough for only neglectable influence on the stability of the structure, this approach may be explained by the imprecision of the parameters, as expectable from experimental determination.

Our system uses random numbers from the interval $[-0.5, 0.5]$, without significant changes in result quality.

3.2.5 Implementation

The implementation of the SCMF method is for most parts straightforward. The whole system depends on the size of the input structure and the sequence to be produced comes with a fixed alphabet. In consequence, all functions working on the sequence matrix should be manageable in linear time while they are executed in each step of the simulation.

Repopulating the matrix with new probabilities by Eqn (3.3), needs to be done for all non-fixed positions. This requires the quasi local mean field of E_{local} for every entry, but not as an online calculation. E may be precomputed and distributed in a second matrix, accessed by Eqn (3.3). Splitting this dependency retains a linear evaluation of the SCMF core routines. Columns which are fixed in the sequence matrix, either during the simulation or requested by the design task, need no further consideration. The implementation of the diverse terms needed to calculate the weighted score E of the system and the annealing procedure, will be discussed in a more separate way. The code of the SCMF design method is available integrated in the CoRB project as a tool called brot (basic RNA optimisation tool) [98].

Nearest Neighbour Contribution

The positive design term is not evaluated along the sequence matrix. Following the standard SCMF scheme here, would mean for each position to look up the structural features it is involved in and calculate the contribution separately. Instead of precomputing the input structure for fast look ups and caching scores to be reused in later calculations, we compute contributions for the structure and then assign them to the matrix.

Initially, the input structure is divided into different structural motifs known by the NN model. Next, the lists of motifs are iterated and contributions of E_{NN} are calculated. For each motif, parameters from the NN tables only need to be retrieved once and are reused for evaluating every sequence position involved. According to the various cases of contributions, some calculations may also be shared, e. g. products of probabilities. The results of each motif evaluation are stored in a two-dimensional array according to sequence positions, later to be used for repopulating the sequence matrix. For positions which are at an

overlap of structural features, the contributions are summed up in the storage matrix.

Evaluating E_{NN} for one step of the simulation also happens in linear time. Obviously, if there were no overlaps of structural features, each position in the sequence and the corresponding motif needs to be visited once. Since motifs only overlap by one base on each side, at maximum those positions need to be considered twice. Because all motifs are treated independent of each other, in the worst case we have to iterate two times the sequence length plus the overhead for maintaining the motifs, which is also linearly bound to the system size according to the Vienna package [15, 17].

The strategy to calculate E_{NN} along the set of structural motifs instead of the sequence matrix, has one advantage concerning a speed-up of the whole simulation. When a column in the matrix is fixed, it is actively skipped during iterations, since it has to stay available for possible data dependencies and the final sequence. Once all participants in a structural motif are fixed, the motif itself needs no further evaluation. Also the result of the simulation is a sequence which does not require for structural information. This means, if all base types within a motif are set, it is removed from the system, speeding up the next step of the simulation requiring less iterations.

NUN Term

The penalty for base types which are not supposed to inhabit paired positions, is applied to the matrix storing E by a linear scan of the sequence matrix.

Negative Design & Heterogeneity Term

In the negative design and heterogeneity term, the contribution of a sequence position is depending on stretches of the sequence matrix itself. For negative design, this is the whole width, for the heterogeneity term the window size. With a naïve approach, these terms would need a full evaluation for every position on the whole system, resulting in a quadratic time consumption (assuming a window size equal to the sequence length). Linear time behaviour is achieved, by calculating the full energy contribution once in the beginning of the simulation and then update them for every position while iterating along the matrix. Thereby an update means to remove the contribution of the current position while reintroducing the one last removed.

Scores for each position are simply added to the values in the matrix storing E.

Annealing

The sequence entropy, needed to control the cooling rate of the system, is calculated for each step of the simulation. By Eqn (3.5), this is already a linear term for a fixed alphabet. All parameters from §3.2.3 depending on the sequence entropy are updated in constant time from previous values. The short and long term observers s_{short}, s_{long} include the sequence entropy directly in their calculation, while the cooling rate depends on the ratio between s_{short} and s_{long}.

Parameter Optimisation

To optimise the parameters of the SCMF simulation, the simplex algorithm [93, 99] was used. Obviously the goal is to find a set of parameters, which leads to sequences resembling the correct shape. Another objective is the stability of the structure, measured by folding probability. Assuming the input structure for a predicted sequence, Gibbs energy is calculated according to the NN model using `er2de` (evaluating RNA 2D energy) of the `CoRB` project. The partition function, needed to calculate probabilities, and a predicted fold, are computed using the Vienna RNA package. To compare a sample structure with the prediction, `RNAdistance` of the Vienna package is utilised to calculate the base pair distance [15]:

$$\Delta_{\text{bp}}(q_u, q_v) = \sum_{\substack{1 \le i \le n \\ i < j,k \le n \\ j \neq k}} y_{ij}(q_u) + y_{ik}(q_v) \tag{3.34}$$

where q_u, q_v are two structures to be compared and $y_{ij}(q) = 1$ if (i, j) are paired in structure q, otherwise $y_{ij}(q) = 0$. n is the size of the smaller structure, then i, j and k are positions in the structures, restricted to count base pairs only once at the opening partner. Since pairs are only considered if closing positions j and k are not the same, the base pair distance describes the number of different pairs in two structures.

With this information provided in each step of the simplex algorithm, the cost function optimised, is calculated as

$$f_{\text{cost}}(X) = \sum_{q \in Q} (1 - P(q|v)) \Delta_{\text{bp}}(q, q_v) \tag{3.35}$$

where v is a sequence predicted by `brot` for structure q using parameters X to be tested and q_v is the structure predicted by `RNAfold` for v. Q is the set of structures listed at [100], used for evaluation. Test structures were chosen to contain all structural motifs of the NN model. The base pair distance of

a structure q to be evaluated and the predicted fold q_v is weighted by the probability for q not being the optimal structure of v as calculated by `RNAfold`. This way, the contribution to costs scales up in case v is a bad prediction, but gets smaller when it becomes more likely to fold into the right shape.

The simplex optimisation is terminated when the difference between best and worst corners is less than a weighted base pair distance of $1 \cdot 10^{-5}$ or the maximum of 10 000 steps is reached. A new run is started, centred on the best point found, but with the scattering of initial points 20 % of the value used in the previous run. A third optimisation is started, if the second run improves the best value found by more than a weighted base pair distance of $1 \cdot 10^{-5}$.

Testing Predicted Sequences

For evaluating predicted sequences and comparing the various tools, the success rate, folding probabilities and the ensemble defect are calculated with help of the Vienna package. Just based on the nucleotides in the sequences, various sequence identities and the average `GC` content are also used.

The *success rate* counts how often a sequence is predicted, likely to fold into the requested structure. Every tested sequence is run through `RNAfold` in probability mode. Predictions with highest probability are compared to the input structure and marked as success if matching. The success rate is calculated as the mean of positive answers over all runs for the same structure.

The *folding probability* is not simply connected to a true positive rate. Instead of the most likely fold, the probability of a predicted sequence assuming the requested structure is calculated. This is more of an overall score showing the stability of predictions. While also sequences contribute which are excluded in the success rate, successful sequences at low probability will lower the score. With the partition function and Gibbs energy derived from the Vienna package, folding probability is calculated using the Boltzmann relation. For a target structure, the result is presented as mean over all runs.

After examining the folding probability of sequences, focus shifts to those excluded by the success rate. A true negative result means a sequence which is not predicted to resemble the target shape. By looking at the probability of true negatives to fold into a target structure instead of its own most probable structure, one may get an idea how badly a prediction fails. More in general, low values mean that a method does not get close to a good solution, while at higher probabilities it gets more likely that the issue is solvable. This would mean changing parameters or using parts of a suboptimal solution as starting point for another run.

Evaluating for positive and negative folding probabilities in separate may not always be the most helpful thing. If a sequence is predicted into a fold

which only slightly differs from the target structure, simply saying that the design process failed could be to coarse grain as a measure. Even if a base pair is missing, it may be present with significant probability in alternative folds. To account for the base pair level, the *ensemble defect* measure was introduced [101]. Using the McCaskill algorithm, one can conceptually visit every possible structure q and calculate its probability $P(q|v)$ for the sequence v. For some base pair (i, j), we then calculate its weighted probability in all structures from

$$P_{ij}(v) = \sum_{q \in Q} P(q|v) y_{ij}(q) \tag{3.36}$$

where $y_{ij}(q)$ is the same as in Eqn (3.34) and Q is the set of all possible structures given sequence v. This is the ensemble-weighted likelihood of a base pair being present. Next the ensemble-weighted defect r is calculated by comparing each base pair in the target q^* to its probability in the ensemble:

$$r(v, q^*) = n - \sum_{\substack{1 \le i \le n-1 \\ 2 \le j \le n}} P_{ij}(v) y_{ij}(q^*) \tag{3.37}$$

where n is the number of bases in the sequence. Clearly, if all desired base pairs are present with a probability of 1, $r(v, q^*) = 0$. Of course, there is always some probability for suboptimal structures, so even in near perfect sequences $r(v, q^*)$ will always have some small positive value.

In the second class of comparisons, only sequences predicted to fold into the target structure are considered. But then the evaluation is solely looking at the base composition of sequences. As discussed when explaining the idea of sequence design, the *GC content* is of some interest. We just count the fraction of `G` and `C` in the sequences and show the average, standard deviation and outliers. This will give an overview if a method always goes for high `GC` content or more moderate levels. Furthermore a high spread could mean that a prediction approach is not driven by the idea of adding up the strongest pairing partners to get to a stable sequence. Since all methods promise to deliver different sequences for the same input on multiple runs, the *number of unique sequences* produced is also considered. Beside the absolute count, the *similarity of different sequences* for the same structure is calculated. That is about the flexibility inside a method itself, while it is also interesting to see how the results between SCMF and the other tools compare. Since we have a completely different approach, we want to know if our sequences are drastically different or very similar to the other tools. To get an overview, the *identity of sequences between SCMF and each of the other methods* is collected.

3.3 Results

When evaluating new prediction software, one usually wants to see two aspects investigated: the performance compared to tools for a similar task and the behaviour on complex systems, or those of special interest. For the performance tests, our approach is compared to the tools discussed in §3.1.3. As case of special interest, amongst others a ribosome sequence will be evaluated. Finally, brot was also used for a real-world case study, synthesising a designed sequence and determining its structure *in vitro*. But before our approach can be used for any testing, the parameters of the simulation need to be defined. Since this means setting up all the terms in the scoring function, we also give a concise overview on the influence of most of them.

3.3.1 Parameter Optimisation

To determine a starting point of the simplex algorithm, already the simplex is used. The very first initial parameters are just guessed and refined by repeated optimisation runs. Values, that are reoccurring in the results, are used to create our starting point for the final simplex iteration. This is then used to create the set of default parameters of brot, as shown in Table 3.1.

Parameter	Default	Description
steps	1000	max. no. of iterations
T	2.000	initial temperature
d	0.420	negative design factor
h	9.730	heterogeneity factor
w	1	heterogeneity window size
seq. entropy thresh.	0.337	sequence entropy threshold
λ	0.627	ratio of the system memory
β_{long}	0.949	long term seq. entropy ratio
β_{short}	0.500	short term seq. entropy ratio
s_c	0.816	threshold to speed up or slow down cooling
c_{min}	0.866	lower bound of the cooling rate

Table 3.1: *Parameters of the SCMF approach.* All parameters, beside the maximal number of steps, as determined by the simplex method [93, 99]. To assure the end of simulations by cooling or convergence, the number of iterations allowed was set to a large number during run-time. Afterwards it was reduced to a value still assuring the optimised results for the test set.

Beside optimising the fitness of predicted sequences by the simplex algo-

rithm, we also examined the progress of temperature and sequence entropy over time for the final set of parameters. As an example, for structure 16 of the evaluation set at [100], Fig. 3.9 shows the observed values plotted against time. Sequence entropy goes down right from the beginning, while the system is still at high temperature. After 35 % of simulation time, it reaches a plateau, with 33 % of the sites already fixed. Towards the almost constant phase, the descent slows down, which is an indicator for the absence of abrupt changes in the system. After 84 % of the simulation have past, cooling increases and the sequence entropy goes down below the convergence threshold. Along the decrease of sequence entropy, cooling is slowed down, as intended. With a final temperature of 0.450 and entropy of 0.334, the simulation is stopped by achieving convergence and not because the maximum number of 1000 steps is reached.

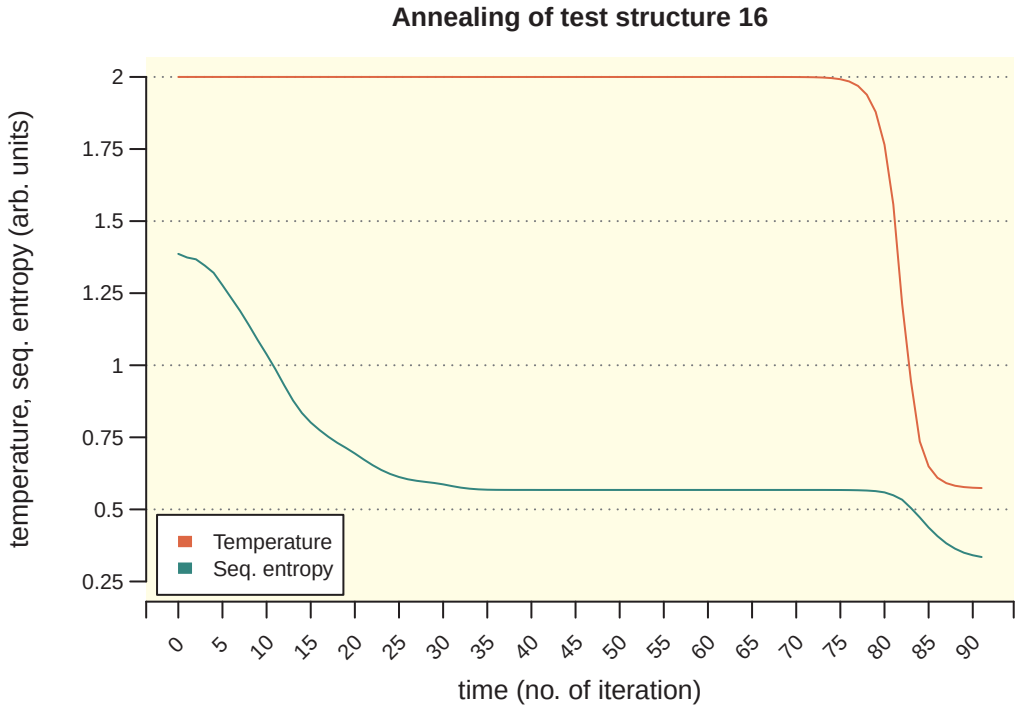


Figure 3.9: *Example of annealing using the default parameters.* Cooling and sequence entropy values for structure 16 of the evaluation set at [100]. In 91 iterations, the temperature cooled down to 0.574 starting from 2.000. In the same time, the sequence entropy dropped from 1.386 to 0.334, below the threshold of 0.337.

For the structures from the test set, all sequence entropy and temperature curves look similar. But this just means, that the default parameters work well

for the structures they are optimised for. While every structural motif covered by the NN model is in the set, for some input systems there may be need to adjust settings. This means repeated simulations with modified parameters, until a satisfactory result is achieved.

3.3.2 Energy Terms

To see the effects of the various terms of our scoring function, the fully parameterised system had to undergo several small-scale tests. Exploring all possible combinations of contributions turned on and off, simulations were run on two RNA motifs, shown in Table 3.2. For the small hairpin loop, the task reduces to get the base pairing right. With all contributions influencing probabilities of single bases, this seems to be the most basic test. To introduce intramolecular dependencies, an internal loop was used. Since the NN model incorporates this motif, it should be able to pick an already good sequence from its parameter set. The remaining terms have no prior knowledge about the constraints but should not disturb a correct answer by design. For base pairs adjacent to the loop region, which are outside the scope of the motif stored in the NN model, all terms should work together to find an answer unlikely to favour an undesired shape.

Table 3.2 shows the results, gathered below the structure they belong to. For each sequence, the folding probability and Gibbs energy are provided, with the last column declaring contributions enabled for the simulation. Using single terms, the system behaves as expected: the terms which reward base pairs find valid answers in both cases, while those targeting other sequence features do not succeed. Highest folding probabilities are achieved for the NUN term by rather artificial looking sequences. With `G` and `C` separated between both sides of a helix, the number of stable folds is obviously limited. Results produced by the NN model come at considerably lower energy and higher sequence variation. This originates from the knowledge of the NN model about stabilising effects of nucleotide-combinations in RNA structures. Opening a loop with an opposing `G` and `C` is one example seen for both motifs tested. While this gives a lower energy by parameters, it introduces additional possibilities for base pairing, enlarging the structural ensemble and lowering folding probability. For the negative design and heterogeneity terms, produced sequences cannot fold. The latter one is designed to create less uniform sequences without paying attention to base pairs. Considering the window size of 1 and the repetitious sequences, this term works as intended. Negative design is also not meant to actively help base pairing but avoid unwanted interactions. Looking at this goal alone, it is also met with sequences unable to form canonical pairs.

Also for two contributions enabled, the negative design and heterogeneity term do not find a valid answer when combined. But as soon as joined by one of the remaining terms, sequences are able to fold. Especially negative design seems to be potent by taking the lead concerning folding probability. With the NN model involved, we also see low energies. Including heterogeneity does introduce more sequence variation at the cost of lower probability for the desired shape. In the hairpin loop, the effect is not exceptionally big, while in the internal loop probability almost cuts half when joined with the NUN term. This should be expected with growing complexity of a structure for two terms which do not prevent population of unpaired positions with corresponding bases. When the NUN term is used together with the NN model, results do not change much compared to only enabling NN contributions. For the internal loop sequences are equal and in the hairpin loop one GC pair is inverted, while folding probabilities and energies do not change for both motifs. Such small or no changes are by intention of the NUN term. It is not designed to drive the system into forming base pairs, but to prevent it from optimising paired positions for non-canonical configurations.

When using three terms enabled in a simulation, one would expect valid sequences for all runs. Every combination of terms contains at least one contribution actively pointing positions towards base pairs. Technically all results are able to fold into the right shape but for high-ranking results the NN term seems to be essential. In both test cases where it is absent, energies are high and for the internal loop, having only the NUN term as matchmaking contribution, folding probability is low. While for a lower number of terms enabled, the NUN term does not perform as badly, here we have two contributions disturbing stability at the same time. But repairing their influence is not the objective of this term. Pointing towards stable folds is left to the NN model. When looking at both test structures, which combination is the most favourable is hard to tell, since none of the selections scores best in both cases. Compared to enabling two terms per simulation, results are rather similar as soon as the NN term is involved. Only the sequence variation seems to be a bit higher when using negative design and heterogeneity together with the NN model.

All four terms enabled leads to folding probabilities above 90 % for both test cases. Energies may be lower as shown by other setups, but are still at least ten times lower than for the worst valid sequence. But showing top performance for the complete scoring function is not the objective in this quick survey. The idea is to verify that every term has its effect on the result when combined. Looking at the single-contribution runs, behaviour is as expected. Going through the various combinations we see that the terms do work along instead of cancelling each other out. While the impact on the result is not the same between contributions. The effect of the NUN term on free energy

and folding probability is small compared to the NN model. Enabling it for simulations which already have NN contributions only swaps one base pair in two cases. For the internal loop, negative design has a higher influence on folding probability than the heterogeneity term which is not as visible for the hairpin loop. The most important observation is that combining all terms does not prevent valid results. None of the other contributions is potent enough to rival the NN model in finding good answers for an input structure.

3.3.3 Performance

To compare our method with the ones described in §3.1.3, all programs are run on equal test sets and the predicted sequences are evaluated for the measures described in §3.2.5. As sets of test structures, the same as by Busch & Backofen [59] and Andronescu et al. [60] are used. Those cover artificial 2D structures (sets Ia, Ib and Ic), designed following certain features, and predicted folds for biological sequences (sets IIa and IIb). An additional set made of 2D representations of experimental RNA structures is left out at this point. One reason is the poor performance participants showed on this set. That is in contrast to published results which could not be reproduced. Inspecting the list, it turned out that the publication shows 10 structures, while the provided set has 13 members. One of them features several non-closed base pairs rendering it unusable for the design task. More in general, the original RNA structures behind the test set feature several pseudoknots which are declared as removed without further information on how exactly they were opened.

The evaluation procedure is the same for all test sets. For each structure, every tool is run 100 times and the predicted sequences are used to calculate our measures. Here, all of the values calculated use folding probability to assess the shape a sequence will adapt and its stability. This is in contrast to earlier evaluations behind the tools we compare to, which all use MFE structures when it comes to structural features. But as explained in §3.1.1, just optimising free energy seems not to be as significant as probability, if available.

Since sequence identities and the GC content are very similar over the various test sets, those measurements will be shown combined.

Another value to be compared is the running time of each tool. Our idea is not a hard contest but providing a feeling on how long the user has to wait for an answer. With the computational prediction part being only the first step in the design process of a new sequence, there is only a technical difference in a running time of a couple of minutes or a few hours. Even for sampling multiple sequences, minutes do not matter since this is usually done in parallel on high performance clusters nowadays. However, the viewpoint changes if

Predicted sequence	$P(q\|v)$	ΔG [kcal mol^{-1}]	Terms
((((((.....))))))			
GCCCGCGACAAGCGGGC	0.99	−12.40	ndhu
GCCGGCGACAAGCCGGC	0.99	−12.40	ndh
CCCCCCGAAAAGGGGGG	0.95	−13.10	ndu
GCGGGCGACAAGCCCGC	0.99	−12.40	nhu
CUAAAUCACACAUUUAG	0.77	−0.80	dhu
GCCCCCGAAAAGGGGGC	0.99	−13.20	nd
GCGGGCGACAAGCCCGC	0.99	−12.40	nh
GCCGCCCAAAGGGCGGC	0.55	−13.10	nu
ACAAAACACCACACACA	0	–	dh
CCGUAGAAAAACUACGG	0.94	−6.80	du
CGCGCGAACAACGCGCG	0.74	−9.50	hu
GCGGGCCAAAGGCCCGC	0.55	−13.10	n
AAAAAAAAAAAAAAAA	0	–	d
AACACACACACACACAC	0	–	h
GGGGGGAAAAACCCCCC	0.85	−12.00	u
((..((((.....))))..))			
CCGAUCGCGACAAGCGAGAGG	0.95	−8.30	ndhu
CCUGAGGCGAUAAGCCUUGGG	0.73	−12.20	ndh
CCGAGCCCGAAAAGGGCGAGG	0.94	−10.60	ndu
GCUGGGGCGACAAGCCCUGGC	0.73	−14.20	nhu
UAGACAUCGAGACGAUGACUA	0.01	−0.80	dhu
GCGAGCCCGAAAAGGGCGAGC	0.97	−10.70	nd
GCUGGGGCGACAAGCCCUGGC	0.73	−14.20	nh
GCUGGGCCCAAAGGGCCUGGC	0.41	−14.90	nu
CACACCACACACACACACACA	0	–	dh
CCAAGUACAAAAAGUACAAGG	0.66	−3.60	du
CGAACGCGAACAACGCGAACG	0.36	−4.60	hu
GCUGGGCCCAAAGGGCCUGGC	0.41	−14.90	n
ACACCACACACACCACACACA	0	–	d
AACACACACACACACACACAC	0	–	h
GGAAGGGGAAAAACCCCAACC	0.84	−8.20	u

Table 3.2: *Effect of energy terms in SCMF.* Various combinations of energy terms turned on and off for the same design task. To show the influence of a term on the stability of a sequence produced, folding probability and Gibbs energy are provided. The last column shows the combination of enabled terms: `n` - NN contribution, `d` - negative design, `h` - heterogeneity, `u` - NUN term.

time consumption reaches amounts where it is not easily possible anymore to rerun, in case of needed adaptation. The whole evaluation presented here is run on the same hardware, so we skip any transformation of values and just base measurements on the real running times. To keep the idea of providing an overview, results are combined over all test sets.

