Universität Leipzig

Fakultät für Mathematik und Informatik

Institut für Informatik

SECONDARY STRUCTURE PREDICTION FOR CIRCULAR RNAS

Diplomarbeit

Leipzig, November, 19. 2007

Vorgelegt von: Ronny Lorenz

geb. am: 02.07.1979

Studiengang: Informatik - Schwerpunkt Bioinformatik

ABSTRACT

RNAs play an important role in bioinformatic applications. Their ability to serve not only as information carrier, but also to develop catalytic properties highlights them in the set of organic macromolecules notably. As these catalytic properties are closely related to the three-dimensional configuration (tertiary structure) of the RNA molecule, the formation and prediction of this tertiary structure - a process called *folding* - is a crucial bioinformatic problem. RNA folding is considered as a hierarchical process, where a secondary structure precedes the tertiary structure, whereas tertiary interactions are energetically weaker than those yielded by the secondary structure. Generally, the secondary structure does not change when tertiary interactions are formed. Because there are efficient methods for predicting the secondary structure of an RNA molecule under certain conditions, but none for the tertiary structure, the secondary structure is used as a first step for the prediction of functional properties of the RNA molecule. However, most methods for the analysis of secondary structure(s) are designed for linear RNAs exclusively. As catalytic active circular RNA molecules occur in nature too, it is necessary to extend these methods. Based on the scheme for a memory efficient extension of previously existing methods for linear RNAs, suggested by the group of Ivo Hofacker, four basic algorithms are introduced, extended and therefore made accessible for circular RNA molecules within this work.

ZUSAMMENFASSUNG

In bioinformatischen Anwendungen spielen RNA Moleküle eine grosse Rolle. Ihre Fähigkeit, nicht nur als Informationsträger für proteinkodierende Erbinformation zu dienen, sondern selbst katalytische Eigenschaften auszuprägen hebt sie aus der Menge der organischen Makromoleküle besonders hervor. Da diese katalytischen Eigenschaften sehr eng mit der dreidimensionalen Beschaffenheit (Tertiärstruktur) des RNA Moleküls verknüpft sind, ist die Ausbildung bzw. die

Vorhersage dieser Tertiärstruktur - ein Prozess der *Faltung* genannt wird - eine wichtige bioinformatische Problemstellung. RNA Faltung wird als hierarchischer Prozess betrachtet, bei dem eine Sekundärstruktur der Tertiärstruktur vorhergeht, wobei die tertiären Interaktionen energetisch schwächer sind, als die durch die Sekundärstruktur hervorgebrachten. Im Allgemeinen erfährt die Sekundärstruktur bei der Ausbildung der tertiären Interaktionen keine Änderung. Da es effiziente Methoden gibt, um die Sekundärstruktur eines RNA Moleüls unter bestimmten Bedingungen vorherzusagen, nicht jedoch für die Tertiärstruktur, wird die Sekundärstuktur als erster Schritt zur Vorhersage der funktionalen Eigenschaften des RNA Moleküls herangezogen. Die meisten Methoden zu Untersuchung der Sekundärstrutur(en) sind jedoch ausschliesslich für lineare RNAs konzipiert. Da in der Natur aber auch katalytisch aktive zirkuläre RNA Moleküle vorkommen, ist es wichtig, diese Methoden zu erweitern. Basierend auf einem Schema der speichereffizienten Erweiterung bisher existierender Methoden für lineare RNAs, vorgeschlagen von der Arbeitsgruppe um Ivo Hofacker, werden in dieser Arbeit vier grundlegende Algorithmen vorgestellt, erweitert und dadurch für die Analyse von zirkulären RNA Molekülen zugänglich gemacht.

There is a theory which states that
if ever anybody discovers exactly
what the Universe is for
and why it is here,
it will instantly disappear
and be replaced by something
even more bizarre and inexplicable.
There is another theory which states that
this has already happened.

— Douglas Adams [1]

ACKNOWLEDGMENTS

In place sincere thanks are given to all who helped me to succeed this work.

First and foremost Peter Stadler, Ivo Hofacker and Stephan Bernhart (Berni) have to be mentioned, who always were available helpfully with their open manner and remarkable scientific competence. In particular, Berni escortet me on my rocky journey through the implementation into the `ViennaRNAPackage` and if necessary helped me to get back on the right way.

Furthermore I want to thank my mother and my brother as my study and the way to its degree would not have been possible without their help.

Last but not least I also want to thank all, who provided me new perspectives and thought-provoking impulses by fruitful discussions about diverse problems: Paul-Robert Kästner, Stefan Kropf, Nico Scherf and Matthias Spitzbarth, to refer only a few of them. I also dont want to forget Claudia Copeland and Berni, who helped me a lot in improving my english within this thesis.

DANKSAGUNG

An dieser Stelle möchte ich allen Personen herzlich danken, die mir zum Gelingen dieser Arbeit verholfen haben.

Zuallererst seien hierbei Peter Stadler, Ivo Hofacker und Stephan Bernhart (Berni) erwähnt, die mir durch ihre offene Art und bemerkenswerter wissenschaftlicher Kompetenz in allen Fragen hilfreich zur Seite standen. Insbesondere hat mich Berni bei meiner felsigen Reise durch die Implementierung in das `ViennaRNAPackage` begleitet und mir wenn nötig auf den rechten Pfad zurück verholfen.

Ausserdem möchte ich mich herzlichst bei meiner Mutter und meinem Bruder bedanken, ohne die mir mein Studium und der Weg zu dessen Abschluss nicht möglich gewesen wäre.

Last but not least gilt mein Dank natürlich auch allen, die mir bei verschiedensten Problemstellungen durch fruchtbare Diskussionen neue Blickwinkel und Denkanstösse geliefert haben: Paul-Robert Kästner, Stefan Kropf, Nico Scherf und Matthias Spitzbarth, um nur ein paar wenige zu nennen. Ich möchte hierbei aber auch nicht Claudia Copeland und Berni vergessen, die mir sehr dabei geholfen haben, mein Englisch in dieser Arbeit zu verbessern.

CONTENTS

LIST OF FIGURES

ACRONYMS

MFE	minimum free energy
RNA	Ribonucleic acid
RNAs	Ribonucleic acids
ASBVd	Avocado sunblotch viroid

PSTVd Potato spindle tuber viroid

CCCVd Coconut cadang-cadang viroid

CSVd Chrysanthemum stunt viroid

HDV Hepatitis delta virus

miRNA micro RNA

rRNA ribosomal RNA

tRNA transfer RNA

snRNA small nuclear RNA

snRNAs small nuclear RNAs

snRNPs small nuclear ribonucleoproteins

mRNA messenger RNA

DNA deoxy-ribonucleic acid

Pol II RNA Polymerase II

NEP nuclear-encoded chloroplast RNA Polymerase

Fig. Figure

Def. Definition

1

INTRODUCTION

1.1 MOTIVATION

Although most RNA molecules are linear, circular single stranded RNAs occur in a few cases. A well known representative is a class of small plant pathogenic RNAs, the so called *viroids*. Another example is the circular genome of the Hepatitis delta virus (HDV) which contains a viroid-like domain. Other RNAs are also known to be circular, e.g. spliced nuclear group I introns whose ability to form circles seems to be a general property [44]. Even tRNA splicing products in archaea have been observed to be circularized [50]. Starostina et al. found circular box C/D RNAs in Pyrococcus furiosus, an archaeon [53], but circular RNAs have also been reported in eukaryotes, e.g. a circular RNA replicon in yeast [40]. Recent studies found exogenous linear RNAs circularized in wheat embryo extract [36]. Even in RNA therapeutics, circular RNAs play an essential role as shown by Umekage et al. who produced a circular streptavidin RNA aptamer in vitro [59]. Referring to circular RNAs, the `Subviral RNA Database` [49] lists more than 1300 circular viroid RNA genomes and hundreds of related objects.

Common to most of these small circular RNAs is that they do not contain any protein coding regions. Viroids for example share common specific secondary structure elements which are important for their replication and processing activity, although they do not code for any protein [20]. They do not have any encapsidation mechanism, do not require helper viruses and still are able to replicate and spread throughout an infected plant [11, 71], causing devastating diseases mediated solely by their secondary and therefore tertiary structure[4, 16, 19, 38, 55, 63]. Their replication, where viroids of the family *Pospoviroidae* or viroid-like RNA (e. g. HDV) utilize DNA-dependant RNA Polymerase II (Pol II) for an RNA-templated rolling circle RNA replication is especially unique, as Pol II is known only for its ability to act on DNA templates. This has been of great interest in the recent years [7, 21, 34, 70]. Viroids of the family *Avsunviroidae*

instead use a nuclear-encoded chloroplast RNA Polymerase (NEP) [42]. These RNAs posess efficient mechanisms for the precise cleavage of monomers from oligomeric replication intermediates [9, 10, 25]. Therefore, most viroids require a host factor but *viroid-like satellite RNAs* and one viroid (Avocado sunblotch viroid (ASBVd)) are self-cleaving RNA enzymes. Investigating viroid and viroid-like RNA secondary and tertiary structures and the role of their metastability is essential for understanding the underlying mechanisms of their pathogenesis and replication [35].

Since the RNA chain has a very large degree of freedom in bending and coiling itself, even more than polypeptides allow, RNA tertiary structures are hard to predict. Thus, known tertiary structures are experimentally determined in almost all cases. Nevertheless, the interpretation of the biochemical function of an RNA molecule can be based on a secondary structure *predicted* without the need of a direct tertiary structure prediction. Secondary structures already cover the major part of free energy of the fold by including base pairing and base stacking energies.

When predicting secondary structures of viroids, the problem arises that most of the RNA folding algorithms are designed for linear RNA molecules only. Although extensions of them exist, e. g. the minimum free energy (MFE) algorithm of Zuker and Stiegler [72] that treats linear RNAs as special cases of circular ones, implemented in the `mfold` package, they result in an enormous additional cost with respect to time and memory requirements. In 2005, Hofacker et al. [28] presented a scheme of *memory efficient folding algorithms for circular RNA secondary structures*. The application and implementation of this scheme to the existing folding algorithms provided in the *ViennaRNAPackage* [27] opens the possiblity to investigate such circular RNAs as well as linear ones according to the recent needs for *in silico* predictions [51, 52].

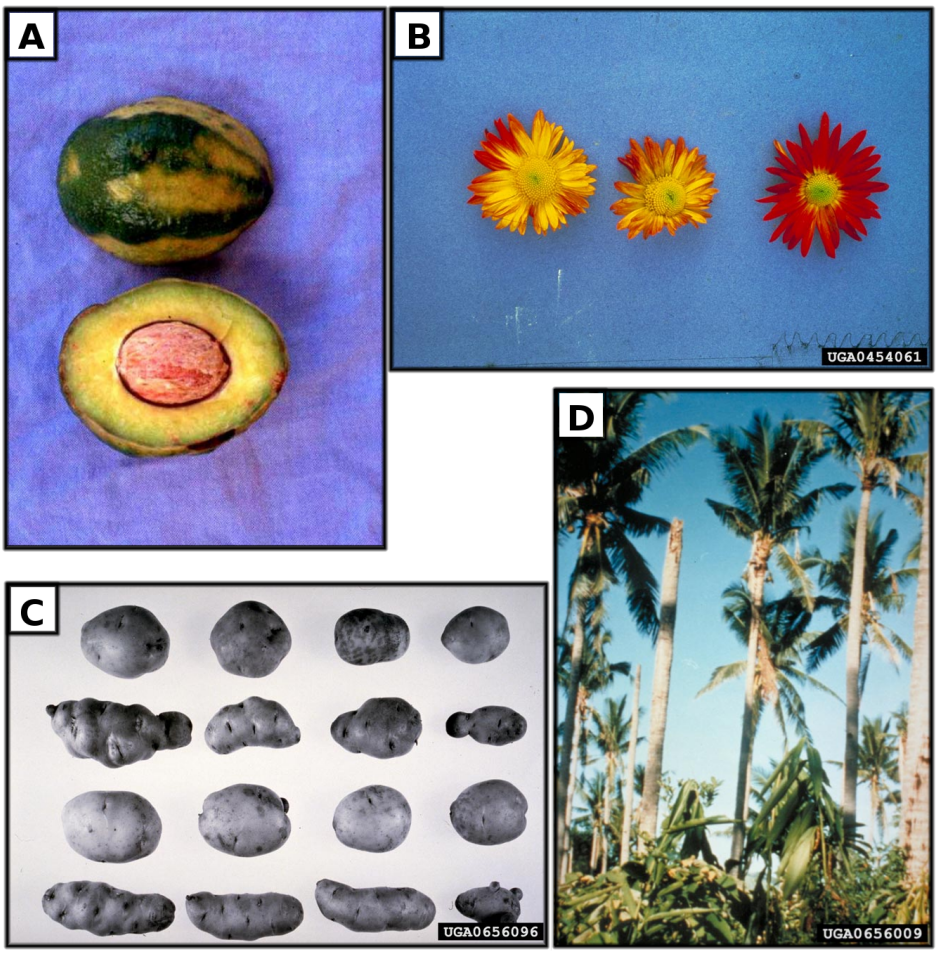

Figure 1: Viroid infections. **(A)** Symptoms of ASBVd on fruit. *(Image taken from "2006 Florida Plant Disease Management Guide: Avocado (Persea americana)", Palmateer, A.J. and Ploetz, R.C. and Harmon, P.F.)* **(B)** Symptoms of CSVd infection on flowers of chrysanthemum cv. Gillglow, showing yellow coloration of petals (on left) compared with the healthy red bloom (on right) *(Image taken by J. Dunez, France, Bugwood.org)* **(C)** PSTVd on potato; left to right: cv. Saco healthy, Saco infected, cv. Kennebec healthy, Kennebec infected. *(Image taken by USDA ARS Archive, USDA Agricultural Research Service, USA, Bugwood.org)* **(D)** Area with cadang-cadang disease (CCCVd) showing trees in the early, medium and late stages of the disease. Herbaceous monocotyledonous plants (Alpinia sp.) in foreground growing in association with infected palms. *(Image taken by M. Holderness, CAB Interational, U.K., Bugwood.org)*

2

BACKGROUND

2.1 RNA

Ribonucleic acids (RNAs) are one of the most important building blocks in living cells and also the carriers of genetic information of some non living virulant particles like viruses and viroids. RNA is a polymer chain- or ring-molecule, consisting of an arrangement of four different nucleic acids, the ribonucleotides *5′-adenylic acid*, *5′-guanylic acid*, *5′-uridylic acid* and *5′-cytidylic acid*. Each ribonucleotide monomer is composed of a ribose, a purine or pyrimidine base, and one phosphate group. Monomers lacking the phosphate group are called nucleosides, namely adenosine (A), uridine (U), cytidine (C) and guanosine (G), where the purine bases adenine (A) and guanine (G) as well as the pyrimidine bases cytosine (C) and uracil (U) are linked to the 1′ carbon atom of the ribose molecule as depicted in Fig. 2. In the polymerized chain the phosphate group interconnects the 3′ carbon of one nucleoside with the 5′ carbon of the next, forming a phosphodiester bond and giving rise to the so called *sugar-phosphate backbone* also referred just as *phosphate backbone*. Furthermore, such a polymer chain has two free ends, labeled, accordingly to the open ribose binding sites for further nucleotides, as 5′-end and 3′-end. The ring polymer, however, has no free ends due to the connection of the first 5′ and the last 3′ carbon closing the ring.

PRIMARY STRUCTURE The primary structure of an RNA molecule is the sequence of ribonucleotides, which can be written down as a sequence of letters A, G, C and U according to the succession of the different bases from the 5′- to its 3′-end. This sequence simplifies the molecule's representation in a way to allow easy exchange or storage in databases of even large primary structures. Typical sequence lengths vary from very small polymers of micro RNA (miRNA) with around 22 nucleotides up to some millions of nucleotides for large messenger RNA (mRNA) gene transcripts.

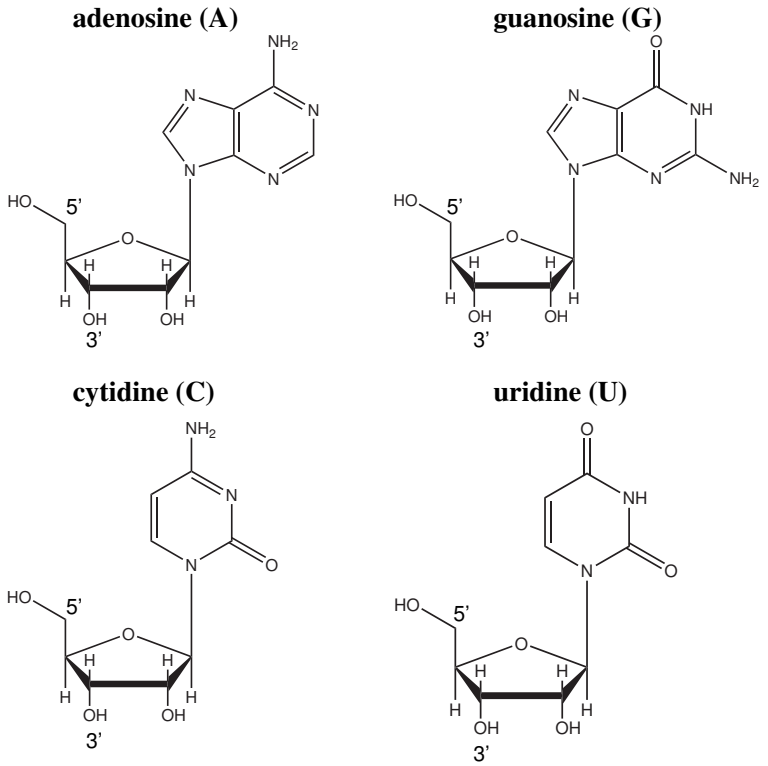

Figure 2: The nucleosides of an RNA molecule. Left ring molecule indicates the ribose with labeled 5′ and 3′ end. Upper right parts of the molecules depict the purine bases adenine (A) and guanine (G) and the pyrimidine bases cytosine (C) and uracil (U) connected to the 1′ carbon of the ribose.

SECONDARY STRUCTURE In contrast to deoxy-ribonucleic acid (DNA), which most often occures as a double stranded helical molecule of two complementary DNA strands, forming intermolecular Watson-Crick base pairs, RNA is found as a single stranded molecule in most cases. The name *Watson-Crick base pairs* dates back to the exploration of the double stranded helical structure of DNA molecules by Watson and Crick [67] in 1953, who suggested that the four different nucleotides in DNA molecules are intermolecularly linked by hydrogen bonds between a pyrimidine and a purine base. Particularly A pairs with T, giving rise to 2 hydrogen bonds and G pairs with C, giving rise to 3 hydrogen bonds. These findings were also concurrent to those of the chemist E. Chargaff in the late 1940s [5], who investigated the relative quantities of A, T, C and G in extracted (double stranded) DNA of ox, human, yeast and the avian tubercle bacilli (a bacterium). His discoveries are well known as *Chargaff's rules* today. The ability of Ribonucleic acid (RNA) to also form base pairs similar to DNA was proposed by Rich and Watson [48] in 1954, whereas in RNA thymine (T) is replaced by uracil (U). Additionally it has been shown [60] that energetically weaker so called *Wobble pairs* may occure, letting guanine (G) pair with uracil (U). This leads to the possible base pairs AU, GC and GU and their reversals for (double stranded) RNA molecules. The structural formulas for AU and GC pairs including their energetically weak hydrogen bonds are shown in Fig. 3.

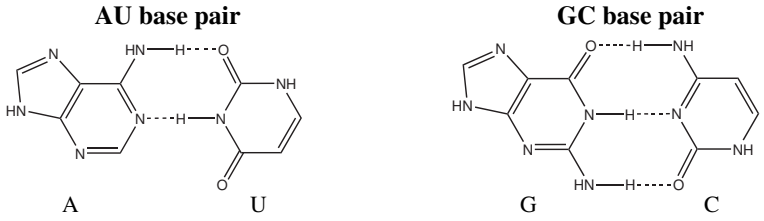

Figure 3: AU and GC base pairs. On the left side of each base pair are the purine bases adenine (A) and guanine (G). To the right hand, the pyrimidines uracil (U) and cytosine (C) are shown. Dashed lines indicate the hydrogen bonds between them.

As mentioned above, RNA most often exists as a single stranded molecule with no complement. This opens the possibility for intramolecular base pairs which occur if the molecule folds back on itself and brings complementary regions of its own sequence close to each other. The resulting two-dimesional structure,

depicted in Fig. 4 (B), is called the *secondary structure* of the RNA sequence, providing information about which bases form base pairs. It has to be noted that base triplets that arise if a base pair interacts with another single nucleotide but also knotted structures and pseudo-knots are excluded from the definition of secondary structures. They are assumed to be part of the tertiary structure as they would complicate the mathematical prediction of secondary structures too much.

TERTIARY STRUCTURE The distribution of the RNA molecule in 3-dimensional space is called *tertiary structure*. This spatial structure, which is usually determined by further intramolecular but also intermolecular forces between RNA and the solvent including its ingredients, is essential for properly fulfilling its role as information carrier or catalytically active particle.

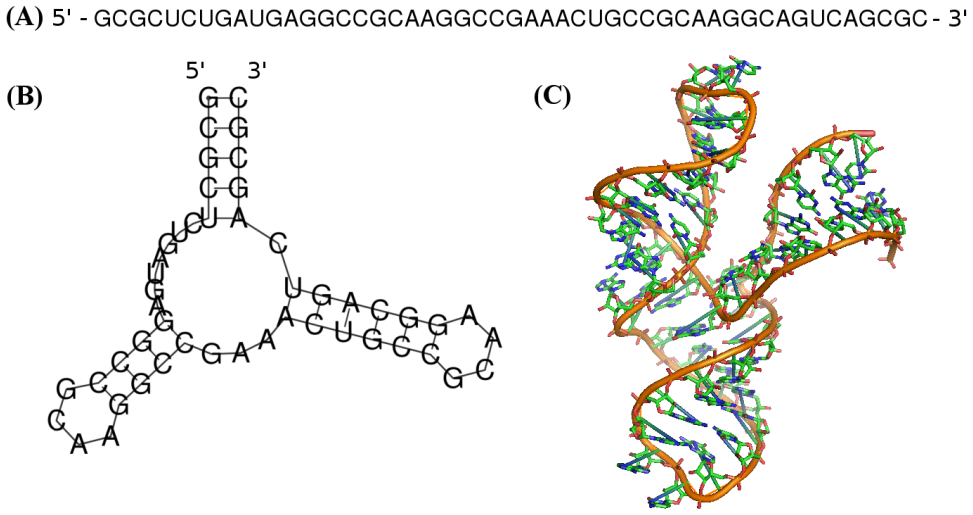

Figure 4: Structures of a 49 nucleotide RNA hammerhead ribozyme taken from [58]. **(A)** Primary structure as supplied in the article. **(B)** Secondary structure predicted with `RNAfold` of the `ViennaRNAPackage` [27] **(C)** Tertiary structure displayed by `PyMOL`, 3-D structure based on PDB-ID 1RMN

FUNCTIONS OF RNA RNAs are multifunctional. As mRNA they carry the genetic information of a polypeptide that is built in the cell from the nucleus to the cytosol, but they also act as catalytic active molecules. Even the transfer of

molecules from one place in the cell to another can be done by RNA, i.e. transfer RNA (tRNA) transfers amino acids to the ribosomal complex. An example for the catalytic activity is the reaction to form the covalent peptide bond between aminoacids in the translation of mRNA to a polypeptide in the ribosomal complex, which is catalysed by ribosomal RNA (rRNA) [45]. Additionally, RNA itself is able to catalyze RNA replication in the absence of proteins [3]. Other kinds of RNA, like small nuclear RNAs (snRNAs) that also exist as complexes of specific proteins and snRNAs, the so called small nuclear ribonucleoproteins (snRNPs), play a significant role in enzymatic reactions of RNA intron splicing, maintenance of DNA telomeres and also the regulation of transcription in the nucleus. RNA is able to catalyze a wide range of reaction types, including phosphoryl group transfer, isomerisation of C-C bond and hydrolytic reactions, thus it can function similar to protein enzymes [61]. Corresponding to protein enzymes, catalytic active RNAs are also known as *ribozymes*.

THE RNA WORLD The ability to act as enzymatic particle and also the template properties raise RNA molecules to the only known biological macromolecules which are able to function as genotype as well as phenotype. This opens the possibility for precellular evolution and Darwin's "survival of the fittest" at the RNA level in the absence of proteins or DNA and supports the idea that an *RNA World* where RNA served as carrier of genetic information and also as a catalytically active unit stood at the origin of life [2, 8, 18, 24, 32]. RNAs therefore have all prerequisites for studies of Darwinian selection at the RNA level, recently investigated by experiments confirming the quasi-species effect of subviral circular RNA plant-pathogens (viroids) [6, 23] and evolutionary effects on replicating RNA sequences [37].

2.2 RNA SECONDARY STRUCTURES

2.2.1 *Secondary Structure Graphs*

From the graph theoretic point of view, a secondary structure is a set of vertices $V = \{1, \ldots, n\}$ of numbered nucleotides, starting with 1 on the 5'-end of the RNA sequence, increasing up to n for the most 3' nucleotide on a sequence with length n and a set of edges $E = \{(i, j)\}$ denoting connected bases. A connection between two bases i and j can be a weak *hydrogen bond* between the two compatible ribonucleotide bases i and j, demarked as a *base pair* (i, j), or a strong *phosphodiester bond* ipj representing the covalent interconnection of two adjacent bases by the phosphate backbone.

Definition 2.1 *The adjacency matrix* A *with entries* $a_{i,j} = 1$ *for all edges* $(i, j) \in E$ *of a valid and feasible secondary structure graph has to fulfill the following properties:* [64, 66]

1. $a_{i,i+1} = 1$ *for* $1 \leqslant i < n$.
2. *For each fixed* i, $1 \leqslant i \leqslant n$, *there is at most one* $a_{i,j} = 1$, *where* $j \neq i \pm 1$.
3. *If* $a_{i,j} = a_{k,l} = 1$ *and* $i < k < j$ *and* $i \neq l \neq j$, *then* $i < l < j$.

As mentioned above, there are two types of bonds in a secondary structure graph. The strong *phosphodiester bond* is fixed by the RNA sequence itself and assured by condition (1) of Def. 2.1, whereas the second type is a bit more interesting by providing information about paired and unpaired bases. Therefore a definition of paired and unpaired is necessary.

Definition 2.2 *A vertex* $i \in V$

- *forms a* ***base pair*** *with vertex* $j \in V$, *if* $\exists (i, j) \in E$ *with* $|i - j| \neq 1$
- *is* ***unpaired*** *otherwise.*

Condition (2) of Def. 2.1 states that each ribonucleotide base is able to form a base pair with at most one other ribonucleotide base. And finally, condition (3) of Def. 2.1 assures that the secondary structure contains no (knots and) pseudoknots as they are assumed to be part of the tertiary structure.

STRUCTURE DECOMPOSITION

Definition 2.3 *A vertex v is interior to a base pair (i,j), if $i < v < j$ and is immediately interior to a base pair (i,j), if there is no (p,q) such as $i < p < v < q < j$. In addition, a base pair (p,q) is called (immediately) interior to a basepair (i,j) if vertex p and q are (immediately) interior to the basepair (i,j). Originating from that aspect, the $k-1$ immediately interior base pairs and the u immediately interior unpaired bases set up the so called k-loop closed by the exterior pair (i,j).[73]*

If $k = 1$, there is no immediately interior base pair (p,q) enclosed by (i,j). So the u vertices immediately interior to (i,j) form a *hairpin loop* of size u with closing pair (i,j). If $k = 2$ and $u = 0$, the enclosed base pair $(i+1, j-1)$ forms a *stacked pair* with the closing base pair (i,j). 2-loops are called a *bulge* if $u > 0$ and either $i+1$ or $j-1$ forms a base pair with another vertex v with $i+1 < v < j$ and $i < v < j-1$. If none of them form such a base pair, the 2-loop is called an *interior loop*. Every k-loop with $k > 2$ is called a *multi loop* or *multiple loop*.

Definition 2.4 *The size of a k-loop is the number u of immediately interior unpaired bases to the k-loop with closing pair (i,j). If $k > 1$ and there are no immediately interior unpaired bases, the loop has a size of zero. For 1-loops the minimal loop size m of immediately interior unpaired bases which arizes from steric effects is generally fixed to $m = 3$.*

Lemma 2.1 *Any secondary structure Ψ can be uniquely decomposed into k-loops with closing base pair (i,j) and external vertices v that are not interior to any other base pair (i,j).*

Proof. Each k-loop is uniquely characterized by its closing pair (i,j). Any vertex p, that is involved in a base pair (p,q) is part of the closing pair of an unique k-loop. Unpaired vertices instead are either immediately interior to a unique basepair and therefor part of an unique k-loop or exterior to any basepair $(i,j) \in \Psi$.

Definition 2.5 *A base pair (i,j) is called terminal if there are no base pairs (p,q) such as $p < i < j < q$. K-loops that have a closing pair which is terminal are denoted as component of Ψ.*

Lemma 2.2 *Any secondary structure Ψ can be uniquely decomposed into components and external vertices. Any k-loop is contained in a component or is a component itself.*

Since components are also k-loops and we have proven the unique decomposition of Lemma 2.1, the proof is trivial.

NOTATIONS In the following sections bases i and j that pair with each other are demarked as a base pair (i, j). Parts (subsequences) of an RNA sequence from position i to j will be marked as $[i, j]$ and are also referred to as *segments* or *intervals*.

2.2.2 *Representation*

Instead of providing a set of vertices and edges it is more useful and intuitive to provide a simplified picture for representation, especially for larger RNA secondary structures, as it was done in some figures before. There are several ways for representing RNA secondary structures.

NORMAL REPRESENTATION The first pictoral representation consists of a curved line - the phosphate backbone - which connects the equidistantly distributed nucleotide labels A,U,C and G. Furthermore, nucleotides are arranged to facilitate the connection of paired bases in the secondary structure by short segments of fixed length. (See Fig. 5)
These 2-D representations are widespread and used by biologists since the first RNA secondary structures were investigated in the 1960s [17]. Also some RNA secondary structure prediction programs, for example `RNAfold` of the `Vienna RNA Package` [27], provide such an output for their predicted secondary structures. The exclusion of knots and pseudoknots from secondary structures, done by condition (3) of Def. 2.1 ensures that each secondary structure graph is planar, i.e. it can be drawn on a plane without overlapping sections.

LINKED GRAPH REPRESENTATION A quite similar way which also directly arises from the graph theoretical view of the secondary structure is the *linked graph representation*. In this representation the vertices 1 to n are drawn vertically

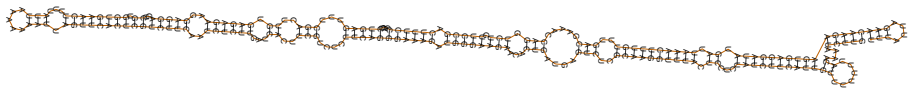

Figure 5: Normal graph representation of predicted Avocado sunblotch viroid secondary structure

on a line, connected by the edges $(i, i+1), \forall 1 \leqslant i < n$, demarking the phosphate backbone of the RNA molecule. These vertices are usually labeled by the shortcut of the nucleotide they are representing (A,U,C,G). All base pair denoting edges (i, j) with $|i - j| \neq 1$ will then be drawn as arcs from vertex i to j. Assuming that the secondary structure is valid in terms of Def. 2.1, none of these "base-pair-arcs" will intersect each other. (See Fig. 6)

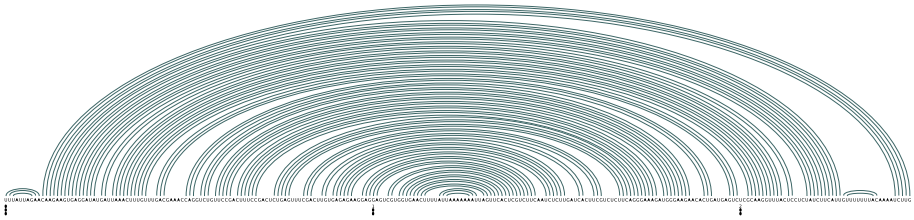

Figure 6: Linked graph representation of predicted Avocado sunblotch viroid secondary structure

CIRCULAR GRAPH REPRESENTATION Instead of drawing the phosphate backbone as a horizontal line as shown above, which is quite a bit space consuming if one wants to represent a secondary structure of a "long" RNA sequence, Ruth Nussinov suggested the usage of a a circle depicting it [47]. Apart from that, arcs were used again to represent basepairs as they do in the previous representation. (See Fig. 7)

DOT-BRACKET NOTATION The *Dot-Bracket Notation* is a string of length n. Although the name suggests the usage of *dots* and *brackets* in this string, parenthesis are used rather than brackets. A base pair (i, j) in the secondary structure is assigned to a pair of parenthesis with opening '(' at character position i and closing ')' at character position j in the string. Unpaired bases u are denoted as a '.' at character position u respectively. (See Fig. 8)

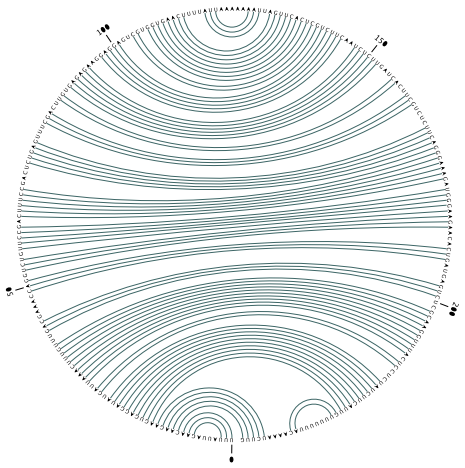

Figure 7: Circular graph representation of predicted Avocado sunblotch viroid secondary structure

```
UUUAUUAGAACAAGAAGUGAGGAUAUGAUUAA
..((((...((.....))...))))........
```

Figure 8: Dot-Bracket Notation of an artificial RNA sequence and its predicted secondary structure

The *Dot-Bracket Notation* is often taken as an easy to use, but human readable, digital interchange format of RNA secondary structures for RNA folding programs like those involved in the ViennaRNA Package [27].

MOUTAIN PLOT Paulien Hogeweg and Danielle Konings [30, 33] devised another 2-D graphical representation - the so called *mountain representation* - which leads to a quite simple method for the comparison of RNA secondary structures. One can derive such a mountain plot from a Dot-Bracket notation by identifying the characters '(', ')' and '.' with moves "up", "down" and horizontal".

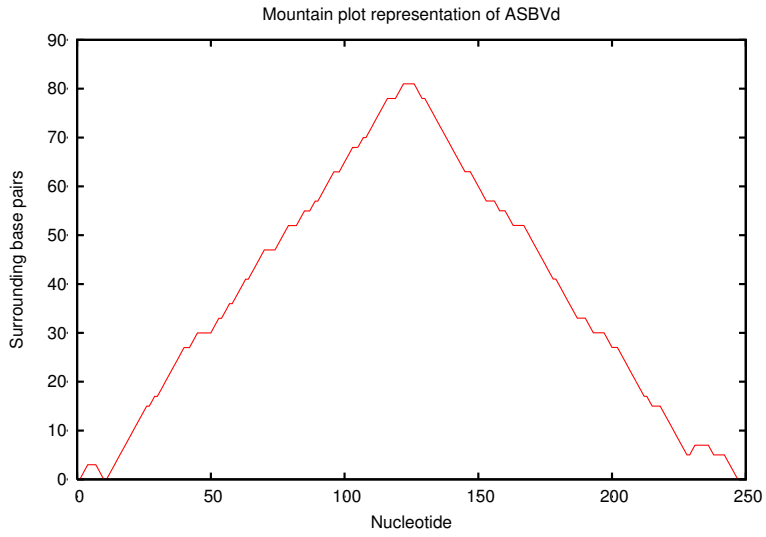

Figure 9: Mountain Plot of predicted Avocado sunblotch viroid secondary structure

While the height (y-coordinate) of the mountain plot at an arbitrary position u (x-coordinate) is the number of surrounding basepairs (i, j) with $i < u < j$, a *peak* in a *mountain representation* indicates *hairpin loops*, showing the unpaired bases as a plateau enclosed by symmetric slopes representing the stem. *Plateaus* always appear in regions which correspond to unpaired bases in the secondary structure. A plateau interrupting a sloped region is the representation of a *bulge*. If the plateau occurs paired with another plateau of the same height on the other side of the mountain it points to an *interior loop*. *Valleys* which are higher than zero represent stem enclosed intermediate unpaired bases in a *multiloop*, whereas if their height equals zero they separate the components of the secondary structure.

This representation was one inspiration for the alignment algorithm for secondary structures introduced in [33].

DOT PLOT The dot plot representation as shown in Fig. 10 consists of an upper and a lower triangle of a quadratic matrix. In both dimensions, each letter of the primary structure is assigned to a matrix index i and j, respectively. Matrix entries at position i, j are filled by black boxes indicating a base pair (i, j). Actually one triangle would be sufficient to represent an RNA secondary structure but as shown in Fig. 10 even more information can be included by using both, the upper as well as the lower triangle of the complete quadratic matrix. In the upper triangle, the size of the boxes depends on the base pairing probability where small boxes indicate low and large boxes high probability to form a base pair (i, j). The lower triangle is filled by boxes of equal size, depicting the secondary structure with minimal free energy. Such diagrams are useful for comparative analyses and their extended design which includes base pairing probabilities allows a compact machine but also human readable representation of much information.

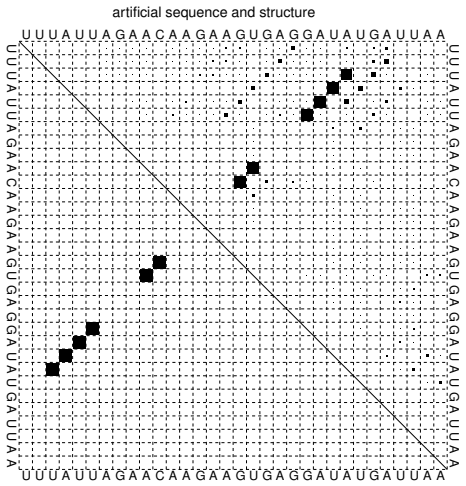

Figure 10: Dot Plot of an artificial RNA sequence and its predicted secondary structure. In the upper triangle the size of the black boxes is proportional to the square of the equilibrium base pairing probabilities. The lower triangle depicts the secondary structure with minimal free energy.

2.3 RNA FOLDING ALGORITHMS

In 1980, Nussinov and Jacobson [46] were the first to design an efficient and precise algorithm for the prediction of secondary structures. Their algorithm fills two matrices M and K with the maximum numbers of base pairs ($M_{i,j}$) which can be formed in the interval $[i, j]$ and the position of base k that pairs with j. After recursively filling the matrices the maximum number of base pairs of the folded sequence from position 1 to n is determined. By applying a trace-back routine afterwards - the so called *backward recursion* - a secondary structure with the maximum number of base pairs is deduced.

Instead of just maximizing the number of base pairs, subsequent RNA folding algorithms take energy rules into account. This allows predictions of secondary structures by regarding their thermodynamic stability. The most often used energy model and its corresponding dynamic programming approach was introduced by Zuker and Stiegler in 1981 [74]. It is explained in more detail on the following pages because all discussed folding algorithms in this work are based on this *loop-based energy model*.

LOOP-BASED ENERGY MODEL As shown in the previous section, an RNA secondary structure graph can be uniquely decomposed into loops and "external" bases. Starting from that point, the main idea behind the loop energy model is that the total free energy of an RNA secondary structure depends on the energies of these loops only. Furthermore the total free energy of a secondary structure is additively composed by the free energy of its loops. External bases are assumed to not contribute to the energy. The free energy $F(S)$ of a possible secondary structure S, can now be expressed as the sum of the contributing free energies F_L of its loops $L \in S$.

$$F(S) = \sum_{L \in S} F_L \tag{2.1}$$

So, for the calculation of the total free energy of an RNA secondary structure, the energies of the formed loops have to be evaluated. For a small set of loop types, the loop energies have been experimantally determined and are used as energy tables in secondary structure predicting programs [31, 39, 57].

LIMITATIONS Some simplifying assumptions have to be made when predicting RNA secondary structures under an energetical point of view. First the relation of the energetically most stable structure and the most likely structure in vivo has to be taken into account. By excluding influences of external forces in the surrounding environment and following Boltzmann's law, they are assumed to be similar. Another assumption is that the energy associated with an arbitrary position in the structure only depends on the local sequence and structure. This means, that the energy associated with a loop and its closing base pair (i, j) is influenced only by previously found base pairs (p, q) with $i < p < q < j$ and not by other elements. It should be noted that the latter assumption becomes a bit less restrictive when introducing energy contributions of dangling end bases.

DYNAMIC PROGRAMMING APPROACH When predicting secondary structures with certain properties, like *minimal free energy* or *suboptimality*, algorithms that find all possible loops L and their associated free energies F_L have to be investigated.

In 1981, Zuker and Stiegler introduced a dynamic programming algorithm to compute the optimal secondary structure according to its free energy for a sequence of n nucleotides in time proportional to $\mathcal{O}(n^3)$ [74]. Their work was based on the dynamic programming algorithm for the computation of the maximum base pairing of a folded RNA molecule formulated by Nussinov et al. [47] and the work of Waterman and Smith [66] whose algorithm takes stacking and destabilizing energies into account.

The idea behind the algorithm of Zuker and Stiegler is to calculate for all possible pairs of bases i and j satisfying $i < j$ two energies $W(i, j)$ and $V(i, j)$. $W(i, j)$ represents the minimum free energy of all possible structures formed on the subsequence $[i, j]$ while $V(i, j)$ is the minimum free energy of all possible structures on the subsequence $[i, j]$ with the requirement that i forms a base pair (i, j) with j. The determiniation of the energy contribution $V(i, j)$ is subdivided into the evaluation of three possible loop-degree-dependant energies E_1, E_2 and E_3, taking $1-\text{loops}$, $2-\text{loops}$ and $k-\text{loops}$ with $k > 2$ into account. Their ansatz to predict a secondary structure with a minimum free energy leads to a minimization of the additive energies of possible substructures which are recursively computed from the smallest subsequences $[i, j]$ up to the complete sequence $[1, n]$.

$$
\begin{aligned}
V(i,j) &= \min\{E_1, E_2, E_3\} \\
E_1 &= \mathcal{H}(i,j) \\
E_2 &= \min_{i<i'<j'<j} \left\{\mathcal{I}(i,j;i',j') + V(i',j')\right\} \\
E_3 &= \min_{i+1<i'<j-2} \left\{W(i+1,i') + W(i'+1,j-1)\right\} \\
W(i,j) &= \min\{W(i+1,j), W(i,j-1), V(i,j), E_4\} \\
E_4 &= \min_{i<i'<j-1} \left\{W(i,i') + W(i'+1,j)\right\}
\end{aligned} \tag{2.2}
$$

The first line of (2.2) shows the evaluation of the minimal free energy of the three loop types, *hairpin loops* (E_1), *interior loops*, *bulges* or *stacked pairs* (E_2) and *multiple loops* (E_3).