The rest of this section shows results. First the test sets are presented, followed by the combined measures.

Test Set Ia

Test set Ia is a set of artificial structures provided by Busch & Backofen [59]. They designed it using an internal tool for overall sizes between 30 and 200 nucleotides, certain loop sizes and stem lengths. It comes with a total of 298 structures listed at [100]. The original set had 300 structures but two of them were redundant.

In contrast to the other sets, here we run `RNAinverse` in one additional mode. Probability and MFE mode will be evaluated everywhere, but when running in MFE mode, treatment of dangling ends will be shown in two ways. The default only considers unpaired bases for dangling ends, affecting how the ends of helices may be treated. But there is a switch to enforce the same energy model as used in probability mode. This is our default for the other sets since SCMF treats the energy model the same way.

brot, the tool running our SCMF simulation, is also shown in three variants. Beside the default parameters, for this set it will also be invoked with the negative design term turned off and a third time without the heterogeneity term.

An effect of the two terms is immediately visible in Fig. 3.10. In its default configuration, brot gets to a success rate of 95 % on average. The 3 standard deviations span from 100 % to around 60 %. But there are 14 outliers, targets without any solution. The standard deviation gets bigger once the negative design term is turned off. Reaching from 100 % down to 0 %, it means that we still can find good sequences without negative design, but with an average of around 30 % this is a rather rare event. Without pushing for heterogeneity, the average gets back to the level of more than 90 %. However, the 3 standard deviations gain 20 % with new outliers at lower success rates. This means turning on the heterogeneity term aids making the desired fold the most probable one more often for test set Ia.

Examining the success rates of INFO-RNA, the probability mode always yields sequences that fold into the target structure. When run in its MFE mode, results are different. The average success rate drops by about 20 % and the 3 standard deviations cover the whole range. This does mean that there are still

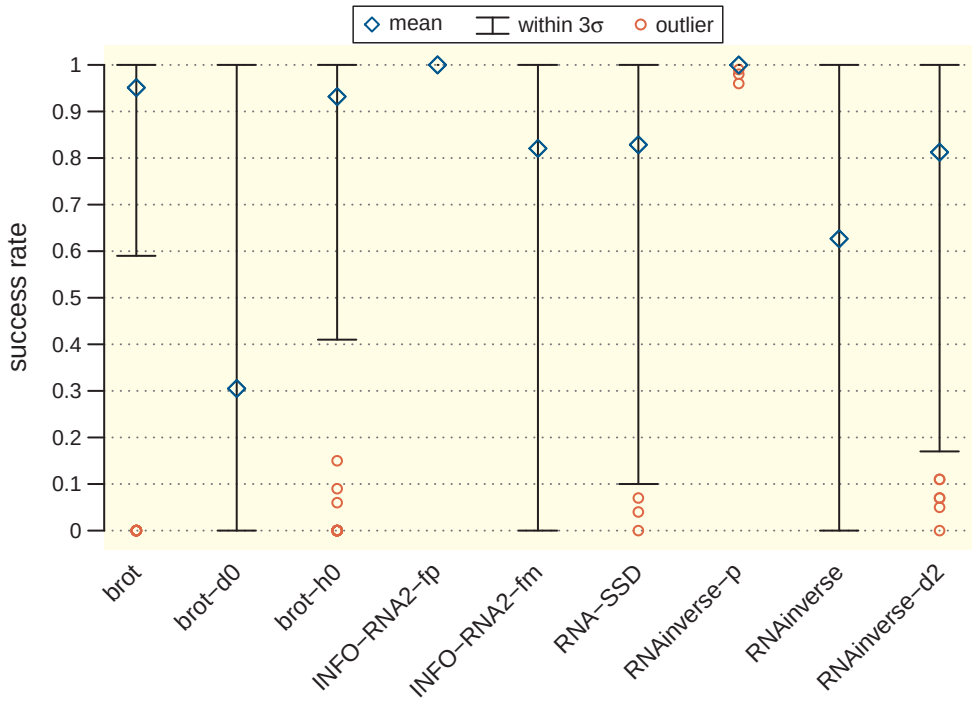


Figure 3.10: *Success rate for test set Ia.* Fractions of sequences which fold into the requested structure. All tools were evaluated with `RNAfold` in probability mode. Diamonds mark the average success rate considering all structures of the test set. Whiskers show the spread of individual success rates within 3 standard deviations in both directions. Outliers are marked by red circles. brot-d0 shows results for brot with negative design turned off, brot-h0 has the heterogeneity term turned off. INFO-RNA2-fp shows results for INFO-RNA version 2 in probability mode, INFO-RNA2-fm represents MFE mode. `RNAinverse` was evaluated in probability (-p) and MFE mode. `RNAinverse`-d2 shows the MFE mode evaluating the NN model the same way `RNAfold` does in probability mode.

targets with matching sequences produced in every run, but at the same time some targets are not covered by any successful result.

RNA-SSD does not provide a switch between a probability and a MFE mode. The usage of the NN model always goes by energy minimisation. Results look similar to INFO-RNA. But the average is slightly higher in the 80 % range and the 3 standard deviations mark the case of complete failure only an outlier.

With its multiple options concerning the energy model, results for `RNAinverse` are shown in the last three columns in Fig. 3.10. Running the partition function approach looks promising. Success rates go up to almost 100 % on average with only a few outliers and the worst still living around 85 %. The last two columns show `RNAinverse` featuring MFE mode. Independent of the way the NN model is treated, the worst result is always complete loss of valid sequences. While achieving success rates of 100 % is still possible, averages stay below 85 %. When treating the energy model the same way as for the evaluation, averages are higher almost by 20 %.

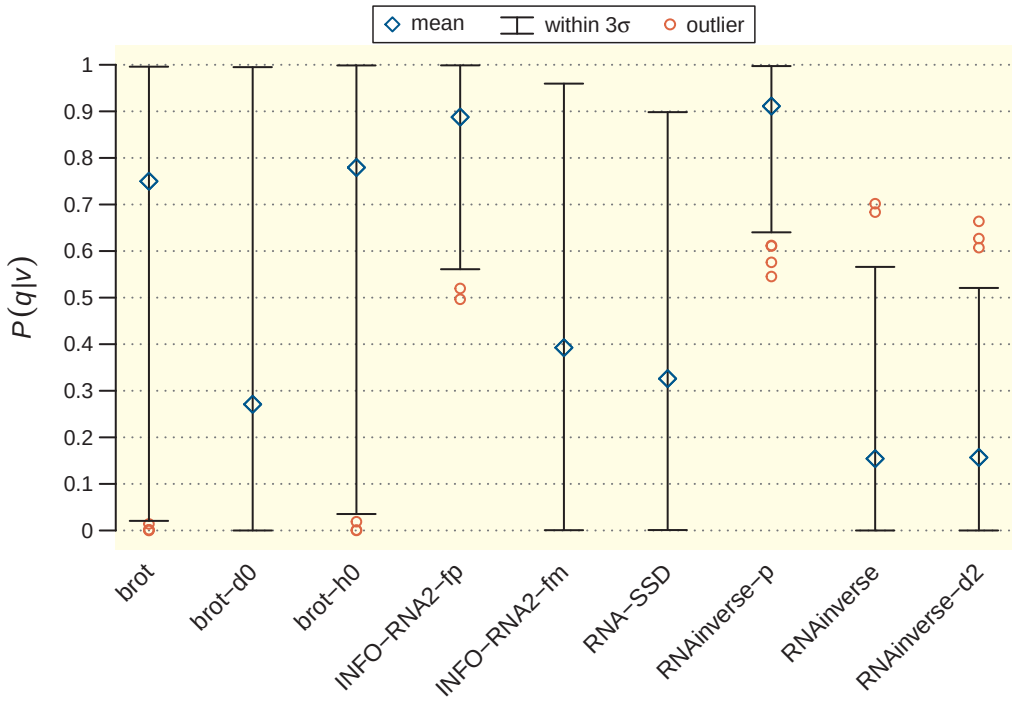


Figure 3.11: *Average folding probability for test set Ia.* Probability of designed sequences to fold into a requested structure. Whiskers and red dots have the same meaning as in Fig. 3.10. Labels for methods have the same meaning as in Fig. 3.10.

After success rates, Fig. 3.11 shows the average folding probabilities. brot

shows similar behaviour as before. The default parameters score between 0.70 and 0.80 with some outliers around 0. The 14 missing targets serve as an explanation here: while they are able to form canonical base pairs where input structures need them, they are predicted with almost zero probability to fold. A reason for those weak results may be a wrong setup of the simulation. Other sequences for the unsolved targets are responsible for the 3 standard deviations reaching down below 0.10. Every missed structure still produces 100 sequences which operate at low probabilities when calculated assuming the target shape. As before, turning off the heterogeneity term shows an effect. This time the average gets slightly better. This could mean, that the sequences which do not fold into the target structure, still have a high probability towards this conformation. Turning off the negative design term shows performance as expected, going considerably down.

INFO-RNA shows that a 100% success rate does not always mean high folding probability. For the probability mode, one outlier goes below 0.50, probably pointing at a large ensemble for sequences of this structure. But 3 standard deviations show almost all of the sequences folding at above 0.55. In MFE mode INFO-RNA performs with a wide spread as expectable from its mixed success rates.

For RNA-SSD in Fig. 3.11 the highest probabilities stay below 0.90. Further, getting down to 0 is not an outlier anymore but within 3 standard deviations of all results. The overall average stays below 0.35.

Top ranking probabilities for test set Ia are produced by `RNAinverse` in probability mode. The success rates are similar to INFO-RNA but with some outliers. Here the average folding probability goes above 0.90. The 3 standard deviations shown stay between 1 and 0.60 and the weakest outlier comes at 0.55. This does look better than INFO-RNA. In contrast for the MFE mode, one could speak of the worst results in the comparison. Regardless of the exact usage of the energy model, when optimising MFE the highest probability is achieved by an outlier at 0.70. The upper border of 3 standard deviations stop around 0.55 and the averages stay below 0.20.

Fig. 3.12 shows the probability of true negative sequences to fold into a requested target structure. On a first impression, `brot` does not look well. An average probability of 0.12 seems low enough to conclude that those sequences are not easily fixed to serve as valid solution. But from the success rates we know that 14 targets are covered here and there are only five more for which true negatives were produced. One of those missed structures is covered by a few sequences which fold at a probability of 0.44. While this value vanishes when averaging, this is the highest value amongst all tested tools for test set Ia. Without the negative design term the standard deviation gets bigger but this may be just because there are much more true negatives to count on. We

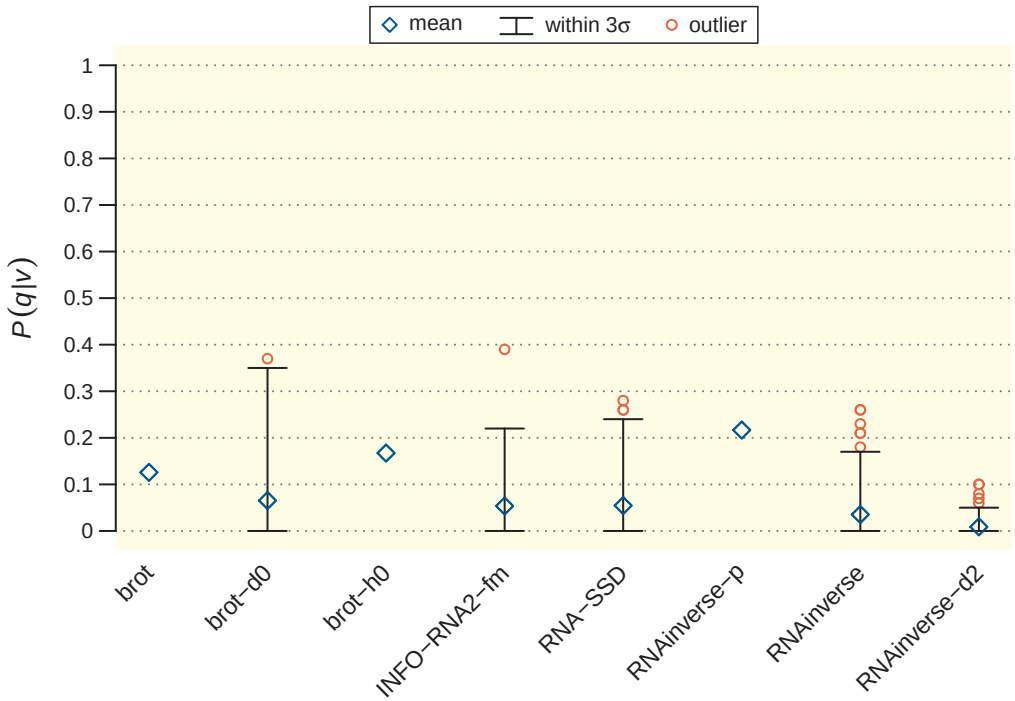


Figure 3.12: *Average folding probability of true negatives for test set Ia.* Probability of sequences to fold into a test structure, predicted by `RNAfold` with a different shape at higher probability. Whiskers and red dots have the same meaning as in Fig. 3.10. Labels for methods have the same meaning as in Fig. 3.10. INFO-RNA2-fp is missing in the set, since all `RNAfold` predictions comply with the desired shape.

find sequences for 244 target structures. The missing heterogeneity term looks again similar to brot. It comes with seven more samples, not being predicted into the target structure. With heterogeneity turned on, they do fold as expected, so if this term does not produce completely different sequences, they should still gain some probability towards valid results.

For INFO-RNA only results of the MFE mode are left. The probability mode does not produce any true negatives. The highest probability is reached by an outlier at 0.40 and the average stays below 0.10. 0.40 is also almost the average folding probability in Fig. 3.11.

As the positive folding probabilities look similar to INFO-RNA, the probabilities of true negatives of RNA-SSD to fold into a target structure look similar again. What is missing is a high outlier, at maximum RNA-SSD stays below 0.30.

RNAinverse in probability mode comes with the highest average. But only for three structures true negatives are produced. Those are visible as outliers in the success rate and folding probability plot. The probabilities are different to the latter plot, since here we do only count sequences failed to fold. Average folding probabilities are instead produced from all sequences for a structure. In MFE mode, both variants of RNAinverse operate at low averages and standard deviations. Only one outlier goes above 0.25. When running RNAinverse with the energy model as used by probability mode, no values higher than 0.10 are observed. When evaluated with RNAfold for probabilities, one would expect a better result when treating energies the same way, not the opposite.

The ensemble distance presented in Fig. 3.13 confirms the trends from the other plots. brot gains an average defect of 0.01 but the space of 3 standard deviations is populated until 0.11. Outliers can be found up to 0.30. Those originate from the 14 structures without well-predicted results. But an average ensemble defect of 0.30 could mean that the majority of base pairs agrees with the target at reasonable probability. What we do not learn from the ensemble defect is the nature of weak base pairs: are single pairs spread over the structure wrong or complete stretches like a flipped helix? If the negative design term is turned off, the average goes up to 0.06 and also the standard deviation is considerably higher. Together with outliers going up to 0.40, this shows that having the negative design term has a positive effect. Without the heterogeneity term, the average is similar as with default parameters, while outliers can be seen above 0.40. The standard deviation is also higher. This may be an effect from the seven additional sequences failing to fold into the target shape.

Results for INFO-RNA in probability mode are as one would expect from the success rates. Obviously, ensemble defect will not be 0 everywhere with the spread of folding probabilities. But still, averages, standard deviation and

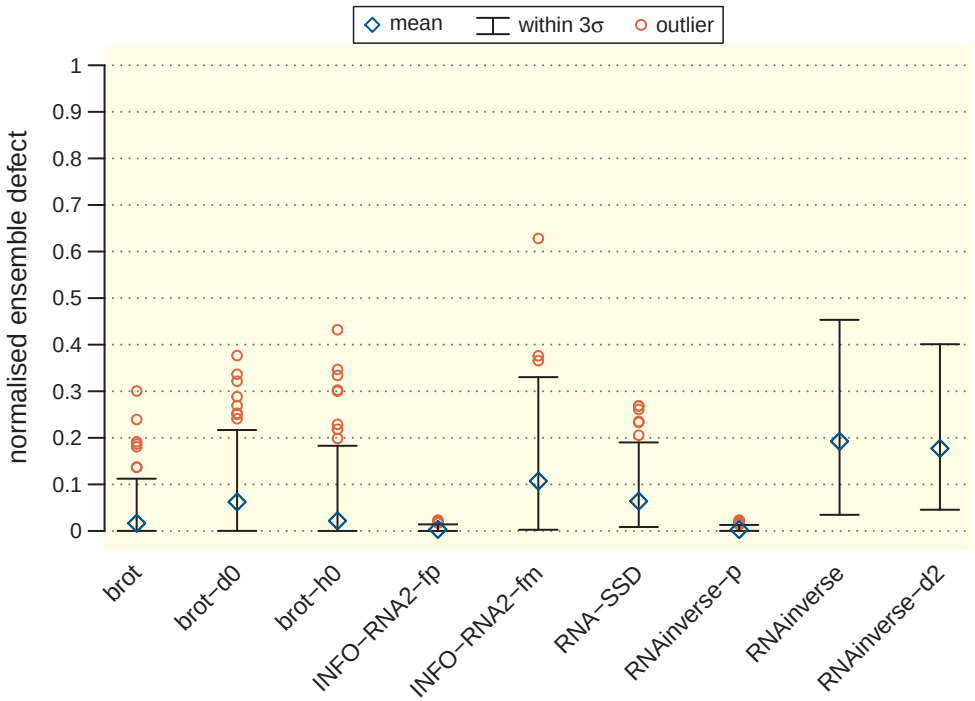


Figure 3.13: *Average normalised ensemble defect for test set Ia.* The average is calculated over all test structures and for all runs of each tool. This includes all designed sequences, regardless if they are corresponding to their input structure. For each structure, ensemble defect is normalised by size. Whiskers and red dots have the same meaning as in Fig. 3.10. Labels for methods have the same meaning as in Fig. 3.10.

outliers are very close together. MFE mode also looks reasonable considering its folding probabilities. The one outlier above 0.60 is the greatest distance of the whole comparison.

With its success rate, the ensemble distance of RNA-SSD occurs a bit to big. When comparing folding probabilities, RNA-SSD already looks similar to brot without negative design. The same holds here, while RNA-SSD comes with outliers at lower distance.

RNAinverse in probability mode is very similar to INFO-RNA. If low performance at folding probability is an indicator for a high ensemble defect, this would explain results for RNAinverse in MFE mode.

If we would have to choose an overall "winner" for test set Ia, this would be RNAinverse in probability mode since it has the highest average folding probability. Also true negatives fold into the right shapes with highest probabilities when compared to the other programs. For the ensemble defect it ties with INFO-RNA and for the success rate INFO-RNA is slightly better. Therefore INFO-RNA would be the second best followed by brot. While for the true negative folding probability brot shows the second highest average. RNA-SSD does not perform as well as other tools or configurations of them.

This is not a bad result for brot. This particular set of structures was designed, and most likely also optimised, for INFO-RNA. RNAinverse just looks better for us, since we switched the evaluation from MFE to probability mode. For this mode, RNAinverse does not split a structure and solves it in separation but samples the full sequence. This may be slow but still means iterating the full sequence until a matching structure is predicted. INFO-RNA does not explain how exactly its probability mode works. They may use the partition function either to evaluate sequences or for ranking candidates for the next search step.

Where brot does not look that well is on the 14 missed targets. Bad structures in test set Ia are small and simple enough for manual inspection. None of the sequences predicted for these structures produced a completely different base pairing pattern. For some, helices are to short or to long and two placed a helix in a different location. Looking at the annealing plots, like the example in Fig. 3.9, descent of entropy looked rather steep for most of those targets. Since this is a hint for too fast cooling or short simulation time, brot was rerun with increased cooling threshold s_c and a lower entropy boundary. This already solved 9 of the targets. Since the remaining ones still had base pairs in wrong positions, they were run again with increased negative design term. That left one target still refusing to fold into the requested structure. That structure contains two neighbouring bulge loops and a hairpin. The solution found by SCMF has the lowest possible energy for bulge loops. When predicting the structure the two loops are merged into one internal loop. Looking

up the energies, it turned out that the particular sequence is a corner case in the NN model. While the parameters for the sequence are optimal when considered as a bulge loop, the same sequence has a lower energy when assuming an internal loop. The reason is a large bonus given to an internal loop, when the first mismatching bases of the loop are GA. Overcoming this energy barrier by the negative design term means scaling it so high that it starts affecting the rest of the structure. When fixing the G in the GA tuple to a C the internal loop loses its bonus and a valid solution is found by brot.

Fixing the 14 missed targets is rather simple, but for the last case. This means, while running an artificial mass test brot does not perform best, but when considering a more real-world scenario it offers the realistic chance to fix problems. The numbers for the annealing plots needed to see how well the simulation worked, can be produced by brot with a command line option. The parameters to be changed are also accessible and our tool works fast enough to rerun. Only the last structure requires special knowledge about the whole method and the energy parameters. But the solution, fixing a base, is simple enough to explain to users if they do not do this by themselves. The other tools in this survey offer a lot less parameters to influence the result, if any at all. They also do not give out any information what may be wrong in the prediction process.

In the additional evaluation for the energy terms of brot, switching negative design on and off shows expected behaviour: without an extra term to avoid unintended interactions, results get considerably worse. For the heterogeneity term the effects are not nearly as big. While this is how the term is intended, it seems that activating this contribution yields little or no improvement in predicted stability. This should produce sequences of higher variability. Looking at repeat numbers, this can be confirmed. The longest single base repetition by brot is 18 nt for both cases. However, without the heterogeneity term this occurs 300 times for test set Ia, while when turned on we only see 4 such sequences. Examining the lower end, we find 3314 repeats of size 3 with the term enabled and 96 528 without.

Test Set Ib

Test set Ib is the second set of artificial structures provided by Busch & Backofen. It was designed with the same tool as test set Ia for sizes from 300 to 698 nucleotides. More information on the exact features was not provided. [100] lists all 300 structures of this set.

The whole evaluation utilises RNAfold in probability mode. brot is tested with its default parameters as shown in Table 3.1. INFO-RNA and RNAinverse run in both modes, MFE and probability.