E_1 as shown in the second line is the tabulated free energy of a *hairpin loop* with closing pair (i,j) and in practice it depends not only on the type of the closing pair, but also on the loop length l indicating the number of unpaired bases $l = j-i-1$. Furthermore, the sequence of a loop may play a role as a destabilizing factor, too. But experimentally determined sequence dependent energy contributions are rare and therefore used in the computation of loops of limited size like *tri-* and *tetra-loops* only. Another property influencing the stability is the *mismatch* energy contribution which adds a further destabilizing factor based on the adjacent nucleotides $i+1$ and $j-1$ of the closing pair (i,j).

The energy table $\mathcal{H}$ used in (2.2) depends on the size of the loop and the type of its closing pair. But in practice, it is limited in the loop size dimension on the one hand due to the limitation of experiments, which can not be performed for the extremely large amount of possible combinations of closing pairs (i,j) and loop sizes l and on the other hand due to the amount of memory that has to be used to store these energy contributions. A work-arround for this situation was introduced in 1988 by Turner et al. [57]. They extrapolated energy values for large hairpin loops with $l > 30$ logarithmically by

$$
\mathcal{H}(i,j,l) = \mathcal{H}(i,j,30) + r \cdot \log(l/30) \tag{2.3}
$$

where r is a constant.

E_2 comprises the free energy of each possible $2-\text{loop}$ with closing pair (i,j) and interior pair (i',j'). This energy is determined by adding the free energy of the structure enclosed by (i',j') to the free energy which arises due to the formation of the $2-\text{loop}$. The latter one, $\mathcal{I}(i,j;i',j')$, mainly depends on the closing pairs and the loop size l and is also tabulated as a set of experimentally determined or mathematically extrapolated energy parameters. Furthermore the size l has to be decomposed into two separate loop lengths $u = i' - i - 1$ and $v = j - j' - 1$ constituting the total length $l = u + v$ to take the asymmetry of a $2-\text{loop}$ into account.

When examining $\mathcal{I}(i,j,i',j')$ the evaluation of the energy contribution for all (i',j') with $i < i' < j' < j$ would let the algorithm's time complexity grow proportional to $\mathcal{O}(n^4)$, so an additional limitation was introduced. It is biologically reasonable that $2-\text{loops}$ are seldom very large in the number of unpaired immediately interior bases. Therefore their loop size can be constrained to be less than some fixed number [74][73]. With this contraint, the dominant term in the complete algorithm becomes the search for multi loops which takes cubic time.

Multi loop energy contribution is shown in the fourth line. Clearly, their simple algorithm does not explicitly handle multi loops as separate loop types but as compositions of $1-$ or $2-\text{loops}$, where the closing pair (i,j) does not contribute any additional free energy. All $k-\text{loops}$ with $k > 2$, which are called *bifurcation loops* in their article, are split into two separate substructures until primitive $k-\text{loops}$ with $k \leqslant 2$ appear.

The different loop types examined with closing pair (i,j) that contribute their free energy in $V(i,j)$ are shown in Fig. 11.

The last lines of (2.2) show the structure decomposition into substructures and their related energy contribution. A graphical description of Zuker's structure decomposition constituting the energy values in $W(i,j)$ is shown in Fig. (12) As mentioned earlier, the algorithm starts with the evaluation of $W(i,j)$ for the smallest subsequences $[i,j]$. These subsequences are normally pentanucleotides according to the minimal loop length of 3 unpaired bases and the 2 bases constituting the base pair. The algorithm then increases the length of these subsequences successively until the complete sequence $[1,n]$ is processed and the upper triangles of the matrices W and V are filled. Afterwards, the minimal free energy of the folding of the complete sequence can be obtained at $W(1,n)$ and a trace-back

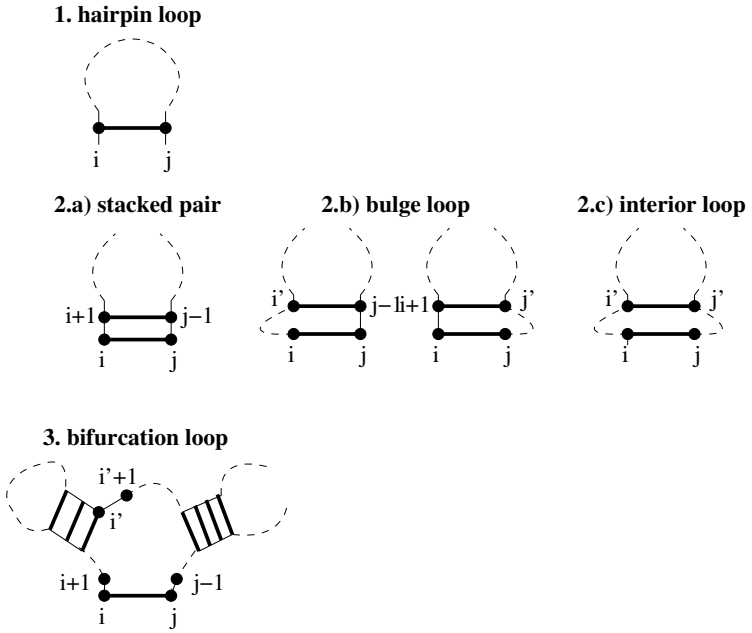

Figure 11: Loop types in Zuker's MFE algorithm

routine similar to the one in [46] is applied to reconstruct the base pair pattern of the appropriate secondary structure. The trace-back routine, which is also called *backtracing* or *backtracking*, will be discussed in detail in 2.3.1.

MULTI LOOP DECOMPOSITION Considering the treatment of multi loops in (2.2) that actually is a simple partition into two substructures, it is evident that multi loops are not really treated as separate loop types in this ansatz. A possibility to take the multi loop energy contributions into account would be to experimentally determine their free energies and to construct a multidimensional energy table as it is done for $1-$ and $2-$loops. But this results in an enormous and practically unmanageable amount of experiments to be done, as each further stem in the multiloop squares the number of experiments and therefore also increases the dimension of the energy table by 2.

The lack of experimentally measured parameters for the computation of multi loop contributions is often bypassed with a linear additive ansatz that only depends on the destabilizing energy constant c of unpaired bases, the degree (number of branches) δ of the multiloop and an energy contribution a for closing

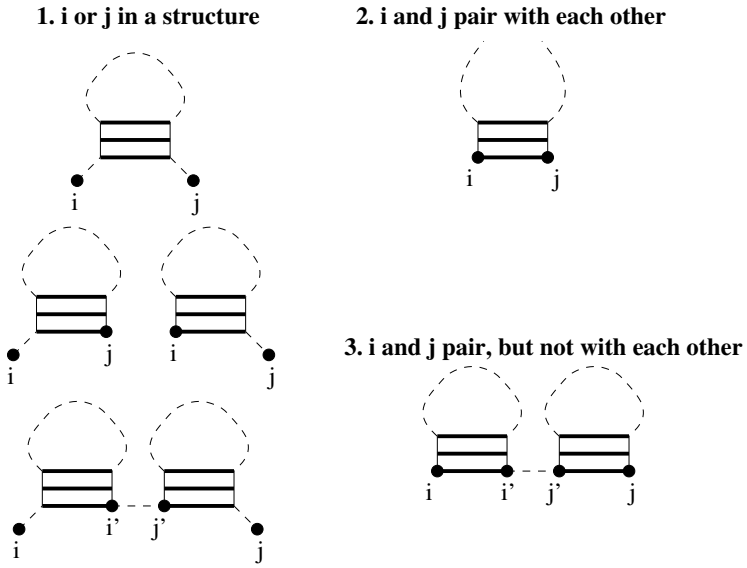

Figure 12: Structure decomposition in Zuker's MFE algorithm

the complete loop.

$$\mathcal{M} \quad = \quad a + b \cdot \delta + c \cdot l \tag{2.4}$$

The constant parameter b is the energy contribution for each branch and l represents the length of the multi loop, thus the number of unpaired bases. This linear ansatz allows fast prediction of multi loop energy contributions and produces good results using $a = 4.6$, $b = 0.4$ and $c = 0.1\text{kcal/mol}$ as shown by Jaeger et al. [31].

RECURSION SCHEME Based on the previously described algorithm a new recursion scheme can be constructed, taking the linear ansatz for multi loops into account. Additionally, the multi loop decomposition which has to be altered anyway will be constructed to uniquely decompose multi loops into rightmost parts with exactly one stem and a left part with at least one stem. This ensures that each structure is counted exactly once in the recursions and thus allows the computation of the *partition function* as described in a later section. It is also necessary for *suboptimal folding* and the *memory efficient folding of circular RNAs*.

First, the last two lines of (2.2), where the best possible energy of the subsequence

$[i,j]$ is determined by scanning for adjacent unpaired bases and components, are taken to formulate recursion (I). An energy array F that is similar to W in (2.2) can now be filled by

$$F_{i,j} = \min\left\{F_{i+1,j}, \min_{i<k\leqslant j} C_{i,k} + F_{k+1,j}\right\} \tag{2.5}$$

The introduced energy array C used in the recursion above is equivalent to the energy array V used by Zuker and Stiegler. It stores the best possible energy contribution on the subsequence $[i,j]$ under the constraint, that i pairs with j. Except for the third part, the multiloop decomposition, recursion (II) equals the computation of $V(i,j)$ in (2.2).

$$\begin{aligned} C_{i,j} = \min\{&\mathcal{H}(i,j),\\ &\min_{i<d<e<j} \mathcal{I}(i,j,d,e) + C_{d,e},\\ &\min_{i+1<u<j-1} M_{i+1,u} + M^1_{u+1,j-1} + a\Big\} \end{aligned} \tag{2.6}$$

Here, the last line shows the decomposition of multi loops into a rightmost part with exactly one stem, storing its energy contribution into an auxilary energy array M^1, and the left multiloop part with at least one stem, using the auxilary energy array M. The energy contribution a which is obtained by closing the multiloop with base pair (i,j) is also taken into account.

The two additional arrays M and M^1 have no equivalent in (2.2) because they originate from the linear ansatz of the multi loop decomposition in (2.4). Therefore two additional recursion (III) and (IV) are used to recursively compute the array elements.

$$\begin{aligned} M_{i,j} = \min\{&M_{i,j-1} + c,\\ &\min_{i<u<j-1} C_{u+1,j} + b + (u-i-1)\cdot c,\\ &\min_{i<u<j-1} M_{i,u} + C_{u+1,j} + b\Big\} \end{aligned} \tag{2.7}$$

$$M^1_{i,j} = \min\{M_{i,j-1} + c, C_{i,j} + b)\} \tag{2.8}$$

After filling all arrays, the minimal free energy of a secondary structure of an RNA sequence can be obtained at $F_{1,n}$

The recursion scheme introduced is graphically represented in Fig. 13 and can be applied not only to the computation of the MFE as it was done in (2.5) - (2.8), but also to other RNA folding algorithms, e.g. the computation of the partition function or the prediction of suboptimal secondary structures.

DANGLING ENDS Additional stabilizing energies which arise if an unpaired nucleotide stacks with an adjacent base pair [62] are called *dangling-end contributions*. When calculating the free energy of a loop, three possibilities for regarding dangling-end contributions exist. Firstly they may be ignored and therefore do not play a role in the algorithm. The second way is to take them into account for every combination of adjacent bases and base pairs. And thirdly, a more complex energy model can be applied letting unpaired bases stack with at most one base pair. The `Vienna RNA Package` implements each of these three models plus an additional energy contribution for coaxial stacking of helices. In the following secondary structure predicting algorithms loop energy contributions are configurated with the second case, taking each combination of unpaired bases and their adjacent base pair into account. Dangling-end contributions where the adjacent unpaired base stacks onto the closing pair of a *hairpin*, *bulge*, *stacked pair* or *interior loop* are assumed to be already included in the appropriate energy table $\mathcal{H}$ and $\mathcal{I}$. All other cases are covered by the parameters $d^5_{i,j,i-1}$ and $d^3_{i,j,j+1}$ according to the location of the unpaired base at the $5'$ or $3'$ direction of the base pair. The first two subscripted sequence positions demark the base pair (i, j), the last position is the unpaired base $i-1$ or $j+1$ which stacks.

2.3.1 *Minimum Free Energy*

The prediction of a secondary structure with minimal free energy based on the recursion scheme in Fig. 13 is divided into two steps again. First, the minimal free energy is determined in the *forward recursion* and afterwards the *backward recursion* reconstructs the base pairing pattern. Taking the linear ansatz for the multiloop decomposition and stabilizing dangling-end contributions into account, the MFE algorithm can be formulated thus:

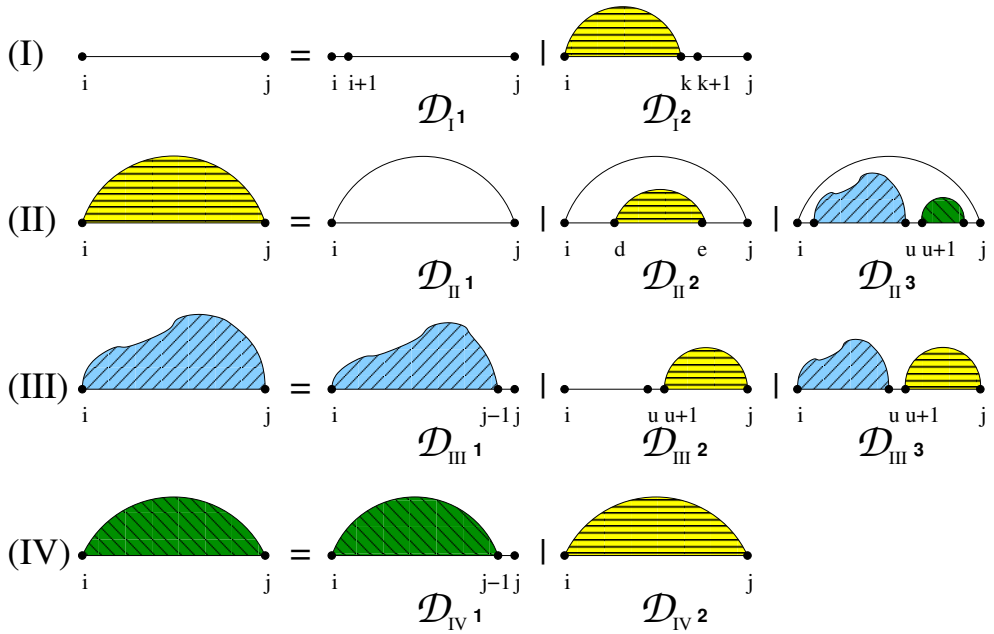

Figure 13: Common recursion scheme for RNA folding

(I) denotes either $F_{i,j}$ in the MFE case or $Q_{i,j}$ when applied to the partition function algorithm. $\mathcal{D}_{I^1}$ and $\mathcal{D}_{I^2}$ are possible decompositions in step (I).

(II) is replaced by either C or Q^B in the MFE case and the partition function case, respectively. The decompositions $\mathcal{D}_{II^x}$ denote the possible decompositions into a *hairpin* ($x = 1$), into an *interior loop* ($x = 2$) or into a *multi loop* structure ($x = 3$).

(III) is subsituted by M in the MFE case or by Q^M when the scheme is applied to the partition function. Decompositions $\mathcal{D}_{III^x}$ denote nibbling of the most 3′ base ($x = 1$), separation into unpaired bases and exactly one stem ($x = 2$) or separation into a left part with at least one stem and a right part with exactly one stem ($x = 3$).

(IV) is either M^1 if the scheme is applied to the MFE algorithm or Q^{M^1} in the partition function case. The two possible decompositions $\mathcal{D}_{IV^1}$ and $\mathcal{D}_{IV^2}$ display nibbling of the most 3′ base or regarding (i, j) as the closing pair of one stem of the multi loop.

FORWARD RECURSION The only additions to (2.5) - (2.8) are the contributions d^5 and d^3 that have to be added when decomposing multi loops or structure segments.

$$
\begin{aligned}
F_{i,j} &= \min\left\{F_{i+1,j}, \min_{i<u\leqslant j} C_{i,u} + d^5_{i,u,i-1} + d^3_{i,u,u+1} + F_{u+1,j}\right\} \\
C_{i,j} &= \min\Bigg\{\mathcal{H}(i,j) \\
&\qquad \min_{i<p<q<j}\{C_{p,q} + \mathcal{I}(i,j;p,q)\}, \\
&\qquad \min_{i<u<j}\left\{M_{i+1,j-1} + M^1_{u+1,j-1} + a + d^5_{j,i,j-1} + d^3_{j,i,i+1}\right\}\Bigg\} \\
M_{i,j} &= \min\{ \\
&\qquad \min_{i<u<j}\left\{(u-i-1)\cdot c + C_{u+1,j} + d^5_{u+1,j,u} + d^3_{u+1,j,j+1} + b\right\}, \\
&\qquad \min_{i<u<j}\left\{M_{i,u} + C_{u+1,j} + d^5_{u+1,j,u} + d^3_{u+1,j,j+1} + b\right\}, \\
&\qquad M_{i,j-1} + c\} \\
M^1_{i,j} &= \min\left\{M^1_{i,j-1} + c, C_{i,j} + d^5_{i,j,i-1} + d^3_{i,j,j+1} + b\right\}
\end{aligned}
\tag{2.9}
$$

BACKWARD RECURSION In contrast to the forward recursion of the MFE algorithm, which starts with the calculation for pentanucleotide sequence segments and ends with the complete sequence $[1,n]$, the backtracking procedure starts with the complete sequence and decreases the sequence length to smaller segments. In detail, it finds all *immediately interior* base pairs for any segment $[i,j]$ which has to be evaluated. *Immediately interior* base pairs found constitute new segments to evaluate further. This strategy is also known as *depth first search*. It ends if there is no segment left for evaluation and the resulting set of base pairs represents the secondary structure with minimum free energy.

Starting with segment $[1,n]_E$ which has been folded to a structure with minimum free energy $F_{1,n}$ and therefore belongs to the F energy array, the backtracking procedure recursively decomposes each segment into subsegments. This decomposition is always done according to the energy array $E = \{F, C, M, M^1\}$ the segment belongs to.

If the segment belongs to F a reversed version of the first equation of algorithm 2.9 is used to decide which segments have to be evaluated further. Base i is

unpaired on the segment $[i,j]_F$ if

$$F_{i,j} = F_{i+1,j} \tag{2.10}$$

The resulting segment for further evaluation in the next recursion step is $[i+1,j]_F$. The other possibility is, that there exists a base u, forming a basepair (i,u) and fulfilling the criterion

$$F_{i,j} = C_{i,u} + d^5_{i,u,i-1} + d^3_{i,u,j+1} + F_{u+1,j}. \tag{2.11}$$

In that case, the resulting segments $[i,u]_C$ and $[u+1,j]_F$ and their appropriate energy arrays C and F are used in the next recursion step.

For backtracking in C, the algorithm decides whether the base pair (i,j) is the closing pair of a hairpin loop

$$C_{i,j} = \mathcal{H}(i,j) \tag{2.12}$$

In this case backtracking on segment $[i,j]_C$ terminates. On the other hand, if the base pair (i,j) is a closing pair of a 2-loop (interior loop, stack or bulge) with interior base pair (p,q) it fulfills

$$C_{i,j} = C_{p,q} + \mathcal{I}(i,j;p,q) \tag{2.13}$$

for some p and q with $i < p < q < j$ and the subsegment $[p,q]_C$ left is processed further in the next backtracking step.

The third loop type where (i,j) may be closing pair of is a multi loop. A position u with $i < u < j$, fulfilling

$$C_{i,j} = M_{i+1,u} + M^1_{u+1,j-1} + a + d^5_{j,i,j-1} + d^3_{j,i,i+1}. \tag{2.14}$$

has to be determined in this case. The resulting multi loop parts $[i+1,u]_M$ with at least one further stem and $[u+1,j-1]_{M^1}$ with exaclty one stem then have to be backtracked further.

When backtracking in M, three cases have to be distinguished, too. Firstly, the $3'$ base j is left unpaired if

$$M_{i,j} = M_{i,j-1} + c \tag{2.15}$$

In that case j will be nibbled and the algorithm evaluates segment $[i, j-1]_M$ next.

In the second case, a position u separating the one and only *immediately interior* base pair $(u+1, j)$ from a segment of unpaired bases $[i, u]$ is searched. The obtained segment $[i, u]$ will not be evaluated further as this segment contains no base pairs. Segment $[u+1, j]_C$ on the other hand is used in the next recursion step.

$$\begin{aligned} M_{i,j} &= (u-i-1)\cdot c + b + C_{u+1,j} \\ &\quad + d^5_{u+1,j,u} + d^3_{u+1.j,j+1} \end{aligned} \tag{2.16}$$

The third and last case of backtracking in M holds, if

$$\begin{aligned} M_{i,j} &= M_{i,u} + b + C_{u+1,j} \\ &\quad + d^5_{u+1,j,u} + d^3_{u+1,j,j+1} \end{aligned} \tag{2.17}$$

This is the case, if $[i, j]_M$ encloses a multi loop part with more than one internal stem. For this evaluation, the algorithm searches for the position u separating segment $[u+1, j]_C$ that represents exactly one stem of the multi loop from the other segment $[i, u]$ that contains at least one further branch.