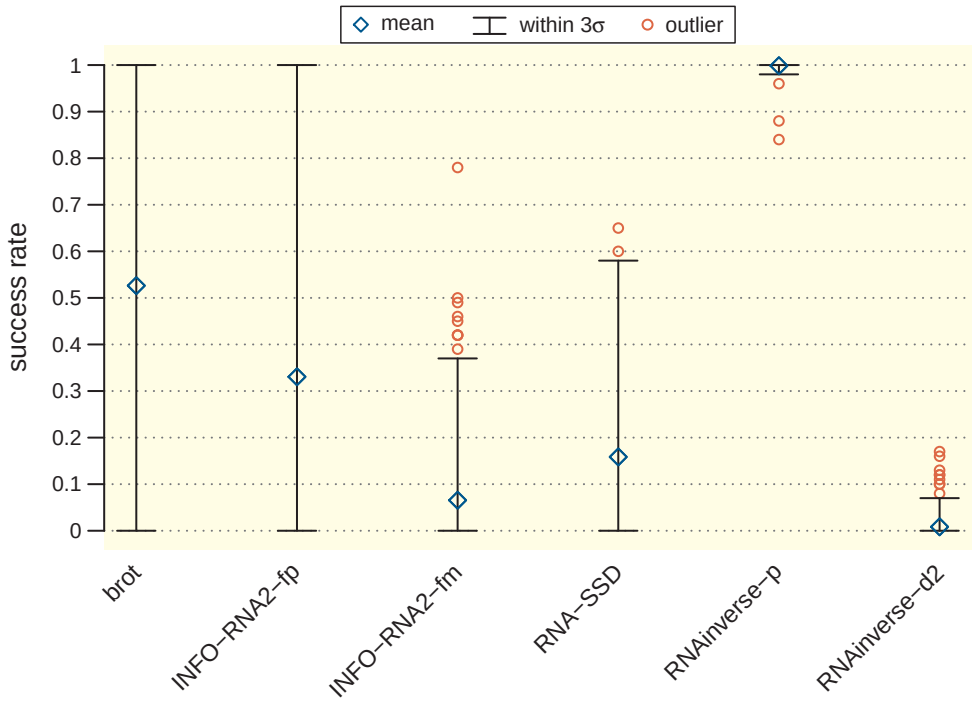


Figure 3.14: *Success rate for test set Ib.* Fractions of sequences which fold into the requested structure. All sequences were tested with `RNAfold` in probability mode. Diamonds mark the average success rate considering all structures of the test set. Whiskers show the spread of individual success rates within 3 standard deviations in both directions. Outliers are marked by red circles. INFO-RNA2-fp shows results for INFO-RNA version 2 in probability mode, INFO-RNA2-fm represents MFE mode. `RNAinverse` was evaluated in probability (-p) and MFE (-d2) mode. For the latter one, the NN model was applied the same way as in probability mode.

The success rates shown in Fig. 3.14 place brot behind RNAinverse in probability mode with an average above 50 %. The results are spread over the full scale with 140 sequences still gaining a success rate bigger than 90 %. INFO-RNA in probability mode has an average success rate of above 30 %, with 78 sequences being right more than 90 % of the time. A drop on averages can be seen for INFO-RNA and RNAinverse running in MFE mode. Both stay below 10 % with RNAinverse being close to 0. The maximum achieved by INFO-RNA is 78 %. RNA-SSD comes with a slightly higher average below 20 % but a maximum value of 65 % for one target. RNAinverse in probability mode gets close to 100 %.

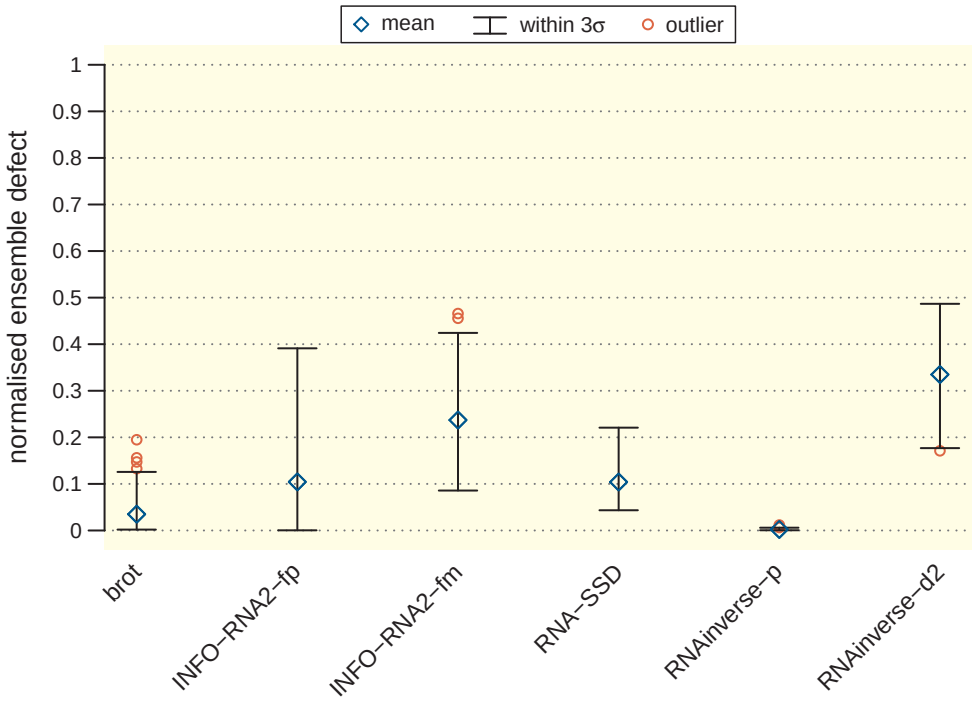


Figure 3.15: *Average normalised ensemble defect for test set Ib.* The average is calculated over all test structures and for all runs of each tool. This includes all designed sequences, regardless if they are corresponding to their input structure. For each structure, ensemble defect is normalised by size. Whiskers and red dots have the same meaning as in Fig. 3.14. Labels for methods have the same meaning as in Fig. 3.14.

The picture drawn by the ensemble distance in Fig. 3.15 follows the success rate. RNAinverse is close to 0 and brot at 0.03 as an average and one outlier at 0.20. INFO-RNA in probability mode is above 0.10 with its average but spreading up to 0.40 within 3 standard deviations. RNA-SSD stays around

0.10 with its average but goes down to 0.02 by its minimum. In MFE mode, INFO-RNA and RNAinverse produce the highest averages.

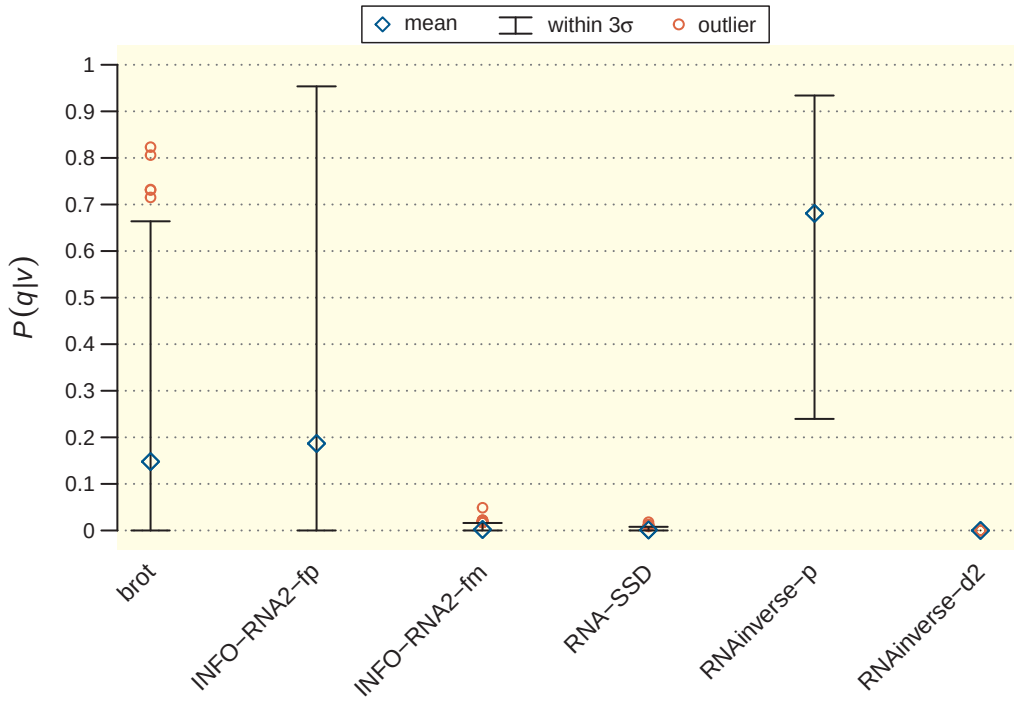


Figure 3.16: *Average folding probability for test set Ib.* Probability of designed sequences to fold into a requested structure. Whiskers and red dots have the same meaning as in Fig. 3.14. Labels for methods have the same meaning as in Fig. 3.14.

The folding probabilities in Fig. 3.16 show brot at 0.15 behind INFO-RNA in probability mode at 0.18 for averages. RNAinverse in probability mode takes the lead at 0.68. The highest outlier of brot is above 0.80 but still below 3 standard deviations of INFO-RNA. With 0.95 this is slightly higher than RNAinverse at 0.93. While brot and INFO-RNA both produce sequences with a folding probability of 0, the lowest probability of RNAinverse is 0.23. The three remaining approaches purely driven by MFE stay close to 0.

Probabilities of true negatives to fold into the desired shape for test set Ib always stay below 0.50. In Fig. 3.17, the folding probabilities of all the MFE based approaches are almost 0. There are some outliers which get some probability but not considerably higher. The average of INFO-RNA in probability mode is also almost 0. However, the highest outlier gets up to about 0.25, while the population within 3 standard deviations ends at a probability of 0.07. brot only looks slightly better with an average of 0.03 and 3 standard

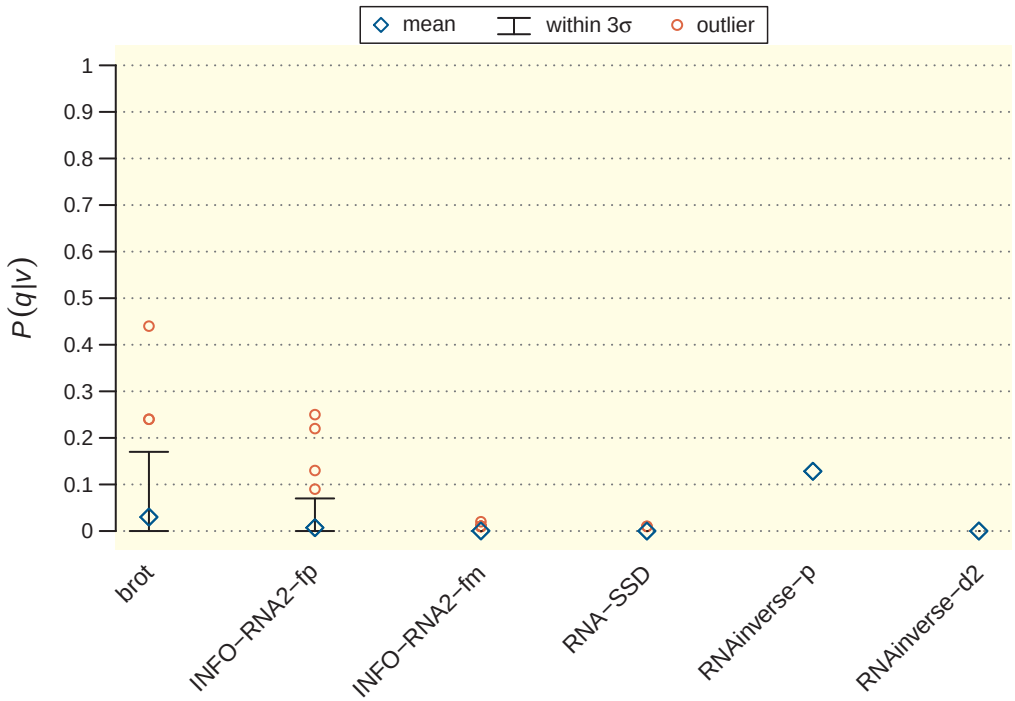


Figure 3.17: *Average folding probability of true negatives for test set Ib.* Probability of sequences to fold into a test structure, predicted by `RNAfold` with a different shape at higher probability. Whiskers and red dots have the same meaning as in Fig. 3.14. Labels for methods have the same meaning as in Fig. 3.14.

deviations up to 0.17. Its outlier at almost 0.45 is the highest value seen here. The few sequences for RNAinverse not predicted with the requested fold, gain an average of 0.12.

RNAinverse delivers a remarkable result for test set Ib. It definitively comes with highest success rates and folding probabilities. But the explanation is rather simple. In probability mode, RNAinverse does not split up the input structure anymore, but tries to optimise the complete sequence in every step. Thereby the design strategy is not complex but rather operating "at random". The initial sequence is filled with random bases and base pairs where needed, mutations are also introduced randomly. Scoring of a new sequence is done with the exactly same method we use to evaluate results. This means, given infinite time, RNAinverse should always find a solution. There are a few missed runs for some targets of this test set, since RNAinverse restricts itself concerning time. After a certain number of steps it will restart the design process and after several restarts, it will stop completely even if no answer was found. The running time is a weak spot, here. In the evaluation by Busch & Backofen, RNAinverse in probability mode was excluded for test set Ib since it "took too long". We found an average running time of 1 hour per target structure and a maximum of 5 hours.

Because of the change in performance between test sets Ia and Ib we had a closer look at how many targets were left without any valid solution here. With 217 unsolved structures, RNAinverse in MFE mode seems to have a general problem. The other two MFE based tools, INFO-RNA and RNA-SSD, seem far better, only missing 94 and 14 targets. But considering the low success rates shows that for the vast majority of runs those two tools also fail. One could say they are finding a valid solution only as an exception. brot is missing 104 and INFO-RNA in probability mode 160 targets. Extending the simulation time for brot would reduce the loss to 66 structures while at the same time the overall success rate goes down below 50 %. To get an idea of what targets the latter two tools fail at, we checked the size of missed structures and their composition concerning structural features. For both tools, no pattern seems to exist. Sequences unable to fold into the desired shape, are produced between 300 to 698 nt. Also all structural features are present in those targets at the same time. Multiloops, marked as "complicated" by Busch & Backofen, are present in those structures, several times with multiple occurrences. But this should not be a general problem. Looking at the solved targets the picture is more than similar. Amongst them, all features are also present, multiloops again with multiple manifestations per structure. The only thing one can see is INFO-RNA not predicting any usable sequence above 420 nt.

Test Set Ic

The last set of artificial structures has 2 members which are closely related to each other and designed to be difficult cases. Structure Ic-1:
`.(((....)))...(((.(((.(((...))).(((...))).(((...))).))).(...).(((...))).))).`
which is described by Busch & Backofen as rather complicated multiloop because of the single base pair towards its 5' end. The second structure Ic-2 fills the small stem with two more base pairs and therefore shifts a hairpin downstream:
`.(((....)))...(((.(((.(((...))).(((...))).(((...))).))).(((...))).(((...))).))).`

The first structure could not be solved by any of the tools within 100 runs. RNAinverse in probability mode at least gets to a very low ensemble defect of 0.08 while brot is in the back of the field with 0.33. Only RNAinverse in MFE mode is worse at 0.46. However, with a folding probability of 0 for every tool this structure does seem to be a hard target. Amongst the true negatives the probability to fold into the desired shape was always 0 for all methods.

brot also found no valid solution, even after manually trying out non-default parameters. Applying slower cooling gives some improvement. One can also put parts of the solution in place an rerun the simulation. To this end, positions 42 and 62 in the sequence where both fixed to `C`, yielding a base pair distance of 1. Obviously this improves folding probabilities and the ensemble defect, but the single pair stem still does not form when predicting a fold. Designing such stems is not impossible for brot since they have been observed when running on various examples.

The results change when looking at the second target structure of test set Ic. This is almost the same as Ic-1 except for the single stem, which is extended to a helix of length three. For this setup, the only tools remaining with a success rate of 0 are brot and RNAinverse in MFE mode. INFO-RNA in probability mode scores a 100 %, MFE mode 64 %, RNA-SSD 53 % and RNAinverse in probability mode goes up to 89 %. Ensemble defect only stays low for RNAinverse, almost unchanged for brot but ranges from 0.11 to 0.50 for the other tools. The folding probabilities only improve slightly. brot still operates at 0, but INFO-RNA with its perfect success rate only goes up to 0.05. RNAinverse takes the lead with 0.11, everybody else stays between 0 and 0.05.

Since all other tools produced some sequences able to fold, one of the results of brot shown in Fig. 3.18 was examined more closely. Gibbs energy of the sequence in the requested structure is $-33.9\,\mathrm{kcal\,mol^{-1}}$ with a folding probability of 0. The predicted fold has an energy of $-41.6\,\mathrm{kcal\,mol^{-1}}$ at a probability of 0.36. When looking at the stems of size three, it stands out that all of them are built by a `GCC - GGC` pattern. This easily increases the populations of alternative folds. A simple idea trying to break the wrong fold

```
CGCCGAAAGGCAACGCCAGCCAGCCACAGGCAGCCACAGGCAGCCACAGGCAGGCAGCCACAGGCAGCCACAGGCAGGCA
.(((....)))...(((.(((.(((...))).(((...))).(((...))).))).(((...))).(((...))).))).
.(((....)))...(((.(((.(((...(((.(((...))).)))...))).(((.(((...))).)))...))).))).
```

Figure 3.18: *Sequence designed for structure Ic-2.* A sequence as designed by `brot` using default parameters. The first Vienna string below the sequence is the requested structure. The second Vienna string is the predicted structure by `RNAfold` in probability mode. Differing pairing sites are highlighted in red.

```
CGCCGAAAGGCAACGCCAGCCAGCCACAGGCAGCCACAGGCACCCACAGGGAGGCAGCCACAGGCACCCACAGGGAGGCA
```

Figure 3.19: *Sequence designed for structure Ic-2 after fixing two sites.* Result of SCMF sequence design after fixing `C` for positions 42 and 66. Positions are highlighted in red.

is to swap some of those stretches into `CGG - CCG`. But this only slightly lowers the energy and does not increase folding probability. At least the prediction of the wrong structure loses folding probability. Inverting too many of the stems again leads to a bigger ensemble of alternative folds. Therefore what is needed are different patterns at some positions in the sequence. As for test set Ia this can be achieved by fixing bases for the simulation and thus making the search space smaller. In terms of SCMF this means lowering the number of base compositions for a structural feature to be chosen from the NN model. For the more complex structure Ic-2, this was done in two positions: a `C` was fixed for position 42 and 66. Fig. 3.19 shows the predicted sequence. This leads to an energy of $-33.10\,\mathrm{kcal\,mol^{-1}}$ and a folding probability of 0.04. The probability seems to be rather low but stays comparable to the 0.05 of INFO-RNA. The problem here is again, SCMF goes for a low energy considering a hairpin loop, while an internal loop of same sequence scores better. This may mean an extension of the negative design term.

Test Set IIa

This set of structures is predicted for ribosomal RNA sequences obtained from the Ribosomal Database Project [102]. It is the same set of sequences as used by Busch & Backofen [59] and Andronescu et al. [60]. Predictions of structures were done with `RNAfold` using probability instead of MFE mode. This affects how dangling ends are treated by the NN model. How exactly `RNAfold` was run in the former evaluations could not be recovered. But in theory, this still leaves an advantage to the other methods excluding SCMF. They apply the same tool to predict the structures for designing sequences. So for them at least one sequence, the biological one, exists that will fold as intended.

The set consists of 24 structures of lengths between 260 and 1475 bases.

They are listed together with identifiers and translated RNA sequences at [100].

Both variants of RNAinverse are excluded for this test set because the running time per sequence was incomparably high.

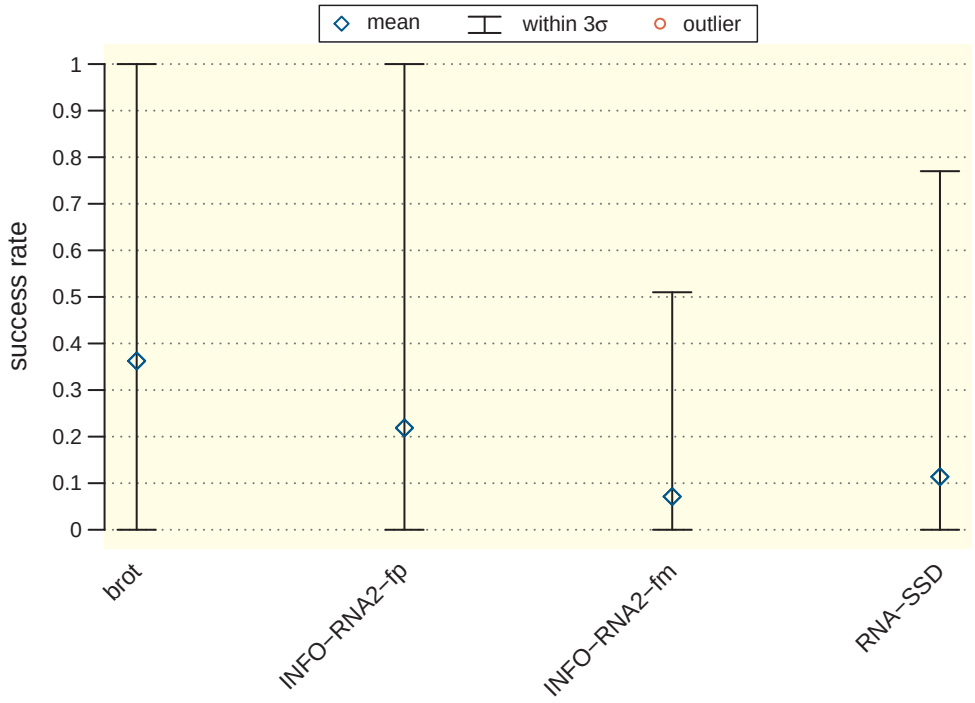


Figure 3.20: *Success rate for test set IIa.* Fractions of sequences which fold into the requested structure. All sequences were tested with RNAfold in probability mode. Diamonds mark the average success rate considering all structures of the test set. Whiskers show the spread of individual success rates within 3 standard deviations in both directions. Outliers are marked by red circles. INFO-RNA2-fp shows results for INFO-RNA version 2 in probability mode, INFO-RNA2-fm represents MFE mode.

The success rates in Fig. 3.20 look different to what is the outcome for the completely artificial test set Ia. While here test structures can not be optimised for a certain tool in this test set, sequences were chosen to present the performance of INFO-RNA and RNA-SSD. brot outperforms both of them with the highest average of almost 40 % or 5 structures with a 100 % success rate. INFO-RNA in probability mode gets above 20 % and 3 structures without any failed sequence. The MFE mode stops below an average of 10 % and does not solve any structure a 100 times successfully. RNA-SSD does achieve more than 10 % but also has not a single 100 % hit. Instead it returned no valid solution

at all for 15 structures. The MFE mode of INFO-RNA loses 18 targets, while the probability mode only slightly better with a count of 17. Again the SCMF method has a better coverage of 50 % of the 24 test structures with at least partially valid answers in 100 tries.