Finally, backtracking in M^1 leads to the case discrimination, whether i forms a base pair with j, constituting a stem of the surrounding multi loop

$$\begin{aligned} M^1_{i,j} &= b + C_{i,j} \\ &\quad + d^5_{i,j,i-1} + d^3_{i,j,j+1} \end{aligned} \tag{2.18}$$

or if j is unpaired and hence must be nibbled.

$$M^1_{i,j} = M^1_{i,j-1} + c. \tag{2.19}$$

The algorithm proceeds by backtracking in the appropriate energy arrays of segments $[i, j]_C$ or $[i, j-1]_{M^1}$, respectively.

After processing all smallest subsequences the algorithm terminates and the base pairs found set up the secondary structure with minimal free energy.

2.3.2 *Partition Function*

In many bioinformatics applications, one is not only interested in the MFE and a related secondary structure, but in the probability of the occurance of a secondary structure s_i in the whole secondary structure space S spanned by the RNA sequence, or in the probability $P_{i,j}$ of the occurance of a single basepair (i, j). To compute such probabilities one needs to have information about the set of all possible secondary structures S. This is collected in the partition function Q of an RNA sequence, where $F(s)$ denotes the free energy of structure $s \in S$, R is the gas constant and T the absolute temperature:

$$Q = \sum_{s \in S} e^{\frac{-F(s)}{RT}} \tag{2.20}$$

The Boltzmann weight $E = e^{\frac{-F(s)}{RT}}$ corresponding to each energy contribution $F(s)$ is the additive term. The partition function also provides the ability to study melting kinetics of RNA molecules, because the Boltzmann weight introduces a temperature dependant energy contribution.

It has been shown that the number of possible secondary structures and thereby the number of summands in Q of an RNA sequence grows exponentially [64] with increasing sequence length n. This fact seems to make the computation of Q impossible in polynomial time. However, McCaskill [41] introduced a dynamic programming algorithm to compute the equilibrium partition function Q and also the basepairing probabilities $P_{i,j}$ of a given RNA molecule in time proportional to $\mathcal{O}(n^3)$.

EQUILIBRIUM PARTITION FUNCTION The computation of the partition function is very similar to the MFE algorithm. The main difference is, that all minima in the MFE algorithm are replaced by sums because each possible secondary structure reflects a contributing part of the partition function. Furthermore, the additivity of free energies implies multiplicativity of the Boltzmann weighted contributions to the partition function.

Introducing a factor $\beta = \frac{1}{RT}$ for simplifying the equations, the recursive algorithm

can be applied formal to the common recursion scheme as follows:

$$
\begin{aligned}
Q_{i,j} &= 1 + \sum_{1<k<j} Q^{B}_{i,k} \cdot Q_{k+1,j} \cdot e^{-\beta\cdot(d^{5}_{i,j,i-1}+d^{3}_{i,j,j+1})} \\
Q^{B}_{i,j} &= e^{-\beta\cdot\mathcal{H}(i,j)} \\
&\quad + \sum_{i<d<e<j} e^{-\beta\cdot\mathcal{I}(i,j;d,e)} \cdot Q^{B}_{d,e} \\
&\quad + \sum_{i<u<j} Q^{M}_{i+1,u} \cdot Q^{M^1}_{u+1,j-1} \cdot e^{-\beta\cdot a} \cdot e^{-\beta\cdot(d^{5}_{i,j,j-1}+d^{3}_{i,j,i+1})} \\
Q^{M}_{i,j} &= Q^{M}_{i,j-1} \cdot e^{-\beta\cdot c} \\
&\quad + \sum_{i<u<j} e^{-\beta\cdot(u-i-1)\cdot c} \cdot Q^{B}_{u+1,j} \cdot e^{-\beta\cdot b} \cdot e^{-\beta\cdot(d^{5}_{u+1,j,u}+d^{3}_{u+1,j,j+1})} \\
&\quad + \sum_{i<u<j} Q^{M}_{i,u} \cdot Q^{B}_{u+1,j} \cdot e^{-\beta\cdot b} \cdot e^{-\beta\cdot(d^{5}_{u+1,j,u}+d^{3}_{u+1,j,j+1})} \\
Q^{M^1}_{i,j} &= Q^{M^1}_{i,j-1} \cdot e^{-\beta\cdot c} + Q^{B}_{i,j} \cdot e^{-\beta\cdot b} \cdot e^{-\beta\cdot(d^{5}_{i,j,i-1}+d^{3}_{i,j,j+1})}
\end{aligned}
\tag{2.21}
$$

Following the recursion scheme introduced in Fig. 13 on page 25, the algorithm fills the upper triangles of four matrices, labeled by Q, Q^{B}, Q^{M} and Q^{M^1}. Corresponding to the energy array F of the MFE algorithm, entries $Q_{i,j} \in Q$ store the partition function of the subsequence $[i,j]$. Q^{B} is the equivalent array to C of the MFE algorithm, filled with the partition function of subsequences $[i,j]$ under the constraint that i and j form a basepair (i,j), while Q^{M} and Q^{M^1} correspond to M and M^1 with an equivalent meaning. By summation of the Boltzmann weighted energies, the algorithm collects the energy contribution of the complete secondary structure ensemble of a particular sequence, beginning with short pentanucleotides and increasing the subsequence length until the partition function $\mathcal{Q} = Q_{1,n}$ of the complete sequence $[1,n]$ has been computed.

BASE PAIRING PROBABILITIES After the computation of the partition function $\mathcal{Q}$ and filling of the appropriate energy arrays Q, Q^{B}, Q^{M} and Q^{M^1}, one is able to calculate the equilibrium probability $P_{i,j}$ of the occurance of each possible base pair (i,j) in the secondary structure ensemble. This is quite simple if (i,j) is not interior to any base pair (p,q), thus $\nexists(p,q) : p < i < j < q$, and can be expressed by

$$
P^{lin}_{i,j} = \frac{Q_{1,i-1} \cdot Q^{B}_{i,j} \cdot Q_{j+1,n}}{Q_{1,n}} \tag{2.22}
$$

This equation also indicates, what is meant by the base pairing probability. It is the proportion of equilibrium weighted structures where the base pair (i,j) occurs compared to the whole secondary structure space.

The computation of $P_{i,j}$ gets more difficult if (i,j) is surrounded by at least one base pair (p,q). This implies that all possible surrounding base pairs and the corresponding loop types where (i,j) becomes part of have to be taken into accout. This additional factor will be added to (2.22) and leads to the common recursive formula [41] for the calculation of $P_{i,j}$ formulated in (2.23).

$$
\begin{aligned}
P_{i,j} \;=\; & P^{\text{lin}}_{i,j} \cdot e^{-\beta\cdot(d^5_{i,j,i-1}+d^3_{i,j,j+1})} \\
& + \sum_{p<i<j<q} P_{p,q} \cdot \frac{Q^B_{i,j}}{Q^B_{p,q}} \cdot e^{-\beta\cdot\mathcal{J}(p,q,i,j)} \\
& + \sum_{p<i} Q_{i,j} \cdot e^{-\beta\cdot(a+b)} \cdot \Bigg\{ \\
& \quad \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot e^{-\beta\cdot((q-j-1)\cdot c+d^5_{q,p,q-1}+d^3_{q,p,p+1})} \cdot Q^M_{p+1,i-1} \\
& \quad + \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot Q^M_{j+1,q-1} \cdot e^{-\beta\cdot((i-p-1)\cdot c+d^5_{q,p,q-1}+d^3_{q,p,p+1})} \\
& \quad + \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot Q^M_{j+1,q-1} \cdot Q^M_{p+1,i-1} \cdot e^{-\beta\cdot(d^5_{q,p,q-1}+d^3_{q,p,p+1})} \Bigg\}
\end{aligned}
\tag{2.23}
$$

The first line of (2.23) covers the contribution of (2.22). The second line shows the probability contributed by all interior loops containing (i,j) by summing over all possible enclosing base pairs (p,q). The last lines take the possibility that (i,j) is part of a multi loop into account and are more complicated. In detail, they cover the cases in which (i,j) delimits the most $3'$, the most $5'$ or an intermediate branch of the multiloop, respectively. On the first view, it seems to be that the overall time complexity to calculate $P_{i,j}$ grows proportinal to $\mathcal{O}(n^4)$.

But if introducing two auxiliary linear probability arrays P^M and P^{M^1} with

$$P^M_{p,j} = \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot Q^M_{j+1,q-1} \cdot e^{-\beta \cdot (d^5_{p,q,p-1} + d^3_{p,q,q+1})} \tag{2.24}$$

$$P^{M^1}_{p,j} = \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot e^{-\beta \cdot ((q-j-1) \cdot c + d^5_{p,q,p-1} + d^3_{p,q,q+1})} \tag{2.25}$$

and by filling these linear arrays at the right point in the recursions - when $P_{p,q}$ is calculated - the time complexity can be reduced to $\mathcal{O}(n^3)$.

2.3.3 *Suboptimal secondary structures*

Although the previously described algorithm for computing the Minimum Free Energy is widely used for prediction of RNA secondary structures, it has been emphasized that it does not really predict the real situation in vivo for two major reasons [72]. The first reason is that all experimentally measured energy parameters used for computation are error-prone and therefore unavoidably imprecise. Thus, the secondary structure associated with the true MFE could be a suboptimal structure within the used energy parameter set and vice versa. Biochemical constraints may alter relative energies within the molecule and are also not taken into account while computing the MFE with energy parameters derived from artificial conditions. The second point which has to be mentioned is that under physiological conditions, an RNA molecule often exists in an equilibrium of alternative states whose energy differences are small, instead of only one particular secondary structure. Hence, for some scientific problems it is a good idea to not only determine one single secondary structure which represents the MFE with respect to the used parameter set, but to determine a set of suboptimal structures within a given energy range arround the MFE.

Different approaches exist for computing suboptimal foldings of an RNA sequence. A widely used algorithm was presented by Zuker in 1989, which is based on an extension of his algorithm for folding of circular RNAs [72]. For each valid base pair within a given RNA sequence it generates the energetically best structure containing that base pair. However, this approach fails to compute all possible suboptimal secondary structures due to the usage of a base pair constraint. In detail, the algorithm will always produce the MFE structure, if the

constrained base pair is included in the MFE structure. Therefore no suboptimal structures which differ in one or more base pairs from the MFE can be generated. Another disadvantage of this constraining base pair is the limitation of the suboptimal structure space to a size of $\frac{n\cdot(n-1)}{2}$ with sequence length n, whereas the number of possible secondary structures grows exponentially with sequence length n [64]. A way out of this limitation was introduced in 1999 by Wuchty et al. [69]. They presented an algorithm to compute the complete suboptimal folding of an RNA sequence, generating structures between the MFE and an arbitrary upper limit. This approach constitutes an extensible basis for the computation of suboptimal structures for circular RNA molecules in 4.2.1.

Complete suboptimal folding

The main idea behind the algorithm of Wuchty et al. is to use the *forward recursion* of the conventional MFE algorithm for filling the energy arrays and to modify the backtracking procedure inspired by the Waterman-Byers scheme [65]. In contrast to the backtracking criterions of equation 2.10 to 2.19 introduced on page 27, they replaced the strict equality conditions by inequalities of the form

$$E_{i,j} + E_{L_S} + \sum_{k,l} E_{k,l} \quad \leqslant \quad E_{min} + \delta \tag{2.26}$$

where $E_{min} + \delta$ is the upper boundary of the energy range from the MFE ($E_{1,n}$) to a userdefined deviation δ. E_{L_S} denotes the summed energy of all substructures already found. $E_{i,j}$ is the energy of the current subsequence $[i, j]$ to evaluate and $\sum_{k,l} E_{k,l}$ labels the best possible energy of all remaining subsequences which have to be evaluated further.

According to the inequalities the possibility to fulfill more than one case in each backtracking step arises and the amount of backtracked structures increases. To handle the amount of data from each recursion step, their algorithm writes found base pairs and subsequences to distinct base pair and interval stacks $\mathcal{P}$ and σ, respectively. These stacks are contained in states $\mathcal{S}$, which are written to a state stack R again. A state $\mathcal{S} = (\sigma, \mathcal{P}, E_{L_S})$ is a triple of an interval stack σ, a base pair stack $\mathcal{P}$ and the summed energy of all substructures already found E_{L_S}. $\mathcal{S}$ will also be called *partial structure* in the following sections. For choosing the right energy array in the backtracking process, each subsequence $[i, j]_E \in \sigma$ is labeled

with the appropriate subscript $E = \{F, C, M, M^1\}$.

The backtracking procedure starts with one initial state $S = ([1, n]_F; \emptyset, 0)$ on stack R. It consists of the interval $[1, n]$ as usual for the initial step in backtracking the MFE structure, an empty stack of base pairs and an energy of 0, because no substructures have been evaluated yet. The algorithm then pops the first element from the state stack R and begins with a *refinement* of the so called *partial structure* S. This *refinement* of S leads to the actual backtracking procedure for one single secondary structure but also generates new *partial structures* which are pushed onto the state stack R, allowing backtracking of more than one secondary structure. A particular *refinement procedure* of a *partial structure* ends, if there is no subsequence left on the interval stack σ. In this case the next state is popped from R and a new *refinement procedure* of the current *partial structure* begins. The complete extended backtracking procedure ends if there is no state left on R.

REFINEMENT OF PARTIAL STRUCTURES The refinement of a partial structure S starts by popping an interval $[i, j]_E$ from the interval stack σ. Corresponding to the type of energy array E, the following inequalities are tested and further operations are performed.

- case $E = F$

  When backtracking in F, the first inequality validated checks, if the 5'-end can be left unpaired.

$$F_{i+1,j} + E_{L_S} + \sum_{[k,l]\in S} E_{k,l} \leqslant E_{min} + \delta \tag{2.27}$$

  If (2.27) is true, the *refinement* $S' = ([i+1, j]_F.\sigma, \mathcal{P}, E_{L_S})$ will be pushed on the partial structure stack R.

  The second possibility checked for acceptance determines all possible left-most basepairs (i, u) which achieve a free energy to fulfill the condition

$$C_{i,u} + F_{u+1,j} + E_{L_S} + \sum_{[k,l]\in S} E_{k,l} + d^5_{i,u,i-1} + d^3_{i,u,u+1} \leqslant E_{min} + \delta \tag{2.28}$$

The *refinement* $\mathcal{S}' = ([i,u]_C.[u+1,j]_F.\sigma, \mathcal{P}, E_{L_s} + d^5_{i,u,i-1} + d^3_{i,u,u+1})$ of each found position u will be pushed on stack R, whereas the evaluated basepair (i,u) is not added to the basepair stack $\mathcal{P}$ in this step.

- case $E = C$
  Backtracking in energy array C is performed if base position i forms a basepair with base position j on the interval $[i,j]_C$. Corresponding to (2.12) - (2.14) three distinct conditions have to be tested for acceptance.

  The first criterion simply checks if the base pair (i,j) may be a hairpin loop with closing pair (i,j):

$$\mathcal{H}(i,j) + E_{L_s} + \sum_{[k,l]\in\mathcal{S}} E_{k,l} \quad \leqslant \quad E_{min} + \delta \tag{2.29}$$

  In this case, the refinement $\mathcal{S}' = (\sigma, \mathcal{P} \cup \{i \cdot j\}, E_{L_s} + \mathcal{H}(i,j))$ is pushed on stack R. The next criterion searchs for base pairs (p,q) with $i < p < q < j$ constituting an interior loop, a bulge or a stacked pair. All base pairs (p,q) fulfilling

$$C_{p,q} + \mathcal{I}(i,j,p,q) + E_{L_s} + \sum_{[k,l]\in\mathcal{S}} E_{k,l} \quad \leqslant \quad E_{min} + \delta \tag{2.30}$$

  lead to a new partial structure $\mathcal{S}' = ([p,q]_C.\sigma, \mathcal{P} \cup \{i \cdot j\}, E_{L_s} + \mathcal{I}(i,j,p,q))$ pushed on R.

  The last condition inspected checks for the possible occurance of a multiloop, closed by basepair (i,j). Every position u that decomposes the interval into a substructure with at least one stem with closing pair (p,q) with $i < p < q \leqslant u$ on the subinterval $[i+1,u]_M$ and a substructure with exactly one stem and closing pair $(u+1,r)$ with $u+1 < r \leqslant j-1$ on the subinterval $[u+1,j-1]_{M^1}$ is examined.

$$\begin{aligned} &M_{i+1,u} + M^1_{u+1,j-1} + a \\ &\qquad + E_{L_s} + \sum_{[k,l]\in\mathcal{S}} E_{k,l} \\ &\qquad + d^5_{i,j,j-1} + d^3_{i,j,i+1} \quad \leqslant \quad E_{min} + \delta \end{aligned} \tag{2.31}$$

  Each u that satisfies (2.31) leads to a new *refinement* $\mathcal{S}' = ([i+1,u]_M.[u+1,j-1]_{M^1}.\sigma, \mathcal{P} \cup \{i \cdot j\}, E_{L_s} + a + d^5_{i,j,j-1} + d^3_{i,j,i+1})$, pushed on the partial structure stack.

- case $E = M$
  Corresponding to (2.15) - (2.17), the following inequalities consider the three possibilities to construct a multiloop part with at least one stem. The first just nibbles the rightmost base j,

$$M_{i,j-1} + c + E_{L_S} + \sum_{[k,l]\in S} E_{k,l} \quad \leqslant \quad E_{min} + \delta \tag{2.32}$$

  yielding $S' = ([i, j-1]_M.\sigma, \mathcal{P}, E_{L_S} + c)$ which is pushed on R. Corresponding to (2.16) the second inequality checks whether there is a base position u separating the interval $[i, j]$ into a region of $(u-i-1)$ unpaired bases and exactly one interior basepair $(u+1, j)$. If

$$\begin{aligned} C_{u+1,j} + (u-i-1)\cdot c + b \\ + E_{L_S} + \sum_{[k,l]\in S} E_{k,l} \\ + d^5_{u+1,j,u} + d^3_{u+1,j,j+1} \quad \leqslant \quad E_{min} + \delta \end{aligned} \tag{2.33}$$

  is accepted the *refinement* $S' = ([u+1, j]_C.\sigma, \mathcal{P}, E_{L_S} + d^5_{u+1,j,u} + d^3_{u+1,j,j+1} + (u-i-1)\cdot c + b)$ is pushed onto the partial structure stack R.

  The third condition may be fulfilled when the interval $[i, j]$ is decomposed into a subinterval $[i, u]$ containing at least one stem, and a loop enclosed by the basepair $(u+1, j)$ constituting exactly one stem of the surrounding multiloop.

$$\begin{aligned} M_{i,u} + C_{u+1,j} + \mathcal{M}_{Stem} \\ + E_{L_S} + \sum_{[k,l]\in S} E_{k,l} \\ + d^5_{u+1.j,u} + d^3_{u+1,j,j+1} \quad \leqslant \quad E_{min} + \delta \end{aligned} \tag{2.34}$$

  The resulting *refinement* $S' = ([i, u]_M.[u+1, j]_C.\sigma, \mathcal{P}, E_{L_S} + \mathcal{M}_{Stem} + d^5_{u+1.j,u} + d^3_{u+1,j,j+1})$ is pushed on R again.

- case $E = M^1$
  The last remaining energy array the algorithm may backbacktrack in is M^1. Only two distinctions have to be done here, because the interval $[i, j]_{M^1}$ contains exactly one terminal basepair by definition and furthermore the

base at position i must be paired with another downstream base u with $i < u \leqslant j$. Hence, the first trivial condition is a check if j may be nibbled

$$M^1_{i,j-1} + c + E_{L_S} + \sum_{[k,l]\in S} E_{k,l} \quad \leqslant \quad E_{min} + \delta \tag{2.35}$$

If the inequality is met it leads to a further *refinement* $S' = ([i, j-1]_{M^1}.\sigma, \mathcal{P}, E_{L_S} + c)$ to be pushed onto R.

The second possibility just handles the case where i pairs with j

$$C_{i,j} + b + d^5_{i,j,i-1} + d^3_{i,j,j+1} + E_{L_S} + \sum_{[k,l]\in S} E_{k,l} \quad \leqslant \quad E_{min} + \delta \tag{2.36}$$

yielding to push the *refinement* $S' = ([i, j]_C.\sigma, \mathcal{P}, E_{L_S} + d^5_{i,j,i-1} + d^3_{i,j,j+1} + b)$ onto the stack R.

If none of the previously introduced conditions holds, hence the best possible energy is too large, the whole partial structure will be dismissed. Else, the *refinement* procedure of S continues until there are no further intervals $[i, j]_E$ in σ. In this case, the base pairs in $\mathcal{P}$ constitute one found suboptimal secondary structure and the algorithm proceeds by popping the next *partial structure* from R. As mentioned before, the algorithm terminates if there is no partial structure S on the stack R for further refinement.