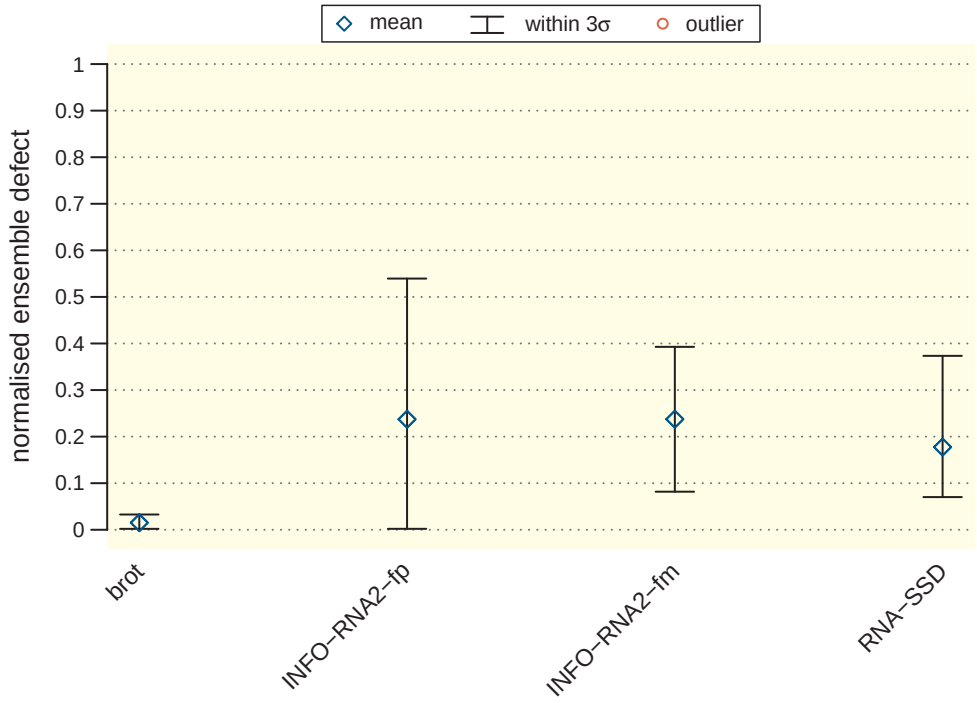


Figure 3.21: *Average normalised ensemble defect for test set IIa.* The average is calculated over all test structures and for all runs of each tool. This includes all designed sequences, regardless if they are corresponding to their input structure. For each structure, ensemble defect is normalised by size. Whiskers and red dots have the same meaning as in Fig. 3.20. Labels for methods have the same meaning as in Fig. 3.20.

For the ensemble distance shown in Fig. 3.21, brot has visibly the lowest average. It stays as low as 0.02, with a very close standard deviation, while both configurations of INFO-RNA operate at 0.23. In probability mode it goes close to 0 by three standard deviations, but at the same time has an upper limit of 0.53. RNA-SSD produces an average ensemble defect of 0.17 without low outliers or extended standard deviation.

The probability plot in Fig. 3.22 does not show any high averages. The lead is taken by the SCMF method again but this also only achieves an average of 0.20. All other tools stay below 0.10. When it comes to three standard deviations, brot and INFO-RNA in probability mode look more promising. They

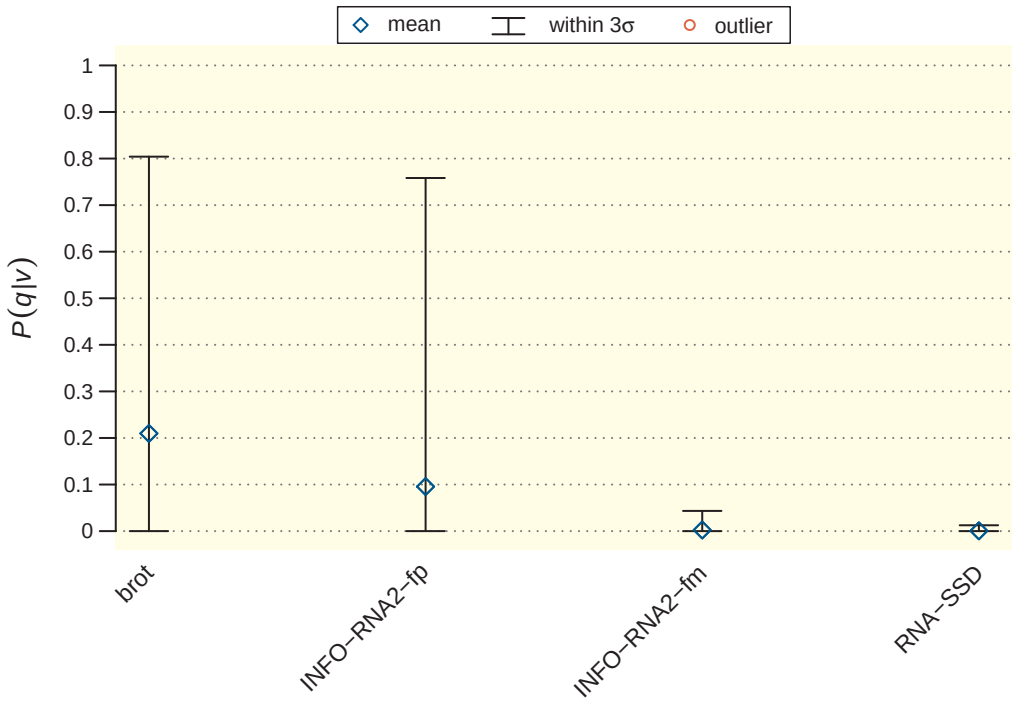


Figure 3.22: *Average folding probability for test set IIa.* Probability of designed sequences to fold into a requested structure. Whiskers and red dots have the same meaning as in Fig. 3.20. Labels for methods have the same meaning as in Fig. 3.20.

spread up to 0.80 and 0.75. But both tools also see probabilities of 0. Averages for the remaining tools stay very close to 0.

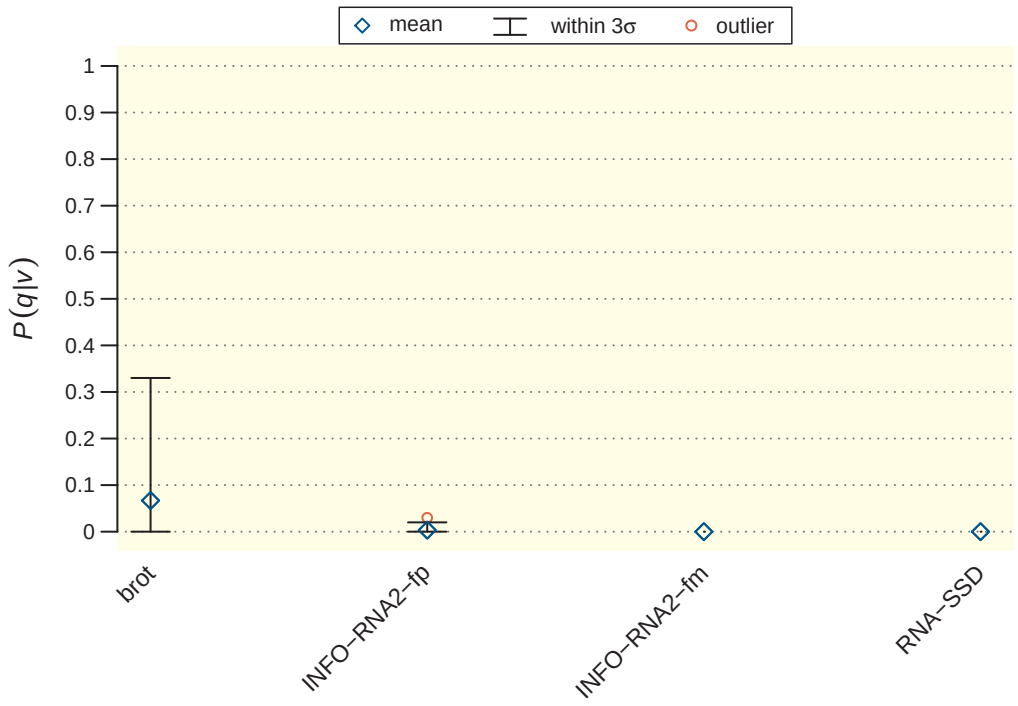


Figure 3.23: *Average folding probability of true negatives for test set IIa.* Probability of sequences to fold into a test structure, predicted by `RNAfold` with a different shape at higher probability. Whiskers and red dots have the same meaning as in Fig. 3.20. Labels for methods have the same meaning as in Fig. 3.20.

Fig. 3.23 shows the probability for true negatives to fold into a target structure. Those are sequences which predict to not fold into the target shape. For all tools but brot, results look like once a predicted fold does not match, sequences would be hard to repair. Methods using the MFE approach get a probability of 0. This could mean that true negative sequences are not able to form base pairs between the right positions. INFO-RNA in probability mode is able to produce probabilities, but a value of 0.02 is already a high outlier. The average stays close to 0. The average of brot is 0.06. While this is still low, by three standard deviations a probability of 0.33 is achieved.

After exploring all the tested values, set IIa seems to be especially hard when compared to former ones. Here, MFE based methods seem to not work well anymore. Their probabilities go down, literally to zero, while the ensemble defect goes up in comparison to former tests. INFO-RNA run in probab-

ility mode does perform slightly better but still stays behind brot. One reason for the change in performance could be the overall occurrence of the structures. While the artificial structures from set Ia and many in set Ib look rather linear, the ones predicted for biological sequences seem to have more multiloops. This may interfere with the splitting strategies of other methods than SCMF, which operates on the full structure.

Test Set IIb

This is the second set of biological sequences with predicted structures from Busch & Backofen [59]. The set comes again from the Ribosomal Database Project [102] and covers 308 annotated eukaryotic rRNA sequences. The lengths vary from 220 to 1975 bases. The prediction procedure is the same as for test set IIa. Sequences and structures are listed at [100].

We only evaluated the SCMF method with this test set. This decision was made after considering running times. Following test runs, a rough time estimate for brot to calculate 100 sequences for the 308 structures was 2000 CPU hours. While brot runs only a couple of minutes per structure, it seemed like the other tools need about an hour or more for the same task. This means 30 800 CPU hours per tools or 92 400 hours combined. Finally the other tools were dropped because of lack of resources.

Since there is only data for one tool, all plots are presented in Fig. 3.24 in a compact way.

Fig. 3.24(a) shows the success rate of brot for test set IIb. The average is slightly above 30 % and the three standard deviations span the full scale. That means that producing 100 sequences predicted into the target fold is not an outlier. But also structures exist without a valid solution. In total, 92 targets are covered a 100 times by a qualified sequence, while in 205 cases success rate was 0.

That sequences for these 205 targets may not be completely bad is the outcome of Fig. 3.24(b). The average ensemble defect is 0.02 with three standard deviations going up to 0.05. Above this, there is only a single outlier but staying below 0.10. This means that only a few base pairs seem to mismatch throughout most of the ensembles of all sequences.

Folding probabilities in Fig. 3.24(c) seem to be low for this test set. The three standard deviations do not even reach 0.5 and the average stays below 0.10. Going to more decent values is always an outlier.

But looking at the values only computed for sequences predicted to not fold into the target shape, could be an explanation. Fig. 3.24(d) covers 205 targets without any solution. Just numerically this can pull the scores down when averaging over all sequences. Especially with the low values found here.

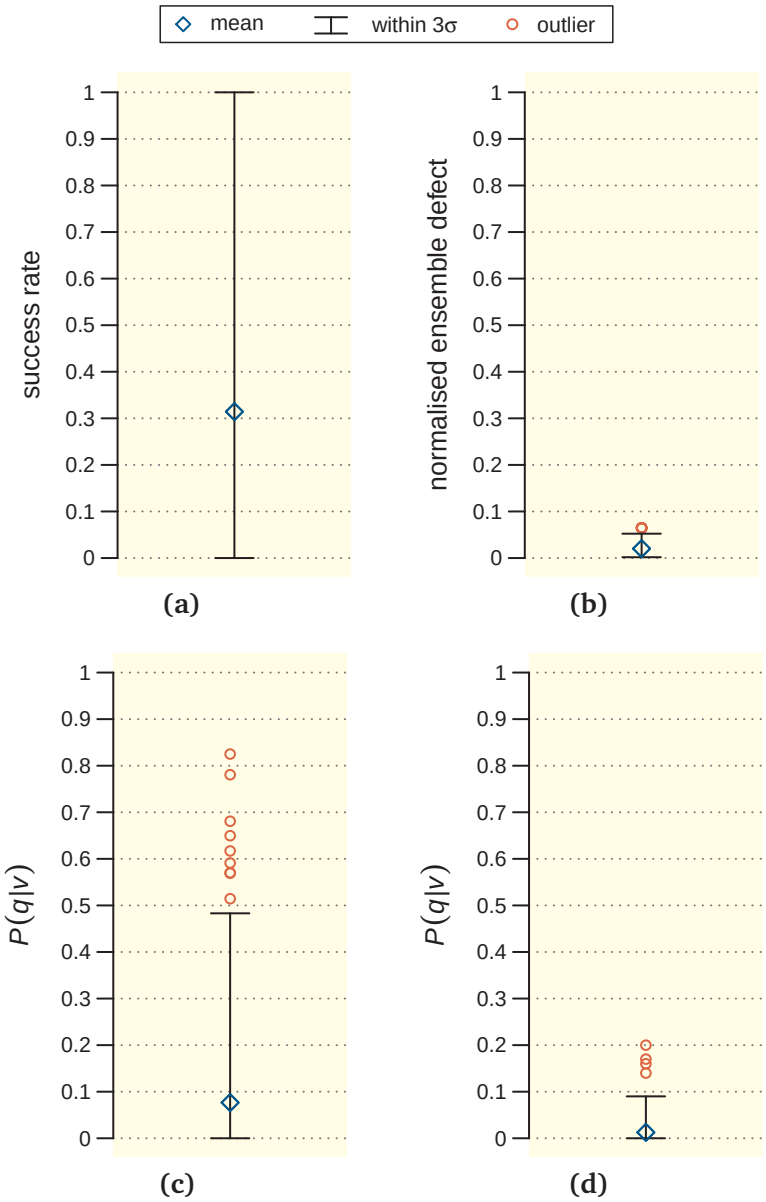


Figure 3.24: *Combined results for test set IIb.* Results of the SCMF method. (a) Average success rate, fractions of sequences predicted by `RNAfold` in probability mode to fold into a requested structure. (b) Average normalised ensemble defect, calculated for all sequences designed for this test set without considering if they predict to fold as requested. Results are normalised by structure size. (c) Average probability of sequences to fold assuming target structures. (d) Average probability of sequences with a different shape than the target one being more probable. For every plot, diamonds mark the mean, whiskers show the spread of three standard deviations and outliers are highlighted as red circles.

As test set IIa, redesigning biological sequences may seem harder than artificial structures. But almost one third of the set was successfully covered without manual adjustment of parameters. For the missing targets, the ensemble defect, accumulated over the whole set, seems to be low enough that not much is needed to fix. When one thinks of a more relevant scenario of designing a single sequence, rather than 308, this is an important result of this evaluation.

Sequence Compositions

This section of the evaluation focuses on how predicted sequences look like. Fig. 3.25 counts how many different sequences are produced for a structure and Fig. 3.26 measures their GC content. For both plots, all sequences able to form base pairs as occur in the corresponding target structure are used. Sequences which are unable to fold are excluded since those are obviously wrong answers. Structure prediction is not considered as an external filter here. Results are based on all test sets since the values do not change much when investigated per set.

Creating a different sequence every time brot is run, is only an outlier in Fig. 3.25. While for hard targets it seems unlikely to produce a completely new and still foldable sequence on every try, outliers go on down to 80. Below begins the range of three standard deviations, including targets where only a single sequence is predicted. On average, the SCMF method creates almost 20 different sequences per structure. This means running brot six times should produce two different sequences.

The other tools all have higher averages, around 70, RNAinverse even above 90. INFO-RNA and RNA-SSD spread within the full range by three standard deviations. This means they also produce single answers for some targets. RNAinverse in both modes only has outliers, spreading around the average of brot.

The GC contents shown in Fig. 3.26 on average stay between 0.50 and 0.80. The minimum is achieved by RNAinverse, the maximum by INFO-RNA, both tools run in MFE mode. INFO-RNA exceeds its own GC content of 0.80 by three standard deviations and outliers above 0.90. All other tools always stay below 0.85. RNA-SSD provides an outlier with the minimum GC content of 0.41.

brot has an average of 0.74 but three standard deviations include a lower border of 0.64. By outliers it even gets down to 0.55.

In terms of number of different sequences produced per target, brot seems to have room for improvement. Considering results for the artificial test sets, it does not seem to be harmful to produce more unique sequences. For the GC

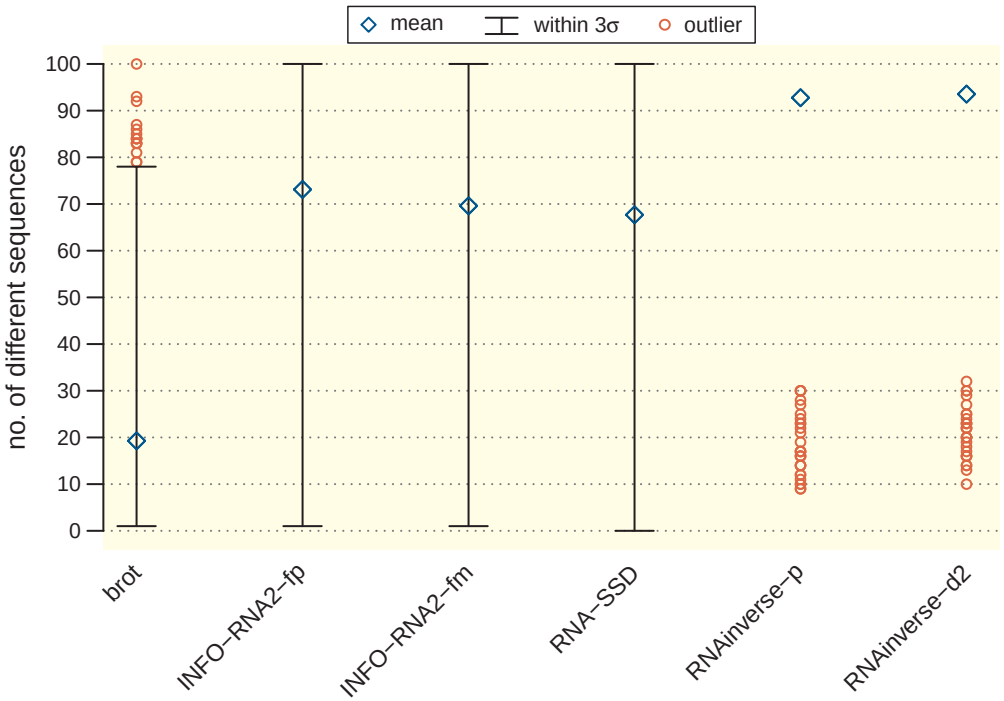


Figure 3.25: *Average no. of unique sequences predicted per target.* Only sequences which are compatible with the desired fold concerning Watson-Crick and GU pairs are counted. Data is used over all test sets. Diamonds mark the averages. Whiskers show the spread within 3 standard deviations in both directions. Outliers are marked by red circles. INFO-RNA2-fp shows results for INFO-RNA version 2 in probability mode, INFO-RNA2-fm represents MFE mode. `RNAinverse` was evaluated in probability (`-p`) and MFE (`-d2`) mode. For the latter one, the NN model was applied the same way as in probability mode.

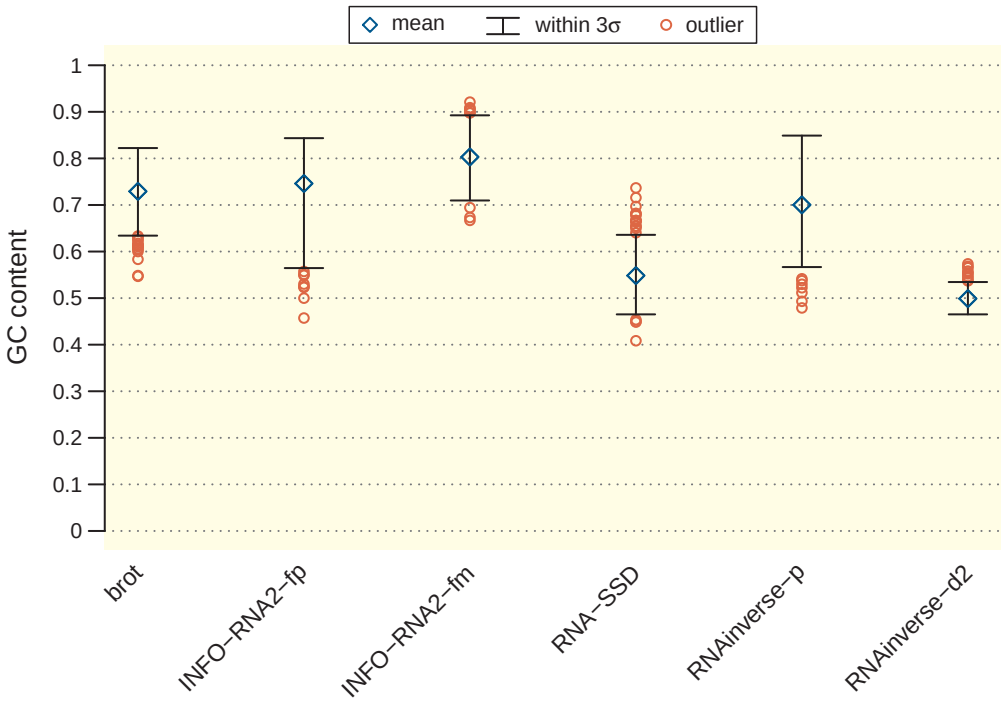


Figure 3.26: *Average GC content of designed sequences.* The share of GC nucleotides is calculated for each tool, only incorporating sequences which are compatible with the desired fold concerning Watson-Crick and GU pairs. Also redundancy is removed from sets of predicted sequences. Sequences are collected over all test sets. Whiskers and red dots have the same meaning as in Fig. 3.25. Labels for methods have the same meaning as in Fig. 3.25.

content all tools seem to be rather close together, while behaviour like RNA-SSD would be preferable if not disturbing performance.

Sequence Identities

When the methods compared produce different sequences for the same target, it is interesting to see how similar they still are. The same holds for sequences designed by different methods for the same target.

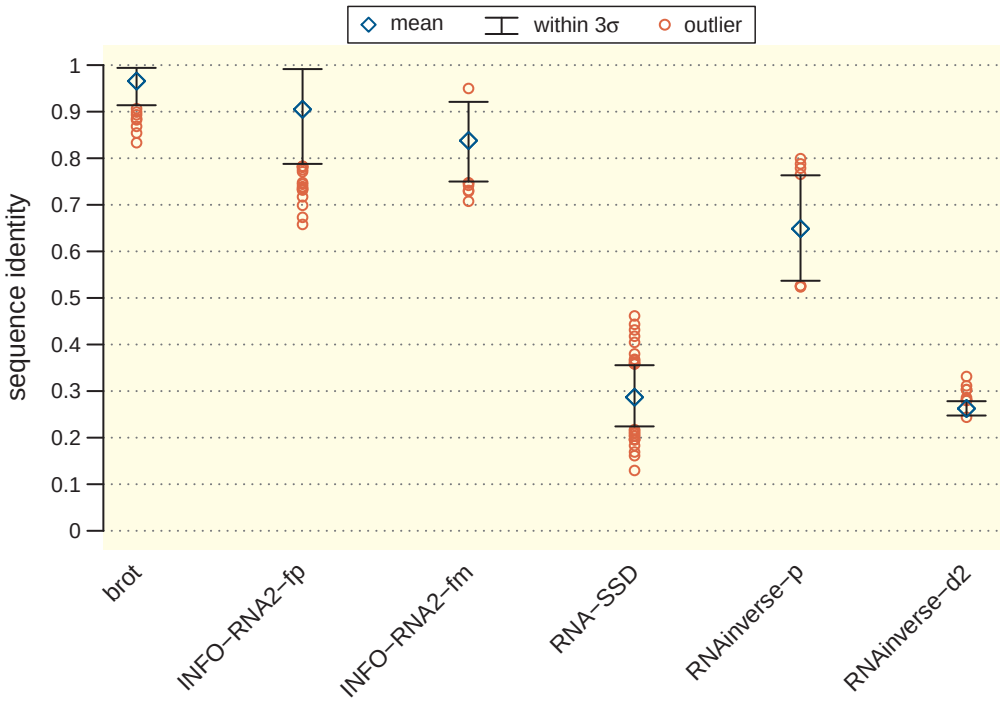


Figure 3.27: *Average identity of sequences predicted for a target.* Sequence identity is only measured between different sequences for the same test structure. Furthermore, only unique sequences are compared. Data is used over all test sets. Diamonds mark the averages. Whiskers show the spread within 3 standard deviations in both directions. Outliers are marked by red circles. INFO-RNA2-fp shows results for INFO-RNA version 2 in probability mode, INFO-RNA2-fm represents MFE mode. `RNAinverse` was evaluated in probability (-p) and MFE (-d2) mode. For the latter one, the NN model was applied the same way as in probability mode.

First, Fig. 3.27 shows identities per tool. That is, average similarities for sequences of the same target by the same tool. An average of above 0.95 for brot means that only a few bases get exchanged in two sequences for a

structure. Three standard deviations reach 0.99. By outliers, not even a value below 0.80 is populated.

In contrast RNAinverse in MFE mode has an average below 0.30. Outliers are not found above 0.35. Only RNA-SSD goes to a lower 0.12 by an outlier. The average is 0.39 and high outliers occur above 0.55. In MFE mode the average of INFO-RNA is 0.85 and one outlier is seen at 0.95. With low outliers, sequence identity goes down to almost 0.70. In probability mode, INFO-RNA and RNAinverse look worse than their MFE pendants. RNAinverse has an average identity of 0.65 and populates the area between 0.50 to 0.80 by three standard deviations and outliers. INFO-RNA has an average of 0.91, going up to 0.98 by three standard deviations.

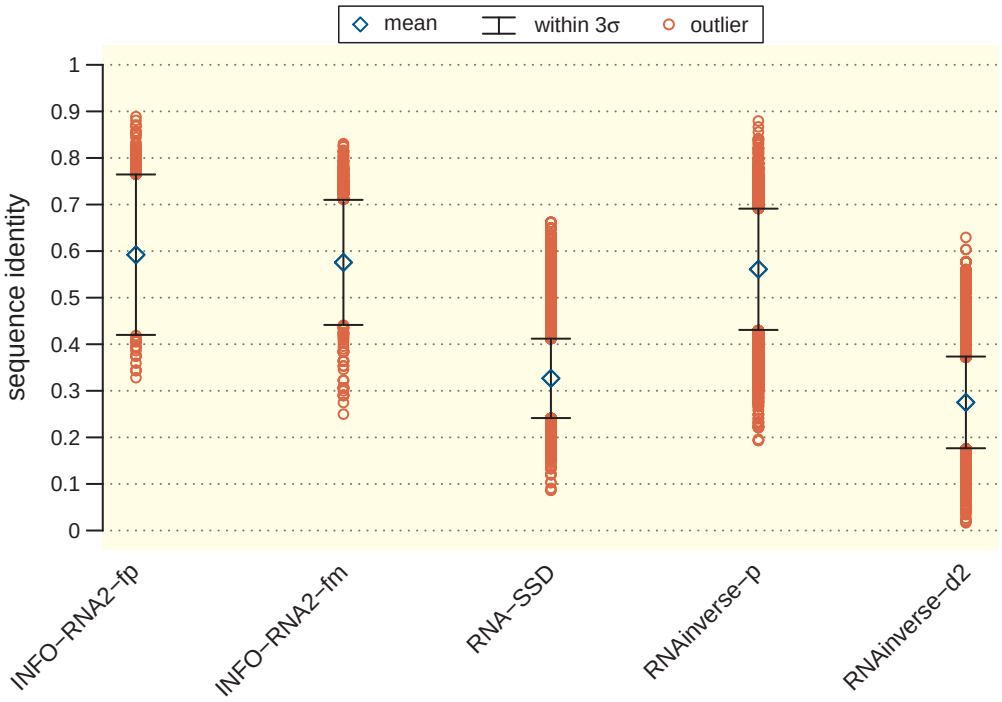


Figure 3.28: *Average sequence identity between various tools and the SCMF method.* Sequence identity is measured between sequences, designed for the same structure, of brot and the tools compared here. Only unique sequences of each tool are considered. Whiskers and red dots have the same meaning as in Fig. 3.27. Labels for methods have the same meaning as in Fig. 3.27.

In Fig. 3.28 sequences from the other tools are compared to the SCMF method. Only unique sequences are used which are compatible with a target. To match with brot, only sequences for the same target are used. That is, we compare unique sequences for the same target of a different method and our

SCMF method.

The most different sequences are produced by RNAinverse in MFE mode. By outliers, it gets close to 0 which would mean a different base in almost every position. The average identity is 0.28. Outliers exist with values up to 0.63. Highest values are found for INFO-RNA in probability mode. The average is close to 0.60, its maximum by an outlier close to 0.90. The MFE mode looks similar with low outliers at 0.25. The minimum for RNA-SSD is below 0.10 and the average at 0.32. RNAinverse in probability mode looks similar to INFO-RNA in MFE mode with more extreme outliers.

Running Times

Since all tools but RNAinverse return a sequence within a decent amount of time, all available data will be used for a quick overview. This means that for brot, also running times are included for test sets IIa and IIb, which did not cover all tools. INFO-RNA and RNA-SSD have measurements for test set IIa, while RNAinverse only has data for sets Ia and Ib.

The averaged running times per run in Fig. 3.29 show the SCMF method ahead of the others. While there are more low average values, three standard deviations plus the outliers are considerably lower than for any other tool. For brot it is already an outlier if it runs for 15 minutes. In the best case it returns instantly. With an average of 46 seconds, it is still more than twice as fast as RNAinverse in MFE mode with 110 seconds.

INFO-RNA in MFE mode has an average of 233 seconds but as outliers already sees runs taking longer than an hour, even without test set IIb. In probability mode, the average raises to 2362 seconds and times of over an hour are included in three standard deviations. RNA-SSD does better in terms of average with 412 seconds and a longest time of 37 minutes. The big outlier in the field is RNAinverse in probability mode. While in MFE mode, the highest value is just a bit more than an hour, the maximum running time in the other mode is close to 5.5 hours. By three standard deviations, a time between 2.5 hours and almost 0 seconds is covered. The average is 1728 seconds. Included for all tools within three standard deviations are cases where the result is delivered almost immediately.

Discussion

In a quick summary of the performance results, brot is not outstandingly better than the other tools on the artificial test sets Ia, Ib and Ic. But also it is not left behind, on set Ib it is only outmatched by RNAinverse. In probability mode, this tool is not really doing sequence design but sampling. Without splitting,

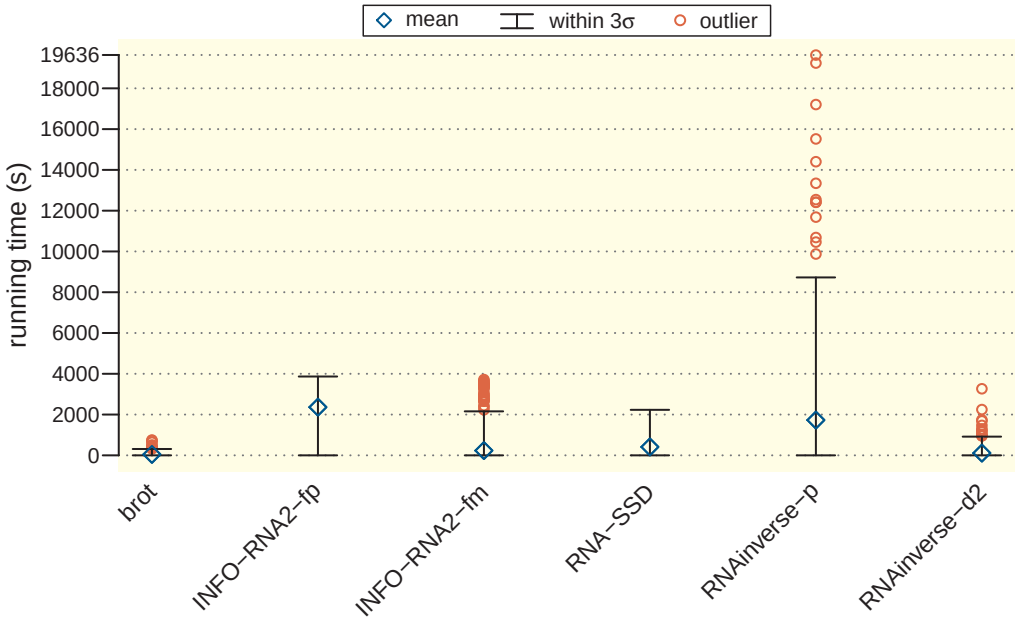


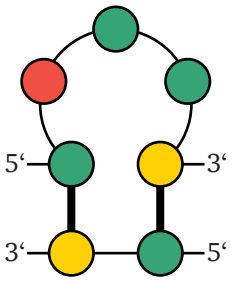
Figure 3.29: *Average running times.* For each tool, running times are shown as average of a 100 runs per sequence. For each tool, all test sets it was evaluated for are used. Diamonds mark the averages. Whiskers show the spread within 3 standard deviations in both directions. Outliers are marked by red circles. INFO-RNA2-fp shows results for INFO-RNA version 2 in probability mode, INFO-RNA2-fm represents MFE mode. `RNAinverse` was evaluated in probability (-p) and MFE (-d2) mode. For the latter one, the NN model was applied the same way as in probability mode.

it exchanges nucleotides in the sequence on mismatching base pairs, which basically means enumerating sequences and testing if they fold as intended. By the running times it is clear that enumerating only works for small sequences. INFO-RNA works well in probability mode, especially on the smaller structures of set Ia while for set Ib it falls behind brot. The same holds for the MFE mode. Only for set Ia it outperforms the other tools, on other sets it falls behind. RNA-SSD looks like it is rivalling INFO-RNA. Being good on the first test set, afterwards it ranks last against INFO-RNA in MFE mode. RNAinverse in MFE mode looks very different to its probability mode. It never performs better than any of the other tools even when treating the NN model the same way as RNAfold in probability mode. Also it is only fast enough for the artificial test sets. When it gets to the bigger and more complex structures based on biological sequences, it gets too slow to compete with the others.

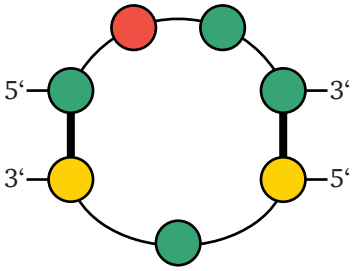
For test set Ic brot and RNAinverse in MFE mode are the only ones to find no answer. Since the second structure of the set is marked as "complicated multiloop", RNAinverse may fail because the splitting strategy could be inadequate. While designing the single branches of the loop, possible cross-interactions are not considered and may prevent joined subsequences from folding correctly. INFO-RNA and RNA-SSD seem to have the better strategy assembling the final sequence.

On test set IIa, based on biological sequences, all tools perform worse than brot. These structures were not designed around INFO-RNA or any other tool, those are predictions on given sequences. While the artificial structures look quite linear, many of the predicted ones seem to have more complicated shapes featuring multiple multiloops. This could be a scenario to complex for the more advanced splitting strategies behind INFO-RNA and RNA-SSD. With every multiloop, the possibilities for unforeseen interactions by a subsequences increase. Especially at the point of joining two multiloop-sequences, wrong base pairs can easily disturb the overall shape of the structure. Since branching loops are joined in later steps of the procedure, such an event would mean throwing away a lot of already made steps. Also if a tool needs to redesign a subsequence, nothing prevents it from running into the same problem again, so the whole process gets infeasible. INFO-RNA tries to prevent bad moves by ranking possible steps ahead. But a sequence for a whole multiloop without knowledge about other components of the structure could already be a local minimum to deep to get out that easily. It just takes more steps to assemble such a loop than INFO-RNA looks ahead.

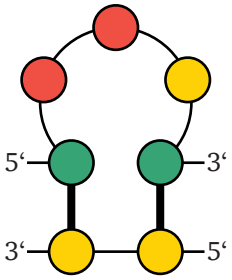
Splitting an input structure into smaller parts may prevent finding a complete solution. But from ensemble distances it also seems like INFO-RNA, RNAinverse and RNA-SSD do not try to find partially good answers. Whenever success rates are not near perfect, the ensemble defect is worse than for brot.



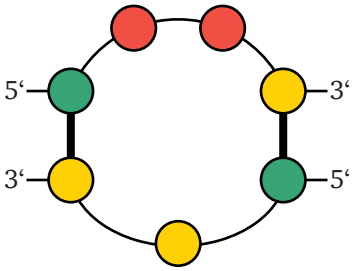
(a) $\Delta G = 7.3\,\text{kcal}\,\text{mol}^{-1}$



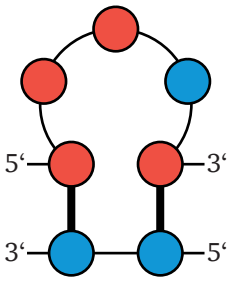
(b) $\Delta G = 4.9\,\text{kcal}\,\text{mol}^{-1}$



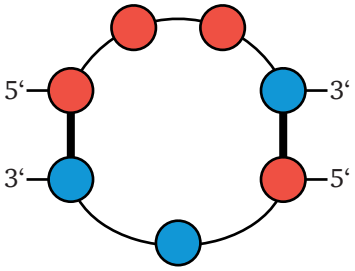
(c) $\Delta G = 7.3\,\text{kcal}\,\text{mol}^{-1}$



(d) $\Delta G = 6.4\,\text{kcal}\,\text{mol}^{-1}$



(e) $\Delta G = 9.3\,\text{kcal}\,\text{mol}^{-1}$



(f) $\Delta G = 10.6\,\text{kcal}\,\text{mol}^{-1}$

Figure 3.30: *Bulge & internal loop energies.* Comparison of Gibbs energies for bulge and internal loops populated with the same bases. (a) shows bases `GAGGC...GC` as bulge loop and (b) `GAGG...CGC` as internal loop. (c) shows bases `GAACG...CC`, (d) `GAAC...GCC`. (e) shows bases `AAAUA...UU`, (f) `AAAU...AUU`. While for (a) to (b) the internal loop always gets the lower energy, for (e) and (f) the bulge loop is more stable. Dots in the sequences indicate the 3' to 5' strand break in the figure. Thick black lines indicate base pairs. Energies were calculated using `RNAeval` of the Vienna RNA package [15].

The trial and error approaches do not have a good sense for the quality of an answer. Either a sequence does fold or a modification step is rejected until some stop criterion. There is no means in choosing a second best solution, e. g. optimise folding probabilities if the overall fold is not right. This would keep the ensemble defect low while instead leaving mismatching sites alone drives it up.

Fig. 3.30 delivers an example what may be done better. It shows bulge and internal loops, paired by equal base populations and the Gibbs energies from the NN model as calculated by `RNAeval` [15]. For bulge loops with three unpaired nucleotides as in Fig. 3.30(a) and Fig. 3.30(c), 7.3 kcal mol^{-1} is the minimum. By the same bases, 2×1 internal loops in Fig. 3.30(b) and Fig. 3.30(d) are formed at a lower energy. If we assume the design task to be a bulge loop of size three, Fig. 3.30(a) and Fig. 3.30(c) are energetically favourable choices. With Fig. 3.30(b) and Fig. 3.30(d) being loops of lower energy, these will be preferred by a folding algorithm discarding the choice of nucleotides by trial and error methods. If we ignore loops in Fig. 3.30(e) and Fig. 3.30(f), no perfect solution in finding the bulge loop may exist. In this case, INFO-RNA, RNA-SSD and `RNAinverse` will stop trying at some point leaving the base population in the state of the last try. What setup this is exactly, is undetermined. It only depends on the order how sequence candidates are iterated rather than some further optimisation criterion. If a tool starts with the best possible choice, the last step may explore the worst possibility, affecting folding probability. What would be a good strategy in this particular scenario, is minimising the difference in energy between the bulge and internal loop. Fig. 3.30(c) and Fig. 3.30(d) have a lower energy threshold to overcome than Fig. 3.30(a) and Fig. 3.30(b). This makes forming a bulge loop at the particular position more probable while not completely solving the issue. At least this would lower the ensemble distance.

Fig. 3.30 also illustrates the problem of the SCMF method with the manually fixed sequences of test sets Ia and Ic. In both cases a sequence pattern is selected for a certain structural motif which has a lower energy when folded into a different conformation. When designing a bulge loop, this is the situation in Fig. 3.30(a) to Fig. 3.30(d). By Fig. 3.30(e) and Fig. 3.30(f) a solution to the problem is provided. There the bulge loop has the lower energy making it more favourable when folding than the corresponding internal loop. While the force field of SCMF favours the lower energy bulge loops from Fig. 3.30(a) and Fig. 3.30(c), the negative design term should fix the situation. This does not happen since this term only prevents stacked base pairs in wrong positions. It does not have the knowledge about any other motif. For the bulge loops all positions are tested to not interact with wrong partners but any contribution assuming an internal loop is not considered. That way the system converges

to a state enabling the lowest energy bulge loop. Since having less guanines in unpaired positions should be preferable to prevent false pairs, SCMF could converge to the setup in Fig. 3.30(c). This still will not fold as intended but it does lower the ensemble defect. A solution for this issue may be an extension of the negative design term. For internal loops, they could be added as second motif to be tested, making the computation of the term more complex. Given the current speed brot operates at, this should be no problem. Considering that the NN model may have more such cases for other structural features, this could be not sufficient. A different solution would be to add all the possible bonuses for certain sequence patterns existing in the NN model to the current negative design term.

In general it seems like that all methods lose performance on larger and more complex structures. For the other methods the main reason seems to be splitting up the input structures. For SCMF some missed targets could be fixed by adapting parameters, enabling more conservative cooling schemes and longer simulation times. This is a hint that the default parameters are not optimal. Probably the structures used for the simplex optimisation are too simple. Creating a new mix of easy and hard targets to optimise parameters on should change the defaults to work better with the difficult cases.

One topic only covered by our method and partially by RNA-SSD is enforcing some sequence composition. RNA-SSD provides the possibility to enforce a certain GC level, we have implemented the heterogeneity term in SCMF. Being an artificial term, not targeting sequence quality directly, the most important note is that it does not have a large effect on target stability. For test set Ia results are very close having the heterogeneity term enabled or disabled. However, it does for sure change the energy landscape and the path the system takes on it. This includes visiting different minima while the term is parameterised to still find a sequence for which the target structure lies in the middle of a set of near-optimal similar structures. But scaling the heterogeneity factor to high can still affect finding good solutions. We would also concede that part of the benefit of this term is aesthetic. Predicted sequences do have less repetitive regions which look extremely unlike biological sequences.

3.3.4 Case Studies

This section leaves the comparisons of the SCMF method behind and goes for scenarios of special interest. Here we use structures with experimental evidence instead of artificial setups. First, a ribosome is investigated as the largest RNA structure one can find. The initial idea was mere curiosity if brot can handle such a large system. After this, we have prepared structures with features the other methods can not handle, namely pseudoknots. As a proof of

concept, a sequence for a ribonuclease is discussed. The last example tests the workflow from design to synthesis and experimental validation by collaborators [103]. This included the design itself, validation *in silico*, synthesis and the analysis of the base pairing pattern of the molecule *in vitro*.

Ribosome

As largest structure, the 23 S rRNA of the 50 S ribosomal subunit of *E. coli* [39] is used as an input to the SCMF method. This corresponds to the PDB entry 1C2W used in the test set to evaluate the H-bond recognition method in Chapter 2. An overview of the secondary structure is presented in Fig. 3.31, taken as is from Mueller et al.. The list of base pairs in a format that brot understands, is provided at [100]. The point of interest in an RNA with 2904 nt, are the vast possibilities of cross-talk between sites, making this a hard target.

Before finding a good solution, a first sequence was predicted using the default parameters listed in Table 3.1. After around 2.6 days, this produced an answer incompatible with the input structure. Some positions were populated with bases which can not form canonical base pairs as required by the 23 S rRNA. Since the running time is to long for just testing multiple parameter sets, the first step to improve the result was to improve the implementation of SCMF, brot, itself. Earlier versions of brot always consider the full system in every step of the simulation. That is, even already fixed sites will still be evaluated while the outcome stays the same, a probability of 1 does not change anymore. For smaller examples the influence of these extra calculations seemed to small to be considered as serious overhead. In contrast the ribosome calculates slow having 2904 nt and all together 808 structural features. Since the number of steps is not to be cut, the time spent per step needs to be reduced for a speed-up. This has been done for the final version of brot as described in §3.2.5: fixed sites are not evaluated anymore, only considered for contributions, and structural motifs are excluded from the simulation once completely solved. This reduces the running time with default parameters to 12 minutes.

With the faster implementation, parameters could be tweaked to produce a compatible sequence. For this large structure, a longer equilibration phase at the beginning of the simulation was set by raising the start temperature to 3. Used with default parameters, the emitted sequence scored a base pair distance of 101 when predicted into the most probable structure. The corresponding annealing plot showed a drop by more than half of the temperature right after the equilibration phase. Entropy, decreasing slowly until this incident, jumped down 20 % afterwards. After an almost constant phase, the simu-

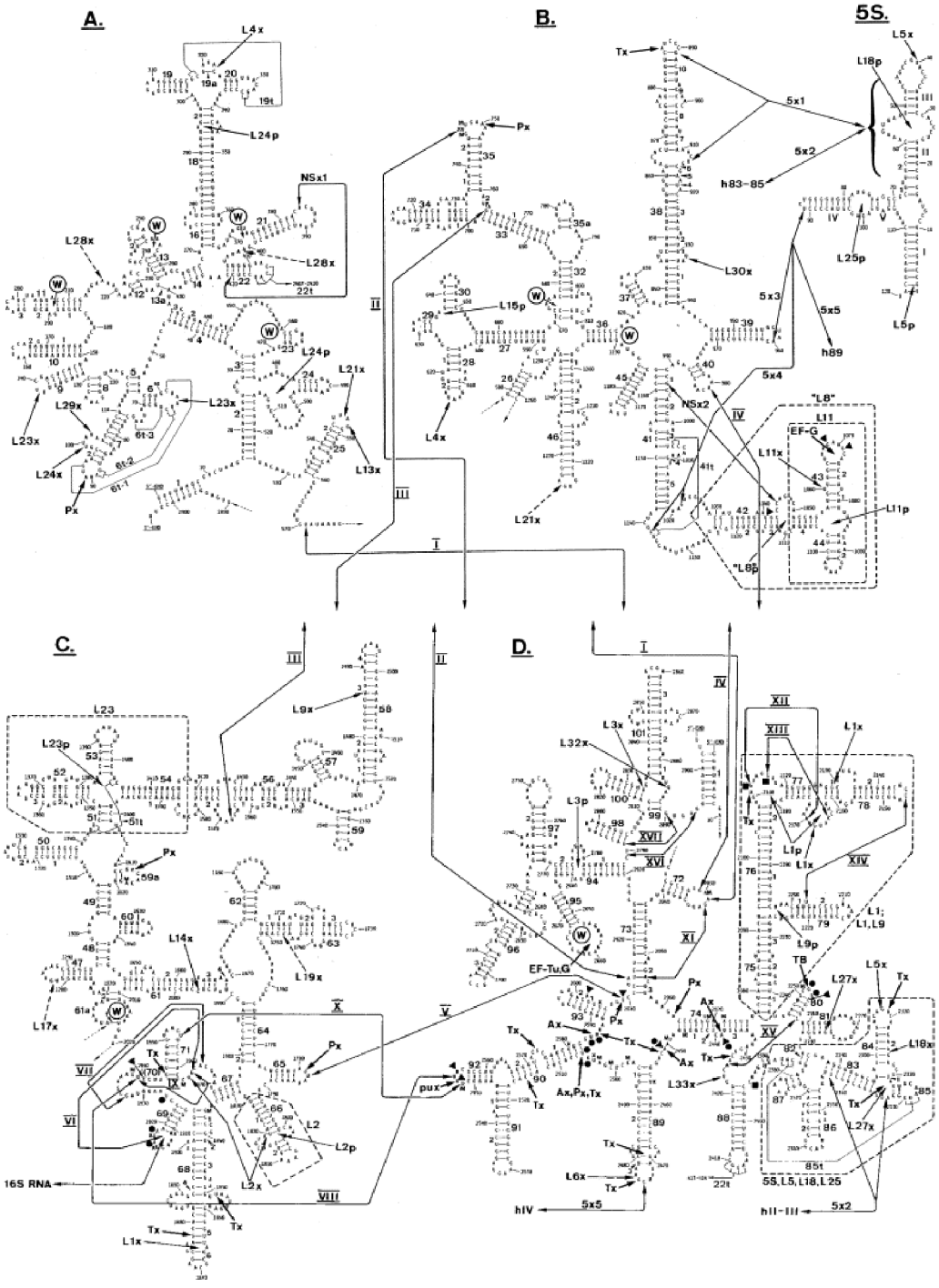


Figure 3.31: *2D structures of 23 S and 5 S rRNA.* Fig. 1 of Mueller et al. [39] shows the 23 S and 5 S rRNA of the 50 S ribosomal subunit of *E. coli* (PDB entry 1C2W) split into four regions. The 5 S structure is shown in the upper right corner. The Vienna string corresponding to the 23 S rRNA is listed at [100].

lation ended by another temperature jump. Certainly, fixing the system in two big steps makes hitting local optima rather likely. Therefore parameters were fitted into producing an almost linear curve for the sequence entropy decrease, shown in Fig. 3.32. Different to default parameters are the start temperature $T = 4$, the maximal cooling rate $c_{\min} = 0.97$, the long term entropy ratio $\beta_{\text{long}} = 0.80$ and the cooling threshold $s_c = 0.85$. The whole simulation needs 89 steps until the entropy drops below the convergence threshold. During this time, it decreases slowly, almost linear. After the equilibration phase of 25 steps, there is no huge speed up following the faster cooling. Towards the end, entropy and temperature slow down slightly instead of "jumping" into the final state. This looks like behaviour to be expected: in the beginning of the simulation more undecided sites are available to cause bigger entropy changes. Once most positions of the sequence are fixed, the impact of the remaining ones on the entropy can only be smaller than before. Therefore slowing down towards the end should mean that the system is almost completely populated with single bases in each site.