Stochastic Backtracking

An additional way to obtain a set of suboptimal secondary structures is the so called *stochastic backtracking*. Considering the previous algorithm for generating the set of secondary structures within an energy range from the MFE, the number of generated structures increases exponentially for larger δ. Also the computation time increases quickly especially for longer RNA sequences, as the algorithm degenerates to an enumeration of all possible secondary structures for $\delta >> E_{min}$. A solution of this dilemma is to generate statistically representative samples of the Boltzmann ensemble of secondary structures. In this ensemble secondary structures s occur with Boltzmann equilibrium probability $P(s)$ where

$$P(s) \quad = \quad \frac{e^{-\frac{F(s)}{RT}}}{Q} \tag{2.37}$$

Representative samples are then chosen by their probability $P(s)$ and a successive run of this sampling method generates a set of suboptimal secondary structures. An algorithm which implements this sampling was described in [12, 13, 29, 56] and is available as part of the `RNAsubopt` program in the `Vienna RNA Package` [27] and the program `Sfold`[12].

Similarly to the *suboptimal backtracking* method of Wuchty et al., the algorithm is a modification of the original MFE *backtracking* procedure. But in contrast to both, the construction of *suboptimal structures* and *structures with minimum free energy*, the partition function Q is calculated in the forward recursion step. Afterwards, backtracking takes place in the partition function arrays Q, Q^B, Q^M and Q^{M1} to compute a base pairing pattern.

PROBABILITY DECOMPOSITION Decomposing a secondary structure s into substructures $s_i \in s$ leads to a decomposition of the Boltzmann equlibrium probability $P(s)$ into conditional equilibrium probabilities $P(s_i|s_j)$ of the occurance of these substructures where s_i results from a decomposition of s_j. When decomposing the secondary structure in the backtracking process by means of the recursion scheme in Fig. 13 each possible decomposition $\mathcal{D}$ into a substructure s_i leads to an equilibrium decomposition probability $P(\mathcal{D})$, where

$$P(\mathcal{D}) = \frac{\text{contribution of } \mathcal{D} \text{ to partition function}}{\text{partition function}} \tag{2.38}$$

Following the recursivity of the backtracking algorithm and the successive order of the decompositions a high-dimensional probability distribution arises, which may make sampling in the Boltzmann ensemble quite difficult. However, the recursion scheme used is capable of dissecting the problem into subproblems, so sampling of a secondary structure becomes successive conditional sampling at lower dimensions. The unique decomposition pattern $\mathcal{D}^* = \{\mathcal{D}^1_{m_1}, \ldots, \mathcal{D}^N_{m_N}\}$ of a secondary structure s, where the superscripted integer indicates the recursion step k and the subscripted variable m_k is the chosen decomposition of all possible decompositions in step k, can then be used to formulate the equilibrium probability $P(s)$ by means of the conditional probabilities of the successive decompositions $D^k_{m_k}$.

$$P(\mathcal{D}^*, s) = \prod_{\mathcal{D}^k_{m_k} \in \mathcal{D}^*} P(\mathcal{D}^k_{m_k} | \mathcal{D}^{k-1}_{m_{k-1}}, s) = P(s) \tag{2.39}$$

At this, the initial probability, thus $P(\mathcal{D}^k_{m_k})$ with $k = 0$, is set to 1 as it denotes the probability to decompose the secondary structure s at all, which is possible in any case. Furthermore, the recursion scheme makes the decomposition probability $P(\mathcal{D}^i)$ according to a certain energy array independant from other decomposition probabilities $P(\mathcal{D}^j)$. This is done by regarding the contribution of $\mathcal{D}^i$ to the appropriate partition function Q, Q^B, Q^M and Q^{M^1} on each interval $[i, j]$ and leads to an extension of equation (2.38).

$$\begin{aligned} P(\mathcal{D}, Q_{i,j}) &= \frac{\text{contribution of } \mathcal{D} \text{ to } Q_{i,j}}{Q_{i,j}} \\ P(\mathcal{D}, Q^B_{i,j}) &= \frac{\text{contribution of } \mathcal{D} \text{ to } Q^B_{i,j}}{Q^B_{i,j}} \\ P(\mathcal{D}, Q^M_{i,j}) &= \frac{\text{contribution of } \mathcal{D} \text{ to } Q^M_{i,j}}{Q^M_{i,j}} \\ P(\mathcal{D}, Q^{M^1}_{i,j}) &= \frac{\text{contribution of } \mathcal{D} \text{ to } Q^{M^1}_{i,j}}{Q^{M^1}_{i,j}} \end{aligned} \tag{2.40}$$

As for a given k each $P(D^k_{m_k}, E)$ is in range $0 \leqslant P(D^k_{m_k}, E) \leqslant 1$ and $\sum_{m_k} P(\mathcal{D}^k_{m_k}, E) = 1$ where $E \in \{Q_{i,j}, Q^B_{i,j}, Q^M_{i,j}, Q^{M^1}_{i,j}\}$, the probabilities $P(\mathcal{D}^k_{m_k}, E)$ can be assigned to successive intervals of length $l = P(\mathcal{D}^k_{m_k}, E)$ in range $[0, 1]$. For z possible decompositions in step k the arrangement would have the form

$$\underbrace{0 \ldots x_1}_{P(D_{m^1_k})} \underbrace{\ldots x_2}_{P(D_{m^2_k})} \ldots x_{z-1} \underbrace{\ldots 1}_{P(D_{m^z_k})} \tag{2.41}$$

An equally distributed random value r_k with $0 \leqslant r_k \leqslant 1$ can then be used to decide which decomposition m_k in each recursion step k has to be taken when backtracking stochastically. Thereby, the decomposition path m_k for each energy array $E = \{Q, Q^B, Q^M, Q^{M^1}\}$ on a sequence interval $[i, j]_E$ is chosen by means of its equilibrium probability. This leads to a recursive algorithm following the scheme depicted in Fig. 13.

- case $E = Q$
  Two different decomposition paths are possible if the interval is related to the Q energy array. A random value r_1 is taken to decide whether base i can be discarded by regarding it to be unpaired.

$$r_1 \cdot Q_{i,j} \leqslant Q_{i+1,j} \tag{2.42}$$

If the inequality is met the interval $[i+1,j]_Q$ remaining is backtracked further.

If (2.42) is not fulfilled, the base at position i must be paired with another downstream base u with $i < u \leqslant j$ and the pairing partner u has to be evaluated. By introducing auxiliary variables $Z[u]$ with

$$Z[u] \quad = \quad Q_{i+1,j} + \sum_{i<v\leqslant u} Q^B_{i,v} \cdot Q_{v+1,j} \cdot e^{-\beta\cdot(d^5_{i,v,i-1}+d^3_{i,v,v+1})} \tag{2.43}$$

the pairing partner for base i can be evaluated by determining the position u that fulfills

$$Z[u-1] \leqslant r_1 \cdot Q_{i,j} \quad < \quad Z[u] \tag{2.44}$$

Afterwards, the algorithm proceeds by backtracking the two intervals $[i,u]_{Q^B}$ and $[u+1,j]_Q$

- case $E = Q^B$
  Backtracking in Q^B leads to 3 different decomposition possibilities originating from the three forward recursion parts of (2.21). The base pair (i,j) delimits a *hairpin* if

$$r_2 \cdot Q^B_{i,j} \quad < \quad e^{-\beta\cdot\mathcal{H}(i,j)} \tag{2.45}$$

  In this case backtracking of $[i,j]_{Q^B}$ terminates.

  For the next possible loop type, *interior loops* including *stacks* and *bulges*, the base pair (p,q) *immediately interior* to (i,j) has to be evaluated. Introducing a function $t_1(p,q) = (j-i-1)\cdot(p-i-1) - \frac{(p-i)\cdot(p-i+1)}{2} + q$ that maps combinations of pairs (p,q) with $i < p < q < j$ to successive numbers in the interval $[1, \frac{n\cdot(n-1)}{2}]$, auxiliary variables $Z[t_1(p,q)]$ with

$$Z[t_1(p,q)] \quad = \quad \sum_{i<u\leqslant p} \sum_{p<v\leqslant q} e^{-\beta\cdot\mathcal{I}(i,j,u,v)} \cdot Q^B_{u,v} \tag{2.46}$$

  can be used to determine the interior base pair (p,q). In detail, a pair of positions p and q satisfying

$$Z[t_1(p,q)-1] \leqslant r_2 \cdot Q^B_{i,j} - e^{-\beta\cdot\mathcal{H}(i,j)} \quad < \quad Z[t_1(p,q)] \tag{2.47}$$

is regarded as the interior base pair of the $2-$loop with closing pair (i,j) and the backtracking procedure proceeds on the interval $[p,q]_{Q^B}$.

If none of the previous conditions hold, the interval $[i,j]_{Q^B}$ is a multiloop with closing pair (i,j). Corresponding to the unique multi loop decomposition of (2.21), a position u which splits the interval into two subintervals $[i+1,u]_M$ and $[u+1,j-1]_{M^1}$ has to be found. Again, auxiliary variables $Z[u]$ with

$$Z[u] = \sum_{i<v\leqslant u} Q^M_{i+1,v} \cdot Q^{M^1}_{v+1,j-1} \cdot e^{-\beta\cdot(d^5_{j,i,j-1}+d^3_{j,i,i+1})} \tag{2.48}$$

and a further variable Z^{aux} with

$$Z^{aux} = e^{-\beta\cdot\mathcal{H}(i,j)} - \sum_{i<p<q<j} e^{-\beta\cdot\mathcal{J}(i,j,p,q)} \cdot Q^B_{p,q} \tag{2.49}$$

are used to determine this position u that has to fulfill

$$Z[u-1] < r_2 \cdot Q^B_{i,j} - Z^{aux} \leqslant Z[u] \tag{2.50}$$

The resulting subintervals $[i+1,u]_{Q^M}$ and $[u+1,j-1]_{Q^{M^1}}$ are backtracked further.

- case $E = Q^M$
  The decomposition of substructures on an interval $[i,j]_{Q^M}$ leads to three possibilities. The first is, that base j does not pair with any other base u with $i \leqslant u < j$ and therefore can be nibbled:

$$r_3 \cdot Q^M_{i,j} < Q^M_{i,j-1} \cdot e^{-\beta\cdot c} \tag{2.51}$$

  This leads to the evaluation of the remaining interval $[i,j-1]_{Q^M}$ in the next recursion step.

  According to the recursion scheme, the second possibility is that there exists a split point u dividing the interval into a rightmost part with exactly one stem and a left part containing $(u-i)$ unpaired bases. This split point is determined by introducing variables $Z[u]$ with

$$Z[u] = \sum_{i<v<u} Q^B_{v,j} \cdot e^{-\beta\cdot((v-i)\cdot c+b)} \cdot e^{-\beta\cdot(d^5_{v,j,v-1}+d^3_{v,j,j+1})} \tag{2.52}$$

and finding the position u with

$$Z[u-1] \leqslant r_3 \cdot Q^M_{i,j} - Q^M_{i,j-1} \cdot e^{-\beta \cdot c} \quad < \quad Z[u] \tag{2.53}$$

Here, the remaining interval $[u,j]_{Q^B}$ has to be backtracked in the next step.

In case that no u fulfilling (2.53) is found, the last possibility holds. It states that there is a position u' separating the interval $[i,j]$ into a rightmost part with exactly one stem with closing pair (u',j) and a left part with at least one additional stem in the interval $[i,u'-1]$. Using $Z[u']$ with

$$Z[u'] \quad = \quad \sum_{i<v\leqslant u'} Q^M_{i,v-1} \cdot Q^B_{v,j} \cdot e^{-\beta \cdot b} \cdot e^{-\beta \cdot (d^5_{v,j,v-1} + d^3_{v,j,j+1})} \tag{2.54}$$

and Z^{aux} with

$$\begin{aligned} Z^{aux} \quad = \quad & Q^M_{i,j-1} \cdot e^{-\beta \cdot c} \\ & + \sum_{i<u<j} Q^B_{u,j} \cdot e^{-\beta \cdot ((u-i)\cdot c + b)} \cdot e^{-\beta \cdot (d^5_{u,j,u-1} + d^3_{u,j,j+1})} \end{aligned} \tag{2.55}$$

this position u' has to meet the condition

$$Z[u'-1] < r_3 \cdot Q^M_{i,j} - Z^{aux} \quad \leqslant \quad Z[u'] \tag{2.56}$$

and leads to a backtracking of the intervals $[i,u'-1]_{Q^M}$ and $[u',j]_{Q^B}$.

- case $E = Q^{M^1}$
  Two inequalities have to be tested if an interval $[i,j]_{Q^{M^1}}$ is related to the Q^{M^1} array. First, j may be unpaired and can therefore be nibbled, which is checked by

$$r_4 \cdot Q^{M^1}_{i,j} \quad < \quad Q^{M^1}_{i,j-1} \cdot e^{-\beta \cdot c} \tag{2.57}$$

  resulting in the interval $[i,j-1]_{Q^{M^1}}$ for further backtracking.

  Second, j may form a base pair with i and the interval $[i,j]_{Q^B}$ is backtracked in the next step.

$$\begin{aligned} r_4 \cdot Q^{M^1}_{i,j} \quad \leqslant \quad & Q^{M^1}_{i,j-1} \cdot e^{-\beta \cdot c} \\ & + Q^B_{i,j} \cdot e^{-\beta \cdot (b + d^5_{i,j,i-1} + d^3_{i,j,j+1})} \end{aligned} \tag{2.58}$$

At the end of this stochastic backtracking process the base pairing pattern found constitutes a secondary structure s sampled with probability $P(s)$. As mentioned before, a successive run of this algorithm produces a set of suboptimal secondary structures.

2.3.4 *Folding of Aligned RNA Sequences*

Considering functional RNA molecules like rRNA, tRNA, small nuclear RNA (snRNA), miRNA, viroids and many others, it is evident that they exhibit characteristic secondary structures which are conserved over clusters of the family they belong to. Creating sequence alignments of such sets of RNA sequences to study the phylogenies is widely used in bioinformatic applications [14]. This for example leads to a better understanding of the ancestral relationships of coding mRNA or noncoding tRNA in an interspecies perspective. Another possibility arises by creating an alignment of a collection of RNA sequences. It allows the prediction of a consensus secondary structure, especially for functional RNAs whose secondary structure motifs seem to be well conserved in evolution, e.g. the cloverleaf structure of tRNA.

An algorithm for the computation of consensus structures from a set of aligned sequences was presented in 2002 by Hofacker et al., combining phylogenetic sequence covariations and thermodynamic stability of RNA molecules into a modified energy model [26]. In contrast to other approaches, their ansatz minimizes the computational effort by allowing to run the algorithm only once for a given alignment. This speeds up calculation especially for alignments of larger sequences. Furthermore it provides a reasonable way for testing the reliability of predictions by providing the base-pairing probability matrix instead of the MFE structure or a set of suboptimal folds.

SEQUENCE COVARIATION Mutations in an RNA sequence which disrupt Watson-Crick base pairs have negative effects as they may prevent the sequence to fold into its desired spatial structure. However, a loss of a base pair can be overcome by a second - so called *compensatory mutation* - restoring the base pair. This kind of two consecutive mutations creates patterns of nucleotide substitutions which are called covariations. Such patterns can be detected in

interspecies sequence alignments of homologous RNAs, where the potential of forming a base pair is retained while sequence similarities are lost. For example, the replacement of a GC pair with an AU pair is a covariation.

For measuring the covariance of two sequences λ and μ in an alignment $\mathcal{A}$ of N sequences, Hofacker et al. first formulated the Hamming distance $d_{i,j}^{\lambda,\mu}$ of the base pairs $(i,j)_\lambda$ and $(i,j)_\mu$

$$d_{i,j}^{\lambda,\mu} = 2 - \delta(a_i^\lambda, a_i^\mu) - \delta(a_j^\lambda, a_j^\mu) \tag{2.59}$$

where $\delta(a', a'') = 1$ if $a' = a''$ and $\delta(a', a'') = 0$ otherwise. This distance measure is able to distinguish between conserved base pairs as well as pairs with consistent mutations and compensatoric mutations. So if $d_{i,j}^{\lambda,\mu} = 0$, position i and j in the aligned sequence λ match with position i and j of sequence μ. If they differ in exactly one position $d_{i,j}^{\lambda,\mu} = 1$ and $d_{i,j}^{\lambda,\mu} = 2$ if they differ in both positions.

Introducing matrices $\prod^\alpha$ where $\prod_{i,j}^\alpha = 1$ if base i may pair with j in the aligned sequence $\alpha \in \mathcal{A}$ and $\prod_{i,j}^\alpha = 0$ otherwise, a measure of covariation is

$$\begin{aligned} \mathcal{C}_{i,j} &= \frac{1}{\binom{N}{2}} \cdot \sum_{\lambda<\mu} d_{i,j}^{\lambda.\mu} \cdot \prod_{i,j}^{\lambda} \cdot \prod_{i,j}^{\mu} \\ &= \sum_{XY,X'Y'} f_{i,j}(XY) \cdot D_{XY,X'Y'} \cdot f_{i,j}(X'Y') \\ &= \langle f_{i,j} D f_{i,j} \rangle \end{aligned} \tag{2.60}$$

D is a 16×16 matrix with entries $D_{XY,X'Y'} = d_H(XY, X'Y')$ if $XY \in B$ and $X'Y' \in B$ and $D_{XY,X'Y'} = 0$, otherwise. The capital letters $\{X, Y, X', Y'\} \in \Sigma$ designate the bases X and Y replaced by bases X' and Y' using the alphabet $\Sigma = \{A, G, C, U\}$ and $d_H(XY, X'Y')$ is the Hamming distance between the base pairs as denoted in (2.59). Therefore D is the distance matrix for each base pair substitution. $XY \in B$ indicates that X may pair with Y and the function $f_{i,j}(XY)$ computes the frequency of finding X in i and Y in j in the complete alignment $\mathcal{A}$. As shown in the second and third line of (2.60), this covariance measure is a scalar product and thus can be evaluated efficiently.

Even though the covariance score $\mathcal{C}_{i,j}$ gives a bonus to compensatory mutations, it does not deal with sequences where no base pair can be formed between position i and j, e.g. due to an inserted gap or a mismatch. A simple extension to

take these cases into account is to count the number of sequences where i, j is a combination of a nucleotide with a gap as it indicates an inconsistent mutation. Whereas gaps in both positions i and j are ignored, as they may constitute a deletion of an entire base pair (i, j) that, in some cases, does not disrupt the structure motif.

$$q_{i,j} = 1 - \frac{1}{N} \cdot \sum_{\alpha} \left\{ \prod\nolimits_{i,j}^{\alpha} + \delta(a_i^{\alpha}, gap) \cdot \delta(a_j^{\alpha}, gap) \right\} \tag{2.61}$$

Combining (2.60) and (2.61) leads to the *combined covariation score* $B_{i,j}$ with

$$B_{i,j} = C_{i,j} - \phi_1 \cdot q_{i,j} \tag{2.62}$$

The factor ϕ_1 represents the relative weight of inconsistent sequences and defaults to 1.0.

In contrast to the previously discussed folding algorithms, where the evaluation of the base pairing energy contribution of a possible base pair (i, j) always depends on the ability to form a Watson-Crick or GU pair between position i and j in the single RNA sequence, the pairing between position i and j depends on all bases $(a_i^{\alpha}, a_j^{\alpha}) \forall \alpha \in \mathcal{A}$ when computing a consensus structure. It is convenient to define a threshold value B^* which indicates whether the energy contribution of a pair of base positions is taken into account when evaluating the consensus structure or not. Using B^* for all positions i, j with $i < j$ the upper triangle of the pairing matrix $\prod^{\mathcal{A}}$ with

$$\prod\nolimits_{i,j}^{\mathcal{A}} = \begin{cases} 0 & \text{if } B_{i,j} < B^* \\ 1 & \text{if } B_{i,j} \geqslant B^* \end{cases} \tag{2.63}$$

is filled and used in the following recursion steps.

The definitions above allow an extension of the *loop-based energy model* to deal with alignments of sequences. Therefore each energy contribution $E_{i,j}$ in the recursion scheme of equation (2.5)-(2.8) becomes an energy contribution of the complete alignment $E_{i,j}^{\mathcal{A}}$ with

$$E_{i,j}^{\mathcal{A}} = \frac{1}{N} \cdot \sum_{\alpha \in \mathcal{A}} \epsilon(a_i^{\alpha}, a_j^{\alpha}) - \phi_2 \cdot B_{i,j} \tag{2.64}$$

Considering an alignment of N sequences, the total energy contribution $E_{i,j}^{\mathcal{A}}$ then is the average of the energy contributions $\epsilon(a_i^{\alpha}, a_j^{\alpha})$ of base pair $(a_i^{\alpha}, a_j^{\alpha})$ in each

sequence α plus the covariance contribution $B_{i,j}$ where the latter is weighted by factor ϕ_2 [26].

PARTITION FUNCTION OF ALIGNED SEQUENCES Using (2.64), the partition function algorithm (2.21) on page 30 can easily be extended to take aligned sequences into account. The main difference then is the evaluation of the energy contribution $Q^{\mathcal{A}B}_{i,j}$ which is 0.0 if position i does not form a base pair with j in the alignment context, indicated by $\prod^{\mathcal{A}}_{i,j} = 0$. The factors $\rho = \frac{1}{RTN}$ and $\beta = \frac{1}{RT}$ are used to simplify the recursive equations.

$$Q^{\mathcal{A}}_{i,j} = 1 + \sum_{i<u\leqslant j} Q^{\mathcal{A}B}_{i,u} \cdot Q^{\mathcal{A}}_{u+1,j} \cdot e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{i_\alpha,u_\alpha,(i-1)_\alpha} + d^3_{i_\alpha,u_\alpha,(u+1)_\alpha}} \tag{2.65}$$

$$\begin{aligned} Q^{\mathcal{A}B}_{i,j} = {} & e^{\rho\cdot\phi_2\cdot B_{i,j}} \cdot \Bigg\{ e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} \mathcal{H}(i_\alpha,j_\alpha)} \\ & + \sum_{i<p<q<j} Q^{\mathcal{A}B}_{p,q} \cdot e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} \mathcal{I}(i_\alpha,j_\alpha,p_\alpha,q_\alpha)} \\ & + \sum_{i<u<j} Q^{\mathcal{A}M}_{i+1,u} \cdot Q^{\mathcal{A}M^1}_{u+1,j-1} \cdot e^{-\beta\cdot(a+b)} \cdot \Bigg\{ \\ & e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^3_{j_\alpha,i_\alpha,(i+1)_\alpha} + d^5_{j_\alpha,i_\alpha,(j-1)_\alpha}} \Bigg\} \Bigg\} \end{aligned} \tag{2.66}$$

$$\begin{aligned} Q^{\mathcal{A}M}_{i,j} = {} & Q^{\mathcal{A}M}_{i,j-1} \cdot e^{-\beta\cdot c} \\ & + \sum_{i<u<j} Q^{\mathcal{A}B}_{u,j} \cdot e^{-\beta\cdot(b+(u-i)\cdot c)} \cdot \Bigg\{ \\ & e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{u_\alpha,j_\alpha,(u-1)_\alpha} + d^3_{u_\alpha,j_\alpha,(j+1)_\alpha}} \Bigg\} \\ & + \sum_{i<u<j} Q^{\mathcal{A}B}_{u,j} \cdot Q^{\mathcal{A}M}_{i,u-1} \cdot e^{-\beta\cdot b} \cdot \Bigg\{ \\ & e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{u_\alpha,j_\alpha,(u-1)_\alpha} + d^3_{u_\alpha,j_\alpha,(j+1)_\alpha}} \Bigg\} \end{aligned} \tag{2.67}$$

$$\begin{aligned} Q^{\mathcal{A}M^1}_{i,j} &= Q^{\mathcal{A}M^1}_{i,j-1} \cdot e^{-\beta \cdot c} \\ &\quad + Q^{\mathcal{A}B}_{i,j} \cdot e^{-\beta \cdot b} \cdot e^{-\rho \cdot \sum_{\alpha \in \mathcal{A}} d^5_{i_\alpha, j_\alpha, (i-1)_\alpha} + d^3_{i_\alpha, j_\alpha, (j+1)_\alpha}} \end{aligned} \tag{2.68}$$

Beginning with all pentanucleotide sequences $[1,5]^{\mathcal{A}}$ to $[n-4,n]^{\mathcal{A}}$ the algorithm again fills the energy matrices $Q^{\mathcal{A}}$, $Q^{\mathcal{A}B}$, $Q^{\mathcal{A}M}$ and $Q^{\mathcal{A}M^1}$, inceasing the sequence length gradually until the complete alignment $[1,n]^{\mathcal{A}}$ has been processed, which is common to the forward recursions of all previously described folding algorithms in this section. The partition function $\mathcal{Q}^{\mathcal{A}}$ of the complete alignment $\mathcal{A}$ is located at $Q^{\mathcal{A}}_{1,n}$ as usual.

BASE PAIRING PROBABILITIES OF ALIGNED SEQUENCES Originating from the four partition function arrays filled by the recursions above, the base pairing probability matrix of the aligned sequences can be obtained. Therefore the backward recursion (2.23) has to be extended in the same way as the forward recursion, leading to

$$\begin{aligned} P^{\mathcal{A}\ lin}_{i,j} &= \frac{Q^{\mathcal{A}}_{1,i-1} \cdot Q^{\mathcal{A}B}_{i,j} \cdot Q^{\mathcal{A}}_{j+1,n}}{Q^{\mathcal{A}}_{1,n}} \cdot e^{\rho \cdot \phi_2 \cdot B_{i,j}} \\ &\quad \cdot e^{-\rho \cdot \sum_{\alpha \in \mathcal{A}} d^5_{i_\alpha, j_\alpha, (i-1)_\alpha} + d^3_{i_\alpha, j_\alpha, (j+1)_\alpha}} \end{aligned} \tag{2.69}$$

for the contribution of secondary structures enclosed by (i,j) that are not enclosed by any other base pair (p,q) with $p < i < j < q$. Taking the contribution of all secondary structures where (i,j) is enclosed by such a base pair (p,q) into account too, the complete recursion to obtain the equilibrium base pairing probability of

the aligned sequences leads to

$$
\begin{aligned}
P^{\mathcal{A}}_{i,j} \;=\; & P^{\mathcal{A}\ lin}_{i,j} \\
& + e^{\rho\cdot\phi_2\cdot B_{i,j}} \cdot \Bigg\{ \\
& \quad \sum_{p<i<j<q} P^{\mathcal{A}}_{p,q} \cdot \frac{Q^{\mathcal{A}B}_{i,j}}{Q^{\mathcal{A}B}_{p,q}} \cdot e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} \mathcal{I}(p_\alpha,q_\alpha,i_\alpha,j_\alpha)} \\
& \quad + \sum_{p<i} Q^{\mathcal{A}}_{i,j} \cdot e^{-\beta\cdot(a+b)} \cdot \Bigg\{ \\
& \qquad + \sum_{q>j} \frac{P^{\mathcal{A}}_{p,q}}{Q^{\mathcal{A}B}_{p,q}} \cdot e^{-\beta\cdot(q-j-1)\cdot c} \cdot Q^{\mathcal{A}M}_{p+1,i-1} \cdot \Big\{ \\
& \qquad\quad e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{q_\alpha,p_\alpha,(q-1)_\alpha} + d^3_{q_\alpha,p_\alpha,(p+1)_\alpha}} \Big\} \\
& \qquad + \sum_{q>j} \frac{P^{\mathcal{A}}_{p,q}}{Q^{\mathcal{A}B}_{p,q}} \cdot Q^{\mathcal{A}M}_{j+1,q-1} \cdot e^{-\beta\cdot(i-p-1)\cdot c} \cdot \Big\{ \\
& \qquad\quad e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{q_\alpha,p_\alpha,(q-1)_\alpha} + d^3_{q_\alpha,p_\alpha,(p+1)_\alpha}} \Big\} \\
& \qquad + \sum_{q>j} \frac{P^{\mathcal{A}}_{p,q}}{Q^{\mathcal{A}B}_{p,q}} \cdot Q^{\mathcal{A}M}_{j+1,q-1} \cdot Q^{\mathcal{A}M}_{p+1,i-1} \cdot \Big\{ \\
& \qquad\quad e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{q_\alpha,p_\alpha,(q-1)_\alpha} + d^3_{q_\alpha,p_\alpha,(p+1)_\alpha}} \Big\} \Bigg\} \Bigg\}
\end{aligned}
\tag{2.70}
$$

The third line of (2.70) shows contributions by *interior loops* and beginning in the fourth line *multi loop* contributions where (i,j) delimits the most $3'$, the most $5'$ or an intermediate branch are handled.

3

FOLDING OF CIRCULAR RNAS (I)

Circular RNAs differ from linear ones in that the first and the last nucleotide are covalently bound, forming an unbroken chain. This leads to a problem in consecutive numbering of their nucleotides, because circles have no beginning. However, one can arbitrary pick one base and denote it to be the first base for writing down the RNA sequence. The direction of numbering from the $5'-$ to the $3'-$ end is uniquely defined as for linear RNAs. Secondary structures of circular RNAs can be defined as in definition 2.1 with the difference that the first and last base may not pair with each other as they are adjecent in the ring molecule.

Simply cutting the circular RNA at an arbitrary point and treating it as linear RNA in secondary structure prediction was found to be highly cut point dependant [54], so the exisiting folding algorithms had to be extended. The first ansatz of Hofman [54] deals with circular RNAs using a modification of the dynamic programming algorithm of Zuker and Stiegler [74] as discussed in 2.3. For each subsequence $[i, j]$ not only the minimal folding energy $V(i, j)$ and $W(i, j)$ is computed, but also the minimum free energy of the so called *exterior loop* from j to i by evaluating $V(j, i)$ and $W(j, i)$. Afterwards, the minimum folding energy of the circular RNA can be obtained with

$$\min_{i<j}\{V(i,j) + V(j,i)\} \tag{3.1}$$

Compared to the prediction of a secondary structure of a linear RNA of the same size, this algorithm doubles the computation time as well as the memory requirements.

3.1 ZUKER'S ALGORITHM

Another method which is used in Zukers `mfold package`, doubles the length of the sequence to $2n$ by concatenating the sequence $[1, n]$ on itself [73]. This results

in a sequence where the nucleotides $n+1,\ldots,2n$ are the same as $1,\ldots,n$. After introducing a new condition that $V(i,j)=\infty$, if $j-i>n-2$, the linear folding algorithm (2.2) is applied to the expanded sequence. Then, assuming that the secondary structure contains at least one base pair (i,j), the minimum free energy can be obtained by

$$\min_{i<j}\{V(i,j)+V(j,i+n)\} \tag{3.2}$$

which quadruples the memory requirements and roughly triples computation time compared to a linear RNA of the same size. In practice, parts of the matrix do not have to be evaluated due to a partial repetition of the matrix entries. At any rate, this algorithm leads to at least a doubling of the memory requirements and a doubling of computation time.

3.2 MEMORY EFFICIENT ALGORITHM

In 2005, Hofacker et al. [28] presented an algorithm which extends the linear folding to take circular RNAs into account by application of a kind of post-processing step. The key observation for the idea of this post-processing step was that the only difference between the handling of linear and circular sequences is the treatment of the *exterior loop* which contains the bases 1 and n. In the linear case, the exterior loop does not contribute any energy whereas in the circular case it has to be treated like any other kind of loop. The precalculated energy arrays of the linear forward recursion are used to calculate the energy contribution of these exterior loops. This memory efficient method allows to compute the structural differences between linear and circular RNAs running the linear forward recursion only once. The extension of the MFE algorithm as presented in their article [28] serves as a pattern for other folding algorithms discussed in the next chapter.

POST-PROCESSING When regarding exterior loops like any other loop in the circular sequence, their types also have to be distinguished. Hence, *exterior hairpin loops*, *exterior interior loops* and *exterior multi loops* occur. After filling the arrays F, C, M and M^1, the energy contribution of these loop types are obtained by evaluating the optimal energy of (a) hairpin structures F^o_H, (b) interior loop

structures F_I^o and (c) multi loop structures F_M^o containing bases 1 and n in their loop.

If the exterior loop is a hairpin, there is a base pair (j, i) with $1 \leqslant i < j \leqslant n$ closing the loop and giving rise to $l = i - 1 + (n - j - 1)$ unpaired bases, where l is the length of the loop. The minimal energy of such an *exterior hairpin loop* is

$$F_H^o = \min_{1 \leqslant i < j \leqslant n} \left\{ C_{i,j} + \mathcal{H}(j, i) \right\} \tag{3.3}$$

and can be calculated in $\mathcal{O}(n^2)$ without additional memory requirements.

Exterior interior loops contain two closing pairs (j, i) and (p, q) constituting the closing pairs of two components with minimal free energy $C_{i,j}$ and $C_{p,q}$. Their optimal energy is

$$F_I^o = \min_{1 \leqslant i < j < p < q \leqslant n} \left\{ C_{i,j} + C_{p,q} + \mathcal{I}(j, i; q, p) \right\} \tag{3.4}$$

and can be obtained in time proportinal to $\mathcal{O}(n^3)$ by limiting the total number of unpaired bases l, as for regular interior loops in the linear case, without additional memory requirements, too. The total length $l = l_1 + l_2$ is composed of the number of unpaired bases on both sides $l_1 = n - q + i - 1$ and $l_2 = p - j - 1$.

The last type, *exterior multi loops*, are loops with at least 3 stems on the sequence $[1, n]$. Therefore, an auxilary energy array M^2 is used that contains the energy contribution of multi loop parts with exactly two stems. $M^2_{k,n}$ is filled based on energy contributions of multiloop stems with exactly one stem (M^1) on the subsequence $[k, n]$.

$$M^2_{k,n} = \min_{k < u < n} \left\{ M^1_{k,u} + M^1_{u+1,n} \right\} \tag{3.5}$$

With this linear array, that requires $\mathcal{O}(n)$ additional memory, the minimal energy of *exterior multi loops* can be calculated proportional to $\mathcal{O}(n^2)$ in time. Using the M array, multi loops with at least three stems are composed by a part with exactly two stems (M^2) at the end of the sequence and another part with at least one stem (M) at the beginning of the sequence, leading to the equation

$$F_M^o = \min_{1 < k < n} \left\{ M_{1,k} + M^2_{k,n} + a \right\} \tag{3.6}$$

The minimal free energy of the folded complete circular sequence is then the minimum of the three possibilities for exterior loops

$$F^o = \min\{F^o_H, F^o_I, F^o_M\} \tag{3.7}$$

A graphical visualization of the exterior loop decomposition can be found in Fig. 14.

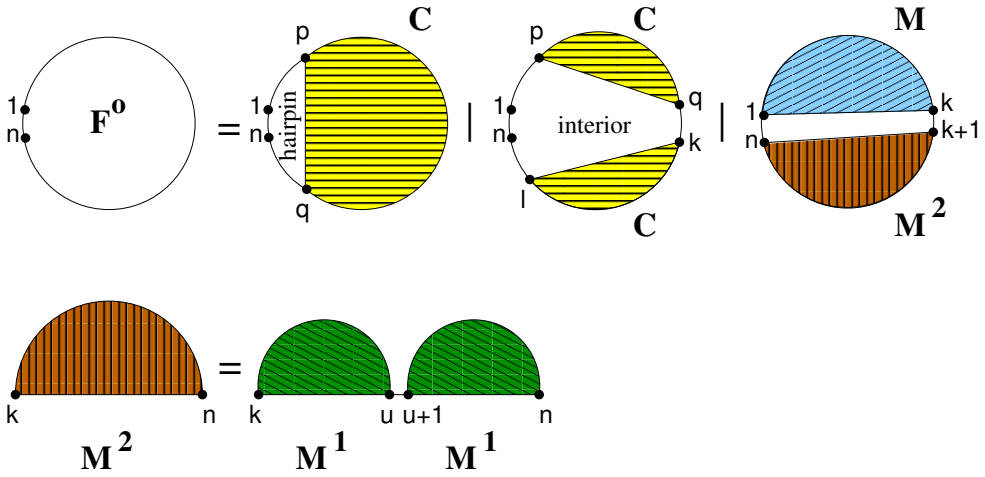

Figure 14: According to the loop types that contain bases 1 and n, a circular fold can be decomposed into three cases:
An exterior *hairpin* loop with closing pair (q, p) and enclosed interval $[p, q]_C$ or an exterior *interior* loop with closing pairs (p, q), (k, l) and enclosed intervals $[p, q]_C$ and $[k, l]_C$ are the first two of them. In the *multi loop* case, there have to be at least three stems, assured by the auxilary array M^2 which is obtained by concatenation of two structures with exactly one stem in the intervals $[k, u]_{M^1}$ and $[u+1, n]_{M^1}$. The exterior multi loop then consists of an interval $[1, k]_M$ with at least one stem and an interval $[k+1, n]_{M^2}$ with exactly two stems.

BACKTRACKING Backtracking a circular secondary structure reverses the post-processing into a pre-processing step. This step has to be performed before the normal backtracking procedure for linear structures as in 2.3.1 on page 26 can be applied. Hence, the optimal exterior loop type has to be determined.

If $F^o = F^o_H$, the exterior loop is a hairpin and its closing pair (i, j), satisfying

$$F^o_H = C_{i,j} + \mathcal{H}(j, i) \tag{3.8}$$

has to be found. This step needs computation time proportional to $\mathcal{O}(n^2)$. In the next step, backtracking of the subsequence $[i,j]_C$ follows the regular backtracking procedure for linear structures.

An *exterior interior loop* is present if $F^o = F^o_I$ holds. In this case, the two base pairs (i,j) and (p,q) which close the 2-loop have to be determined.

$$F^o_I = C_{i,j} + C_{p,q} + \mathcal{I}(j,i;q,p) \tag{3.9}$$

Constraining the total loop length l as in the post-processing step, the search for both closing pairs needs time proportinal to $\mathcal{O}(n^3)$ and the subsequences $[i,j]_C$ and $[p,q]_C$ have to be backtracked as in the linear case.

The exterior loop is a multi loop if $F^o = F^o_M$. In such a case, the first step is to find the position k which separates the multi loop into a left part with at least one stem and a right part with exactly two stems.

$$F^o_M = M_{1,k} + M^2_{k,n} + a \tag{3.10}$$

Once the position k is determined, the substructure with exactly two stems has to be split into two substructures representing exactly one stem each. This is done by looking for the separation position u which has to satisfy

$$M^2_{k,n} = M^1_{k,u} + M^1_{u+1,n} \tag{3.11}$$

The remaining subsequences $[1,k]_M$, $[k,u]_{M^1}$ and $[u+1,n]_{M^1}$ are treated like linear ones in the further backtracking steps following the backward recursions in 2.3.1 on page 26 again. All found base pairs constitute the secondary structure of the circular RNA sequence.

4

FOLDING OF CIRCULAR RNAS (II)

The memory efficient circular extension of the MFE algorithm discussed in the previous chapter can be applied to other folding algorithms too. An easy extension is possible if the folding algorithms follow the recursion scheme depicted in Fig. 13 on page 25. As all folding algorithms previously discussed are already converted into this scheme, their extension to take circular RNA sequences into account can also be treated by the application of a post-processing step in the forward- and a pre-processing step in the backward-recursions. This opens the chance to investigate circular RNA sequences and even alignments of them in the same way as linear RNAs.

4.1 PARTITION FUNCTION

The partition function $\mathcal{Q}^o$ of a circular RNA sequence can now be calculated in the same way as the MFE by applying a post-processing step. As the recursions (3.3) - (3.6) already provide a partition of the set of all secondary structures, their modification is straightforward.

$$Q^{M^2}_{k,n} = \sum_{k<u<n} Q^{M^1}_{k,u} \cdot Q^{M^1}_{u+1,n} \tag{4.1}$$

$$Q^o_H = \sum_{1\leqslant i<j\leqslant n} Q^B_{i,j} \cdot e^{-\beta\cdot\mathcal{H}(j,i)} \tag{4.2}$$

$$Q^o_I = \sum_{i<j<p<q} Q^B_{i,j} \cdot Q^B_{p,q} \cdot e^{-\beta\cdot\mathcal{I}(i,j;p,q)} \tag{4.3}$$

$$Q^o_M = \sum_k Q^M_{1,k} \cdot Q^{M^2}_{k+1,n} \cdot e^{-\beta\cdot a} \tag{4.4}$$

$$\mathcal{Q}^o = Q^o_H + Q^o_I + Q^o_M \tag{4.5}$$

BASE PAIRING PROBABILITIES Calculating the equilibrium base pairing probabilities $P_{i,j}$ of the circular RNA sequence differs in the calculation of the contri-

bution of the *exterior loop* the base pair (i,j) is part of. In the linear case (2.23) this contribution is handled by the term $P^{lin}_{i,j}$ where the *exterior loop* with a closing pair (i,j) does not contribute any energy. The complete equation for the calculation of $P^{lin}_{i,j}$ can also be formulated in the following way to take the Boltzmann weighted energy contribution $e^{\frac{0}{RT}} = 1$ of the (non existant) exterior loop into account.

$$P^{lin}_{i,j} = \frac{Q_{1,i-1} \cdot Q^B_{i,j} \cdot Q_{j+1,n} \cdot 1}{Q_{1,n}} \tag{4.6}$$

In the circular case, there are several different *exterior loops* possible so their contribution plays an important role in the computation of the equilibrium base pairing probability. A replacement of the term $P^{lin}_{i,j}$ by a contribution $P^{circ}_{i,j}$ that takes all possible exterior loops into account allows an easy extension of the recursions in (2.23). The equilibrium base pairing probability $P_{i,j}$ of a circular sequence can then be calculated by the recursive equation

$$\begin{aligned}
P_{i,j} = {} & P^{circ}_{i,j} \\
& + \sum_{p<i<j<q} P_{p,q} \cdot \frac{Q^B_{i,j}}{Q^B_{p,q}} \cdot e^{-\beta \cdot \mathcal{I}(p,q,i,j)} \\
& + \sum_{p<i} Q_{i,j} \cdot e^{-\beta \cdot (a+b)} \cdot \Bigg\{ \\
& \quad \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot e^{-\beta \cdot ((q-j-1)\cdot c + d^5_{q,p,q-1} + d^3_{q,p,p+1})} \cdot Q^M_{p+1,i-1} \\
& \quad + \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot Q^M_{j+1,q-1} \cdot e^{-\beta \cdot ((i-p-1)\cdot c + d^5_{q,p,q-1} + d^3_{q,p,p+1})} \\
& \quad + \sum_{q>j} \frac{P_{p,q}}{Q^B_{p,q}} \cdot Q^M_{j+1,q-1} \cdot Q^M_{p+1,i-1} \cdot e^{-\beta \cdot (d^5_{q,p,q-1} + d^3_{q,p,p+1})} \Bigg\}
\end{aligned} \tag{4.7}$$

As mentioned above, the equilibrium probability $P^{circ}_{i,j}$ has to take into account each possible *exterior loop* the base pair (i,j) can be part of. A base pair (i,j) may be the closing pair of an *exterior hairpin loop* with contribution of $Q^B_{i,j} \cdot e^{-\beta \cdot \mathcal{H}(j,i)}$ to the partition function $\mathcal{Q}^o$, a closing pair of an *exterior interior loop* giving rise to two contributions $Q^B_{p,q} \cdot e^{-\beta \cdot \mathcal{I}(q,p,j,i)}$ or $Q^B_{p,q} \cdot e^{-\beta \cdot \mathcal{I}(p,q,j,i)}$ corresponding to whether (i,j) delimits the "left" or "right" part of the exterior interior loop. Considering

exterior multi loops three contributions appear according to the part the base pair (i,j) delimits. Firstly the base pair may delimit the left most branch of the multi loop, secondly it may delimit an intermediate part and thirdly the right most part. These contributions lead to the equation replacing the calculation of the *exterior loop* contribution [28]:

$$
\begin{aligned}
P_{i,j}^{circ} \quad = \quad & \frac{Q_{i,j}^{B}}{Q^{o}} \cdot \Bigg\{ \\
& e^{-\beta \cdot \mathcal{H}(j,i)} \\
& + \sum_{p<q<i<j} Q_{p,q}^{B} \cdot e^{-\beta \cdot \mathcal{I}(q,p,j,i)} \\
& + \sum_{i<j<p<q} Q_{p,q}^{B} \cdot e^{-\beta \cdot \mathcal{I}(p,q,j,i)} \\
& + Q_{1,i-1}^{M} \cdot Q_{j+1,n}^{M} \cdot e^{-\beta \cdot (a+b)} \cdot e^{d_{j,i,j-1}^{5} + d_{j,i,i+1}^{3}} \\
& + \sum_{u<k} Q_{1,u}^{M} \cdot Q_{u+1,i-1}^{M_1} \cdot e^{-\beta \cdot (a+b+(n-u)\cdot c)} \cdot e^{d_{j,i,j-1}^{5} + d_{j,i,i+1}^{3}} \\
& + \sum_{u>l} Q_{u+1,n}^{M} \cdot Q_{j+1,u}^{M_1} \cdot e^{-\beta \cdot (a+b+(i-1)\cdot c)} \cdot e^{d_{j,i,j-1}^{5} + d_{j,i,i+1}^{3}} \Bigg\}
\end{aligned}
\tag{4.8}
$$

4.2 SUBOPTIMAL SECONDARY STRUCTURES

Complete suboptimal folding and *stochastic backtracking* as discussed in the previous chapters follow the recursion scheme in figure 13. So, they can also be extended to take circular RNA sequences into account. This enables the investigation of their energy landscape [68], e.g. for viroid sequences, and the design of multistable RNA molecules [15].