In terms of quality of the designed sequence, it is able to form canonical base pairs where the target requires them. There is still a base pair distance for the predicted structure but going down to 87 for a total of 808 base pairs in the rRNA. Considering the ensemble distance as more appropriate, it is 148.48 in total and 0.05 normalised. With the vast ensemble of possible structures a sequence with 2904 nt can produce, this does not seem to be a bad result. For this low defect, the majority of base pairs will exist in many conformations. Since folding probabilities get to small to be compared with such big ensembles, energies have to serve here. The most probable structure for our sequence has a Gibbs energy of $-1840.51\,\text{kcal}\,\text{mol}^{-1}$. Assuming the target structure for our design the energy is $-1813.09\,\text{kcal}\,\text{mol}^{-1}$. This does not look like a big difference but those $27.42\,\text{kcal}\,\text{mol}^{-1}$ cover a range of too many structures to be computationally enumerated in decent time. The native sequence has an energy of $-962.39\,\text{kcal}\,\text{mol}^{-1}$ in the target conformation. Predicting a fold for the rRNA sequence, an energy of $-1173.71\,\text{kcal}\,\text{mol}^{-1}$ is reached. Obviously the most probable and the real structure are not identical and differ at a base pair distance of 862. The corresponding ensemble defect is 1254.04, 0.43 normalised. The similarity between the designed and the native sequence is 47.7 %.

One could argue that our result looks good, while not perfect, since the difference in energy between the predicted and the target structure is orders of magnitude smaller compared to the native sequence. Also our low ensemble defect indicates that our ensemble has the base pairs more conserved than the native setup. A conclusion may be that the SCMF approach can give a promising starting point for real physical exploration of designing RNA sequences.

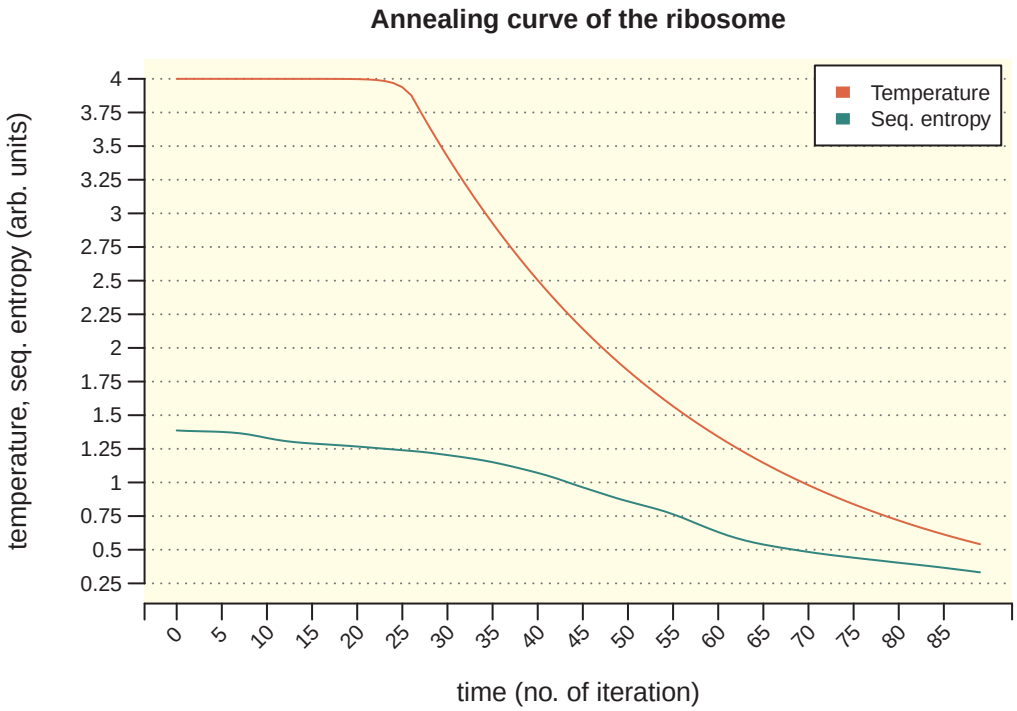


Figure 3.32: *Annealing of the 23 S rRNA of E. coli.* Temperature and entropy changes during time, predicting a sequence for ribosomal RNA using SCMF. Instead of the default parameters of brot, this simulation was run with start temperature $T = 4$, maximal cooling rate $c_{\min} = 0.97$, long term entropy memory rate $\beta_{\text{long}} = 0.80$ and the threshold to speed up or slow down cooling $s_c = 0.85$.

Pseudoknots

As working example of a pseudoknot, the structure of a ribonuclease P RNA by Kazantsev et al. [104] was used. With 417 nt this structure seems of relevant size and provides all structural features but internal loops. The pseudoknot to be included in the design is marked as helix P4 in Fig. 3.33. The structure is listed at [100] as Connectivity Table.

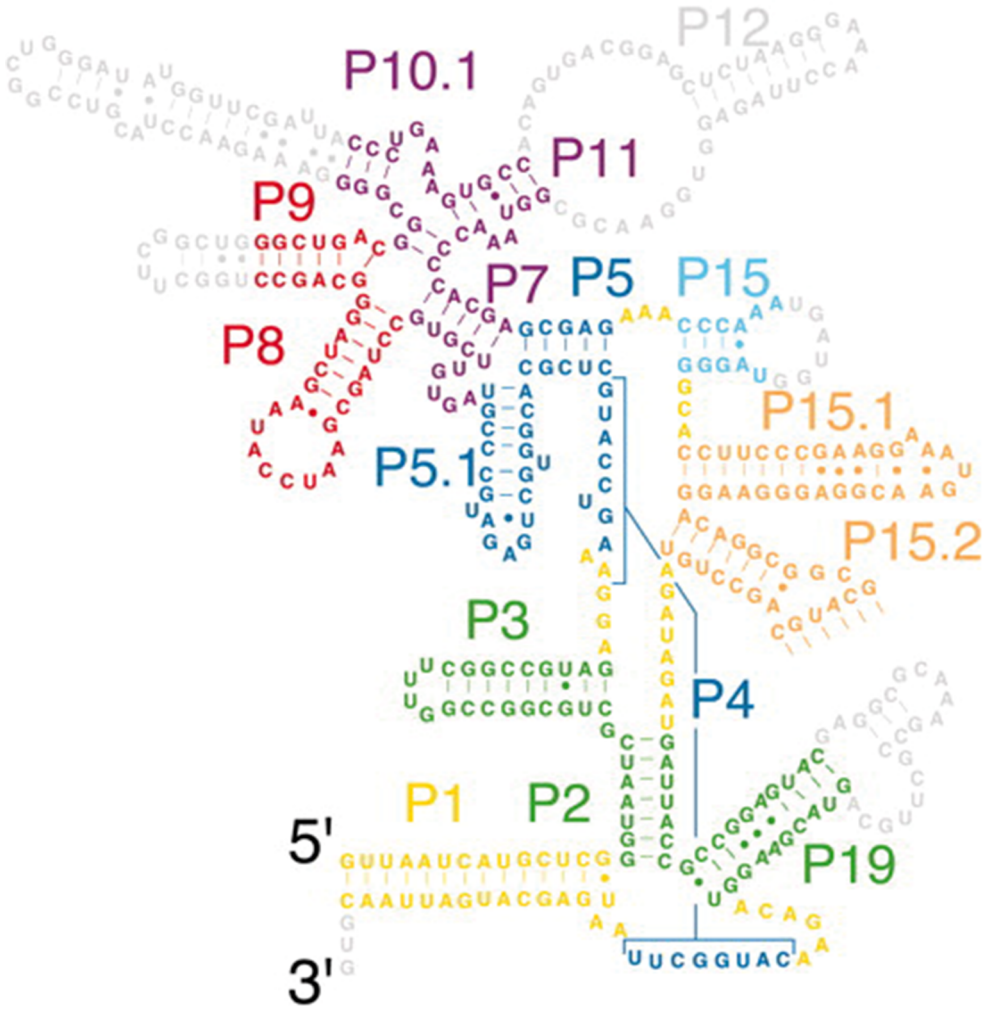


Figure 3.33: *2D structure of bacterial ribonuclease P RNA.* Fig. 1 of Kazantsev et al. of the *B. stearothermophilus* RNase P RNA structure, PDB entry 2A64 [104]. The most notable feature is the pseudoknot formed by helix P4. Base pairs are listed at [100] as Connectivity Table.

To be able to process the structure, the implementation of the SCMF method, brot, needed some adaptation. Just for the proof of principle, the input parser

was preloaded with the structure and the analysis of structural features was modified. When fetching features from the input structure without pseudoknots, the case that a new helix may spawn from within a helix is not covered. For this one example this was enabled in a way that it works for the ribonuclease P structure. Therefore this functionality is highly experimental in terms of program code.

With those extensions brot is able to produce a sequence within 15 minutes. The result is able to form Watson-Crick and GU pairs in positions where the input structure requests them. It was not possible to check if the designed sequence would fold to the target structure, since there are no reliable tools for predictions including pseudoknots. Those kind of long range interactions are not supported by many tools predicting secondary structure. The one which was tried with the designed sequence is pKiss by Janssen & Giegerich [105], the successor of pknotsRG by Reeder et al. [79] but without a positive result. Every structure predicted by pKiss does look rather different to the ribonuclease P structure. Predictions do contain pseudoknots, some even several, but the 5' end of helix P4 always stays unpaired.

Since it was unclear if the issue was the sequence or the structure prediction, brot was not further tested with different parameters. The sequence designed by default parameters already calculates to a Gibbs energy of 256.2 kcal mol^{-1} while the native sequence gets 190.0 kcal mol^{-1}, using the NN model as energy function. In theory this makes our design more stable than nature.

tRNA Design & Validation

This study is a joint work between the group of Ulrich Hahn (University of Hamburg, Institute for Biochemistry and Molecular Biology) and our group. The subject is the redesign of a 76 nt yeast phenylalanine tRNA [107] and the experimental validation. The structure is available as PDB entry 6TNA. Fig. 3.34 shows the secondary structure after stripping from the PDB file using s2s by Jossinet & Westhof [106].

The work is split into two parts: the first task is the computational design of a new sequence folding into a tRNA cloverleaf secondary structure, the second was done by Kristina Dorothée Gorkotte-Szameit, experimentally validating base pairs.

As a first step, a pool of sequence candidates was created. This was done by SCMF and a second method based on Newtonian dynamics operating on the same sequence space as our method, developed by Marco Matthies in our group [108]. The idea behind using two tools was to gain confidence in the solution by choosing a sequence predicted by both methods. brot was used with various parameters to create several sequences which are stable for dif-

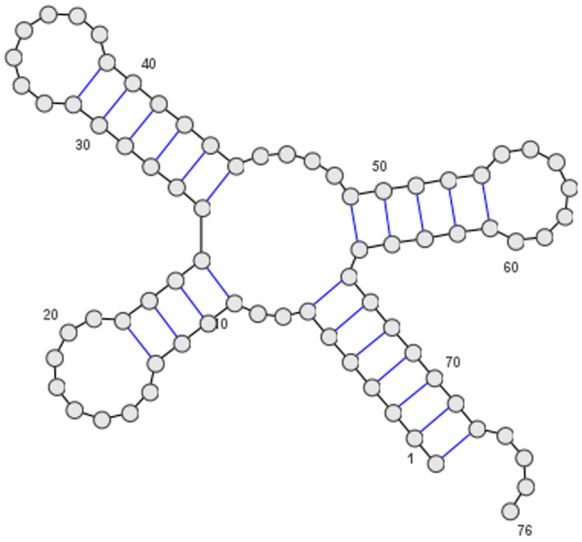


Figure 3.34: *2D structure of yeast tRNA-Phe.* Secondary structure of PDB entry 6TNA as stripped by `s2s` [106].

ferent settings. The list of sequences both design methods agreed up on was then run through various computational tools to chose a single sequence.

GAGCGCCACGGACGAACACAAGUCCGAUCGCGACAACAGCGAUCAACAGCCACGACAACAGUGGCGGCGCUCACAA

Figure 3.35: *Designed sequence for the yeast tRNA-Phe.* The final sequence as the outcome of the design process and *in silico* validation. Predicted structures with `RNAfold` and `CONTRAfold` have a base pair distance of 0 compared to the target structure. Folding probability is 0.904.

To predict a fold for the sequences, not only `RNAfold` was used. `CONTRAfold` [109] predicts secondary structure based on a machine learning approach and `MC-Fold` [11] allows structures with non-canonical base pairs. Acceptance criteria for `RNAfold` were a base pair distance of 0 and a folding probability above 0.9. Since `CONTRAfold` does not provide probabilities, only the base pair distance was tested. A distance measurement with results of `MC-Fold` would have rejected all sequences. This originates form the tendency of this tool to establish too many base pairs. Since the final sequence shown in Fig. 3.35 passed the first two tools, we assumed a ranking in the top ten scoring of `MC-Fold` as sufficient. After passing all tests, we made sure that we look at a really unique sequence by running a BLAST search on the NCBI web page [110] finding no hits.

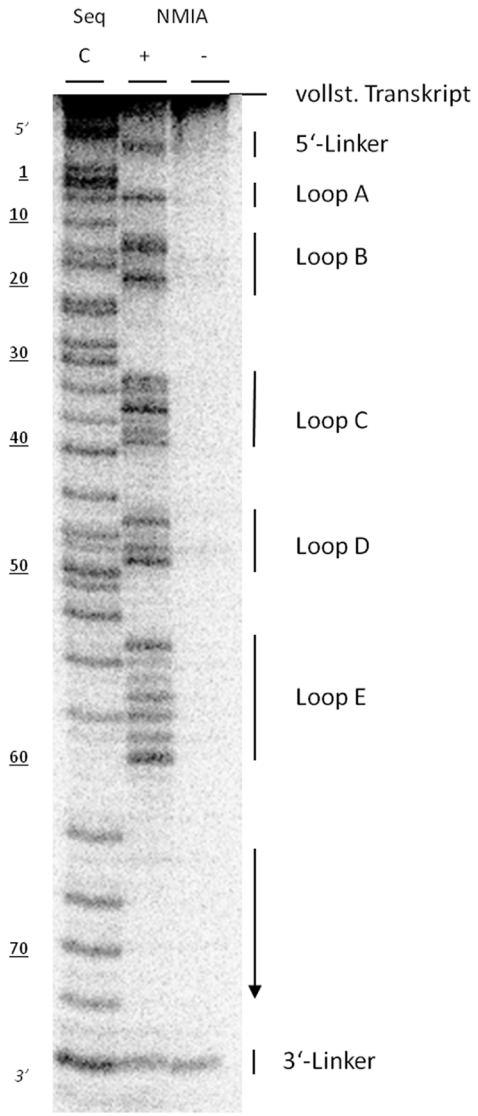


Figure 3.36: *Sequencing gel of the designed yeast tRNA-Phe sequence.* The left lane shows a sequencing reaction locating guanines in the sequence. On the middle lane reactivity of SHAPE fragments is shown. The right lane is a control for the absence of signals without inducing SHAPE reactions. Image used unmodified from [111].

In the second part of this study, the structure of the synthesised RNA is analysed utilising the SHAPE method [112, 113]. This experimental part is entirely covered by the diploma thesis of Kristina Dorothée Gorkotte-Szameit [111], so here we only give a quick overview on the method. SHAPE (Selective 2'-Hydroxyl Acylation analysed by Primer Extension) exploits the different flexibility in paired and unpaired RNA nucleotides. Chemical reagents are added to the solution and react with the 2'-hydroxyl group of nucleotides, if those are flexible. That way SHAPE is detecting unpaired nucleotides while the absence of signals is interpreted as paired regions. To identify locations of unpaired sites a sequencing run is needed on the same gel SHAPE fragments are distributed on. This is shown in the first two columns of Fig. 3.36. On the gel, loop and paired regions show good agreement with sequence positions, providing evidence that the secondary structure was established as intended.

The conclusion of the experimental validation of the designed sequence is that designing a sequence works in principle. Other sequence prediction tools always stayed on the hypothetical level, while here it has been shown for SCMF and the Newtonian dynamics methods that sequence design *in silico* works. While it seems to be also evident that a single tool will not be enough for real world scenarios given the amount of additional testing done on sequence candidates.

Chapter 4

Discussion

When surveying the field of structure analysis and design, there are still problems at the fundamental and technical levels.

As a first idea, thinking of the definition of sequence design in Chapter 3. This assumes a rigid world by considering a base pair as discrete state. But the work on hydrogen bonding shows, these rigid definitions are a convenience and not the chemical truth. Following the evaluation in Chapter 2 the energy associated with a hydrogen bond is a continuous function of geometry and not a discrete property. This is clear from the distributions of quasi-energy from the simple model in §2.5. To get to a better model of the real world would mean using a probabilistic description over correct physical models. Unfortunately, this will lead to a sea of probability distributions which quickly renders this kind of problem intractable. As an example, if one considers extending the hydrogen bond model for all possible interaction edges in Fig. 2.3 instead of only treating the Watson-Crick edge, this ends up in several additional quasi-energy distributions per edge.

This leads to the general question of the energy models used in this work. For hydrogen bond recognition, a model with the same idea of Kabsch & Sander [8], resembling Coulomb energies was used. For proteins, this is well established and has become a standard tool for secondary structure assignment. To deal with the sequence design problem, the NN model was used. It is a simple choice with the virtue of being the most widely used model in the literature. But there is the severe disadvantage of having little physical basis. Parameters do come from experimental measurements, but then they are treated somewhat out of the context of the system they originate from. Each structural motif measured is assumed to be self-contained and can be combined with every other motif. Within loops, including base pairs as stack loops, that goes to the extend that Gibbs energies are approximated as simple

summation. This completely ignores the fact that entropy is not a linear additive quantity.

In most molecular modelling exercises, there are two aspects which can be difficult to disentangle. Firstly, there is the question of force field or scoring function. How close is the model to the real world, in terms of true physics or minima on the computed energy landscape? Secondly, one wants to know how well the energy model is explored. That is a question of search method. Does the choice of Monte Carlo [114], SCMF or another search strategy find the best minimum on a computed energy landscape? In this work, the scoring function is not a question since all methods use the NN model. Also for the evaluation known cases were used, calculated using the same model. Obviously this is not a test of real world behaviour, but a traditional test of search methods in computational biology & chemistry. Interpreting test results is slightly more complicated in our particular case, since there are two goals in literature. Does a method design a sequence with the target structure at minimum free energy or is success judged by the calculated probability of the target structure? For us the decision was clear. The probability of a sequence to adopt the target conformation is most important. But one also has to consider the success rate for MFE since this is often discussed in the literature.

Given our criterion of success, we can say that the SCMF methods by its implementation brot works very well. On the artificial structures, the other methods compared may be better. One could even go so far as to say that on small structures `RNAinverse` in probability mode is unbeatable. That is just because it is not really searching but enumerating full sequences to pick a matching one. Given enough time, this should always converge to a good solution. But considering time consumption, this is the limiting factor. Concerning structure size, `RNAinverse` gets quickly impracticable. INFO-RNA does look better on its own publication [59], only evaluating MFE and version 1 of the tool. In this work version 2 was used and testing for probability. It turns out, optimising for probability seems to be more challenging. Where brot is not the top-performer is on the cases of pure search method comparison with less real world relevance. When it comes to biological sequences, producing structures of higher complexity, brot takes the lead. Additionally our tool has shown its potential in three case studies. Firstly, we are able to compute a ribosomal RNA of 2904 nt in less than half an hour. The result is compatible with the target structure and the most probable fold gets only 87 out of 808 base pairs wrong at low normalised ensemble defect of 0.05. In a proof of principle manner, we have shown that pseudoknots are not a problem. Once adapted to the constraints in difference to knot-free structures, sequences fitting the target pairing pattern are produced. This is something impossible for the other tools. As a last dedicated study, the secondary structure of a

sequence designed with the SCMF method was successfully validated *in vitro*. Experimental evidence is another topic the other tools are missing.

In general, SCMF seems to have some advantages over the discrete approaches. As explained in §3.3.3 splitting of structures, necessary to keep running times manageable, may prevent some good results. Only optimising parts of a structure can easily end up in a local minimum. Another disadvantage is to rely on a trial and error strategy. This limits the classes of structures which can be handled to motifs which can be predicted. Namely this forbids pseudoknots and the possibility to move to three-dimensional space. In contrast we do not test intermediate results, we can converge on the full-size structure without testing folding during the process. What is crucial in SCMF is a scoring scheme for what should be designed and a representation fitting our sequence matrix.