4.2.1 *Complete suboptimal folding*

The modification of the backtracking algorithm for finding all suboptimal secondary structures discussed in 2.3.3 is straight-forward. The pre-processing step alters the condition that $E = F$ of this linear backtracking process as this is the

point where energy contributions of the exterior loop play a role. In detail, the recursions of this condition have to be replaced by the evaluation of the exterior loop type(s). Of course, the appropriate energy variables F^o_H, F^o_I and F^o_M and also the energy array $M^2_{k,n}$ have to be filled in the forward recursion. The pre-processing step then is used to find all possible exterior loops where the energy contribution of the exterior and the interior loop is in range δ from the minimum free energy.

PRE-PROCESSING The first exterior loop type which may arise is the exterior hairpin and will be considered if $F^o_H \leqslant E_{min} + \delta$. To find all exterior hairpins and their enclosed interior structure constituting a suboptimal secondary structure, all possible base pairs (i, j) with $i < j$ closing the hairpin have to be determined. Each found base pair (i, j) satisfying equation (4.9) then leads to the *refinement* $\mathcal{S}' = ([i, j]_C.\sigma, \mathcal{P}, E_{L_\mathcal{S}} + \mathcal{H}(j, i))$ which is pushed on the partial structure stack R.

$$\mathcal{H}(j, i) + C_{i,j} \quad \leqslant \quad E_{min} = \delta \tag{4.9}$$

If $F^o_I \leqslant E_{min} + \delta$ holds, the second possible exterior loop type is treated. Therefore all possible closing pairs (i, j) and (p, q) with $i < j < p < q$ are evaluated. Each pair of base pairs fulfilling equation (4.10) results in a *refinement* $\mathcal{S}' = ([i, j]_C.[p, q]_C.\sigma, \mathcal{P}, E_{L_\mathcal{S}} + \mathcal{I}(j, i, q, p))$ pushed on stack R.

$$\mathcal{I}(j, i; q, p) + C_{i,j} + C_{p,q} \quad \leqslant \quad E_{min} + \delta \tag{4.10}$$

That a suboptimal secondary structure with an exterior multi loop exists is obvious if $F^o_M \leqslant E_{min} + \delta$. In this case two steps have to be done according to the post-processing step in the forward recursion. First, all positions k with $1 < k < n$ satisfying

$$M_{1,k} + M^2_{k+1,n} + a \quad \leqslant \quad E_{min} + \delta \tag{4.11}$$

must be investigated. Accepted positions k lead to a further decompostion of the substructure with exactly two stems of the multi loop in the interval $[k + 1, n]$. Therefore the position u with $k + 1 < u < n$ which splits this substructure into two substructures each with exactly one stem has to be found. Each u where

$$M_{1,k} + M^1_{k+1,u} + M^1_{u+1,n} + a \quad \leqslant \quad E_{min} + \delta \tag{4.12}$$

holds gives rise to a partial structure $\mathcal{S}' = ([1,k]_M.[k+1,u]_{M^1}.[u+1,n]_{M^1}\sigma, \mathcal{P}, E_{L_S} + a)$.

Each refinement of all generated partial structures $\mathcal{S}'$ pushed on R during the exterior loop determination is covered by the linear suboptimal backtracking algorithm of Wuchty et al. [69]. After performing the pre-processing step, the regular suboptimal backtracking follows, generating a base pair pattern of suboptimal secondary structures of a circular RNA sequence.

4.2.2 *Stochastic backtracking*

PRE-PROCESSING Extending the stochastic backtracking algorithm introduced in 2.3.3 on page 37 leads to two modifications. The first modification replaces the case where $E = Q$ with the pre-processing step to find an exterior loop. The second modification is the introduction of a new case $E = Q^{M^2}$ as there appear sequence fragments $[k,n]_{Q^{M^2}}$ which belong to the linear Q^{M^2} array in the pre-processing step. This newly introduced case is used to sample a statistically representative decomposition of a structure element with exactly two components on the interval $[k,n]$ constituting a part of the exterior multi loop. Reverting recursion (4.1) which fills the partition function array Q^{M^2} leads to

$$r_5 \cdot Q^{M^2}_{k,n} \leqslant \sum_{k<u<n} Q^{M^1}_{k,u} \cdot Q^{M^1}_{u+1,n} \tag{4.13}$$

where r_5 is a random number with $0 \leqslant r_5 \leqslant 1$. By introducing variables $Z[u]$ with

$$Z[u] = \sum_{k<v\leqslant u} Q^{M^1}_{k,v} \cdot Q^{M^1}_{v+1,n} \tag{4.14}$$

the stochastically sampled cutpoint between two components is a position u with

$$Z[u-1] < r_5 \cdot Q^{M^2}_{k,n} \leqslant Z[u] \tag{4.15}$$

The remaining sequence intervals $[k,u]_{M^1}$ and $[u+1,n]_{M^1}$ then have to be backtracked in the next step following the linear stochastic backtracking algorithm again.

The pre-processing step that replaces the case $E = Q$ of the linear stochastic backtracking algorithm samples one of the three possible exterior loop types. Following equation (2.38) the equilibrium probabilities for forming an *exterior hairpin* P_H^o, an *exterior interior loop* P_I^o or an *exterior multi loop* P_M^o then are

$$\begin{aligned} P_H^o &= \frac{Q_H^o}{Q^o} \\ P_I^o &= \frac{Q_I^o}{Q^o} \\ P_M^o &= \frac{Q_M^o}{Q^o} \end{aligned} \tag{4.16}$$

Using a random number r_6 with $0 \leqslant r_6 \leqslant 1$ the exterior loop is assumed to be a hairpin if

$$r_6 \cdot Q^o < Q_H^o \tag{4.17}$$

In this case the closing pair (j, i) of this hairpin has to be determined. Therefore, the Boltzmann weighted energy contributions of all possible exterior hairpin loops are successively added up until the first pair (i, j), fulfilling equation (4.20), is found.

$$r_6 \cdot Q^o < \sum_{i<j} Q_{i,j}^B \cdot e^{-\beta \cdot \mathcal{H}(j,i)} \tag{4.18}$$

Using a mapping function $t_2(i, j)$ with $t_2(i, j) = n \cdot (i-1) - \frac{i^2+i}{2} + j$ and auxiliary variables $Z[t_2(i, j)]$ with

$$Z[t_2(i, j)] = \sum_{i<j} Q_{i,j}^B \cdot e^{-\beta \cdot \mathcal{H}(j,i)} \tag{4.19}$$

this base pair (i, j) has to meet the following condition:

$$Z[t_2(i, j) - 1] \leqslant r_6 \cdot Q^o < Z[t_2(i, j)] \tag{4.20}$$

As soon as this pair is determined, sampling of the exterior loop terminates and the resulting (interior) sequence fragment $[i, j]_{Q^B}$ is backtracked stochastically.

Whenever equation (4.17) is false, the next possible loop type is tested. The statistically representative secondary structure contains an *exterior interior loop* if

$$r_6 \cdot Q^o < Q_H^o + Q_I^o \tag{4.21}$$

Is this case, both closing pairs (i,j) and (p,q) separating the enclosed intervals $[i,j]_{Q^B}$ and $[p,q]_{Q^B}$ have to be discovered.

Similarly to the previous loop type the Boltzmann weighted energy contributions of all possible exterior interior loops and their enclosed intervals are summed up until the first pair of base pairs matching

$$r_6 \cdot Q^o - Q^o_H \quad < \quad \sum_{i<j<p<q} Q^B_{i,j} \cdot Q^B_{p,q} \cdot e^{-\beta \cdot \mathcal{J}(i,j;p,q)} \tag{4.22}$$

is detected. Again, a function $t_3(i,j,p,q)$ that maps the loop cycles of $\sum_{i<j<p<q}$ to consecutive natural numbers can be constructed. Utilizing this function t_3, auxiliary variables $Z[t_3(i,j,p,q)]$ with

$$Z[t_3(i,j,p,q)] \quad = \quad \sum_{i' \leqslant i<j' \leqslant j<p' \leqslant p<q' \leqslant q} Q^B_{i',j'} \cdot Q^B_{p',q'} \cdot e^{-\beta \cdot \mathcal{J}(i',j';p',q')} \tag{4.23}$$

can be filled. The base pairs (i,j) and (p,q) wanted are determined if they fulfill the next inequality:

$$Z[t_3(i,j,p,q)-1] \leqslant r_6 \cdot Q^o - Q^o_H \quad < \quad Z[t_3(i,j,p,q)] \tag{4.24}$$

In the next recursion step, the remaining intervals $[i,j]_{Q^B}$ and $[p,q]_{Q^B}$ are backtracked further.

If neither (4.17) nor (4.21) are true, the exterior loop has to be a multi loop. To sample a statistically representative *exterior multi loop*, a split point k that separates a rightmost part with exactly two stems from the left part with at least one stem has to be determined. Regarding the energy contributions of exterior multi loops Q^o_M and its construction, the equilibrium probabilities P_k for each possible split point k is

$$P_k \quad = \quad \frac{Q^M_{1,k} \cdot Q^{M^2}_{k+1,n}}{Q^o_M} \tag{4.25}$$

Thus, using auxiliary variables $Z[k]$ with

$$Z[k] \quad = \quad \sum_{\nu \leqslant k} Q^M_{1,\nu} \cdot Q^{M^2}_{\nu+1,n} \tag{4.26}$$

and a further random number r_7 a split point k has to fulfill the following condition:

$$Z[k-1] < r_7 \cdot Q^o_M \quad \leqslant \quad Z[k] \tag{4.27}$$

After the position k is found the left part on the subsequence $[1,k]_{Q^M}$ is backtracked further by the regular linear stochastic backtracking recursions. The right part on subsequence $[k+1,n]_{Q^{M^2}}$ is backtracked according to recursion (4.15).

4.3 CONSENSUS STRUCTURE OF ALIGNED SEQUENCES

Alignments of circular RNA sequences are quite difficult to compute due to the arbitrary start point of each sequence. Nevertheless, the program `cyclope` [43] provides such circular alignments which opens up the investigation of consensus structures especially for viroids or classes of viroids. Particularly the computation of the partition function for the circular alignment $\mathcal{Q}^{\mathcal{A}o}$ and the resulting ability to compute the base pairing probabilities are of great interest.

4.3.1 *Partition function*

The already discussed partition function algorithm for aligned RNA sequences of section 2.3.4 can also be easily extended to take circular alignments into account using the post-processing scheme of 3.2. Therefore the recursions for the energy contributions of the exterior loops have to be altered according to equation (2.64).

POST-PROCESSING The first step in the post-processing prepares the evaluation of exterior multi loops by the construction of the auxiliary linear energy array $Q^{\mathcal{A}M^2}$. As multi loop parts with exactly two stems are derived by the concatenation of two parts with exactly one stem the recursion to fill $Q^{\mathcal{A}M^2}$ remains trivial.

$$Q^{\mathcal{A}M^2}_{k,n} = \sum_{k<u<n} Q^{\mathcal{A}M^1}_{k,u} \cdot Q^{\mathcal{A}M^1}_{u+1,n} \tag{4.28}$$

The energy contributions of *exterior hairpin loops* can also be extended easily to take all sequences in the alignment into account. Therefore the weighted energy contributions over all sequences have to be summed up. As the covariance contribution $e^{\rho\cdot\phi_2\cdot B_{i,j}}$ with $\rho = \frac{1}{RTN}$ of each closing base pair (i,j) is already added in the energy contribution $Q^{\mathcal{A}B}_{i,j}$, it does not have to be treated again. The

extended recursion can now be formulated as

$$Q_H^{\mathcal{A}o} = \sum_{1 \leqslant i < j \leqslant n} Q_{i,j}^{\mathcal{A}B} \cdot e^{-\rho \cdot \sum_{\alpha \in \mathcal{A}} \mathcal{H}(j_\alpha, i_\alpha)} \tag{4.29}$$

Applying the same method to the recursion for the *exterior interior loop* contribution results in the following extended equation.

$$Q_I^{\mathcal{A}o} = \sum_{i < j < p < q} Q_{i,j}^{\mathcal{A}B} \cdot Q_{p,q}^{\mathcal{A}B} \cdot e^{-\rho \cdot \sum_{\alpha \in \mathcal{A}} \mathcal{I}(i_\alpha, j_\alpha, p_\alpha, q_\alpha)} \tag{4.30}$$

Considering *exterior multi loops*, equation (4.4) can be extended without additional modifications.

$$Q_M^{\mathcal{A}o} = \sum_{k} Q_{1,k}^{\mathcal{A}M} \cdot Q_{k+1,n}^{\mathcal{A}M^2} \cdot e^{-\beta \cdot a} \tag{4.31}$$

Using the *memory efficient post-processing step*, the partition function of the circular alignment $\mathcal{Q}^{\mathcal{A}o}$ then is obtained by summation of the loop type dependent partition functions $Q_H^{\mathcal{A}o}$, $Q_I^{\mathcal{A}o}$ and $Q_M^{\mathcal{A}o}$.

$$\mathcal{Q}^{\mathcal{A}o} = Q_H^{\mathcal{A}o} + Q_I^{\mathcal{A}o} + Q_M^{\mathcal{A}o} \tag{4.32}$$

BASE PAIRING PROBABILITIES Calculating the equilibrium base pairing probabilities $P_{i,j}^{\mathcal{A}o}$ of the aligned circular RNA sequences differs in the calculation of the contribution of the *exterior loop* the base pair (i, j) is part of. As this contribution is covered by $P_{i,j}^{\mathcal{A}\ lin}$ in recursion (2.70) only this term has to be altered. Similarly to the calculation of the equilibrium base pairing probability for circular

RNAs in 4.7, $P_{i,j}^{\mathcal{A}\ lin}$ is subsituted by $P_{i,j}^{\mathcal{A}\ circ}$ with

$$
\begin{aligned}
P_{i,j}^{\mathcal{A}\ circ} \quad = \quad & \frac{Q_{i,j}^{\mathcal{A}B}}{Q^{\mathcal{A}o}} \cdot e^{\rho\cdot\phi_2\cdot B_{i,j}} \cdot \Bigg\{ \\
& e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} \mathcal{H}(j_\alpha,i_\alpha)} \\
& + \sum_{p<q<i<j} Q_{p,q}^{\mathcal{A}B} \cdot e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} \mathcal{I}(q_\alpha,p_\alpha,j_\alpha,i_\alpha)} \\
& + \sum_{i<j<p<q} Q_{p,q}^{\mathcal{A}B} \cdot e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} \mathcal{I}(p_\alpha,q_\alpha,j_\alpha,i_\alpha)} \\
& + Q_{1,i-1}^{\mathcal{A}M} \cdot Q_{j+1,n}^{\mathcal{A}\ M} \cdot e^{-\beta\cdot(a+b)} \cdot \{ \\
& \quad e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{j_\alpha,i_\alpha,(j-1)_\alpha}+d^3_{j_\alpha,i_\alpha,(i+1)_\alpha}} \} \\
& + \sum_{u<k} Q_{1,u}^{\mathcal{A}M} \cdot Q_{u+1,i-1}^{\mathcal{A}M^1} \cdot e^{-\beta\cdot(a+b+(n-u)\cdot c)} \cdot \{ \\
& \quad e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{j_\alpha,i_\alpha,(j-1)_\alpha}+d^3_{j_\alpha,i_\alpha,(i+1)_\alpha}} \} \\
& + \sum_{u>l} Q_{u+1,n}^{\mathcal{A}M} \cdot Q_{j+1,u}^{\mathcal{A}M^1} \cdot e^{-\beta\cdot(a+b+(i-1)\cdot c)} \cdot \{ \\
& \quad e^{-\rho\cdot\sum_{\alpha\in\mathcal{A}} d^5_{j_\alpha,i_\alpha,(j-1)_\alpha}+d^3_{j_\alpha,i_\alpha,(i+1)_\alpha}} \} \Bigg\}
\end{aligned}
\tag{4.33}
$$

5

RESULTS

5.1 IMPLEMENTATION

All previously discussed extensions of the folding algorithms in 4 were implemented in `C` using the source code version 1.6.4 of the `ViennaRNAPackage` and will be available in one of its next releases. Details about the implementation are shown in appendix A.

The circular variants of the discussed algorithms are available by using the command line parameter `-circ` to the appropriate programs `RNAfold`, `RNAsubopt` and `RNAalifold`. The modified manual pages (man pages) for these programs are listed in appendix B.

5.2 VALIDATION

To validate the implemented algorithms in terms of correctness, several testing scenarios were applied. All of them can be summarized by the principle that the results of these algorithms were compared with results that appear if the arbitrary cut point, which designates position 1 of the RNA sequence, is shifted throughout the complete sequence. If the algorithms work well, each result will be cut point independent. Specifically, single circular RNA sequences were used to test the implementation of the partition function, the base pairing probabilities and the computation of suboptimal secondary structures.

Furthermore, base pairing probabilites and partition function were calculated for comparison by taking the output of the algorithm of Wuchty et al. [69], which was set up to generate the complete secondary structure space. This was done for some random small artificial RNA sequences (up to 60 nucleotides) only because of the massive rise of possible secondary structures with increasing sequence length. Testing the reliability of the implemented circular extension of the folding

algorithms for aligned RNA sequences was performed in the same manner, except for the estimation of the base pairing probabilities, as the algorithm of Wuchty et al. does not deal with aligned RNA sequences.

All testing scenarios were executed by a collection of `perl`-scripts, that are available on request.

5.3 CUT-POINT SPECIFICITY

As already mentioned in [54] the computation of MFE secondary structures is highly cut-point dependant. This effect is not limited to the MFE calculations only. The partition function algorithms for single and aligned RNA sequences as well as the algorithms for the prediction of suboptimal secondary structures discussed are highly cut-point dependant, too. This results from the energy contributions of exterior loops and prohibited structures which are not taken into account without the circular extension.