This already points at extensions of brot. But before increasing the functionality, some ideas for improvements on the current status. What is most unsatisfying in the current implementation, is the low level of different sequences produced for the same target structure. On average, currently five to six runs are needed to produce two different sequences. While SCMF is known to be deterministic in its end point, adding thermal noise to the NN model has an effect. Probably one needs to look into which are the exact parameters changed, once a new sequence is produced. Maybe some structural motifs inside the model are more likely to favour new sequences if equipped with a certain amount of noise. Another idea concerning the use of the NN model is an improvement of the negative design term. As discussed in §3.3.3, some base compositions exist in multiple states with different energies. What strategy prevents accepting a pattern more favourable in a different than the intended motif needs further investigation. A simple extension to the sequence matrix would be moving towards modelling multimers. Multiple sequence can be placed one after another in the sequence matrix. What needs some consideration is the negative design term. It needs to be aware of the ends of sequences while a private value per sequence seems not to be necessary. Just the update routine moving the term over the sequence matrix needs to be aware of sequence starts and ends. At those points the calculation can not assume a continuous sequence anymore and skip two stacked base pairs. For intended interactions nothing should change. This term only considers interacting sites, not complete sequences.

A real challenging extension would be sequence design in three-dimensional space. With the current approach, only secondary structure is covered without knowledge about tertiary interactions. This means every design needs to be thoroughly evaluated *in vitro* before it can be used for its purpose. That will not entirely vanish with the possibility to create three-dimensional designs,

but speed up the process by suggesting solutions which are less likely to fail. A huge practical advantage could be to include ligands in the design step, in the future. This is impossible with two-dimensional designs. To get to this state bears some difficulties. Firstly there is the representation of the scaffold. This could be provided by something like the standardised nucleotide presented in Fig. 2.2. At least the orientation of nucleotides in space could be described with this. But when describing interactions, this also needs some coarse-grained representation. For secondary structure design, the states are "paired" and "unpaired", easy to reflect with bases on a non-atomistic level. In tertiary structures, states go into the direction of "H-bond" or "no H-bond". But this would already determine where in space atoms are needed, limiting design possibilities. Instead, the interaction edges discussed in Chapter 2 could be useful to declare the interaction pattern. Since building a RNA structure just from known interactions seems not to be doable in a reliable way for now, this would not eradicate the need for a scaffold. Then, also a scoring function would be needed to evaluate the system. Assuming a completely rigid molecule, calculating H-bond interactions may be sufficient. This would push the whole idea quite a bit away from the real world. But when insisting on a flexible backbone calculating the force field for SCMF becomes computationally challenging. Flexibility would also mean to employ a minimiser to relax the structure which needs to be able to deal with our four-dimensional nucleotides. Beside fixing nucleotides in the sequence matrix, this would mean over time also the structure settles into a final state. Three-dimensional sequence design does not seem to be impossible but it does require a lot of further consideration before one can state how usable a result would be in the end.

To some extend the H-bond probability distributions could serve for the scoring of three-dimensional design. In terms of extending the H-bond terms in this work, the next step would be to lower the number of probability distributions as described in §2.5.6. After this, the Hoogsteen and sugar edge should be incorporated, since they are crucial in tertiary interactions of RNA molecules. Once all interaction sites are described, favourably by a low number of probability distributions, the whole model could be assembled into a tool for automated RNA structure assignment.

Bibliography

[1] Crick, F. Central Dogma of Molecular Biology. *Nature* **227** 561–562 (1970). doi:10.1038/227561a0. 9

[2] Berg, J. M., Tymoczko, J. L. & Stryer, L. Biochemistry (W.H. Freeman, New York, 2002), 5th edn. ISBN 0-7167-3051-0. 9, 12, 64

[3] Jaeger, L. The new world of ribozymes. *Curr. Opin. Struct. Biol.* **7** 324 – 335 (1997). doi:10.1016/S0959-440X(97)80047-4. 9

[4] Gesteland, R. F. The RNA World (Cold Spring Harbor Laboratory Press, Cold Spring Harbor, 2005), 3rd edn. ISBN 978-0879697396. 9

[5] Lee, V. S., Nimmanpipug, P., Aruksakunwong, O., Promsri, S., Sompornpisut, P. & Hannongbua, S. Structural analysis of lead fullerene-based inhibitor bound to human immunodeficiency virus type 1 protease in solution from molecular dynamics simulations. *J. Mol. Graphics Modell.* **26** 558 – 570 (2007). doi:10.1016/j.jmgm.2007.03.013. 9, 73

[6] Cooper, T. A., Wan, L. & Dreyfuss, G. RNA and Disease. *Cell* **136** 777–93 (2009). doi:10.1016/j.cell.2009.02.011. 9

[7] Osborne, R. J. & Thornton, C. A. RNA-dominant diseases. *Human Molecular Genetics* **15** R162–R169 (2006). doi:10.1093/hmg/ddl181. 9

[8] Kabsch, W. & Sander, C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22** 2577–2637 (1983). doi:10.1002/bip.360221211. 9, 11, 12, 21, 159

[9] Cech, T. R. Ribozyme engineering. *Curr. Opin. Struct. Biol.* **2** 605 – 609 (1992). doi:10.1016/0959-440X(92)90093-M. 10, 73

[10] Nussinov, R. & Jacobson, A. B. Fast algorithm for predicting the secondary structure of single-stranded

RNA. *Proc. Natl. Acad. Sci. U. S. A.* **77** 6309–6313 (1980). eprint:http://www.pnas.org/content/77/11/6309.full.pdf+html. 10, 38, 76, 79, 84, 93

[11] Parisien, M. & Major, F. The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature* **452** 51–55 (2008). doi:10.1038/nature06684. 10, 11, 85, 156

[12] Kim, S. H., Quigley, G., Suddath, F. L., McPherson, A., Sneden, D., Kim, J. J., Weinzierl, J., Blattmann, P. & Rich, A. The three-dimensional structure of yeast phenylalanine transfer RNA: shape of the molecule at 5.5 Å resolution. *Proc. Natl. Acad. Sci. U. S. A.* **69** 3746–3750 (1972). eprint:http://www.pnas.org/content/69/12/3746.full.pdf+html. 11

[13] Kim, S. H., Quigley, G. J., Suddath, F. L., McPherson, A., Sneden, D., Kim, J. J., Weinzierl, J. & Rich, A. Three-dimensional structure of yeast phenylalanine transfer RNA: folding of the polynucleotide chain. *Science* **179** 285–288 (1973). doi:10.1126/science.179.4070.285. 11

[14] Ladner, J. E., Jack, A., Robertus, J. D., Brown, R., Rhodes, D., Clark, B. & Klug, A. Atomic co-ordinates for yeast phenylalanine tRNA. *Nucleic Acids Res.* **2** 1629–1638 (1975). doi:10.1093/nar/2.9.1629. 11

[15] Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M. & Schuster, P. Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.* **125** 167–188 (1994). doi:10.1007/BF00818163. 11, 74, 76, 79, 80, 81, 85, 103, 107, 108, 147, 148

[16] Hofacker, I. L. Vienna RNA secondary structure server. *Nucleic Acids Res.* **31** 3429–3431 (2003). doi:10.1093/nar/gkg599. 11, 79

[17] Lorenz, R., Bernhart, S., Honer zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. & Hofacker, I. ViennaRNA package 2.0. *Algorithms Mol. Biol.* **6** 26 (2011). doi:10.1186/1748-7188-6-26. 11, 74, 76, 79, 80, 85, 107

[18] Zuker, M. & Stiegler, P. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* **9** 133–148 (1981). doi:10.1093/nar/9.1.133. 11, 76, 79, 84, 93

[19] Zuker, M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* **31** 3406–3415 (2003). doi:10.1093/nar/gkg595. 11, 76, 79

[20] Markham, N. R. & Zuker, M. UNAFold: software for nucleic acid folding and hybridization, chap. 1, 3–31. Bioinformatics, Volume II. Structure, Function and Applications, no. 453 in *Methods Mol. Biol.* (Humana Press, Totowa, NJ., 2008). ISBN 978-1-60327-428-9. doi:10.1007/978-1-60327-429-6_1. 11, 76, 79

[21] Freier, S. M., Kierzek, R., Jaeger, J. A., Sugimoto, N., Caruthers, M. H., Neilson, T. & Turner, D. H. Improved free-energy parameters for predictions of RNA duplex stability. *Proc. Natl. Acad. Sci. U. S. A.* **83** 9373–9377 (1986). eprint:http://www.pnas.org/content/83/24/9373.full.pdf+html. 11, 27, 76, 77

[22] Jaeger, J. A., Turner, D. H. & Zuker, M. Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci. U. S. A.* **86** 7706–7710 (1989). eprint:http://www.pnas.org/content/86/20/7706.full.pdf+html. 11, 27, 76

[23] Mathews, D. H., Sabina, J., Zuker, M. & Turner, D. H. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.* **288** 911–940 (1999). doi:10.1006/jmbi.1999.2700. 11, 27, 76

[24] Mills, J. E. J. & Dean, P. M. Three-dimensional hydrogen-bond geometry and probability information from a crystal survey. *J. Comput.-Aided Mol. Des.* **10** 607–622 (1996). doi:10.1007/BF00134183. 12, 13, 15, 20

[25] Yang, H., Jossinet, F., Leontis, N., Chen, L., Westbrook, J., Berman, H. & Westhof, E. Tools for the automatic identification and classification of RNA base pairs. *Nucleic Acids Res.* **31** 3450–3460 (2003). doi:10.1093/nar/gkg529. 12, 13, 16, 17, 20, 27

[26] Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C. & Ferrin, T. E. UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.* **25** 1605–1612 (2004). doi:10.1002/jcc.20084. url:www.cgl.ucsf.edu/chimera 13

[27] Allen, F. H., Davies, J. E., Galloy, J. J., Johnson, O., Kennard, O., Macrae, C. F., Mitchell, E. M., Mitchell, G. F., Smith, J. M. & Watson, D. G. The development of versions 3 and 4 of the Cambridge structural database system. *J. Chem. Inf. Comput. Sci.* **31** 187–204 (1991). doi:10.1021/ci00002a004. 13

[28] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. & Bourne, P. E. The protein data bank. *Nucleic Acids Res.* **28** 235–242 (2000). doi:10.1093/nar/28.1.235. url:www.pdb.org 13, 39

[29] Olson, W. K., Bansal, M., Burley, S. K., Dickerson, R. E., Gerstein, M., Harvey, S. C., Heinemann, U., Lu, X.-J., Neidle, S., Shakked, Z., Sklenar, H., Suzuki, M., Tung, C.-S., Westhof, E., Wolberger, C. & Berman, H. M. A standard reference frame for the description of nucleic acid base-pair geometry. *J. Mol. Biol.* **313** 229 – 237 (2001). doi:DOI:10.1006/jmbi.2001.4987. 16, 17

[30] Leontis, N. B. & Westhof, E. Geometric nomenclature and classi-

fication of RNA base pairs. *RNA* **7** 499–512 (2001). eprint:http://rnajournal.cshlp.org/content/7/4/499.full.pdf+html. 16, 18, 19

[31] Chemical component dictionary (2011). url:http://www.wwpdb.org/ccd.html. [Online; accessed 04-March-2011]. 24

[32] Henrick, K., Feng, Z., Bluhm, W. F., Dimitropoulos, D., Doreleijers, J. F., Dutta, S., Flippen-Anderson, J. L., Ionides, J., Kamada, C., Krissinel, E., Lawson, C. L., Markley, J. L., Nakamura, H., Newman, R., Shimizu, Y., Swaminathan, J., Velankar, S., Ory, J., Ulrich, E. L., Vranken, W., Westbrook, J., Yamashita, R., Yang, H., Young, J., Yousufuddin, M. & Berman, H. M. Remediation of the protein data bank archive. *Nucleic Acids Res.* **36** D426–D433 (2008). doi:10.1093/nar/gkm937. 24

[33] Gremme, G. The GenomeTools genome analysis system. (2011). url:http://genometools.org. [Online; accessed 24-March-2011]. 27

[34] Wurst, H. Everything has an end but the Wurst has 2. (2011). url:http://wurst.org. [Online; accessed 24-March-2011]. 27

[35] Reyes, F. E., Garst, A. D. & Batey, R. T. Strategies in RNA crystallography. In Herschlag, D. (ed.), Biophysical, Chemical, and Functional Probes of RNA Structure, Interactions and Folding: Part B, vol. 469 of *Methods in Enzymology*, 119 – 139 (Academic Press, 2009). doi:DOI:10.1016/S0076-6879(09)69006-6. 27

[36] Spears, J. L., Gaston, K. W. & Alfonzo, J. D. Analysis of tRNA editing in native and synthetic substrates. In Aphasizhev, R. (ed.), RNA and DNA Editing, vol. 718 of *Methods in Molecular Biology*, 209–226 (Humana Press, 2011). ISBN 978-1-61779-018-8. url:http://dx.doi.org/10.1007/978-1-61779-018-8_13. 27

[37] Protein data bank contents guide: atomic coordinate entry format description (2008). url:http://www.wwpdb.org/documentation/format32/v3.2.html. [Online; accessed 06-April-2011]. 27, 29, 34

[38] Hopcroft, J., Motwani, R. & Ullman, J. Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie (Oldenbourg R. Verlag GmbH, 2000), 3rd edn. ISBN 3827370205. 28

[39] PDB ID: 1C2W
Mueller, F., Sommer, I., Baranov, P., Matadeen, R., Stoldt, M., Wöhnert, J., Görlach, M., van Heel, M. & Brimacombe, R. The 3D arrangement of the 23 S and 5 S rRNA in the *Escherichia coli* 50 S ribosomal subunit based on a cryo-electron microscopic reconstruction at 7.5 Å resolution. *J. Mol. Biol.* **298** 35 – 59 (2000). doi:10.1006/jmbi.2000.3635. 39, 64, 150, 151

[40] PDB ID: 157D
Leonard, G. A., McAuley-Hecht, K. E., Ebel, S., Lough, D. M., Brown, T. & Hunter, W. N. Crystal and molecular structure of r(CGCGAAUUAGCG): an RNA duplex containing two G(*anti*)·A(*anti*) base

pairs. *Structure* **2** 483–94 (1994). doi:10.1016/S0969-2126(00)00049-6. 39, 61

[41] PDB ID: 165D
Cruse, W. B., Saludjian, P., Biala, E., Strazewski, P., Prangé, T. & Kennard, O. Structure of a mispaired RNA double helix at 1.6-A resolution and implications for the prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. U. S. A.* **91** 4160–4164 (1994). eprint:http://www.pnas.org/content/91/10/4160.full.pdf+html. 39, 61, 62

[42] PDB ID: 1A51
Dallas, A. & Moore, P. B. The loop E–loop D region of *Escherichia coli* 5 S rRNA: the solution structure reveals an unusual loop that may be important for binding ribosomal proteins. *Structure* **5** 1639 – 1653 (1997). doi:10.1016/S0969-2126(97)00311-0. 39, 62

[43] PDB ID: 1B36
Butcher, S. E., Allain, F. H. & Feigon, J. Solution structure of the loop B domain from the hairpin ribozyme. *Nat. Struct. Biol.* **6** 212–6 (1999). doi:10.1038/6651. 39

[44] PDB ID: 1DDY
Sussman, D., Nix, J. C. & Wilson, C. The structural basis for molecular recognition by the vitamin B_{12} RNA aptamer. *Nat. Struct. Mol. Biol.* **7** 53–7 (2000). doi:10.1038/71253. 39, 65

[45] Brodsky, A. S. & Williamson, J. R. Solution structure of the HIV-2 TAR-argininamide complex. *J. Mol. Biol.* **267** 624 – 639 (1997). doi:10.1006/jmbi.1996.0879. 39, 63

[46] PDB ID: 1FIR
Bénas, P., Bec, G., Keith, G., Marquet, R., Ehresmann, C., Ehresmann, B. & Dumas, P. The crystal structure of HIV reverse-transcription primer $tRNA^{Lys,3}$ shows a canonical anticodon loop. *RNA* **6** 1347–1355 (2000). eprint:http://rnajournal.cshlp.org/content/6/10/1347.full.pdf+html. 39, 67, 68

[47] PDB ID: 1EHZ
Shi, H. & Moore, P. B. The crystal structure of yeast phenylalanine tRNA at 1.93 Å resolution: a classic structure revisited. *RNA* **6** 1091–1105 (2000). eprint:http://rnajournal.cshlp.org/content/6/8/1091.full.pdf+html. 39, 66, 67

[48] PDB ID: 170D
Schweitzer, B. I., Mikita, T., Kellogg, G. W., Gardner, K. H. & Beardsley, G. P. Solution structure of a DNA dodecamer containing the anti-neoplastic agent arabinosylcytosine: Combined use of NMR, restrained molecular dynamics, and full relaxation matrix refinement. *Biochemistry* **33** 11460–11475 (1994). doi:10.1021/bi00204a008. 39, 61

[49] PDB ID: 1BZT
Durant, P. C. & Davis, D. R. Stabilization of the anticodon stem-loop of $tRNA^{Lys,3}$ by an A^{+}-C base-pair and by pseudouridine. *J. Mol. Biol.* **285** 115 – 131 (1999). doi:10.1006/jmbi.1998.2297. 39, 65

[50] Luebke, K. J., Landry, S. M. & Tinoco, I. Solution conformation of a five-nucleotide RNA bulge loop

from a group I intron. *Biochemistry* **36** 10246–10255 (1997). doi:10.1021/bi9701540. 63, 64

[51] Levitt, M. Detailed molecular model for transfer ribonucleic acid. *Nature* **224** 759 – 763 (1969). doi:10.1038/224759a0. 67

[52] Forrest, L. R. & Honig, B. An assessment of the accuracy of methods for predicting hydrogen positions in protein structures. *Proteins: Struct., Funct., Bioinf.* **61** 296–309 (2005). doi:10.1002/prot.20601. 69

[53] Leontis, N. B. & Westhof, E. Geometric nomenclature and classification of RNA base pairs. *RNA* **7** 499–512 (2001). eprint:http://rnajournal.cshlp.org/content/7/4/499.full.pdf+html. 70

[54] West, M. L. & Fairlie, D. P. Targeting HIV-1 protease: a test of drug-design methodologies. *Trends Pharmacol. Sci.* **16** 67 – 75 (1995). doi:10.1016/S0165-6147(00)88980-4. 73

[55] Wang, J. Can man-made nanomachines compete with nature biomotors? *ACS Nano* **3** 4–9 (2009). doi:10.1021/nn800829k. 73

[56] Farokhzad, O. C. & Langer, R. Impact of nanotechnology on drug delivery. *ACS Nano* **3** 16–20 (2009). doi:10.1021/nn900002m. 73

[57] Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* **16** 3–50 (1996). doi:10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6. 73

[58] Dirks, R. M., Lin, M., Winfree, E. & Pierce, N. A. Paradigms for computational nucleic acid design. *Nucleic Acids Res.* **32** 1392–1403 (2004). doi:10.1093/nar/gkh291. 74, 75

[59] Busch, A. & Backofen, R. INFO-RNA–a fast approach to inverse RNA folding. *Bioinformatics* **22** 1823–1831 (2006). doi:10.1093/bioinformatics/btl194. 74, 79, 81, 84, 115, 117, 132, 137, 160

[60] Andronescu, M., Fejes, A. P., Hutter, F., Hoos, H. H. & Condon, A. A new algorithm for RNA secondary structure design. *J. Mol. Biol.* **336** 607–624 (2004). doi:10.1016/j.jmb.2003.12.041. 74, 79, 81, 83, 115, 132

[61] Wuchty, S., Fontana, W., Hofacker, I. L. & Schuster, P. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers* **49** 145–165 (1999). doi:10.1002/(SICI)1097-0282(199902)49:2<145::AID-BIP4>3.0.CO;2-G. 74, 76, 79

[62] Bolton, E. T. & McCarthy, B. J. A general method for the isolation of RNA complementary to DNA. *Proc. Natl. Acad. Sci. U. S. A.* **48** 1390–1397 (1962). eprint:http://www.pnas.org/content/48/8/1390.full.pdf+html. 75

[63] Li, K., Brownley, A., Stockwell, T., Beeson, K., McIntosh, T., Busam,

D., Ferriera, S., Murphy, S. & Levy, S. Novel computational methods for increasing PCR primer design effectiveness in directed sequencing. *BMC Bioinf.* **9** (2008). url:http://ukpmc.ac.uk/abstract/MED/18405373. 75

[64] Mitsuhashi, M. Technical report: part 1. basic requirements for designing optimal oligonucleotide probe sequences. *J. Clin. Lab. Anal.* **10** 277–284 (1996). doi:10.1002/(SICI)1098-2825(1996)10:5<277::AID-JCLA8>3.0.CO;2-5. 75

[65] Breslauer, K. J., Frank, R., Blöcker, H. & Marky, L. A. Predicting DNA duplex stability from the base sequence. *Proc. Natl. Acad. Sci. U. S. A.* **83** 3746–3750 (1986). eprint:http://www.pnas.org/content/83/11/3746.full.pdf+html. 76, 77

[66] Gotoh, O. & Tagashira, Y. Stabilities of nearest-neighbor doublets in double-helical DNA determined by fitting calculated melting profiles to observed profiles. *Biopolymers* **20** 1033–1042 (1981). doi:10.1002/bip.1981.360200513. 76, 77

[67] Walter, A. E., Turner, D. H., Kim, J., Lyttle, M. H., Müller, P., Mathews, D. H. & Zuker, M. Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proc. Natl. Acad. Sci. U. S. A.* **91** 9218–9222 (1994). eprint:http://www.pnas.org/content/91/20/9218.full.pdf+html. 76, 77

[68] Wu, M., McDowell, J. A. & Turner, D. H. A periodic table of tandem mismatches in RNA. *Biochemistry* **34** 3204–3211 (1995). doi:10.1021/bi00010a009. 76, 77

[69] Antao, V. P. & Tinoco, I. Thermodynamic parameters for loop formation in RNA and DNA hairpin tetraloops. *Nucleic Acids Res.* **20** 819–824 (1992). doi:10.1093/nar/20.4.819. 76, 77, 96

[70] Sheehy, J. P., Davis, A. R. & Znosko, B. M. Thermodynamic characterization of naturally occurring RNA tetraloops. *RNA* (2010). doi:10.1261/rna.1773110. 76, 77, 96

[71] Zuker, M. On finding all suboptimal foldings of an RNA molecule. *Science* **244** 48–52 (1989). doi:10.1126/science.2468181. 76, 79

[72] He, L., Kierzek, R., SantaLucia, J., Walter, A. E. & Turner, D. H. Nearest-neighbor parameters for G·U mismatches: $\overset{5'GU3'}{3'UG5'}$ is destabilizing in the contexts $\overset{CGUG}{\dot{g}UG\dot{c}}$, $\overset{UGUA}{\grave{a}UG\dot{u}}$, and $\overset{AGUU}{\dot{u}UG\grave{a}}$ but stabilizing in $\overset{GGUC}{\dot{c}UG\dot{g}}$. *Biochemistry* **30** 11124–11132 (1991). doi:10.1021/bi00110a015. 77

[73] Gultyaev, A. P., van Batenburg, F. H. & Pleij, C. W. An approximation of loop free energy values of RNA H-pseudoknots. *RNA* **5** 609–617 (1999). eprint:http://rnajournal.cshlp.org/content/5/5/609.full.pdf+html. 77

[74] Andronescu, M. S., Pop, C. & Condon, A. E. Improved free energy parameters for RNA pseudoknotted secondary structure prediction.

RNA **16** 26–42 (2010). doi:10.1261/rna.1689910. 77, 84

[75] Bellman, R. On the theory of Dynamic Programming. *Proc. Natl. Acad. Sci. U. S. A.* **38** 716–719 (1952). eprint:http://www.pnas.org/content/38/8/716.full.pdf+html. 79

[76] Bellman, R. Dynamic Programming. *Science* **153** 34–37 (1966). doi:10.1126/science.153.3731.34. 79

[77] Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48** 443 – 453 (1970). doi:10.1016/0022-2836(70)90057-4. 79

[78] Nussinov, R., Pieczenik, G., Griggs, J. R. & Kleitman, D. J. Algorithms for loop matchings. *SIAM J. Appl. Math.* **35** 68–82 (1978). url:http://www.jstor.org/stable/2101031. 79

[79] Reeder, J., Steffen, P. & Giegerich, R. pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows. *Nucleic Acids Res.* **35** W320–W324 (2007). doi:10.1093/nar/gkm258. 79, 84, 155

[80] Reidys, C. M., Huang, F. W. D., Andersen, J. E., Penner, R. C., Stadler, P. F. & Nebel, M. E. Topology and prediction of RNA pseudoknots. *Bioinformatics* **27** 1076–1085 (2011). doi:10.1093/bioinformatics/btr090. 79, 84

[81] McCaskill, J. S. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers* **29** 1105–1119 (1990). doi:10.1002/bip.360290621. 82

[82] Bernhart, S., Tafer, H., Mückstein, U., Flamm, C., Stadler, P. & Hofacker, I. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms Mol. Biol.* **1** 3 (2006). doi:10.1186/1748-7188-1-3. 85

[83] Mückstein, U., Tafer, H., Hackermüller, J., Bernhart, S. H., Stadler, P. F. & Hofacker, I. L. Thermodynamics of RNA–RNA binding. *Bioinformatics* **22** 1177–1182 (2006). doi:10.1093/bioinformatics/btl024. 85

[84] Vieregg, J., Cheng, W., Bustamante, C. & Tinoco, I. Measurement of the effect of monovalent cations on RNA hairpin stability. *J. Am. Chem. Soc.* **129** 14966–14973 (2007). doi:10.1021/ja074809o. 85

[85] Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by Simulated Annealing. *Science* **220** 671–680 (1983). doi:10.1126/science.220.4598.671. 86, 88, 92

[86] Leach, A. R. Molecular modelling: Principles and applications (Prentice Hall, Harlow, England, 2001), 2nd edn. ISBN 0582382106. 86, 88, 90, 92

[87] Koehl, P. & Delarue, M. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their

conformational entropy. *J. Mol. Biol.* **239** 249 – 275 (1994). doi: 10.1006/jmbi.1994.1366. 86, 87, 88, 89, 90

[88] Koehl, P. & Delarue, M. A self consistent mean field approach to simultaneous gap closure and side-chain positioning in homology modelling. *Nat. Struct. Biol.* **2** 163–170 (1995). doi: 10.1038/nsb0295-163. 86, 87, 88, 90

[89] Koehl, P. & Delarue, M. Mean-field minimization methods for biological macromolecules. *Curr. Opin. Struct. Biol.* **6** 222 – 226 (1996). doi:10.1016/S0959-440X(96) 80078-9. 86, 87, 88

[90] Delarue, M. & Koehl, P. The inverse protein folding problem: self consistent mean field optimization of a structure specific mutation matrix. In Altman, R., Dunker, A., Hunter, L. & Klein, T. (eds.), Proceedings of the Pacific Symposium on Biocomputing, 109–121 (World Scientific, Singapore, 1997). 86, 87, 88, 89, 90, 104

[91] Voigt, C. A., Gordon, D. & Mayo, S. L. Trading accuracy for speed: a quantitative comparison of search algorithms in protein sequence design. *J. Mol. Biol.* **299** 789 – 803 (2000). doi:10.1006/jmbi.2000. 3758. 86, 87, 88, 89, 90, 93

[92] Cramer, C. J. Essentials of Computational Chemistry (John Wiley & Sons, Chichester, 2004), 2nd edn. ISBN 0470091827. 86

[93] Papadimitriou, C. H. & Steiglitz, K. Combinatorial optimization: algorithms and complexity (Courier Dover Publications, Dover, 1998). ISBN 0486402584. 86, 108, 111

[94] Roitberg, A. & Elber, R. Modeling side chains in peptides and proteins: Application of the locally enhanced sampling and the simulated annealing methods to find minimum energy conformations. *J. Chem. Phys.* **95** 9277–9287 (1991). doi:10.1063/1.461157. 87

[95] Bordoli, L., Kiefer, F., Arnold, K., Benkert, P., Battey, J. & Schwede, T. Protein structure homology modeling using SWISS-MODEL workspace. *Nat. Protoc.* **4** 1–13 (2009). doi:10.1038/nprot.2008.197. 87

[96] Eswar, N., Webb, B., Marti-Renom, M. A., Madhusudhan, M., Eramian, D., Shen, M.-y., Pieper, U. & Sali, A. Comparative Protein Structure Modeling Using Modeller, chap. 5, 5.6.1–5.6.30. Curr Protoc Bioinformatics (John Wiley & Sons, Inc., 2006). ISBN 9780471250951. doi: 10.1002/0471250953.bi0506s15. 87

[97] Läuger, P., Stark, G. & Adam, G. Physikalische Chemie und Biophysik. Springer-Lehrbuch (Springer-Verlag, Berlin, 2003), 4th edn. ISBN 3-540-00066-6. 87, 93

[98] Bienert, S. Collection of RNAanalysis Binaries. (2010). url:`https://github.com/bienchen/corb`. [Online; accessed 02-June-2013]. 106

[99] Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. Nu-

merical recipes in C: the art of scientific computing (Cambridge University Press, New York, NY, USA, 1992), 2nd edn. ISBN 0-521-43108-5. url:http://www.nr.com. 108, 111

[100] Test data for "rna energetics and sequence design" (2015). url:http://www.zbh.uni-hamburg.de/fileadmin/bm/Data/rna_test_data.tar.gz. [Online; accessed 27-November-2015]. 108, 112, 117, 125, 133, 137, 150, 151, 154

[101] Zadeh, J. N., Wolfe, B. R. & Pierce, N. A. Nucleic acid sequence design via efficient ensemble defect optimization. *J. Comput. Chem.* **32** 439–452 (2011). doi:10.1002/jcc.21633. 110

[102] Cole, J. R., Wang, Q., Fish, J. A., Chai, B., McGarrell, D. M., Sun, Y., Brown, C. T., Porras-Alfaro, A., Kuske, C. R. & Tiedje, J. M. Ribosomal Database Project: data and tools for high throughput rRNA analysis. *Nucleic Acids Res.* **42** D633–D642 (2014). doi:10.1093/nar/gkt1244. 132, 137

[103] Gorkotte-Szameit, K., Meyer, C., Hahn, U., Matthies, M. & Torda, A. personal communication. 150

[104] PDB ID: 2A64
Kazantsev, A. V., Krivenko, A. A., Harrington, D. J., Holbrook, S. R., Adams, P. D. & Pace, N. R. Crystal structure of a bacterial ribonuclease P RNA. *Proc. Natl. Acad. Sci. U. S. A.* **102** 13392–13397 (2005). doi:10.1073/pnas.0506662102. 154

[105] Janssen, S. & Giegerich, R. The RNA shapes studio. *Bioinformatics* **31** 423–425 (2015). doi:10.1093/bioinformatics/btu649. 155

[106] Jossinet, F. & Westhof, E. Sequence to Structure (S2S): display, manipulate and interconnect RNA data from sequence to structure. *Bioinformatics* **21** 3320–3321 (2005). doi:10.1093/bioinformatics/bti504. 155, 156

[107] PDB ID: 6TNA
Sussman, J. L., Holbrook, S. R., Warrant, R., Church, G. M. & Kim, S.-H. Crystal structure of yeast phenylalanine transfer RNA: I. Crystallographic refinement. *J. Mol. Biol.* **123** 607 – 630 (1978). doi:http://dx.doi.org/10.1016/0022-2836(78)90209-7. 155

[108] Matthies, M. C., Bienert, S. & Torda, A. E. Dynamics in Sequence Space for RNA Secondary Structure Design. *JĊhemİheory Comput.* **8** 3663–3670 (2012). doi:10.1021/ct300267j. 155

[109] Do, C. B., Woods, D. A. & Batzoglou, S. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics* **22** e90–98 (2006). doi:10.1093/bioinformatics/btl246. 156

[110] Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K. & Madden, T. L. BLAST+: architecture and applications. *BMC Bioinf.* **10** 421 (2009). doi:10.1186/1471-2105-10-421.

url:http://blast.ncbi.nlm.nih.gov 156

[111] Gorkotte-Szameit, K. D. Sekundärstrukturanalysen modellierter tRNA-Varianten (2011). 157, 158

[112] Merino, E. J., Wilkinson, K. A., Coughlan, J. L., & Weeks, K. M. RNA Structure Analysis at Single Nucleotide Resolution by Selective 2'-Hydroxyl Acylation and Primer Extension (SHAPE). *J. Am. Chem. Soc.* **127** 4223–4231 (2005). doi: 10.1021/ja043822v. 158

[113] Wilkinson, K. A., Merino, E. J. & Weeks, K. M. Selective 2'-hydroxyl acylation analyzed by primer extension (shape): quantitative RNA structure analysis at single nucleotide resolution. *Nat. Protoc.* **1** 1610–1616 (2006). doi:10.1038/nprot.2006.249. 158

[114] Lester, W. A., Hammond, B. & Reynolds, P. Monte Carlo methods in AB initio quantum chemistry: quantum Monte Carlo for molecules. World Scientific Lecture and Course Notes in Chemistry (World Scientific, Singapore, 1994). url:http://cds.cern.ch/record/1603741. 160

Appendix A

Gefahrstoffe und KMR-Substanzen

Die vorliegende Arbeit ist rein theoretischer Natur. Es wurden daher keinerlei Laborexperimente mit chemischen oder biologischen Materialien durchgeführt. Aus diesem Grund werden keine Gefahrstoffe, krebserzeugende, erbgutverändernde oder fortpflanzungsgefährdende (KMR) Stoffe angegeben.

Appendix B

PDB Structures Used For H-Bond Evaluation

This list was compiled in August 2008 by searching the PDB for structures containing RNA.

100D	104D	124D	157D	161D	165D	168D	170D	176D	17RA	1A1T
1A34	1A3M	1A4D	1A4T	1A51	1A60	1A9L	1A9N	1AC3	1AFX	1AJF
1AJL	1AJT	1AJU	1AKX	1AL5	1AM0	1ANR	1APG	1AQ3	1AQ4	1AQ0
1ARJ	1ASY	1ASZ	1ATO	1ATV	1ATW	1AUD	1AV6	1B23	1B2M	1B36
1B7F	1BAU	1BGZ	1BIV	1BJ2	1BMV	1BNO	1BR3	1BVJ	1BYJ	1BYX
1BZ2	1BZ3	1BZT	1BZU	1C04	1C0A	1C0O	1C2Q	1C2W	1C2X	1C4L
1C9S	1CGM	1CQ5	1CQL	1CSL	1CVJ	1CWP	1CX0	1CX5	1D0T	1D0U
1D4R	1D6K	1D87	1D88	1D96	1D9H	1DDL	1DDY	1DFU	1DI2	1DK1
1DNO	1DNT	1DNX	1DQF	1DQH	1DRR	1DRZ	1DUH	1DUL	1DUQ	1DV4
1DXN	1DZ5	1E4P	1E7K	1E8O	1E8S	1E95	1EBQ	1EBR	1EBS	1EC6
1EFO	1EFS	1EFW	1EGO	1EHT	1EHZ	1EI2	1EIY	1EJZ	1EKA	1EKD
1EKZ	1ELH	1EMI	1EOR	1EQQ	1ESH	1ESY	1ET4	1ETF	1ETG	1EUQ
1EUY	1EVP	1EVV	1EXD	1EXY	1F1T	1F27	1F5G	1F5H	1F5U	1F6U
1F6X	1F6Z	1F78	1F79	1F7F	1F7G	1F7H	1F7I	1F7U	1F7V	1F7Y
1F84	1F85	1F8V	1F9L	1FC8	1FCW	1FEQ	1FEU	1FFK	1FFY	1FFZ
1FGO	1FHK	1FIR	1FIX	1FJE	1FJG	1FKA	1FL8	1FMN	1FNX	1FOQ
1FQZ	1FUF	1FYO	1FYP	1G1X	1G2E	1G2J	1G3A	1G4Q	1G59	1G70
1GAX	1GID	1GIX	1GIY	1GRZ	1GSG	1GTF	1GTN	1GTR	1GTS	1GUC
1GV6	1HOQ	1H1K	1H2C	1H2D	1H38	1H3E	1H4Q	1H4S	1HC8	1HG9
1HHW	1HHX	1HJI	1HLX	1HMH	1HNW	1HNX	1HNZ	1HO6	1HOQ	1HQ1

1HR0 1HR2 1HS1 1HS2 1HS3 1HS4 1HS8 1HVU 1HWQ 1HYS 1I2X
1I2Y 1I3X 1I3Y 1I46 1I4B 1I4C 1I5L 1I6H 1I6U 1I7J 1I94
1I95 1I96 1I97 1I9F 1I9K 1I9V 1I9X 1IBK 1IBL 1IBM 1ICG
1ID9 1IDV 1IDW 1IE1 1IE2 1IHA 1IK1 1IK5 1IKD 1IL2 1IVS
1J1U 1J2B 1J4Y 1J5A 1J5E 1J6S 1J7T 1J8G 1J9H 1JB8 1JBR
1JBT 1JGO 1JGP 1JGQ 1JID 1JJ2 1JO7 1JOX 1JPO 1JTJ 1JTW
1JU1 1JU7 1JUR 1JWC 1JZC 1JZV 1JZX 1JZY 1JZZ 1KO1 1K1G
1K2G 1K4A 1K4B 1K5I 1K6G 1K6H 1K73 1K8A 1K8S 1K8W 1K9M
1K9W 1KAJ 1KC8 1KD1 1KD3 1KD4 1KD5 1KFO 1KH6 1KIS 1KKA
1KKS 1KNZ 1KOC 1KOD 1KOG 1KOS 1KP7 1KPD 1KPY 1KPZ 1KQ2
1KQS 1KUO 1KUQ 1KXK 1L1C 1L1W 1L2X 1L3D 1L3M 1L3Z 1L8V
1L9A 1LAJ 1LC4 1LC6 1LDZ 1LMV 1LNG 1LNT 1LPW 1LS2 1LU3
1LUU 1LUX 1LVJ 1M1K 1M5K 1M5L 1M5O 1M5P 1M5V 1M82 1M8V
1M8W 1M8X 1M8Y 1M9O 1MDG 1MEO 1ME1 1MFJ 1MFK 1MFQ 1MFY
1MHK 1MIS 1MJ1 1MJI 1ML5 1MME 1MMS 1MNB 1MNX 1MSW 1MSY
1MT4 1MUV 1MV1 1MV2 1MV6 1MVR 1MWG 1MWL 1MY9 1MZP 1N1H
1N32 1N33 1N34 1N35 1N36 1N38 1N53 1N66 1N77 1N78 1N7A
1N7B 1N8R 1N8X 1NA2 1NAO 1NB7 1NBK 1NBR 1NBS 1NCO 1NEM
1NJI 1NJM 1NJN 1NJO 1NJP 1NKW 1NLC 1NTA 1NTB 1NTQ 1NTS
1NTT 1NUJ 1NUV 1NWX 1NWY 1NXR 1NYB 1NYI 1NZ1 1OOB 1OOC
1O15 1O3Z 1O9M 1OB2 1OB5 1OKF 1OLN 1OND 1OO7 1OOA 1OQO
1OSU 1OSW 1OW9 1P5M 1P5N 1P5O 1P5P 1P6G 1P6V 1P79 1P85
1P86 1P87 1P9X 1PBL 1PBM 1PBR 1PGL 1PJG 1PJO 1PJY 1PN7
1PN8 1PNS 1PNU 1PNX 1PNY 1PVO 1Q29 1Q2S 1Q75 1Q7Y 1Q81
1Q82 1Q86 1Q8N 1Q93 1Q96 1Q9A 1QA6 1QBP 1QCO 1QC8 1QCU
1QD3 1QD7 1QES 1QET 1QF6 1QFQ 1QLN 1QRS 1QRT 1QRU 1QTQ
1QU2 1QU3 1QVF 1QVG 1QWA 1QWB 1QZA 1QZB 1QZC 1QZW 1R2P
1R2W 1R2X 1R3E 1R3O 1R3X 1R4H 1R7W 1R7Z 1R9S 1R9T 1RAU
1RAW 1RC7 1RFR 1RGO 1RHT 1RKJ 1RLG 1RMN 1RMV 1RNA 1RNG
1RNK 1ROQ 1RPU 1RRD 1RRR 1RXA 1RXB 1RY1 1SO3 1SOV 1S1H
1S1I 1S2F 1S34 1S72 1S76 1S77 1S9L 1S9S 1SA9 1SAQ 1SCL
1SDR 1SDS 1SER 1SFO 1SI3 1SJ3 1SJ4 1SJF 1SLO 1SLP 1SM1
1SY4 1SYZ 1SZ1 1SZY 1TOD 1TOE 1TOK 1T1M 1T1O 1T28 1T2R
1T4L 1T4X 1TBK 1TFN 1TFW 1TFY 1TJZ 1TLR 1TN2 1TOB 1TRA
1TRJ 1TTT 1TUT 1TXS 1UOB 1U2A 1U3K 1U63 1U6B 1U6P 1U8D
1U9S 1ULL 1UN6 1UON 1UTD 1UTS 1UUD 1UUI 1UUU 1UVI 1UVJ
1UVK 1UVL 1UVM 1UVN 1VBX 1VBY 1VBZ 1VCO 1VC5 1VC6 1VC7
1VFG 1VOP 1VOQ 1VOR 1VOS 1VOU 1VOV 1VOW 1VOX 1VOY 1VOZ
1VPO 1VQ4 1VQ5 1VQ6 1VQ7 1VQ8 1VQ9 1VQK 1VQL 1VQM 1VQN
1VQO 1VQP 1VS5 1VS6 1VS7 1VS8 1VS9 1VTM 1W2B 1WKS 1WMQ

1WNE 1WPU 1WRQ 1WSU 1WTS 1WTT 1WVD 1WWD 1WWE 1WWF 1WWG
1WZ2 1X18 1X1L 1X8W 1X9C 1X9K 1XBP 1XHP 1XJR 1XMO 1XMQ
1XNQ 1XNR 1XOK 1XP7 1XPE 1XPF 1XPO 1XPR 1XPU 1XSG 1XSH
1XST 1XSU 1XVO 1XV6 1XWP 1XWU 1YOQ 1Y1W 1Y26 1Y27 1Y39
1Y3O 1Y3S 1Y69 1Y6S 1Y6T 1Y73 1Y77 1Y9O 1Y95 1Y99 1YFG
1YFV 1YG3 1YG4 1YHQ 1YI2 1YIJ 1YIT 1YJ9 1YJN 1YJW 1YKV
1YL3 1YL4 1YLG 1YMO 1YN1 1YN2 1YNC 1YNE 1YNG 1YRJ 1YSH
1YSV 1YTU 1YTY 1YVP 1YXP 1YYK 1YYO 1YYW 1YZ9 1Z2J 1Z3O
1Z31 1Z43 1Z58 1Z7F 1ZBH 1ZBI 1ZBL 1ZBN 1ZC5 1ZC8 1ZCI
1ZDH 1ZDI 1ZDJ 1ZDK 1ZE2 1ZEV 1ZFT 1ZFV 1ZFX 1ZH5 1ZHO
1ZIF 1ZIG 1ZIH 1ZJW 1ZL3 1ZNO 1ZN1 1ZO1 1ZO3 1ZSE 1ZX7
1ZZ5 1ZZN 205D 216D 217D 219D 222D 246D 247D 248D 255D
259D 280D 283D 28SP 28SR 299D 2A04 2AOP 2A1R 2A2E 2A43
2A64 2A8V 2A9L 2A9X 2AAR 2AB4 2AD9 2ADB 2ADC 2ADT 2AGN
2AHT 2AKE 2ANN 2ANR 2AO5 2APO 2AP5 2ASB 2ATW 2AU4 2AVY
2AW4 2AW7 2AWB 2AWE 2AWQ 2AZO 2AZ2 2AZX 2B2D 2B2E 2B2G
2B3J 2B57 2B63 2B64 2B66 2B6G 2B7G 2B8R 2B8S 2B9M 2B9N
2B9O 2B9P 2BBV 2BCY 2BCZ 2BEO 2BEE 2BGG 2BH2 2BJ2 2BJ6
2BNY 2BQ5 2BSO 2BS1 2BTE 2BU1 2BX2 2BYT 2CO6 2COB 2C4Q
2C4R 2C4Y 2C4Z 2C50 2C51 2CJK 2CKY 2CSX 2CT8 2CVO 2CV1
2CV2 2CZJ 2D17 2D18 2D19 2D1A 2D1B 2D2K 2D2L 2D3O 2D6F
2DB3 2DD1 2DD2 2DD3 2DER 2DET 2DEU 2DQO 2DQP 2DQQ 2DR2
2DR5 2DR7 2DR8 2DR9 2DRA 2DRB 2DU3 2DU4 2DU5 2DU6 2DVI
2DXI 2E2H 2E2I 2E2J 2ERR 2ES5 2ESE 2ESI 2ESJ 2ET3 2ET4
2ET5 2ET8 2EUY 2EVY 2EZ6 2F4S 2F4T 2F4U 2F4V 2F4X 2F87
2F88 2F8K 2F8S 2F8T 2FCX 2FCY 2FCZ 2FDO 2FDT 2FEY 2FGP
2FK6 2FMT 2FQN 2FRL 2FTC 2FY1 2FZ2 2G1G 2G1W 2G32 2G3S
2G4B 2G5K 2G5Q 2G8F 2G8H 2G8I 2G8K 2G8U 2G8V 2G8W 2G91
2G92 2G9C 2GBH 2GCS 2GCV 2GDI 2GIC 2GIO 2GIP 2GIS 2GJE
2GJW 2GMO 2GO5 2GOZ 2GPM 2GQ4 2GQ5 2GQ6 2GQ7 2GRB 2GRW
2GTT 2GUN 2GV3 2GV4 2GY9 2GYA 2GYB 2GYC 2HOS 2HOW 2HOX
2HOZ 2H1M 2H2X 2H49 2HEM 2HGH 2HGI 2HGJ 2HGP 2HGQ 2HGR
2HGU 2HHH 2HNS 2HO6 2HO7 2HOJ 2HOK 2HOL 2HOM 2HOO 2HOP
2HT1 2HUA 2HVR 2HVS 2HVY 2HW8 2HYI 2I1C 2I2P 2I2T 2I2U
2I2V 2I2Y 2I7E 2I7Z 2I82 2I91 2IHX 2IL9 2IPY 2IRN 2IRO
2IX1 2IXY 2IXZ 2IY3 2IY5 2IZ8 2IZ9 2IZN 2JOO 2JO1 2JO2
2JO3 2JOQ 2JOS 2J28 2J37 2JA5 2JA6 2JA7 2JA8 2JEA 2LDZ
2NOK 2NOQ 2NVQ

Appendix C

Selbstständigkeitsversicherung

Versicherung an Eides statt

Hiermit versichere ich an Eides statt, die vorliegende Dissertation selbst verfasst und keine anderen als die angegebenen Hilfsmittel benutzt zu haben. Ich versichere, dass diese Dissertation nicht in einem früheren Promotionsverfahren eingereicht wurde.

_____

(Stefan Bienert)

Hamburg, den 27. November 2015