To show this dependancy for the two discussed partition function algorithms, a single RNA sequence *(Acc. No. M16826)* [22] and 134 PSTVd RNA sequences taken from the `Subviral RNA Database` [49] to produce an alignment were used. The free energies of the ensemble $F = -kT \cdot \ln(Q)$ and $F^{\mathcal{A}} = -kT \cdot \ln(Q^{\mathcal{A}})$ were obtained from `RNAfold` and `RNAalifold` with parameters `-p -d2` at a default temperature of 37° Celsius. The starting point of the sequence(s) was shifted throughout the complete genome length. Resulting differences to the predicted "circular" free energy of the ensemble $F^{\circ} = -kT \cdot \ln(Q^{o})$ and $F^{\mathcal{A}\circ} = -kT \cdot \ln(Q^{\mathcal{A}o})$ are depicted in Fig. 15 and 16. In both plots a heavy cut-point specificity of F and $F^{\mathcal{A}}$ appears which clearly shows the necessity of an extension of the linear folding algorithms when investigating circular RNA sequences.

When predicting suboptimal secondary structures this effect can be crucial. Due to the different MFE for each base used as starting point of the RNA sequence, the amount of secondary stuctures within a percentage interval arround the MFE highly differs. This effect is depicted in Fig. 17. As a consequence of the cut-point dependancy of Q, statistically sampled representative secondary structures, obtained by the stochastic backtracking algorithm, differ in the distribution of their free energies as shown in Fig. 18. In this example, 100 samples per cut-point

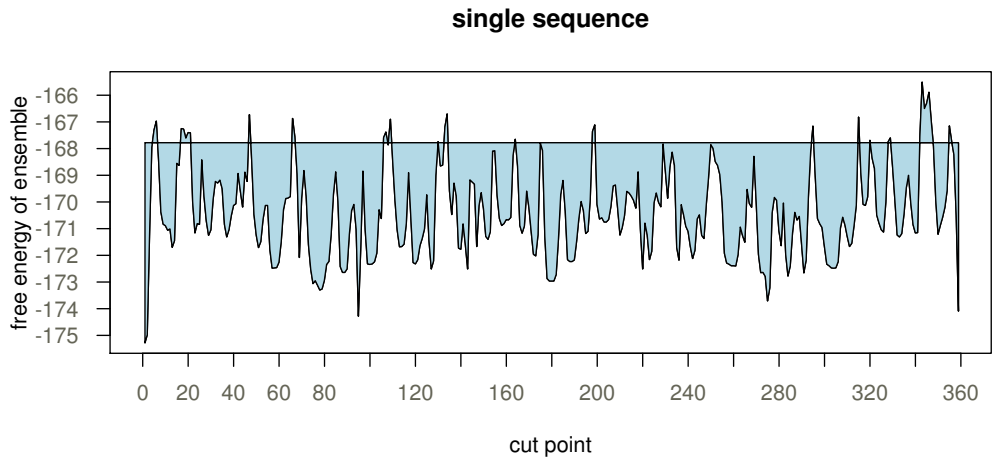

Figure 15: Free energy of the ensemble $F = -kT \cdot \ln(Q)$ in *kcal/mol* as a function of the cut-point of a circular RNA sequence.
Predicted F of PSTVd *(Acc. No. M16826)* [22] without exterior loop extension relative to predicted "circular" free energy of the ensemble $F^{\circ} = -167.78$*kcal/mol*.

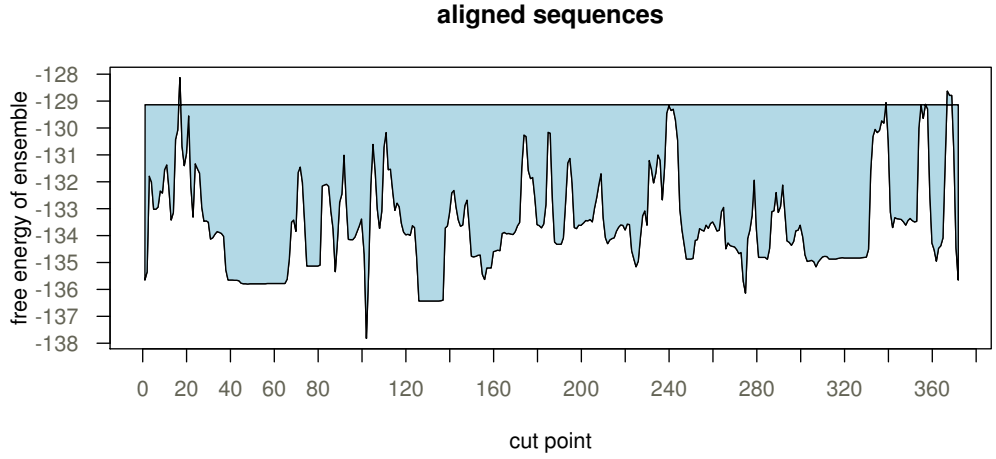

Figure 16: Free energy of the ensemble $F^{\mathcal{A}} = -kT * \ln(Q^{\mathcal{A}})$ in *kcal/mol* as a function of the cut-point of a circular RNA sequence alignment.
Alignment contains 134 sequences of PSTVd obtained from `Subviral RNA database` [49]. Depicted is the predicted F without exterior loop extension relative to the predicted "circular" free energy of the ensemble $F^{\mathcal{A}\circ} = -129.14$*kcal/mol*.

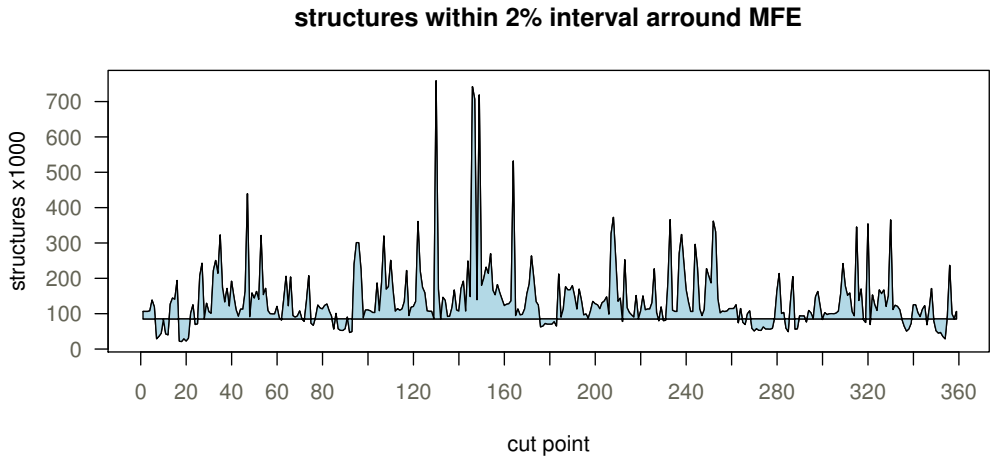

Figure 17: Number of secondary structures within 2% interval arround corresponding MFE as function of the cut-point.
Circular RNA sequence used is PSTVd *(Acc. No. M16826)* [22]. Number of structures without exterior loop extension relative to the amount of 85085 predicted secondary structures when folding with circular extension.

were generated. As expected, the distribution of the free energies per cut-point are quite homogeneous if the extension for circular RNAs is applied (Fig. 18 left). Without this extension, the cut-point induces shifts in the distribution of free energies (Fig. 18 right).

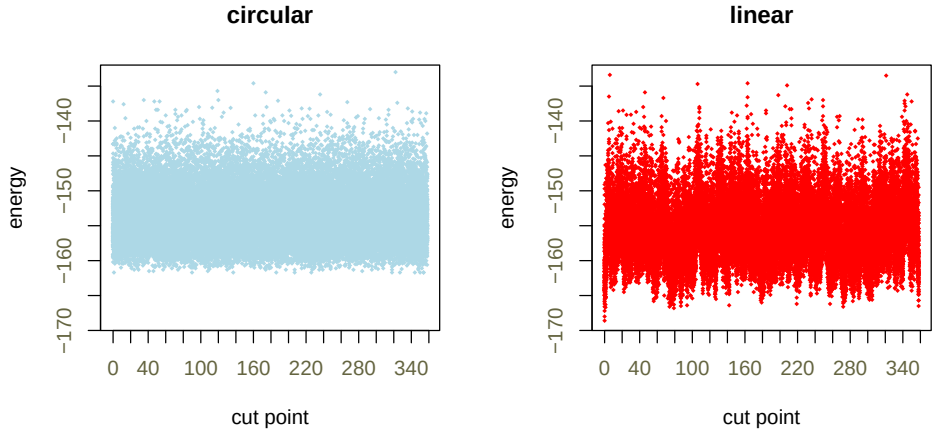

Figure 18: Free energies in *kcal/mol* of stochastically sampled suboptimal secondary structures depending on cut-point of circular PSTVd RNA sequence *(Acc. No. M16826)* [22].
Left: 100 samples per cut-point with circular option. *Right:* 100 samples per cut-point without circular option.

6

CONCLUSION AND OUTLOOK

RNAs play an important role in living cells. Their capability to operate as information carrier and to exhibit catalytic activity highlights them for studies of genotype-phenotype relationships. Circular RNAs like those of viroids which are suggested to be *living fossils* of a pre-cellular world [10] are predestined for such fields of investigation due to their small genome size that merely codes for its structure, and its concurrent enzymatic activity.

Studying RNA secondary structures yields useful information for the prediction of tertiary structure, and therefore the interpretation of biochemical interactions of the molecules. The discussed graph theoretic representation of discrete secondary structures makes them well compatible to efficient algorithms that compute thermodynamic quantities like the equilibrium partition function, minimum free energy of consensus structures or metastable states.

Several so called *dynamic programming* algorithms exist that, originating from the loop-based energy model, compute the secondary structure fold of an RNA sequence or an alignment of RNA sequences under certain thermodynamic circumstances. In this work, the partition function algorithm and the computation of equilibrium base pairing probabilities for single RNA sequences and alignments, the computation of suboptimal secondary structures with a free energy within an interval around the MFE and stochastic backtracking were formal applied to a common recursion scheme. Each of these algorithms was extended to not only take stabilizing base pairing or base pair stacking energies and destabilizing loop energies into account but also stabilizing contributions of so called *dangling ends* that occur if a nucleotide stacks onto an adjacent base pair.

A scheme of memory efficient folding algorithms for circular RNAs was taken to extend the algorithms discussed for acting on circular RNA sequences. The resulting algorithms were implemented in `C` as an extension of the available secondary structure predicting programs `RNAfold`, `RNAsubopt` and `RNAalifold` of the `ViennaRNAPackage`.

Thus, this work unlocks the application of widely-used secondary structure prediction algorithms to circular RNA molecules. As an integral part of the *ViennaRNAPackage* the `RNAsubopt` program now opens up the possibility to investigate the energy landscape of secondary structures of circular RNAs. This may lead to findings of bi- or multi-stable secondary structure states *in silico*, e. g. for viroid RNAs or even for artificially constucted circular RNA aptamers or riboswitches. The implementation of the memory efficient folding algorithm scheme for circular RNAs to `RNAalifold` furthermore facilitates the prediction of conserved structural elements in viroid families, or other circular RNA families where not the sequence itself but the adopted secondary structure is conserved.

A

IMPLEMENTATION

For the implementation of the circularized versions of the suboptimal algorithms, the partition function for single RNA strands and alignments some changes in the program code of the `ViennaRNAPackage` were necessary.

A.1 GENERAL CHANGES

To avoid redefinition of functions and variables during the compilation steps, the headerfile `fold_vars.h` was modified by enclosing all previous code with:

```
#ifndef __FOLD_VARS__
#define __FOLD_VARS__
/* previously existing code follows here */
<code>
#endif
```

A.2 CHANGES CONCERNING PARTITION FUNCTION

Obtaining the partition function Q or the base paring probabilities $P_{i,j}$ of a given RNA sequence is done with the program `RNAfold`. Therefore, its source code had to be modified for the extension with a post-processing step for the calculation of Q^o and pre-processing steps for the determination of $P^o_{i,j}$ and *stochastic backtracking*. Although the latter - *stochastic backtracking* - relates to the program `RNAsubopt` that summarizes the algorithms for suboptimal RNA folding in this package, the implementation of *stochastic backtracking* is done in the source code files that constitute `RNAfold`. The following files were modified.

A.2.1 *RNAfold.c*

Function calls to get MFE and partition function in condition

```
if(pf){
  ...
}
```

were changed to take circularized version into account.

A.2.2 *part_func.c*

At the top of this source file some new global variables and function definitions were added

```
/* new variables */
int circ = 0;
FLT_OR_DBL qo, qho, qio, qmo, *qm2;
/* new functions */
PUBLIC  float pf_circ_fold(char *sequence, char *structure);
PRIVATE void  pf_circ(char *sequence, char *structure);
PRIVATE void  pf_linear(char *sequence, char *structure);
PRIVATE void  pf_create_bppm(char *sequence, char *structure);
PUBLIC  char  *pbacktrack_circ(char *seq);
static  void  backtrack_qm(int i, int j);
static  void  backtrack_qm2(int u, int n);
```

Due to the implementation of the post- and pre-processing step some existing functions had to be modified to reuse as much existing code as possible.

Taking out the forward and backward recursions previously combined in

```
pf_fold(char *sequence, char *structure)
```

into own functions was one of the next modifications. For compatibility with other parts of the `ViennaRNAPackage` this function still performs the separated recursions by calling the two additional new functions

```
pf_linear(char *sequence, char *structure) /* forward recursion to fill arrays */
pf_create_bppm(char *sequence, char *structure) /* backward recursion to calculate
     base paring probabilities */
```

Newly introduced was the function

```
pf_circ_fold(char *sequence, char *structure)
```

that controls the execution of the linear forward recursion and the post-processing step afterwards when considering a circular RNA.

For the allocation and freeing of memory used for the auxilary energy array Q^{M^2} the existing functions

```
void get_arrays(unsigned int length) /* Allocation of memory for qm1 and qm2 array in
     case of circular folding */
void free_pf_arrays(void) /* Freeing of qm2 array if it was initialized before */
```

were extended, too.

The code inside the newly introduced

```
pf_linear(char *sequence, char *structure) /* perform forward recursion for linear RNA
    */
```

previously was part of the "old" `pf_fold()` function. It just implements the forward recursion to fill all energy arrays. Changes of the exisiting code were done concerning the allocation of memory for the two arrays that contain the encoded RNA sequence S and S1. Their length was extended to provide the last base n of the sequence in front of it (0) and the first base 1 at the end ($n+1$). This avoids exhaustive if-conditions within the code especially when calculating dangling-end contributions. Additionally, the dangling-end calculations were modified to take the contribution of base n stacking on base pair $(1, i)$ and base 1 stacking onto base pair (j, n) into account when analyzing a circular RNA.

The function

```
void pf_circ(char *sequence, char *structure) /* perform post-processing for circular
    RNA */
```

implements the post-processing step as discussed in 4.1. It fills the Q^{M^2} array and also stores the contribution of exterior hairpin loops Q^o_H, exterior interior loops Q^o_I and exterior multiloops Q^o_M in the appropriate variables `qho`, `qio` and `qmo`, respectively.

As already mentioned, the computation of the base pairing probability matrix was externalized to allow a separate call of this backward recursion. Therefore,

```
pf_create_bppm(char *sequence, char *structure) /* compute base pairing probabilities
     */
```

was created. It implements previously existing code of the "old" `pf_fold()` function with some slight changes. The main modification was done in the calculation of the contribution of the exterior loop that has to be replaced by the pre-processing step when considering circular RNAs.

Some of the existing variables and function definitions neccessary for stochastic backtracking were moved from within the code to the top of the source file.

```
static void backtrack(int i, int j);
static void backtrack_qm1(int i,int j);
static char *pstruc;
static char *sequence;
```

Stochastic backtracking of a secondary structure of a circular RNA sequence also leads to the chance to reuse the program block that is neccessary for backtracking in the Q^M array. This block is now implemented as an own function, namely

```
static void backtrack_qm(int i, int j) /* was previously part of the function
     backtrack(int i,int j) */
```

Other new functions for stochastic backtracking of circular RNA secondary structures were implemented, too.

```
 char *pbacktrack_circ(char *seq) /* get statistically representative circular RNA
      secondary structure */
```

This function backtracks the exterior part of a circular RNA. It calls the regular `backtrack(int i, int j)` function for the interior part(s) and returns a statistically representative secondary structure as `char *pbacktrack(int i, int j)` does in the linear case.

Backtracking in the auxilary array Q^{M^2} array leads to the following function.

```
static void backtrack_qm2(int k, int n) /* backtracking in qm2 */
```

It determines the barrier position u between the two concatenated $Q^{M^1}_{k,u}$ and $Q^{M^1}_{u+1,n}$ and continues with backtracking in both of the them.

A.2.3 *part_func.h*

Previous code inside this header file is now enclosed by

```
#ifndef __PART_FUNC__
#define __PART_FUNC__
<code>
#endif
```

to avoid redefinitions. Additionally the definitions for the newly implemented functions in `part_func.c` are added to make them available for other parts of the `ViennaRNAPackage`.

```
extern  char  *pbacktrack(char *sequence);
extern  float pf_circ_fold(char *sequence, char *structure);
extern  char  *pbacktrack_circ(char *seq);
```

A.3 CHANGES CONCERNING SUBOPTIMAL FOLDING

The `ViennaRNAPackage` implements two algorithms for the calculation of suboptimal secondary structures. The algorithm of Wuchty et al. [69] and *stochastic backtracking* as introduced in 2.3.3. Both algorithms are available via the program `RNAsubopt`.

For the implementation of the pre-processing step that modify these algorithms to take circular RNA into account as shown in 4.2.1 and 4.2.2, the appropriate program source has to be modified too.

A.3.1 *RNAsubopt.c*

Some small modifications were done in this source code file. First, the function definition

```
extern char *pbacktrack(char *sequence);
```

was removed due to its insertion into `part_func.h`. Furthermore, in function

```
int main(int argc, char *argv[])
```

that controls the program flow, a new state variable

```
int circ=0;
```

was inserted, providing a switch to toggle between the linear and circular algorithm. Along with this, some slight other modifications were neccessary, controling the function calls according to the selection.

A.3.2 *subopt.c*

In this file, the complete algorithm of Wuchty et al. [69] is implemented. According to 4.2.1, new definitions of variables and functions in the head of the source code file were neccessary.

```
extern    int circ;
SOLUTION  *subopt_circ(char *seq, char *sequence, int delta, FILE *fp);
int       *fM2; /* energies of M2 */
int       Fc, FcH, FcI, FcM; /* parts of the exterior loop energies  */
```

Modifications were done in

```
void encode_seq(char *sequence)
```

that allocates memory for the RNA sequence containing arrays used and in

```
int best_attainable_energy(STATE * state)
```

where the best attainable energy of all remaining subsequences of the current state is determined.

Since

```
SOLUTION *subopt(char *seq, char *structure, int delta, FILE *fp)
```

handles the algorithms function call procedure several changes had to be done to execute the appropriate modifications for circular RNAs.

The pre-processing step that computes the exterior loop contributions as shown in 4.2.1 was implemented by modifying

```
void scan_interval(int i, int j, int array_flag, STATE * state)
```

An overloading of `SOLUTION *subopt(char *seq, char *structure, int delta, FILE *fp)` sets the external state variable `extern int circ = 1;` and returns the original function call afterwards.

```
SOLUTION *subopt_circ(char *seq, char *structure, int delta, FILE *fp)
```

A.3.3 *subopt.h*

The only modification in this header file was the insertion of the function definition

```
extern SOLUTION *subopt_circ (char *seq, char *sequence, int delta, FILE *fp);
```

to make it available for other contexts within the `ViennaRNAPackage`.

Some major changes were done in the source files constituting the implementation of the circularized MFE algorithm as they are also part of the `RNAsubopt` program.

A.3.4 *fold.c*

Forward and backward recursions of the circular MFE algorithm in the `ViennaRNAPackage` are implemented in the files `fold.c` and `circfold.inc`. To make both of them available separately as needed for `RNAsubopt` some former private and also new variables were introduced as global ones

```
int circ = 0; /* state variable to toggle circfold ON/OFF */
int *fM2; /* linear auxilary multiloop array M2 */
int Fc, FcH, FcI, FcM; /* exterior loop energies */
```

Existing functions had to be modified, especially due to the memory management of the arrays used.

```
void get_arrays(unsigned int size)
void free_arrays(void)
void encode_seq(const char *sequence)
float fold(const char *string, char *structure)
```

A new function that allows the memory efficient export of all energy variables and arrays after the forward recursion of the MFE algorithm including the post-processing step for circular RNA was implemented, too.

```
void export_circfold_arrays(int *Fc_p, int *FcH_p, int *FcI_p, int *FcM_p, int **
    fM2_p, int **f5_p, int **c_p, int **fML_p, int **fM1_p, int **indx_p, char **
    ptype_p):
```

A.3.5 *fold.h*

To make the new functions available for other programs, linked against the implementations in `fold.c` or linked against the provided `libRNA` that provides an interface to the folding algorithms, the following function definitions were added

```
extern void  export_fold_arrays(int **f5_p, int **c_p, int **fML_p, int **fM1_p, int
    **indx_p, char **ptype_p);
extern float circfold(const char *string, char *structure);
extern float energy_of_circ_struct(const char *string, const char *structure);
extern void  export_circfold_arrays(int *Fc_p, int *FcH_p, int *FcI_p, int *FcM_p,
    int **fM2_p, int **f5_p, int **c_p, int **fML_p, int **fM1_p, int **indx_p, char
    **ptype_p);
```

A.3.6 *circfold.inc*

Due to the changes in `fold.c` that assimilated some previously defined variables and even code fragments of this file, several modifications were required.

A.4 CHANGES CONCERNING ALIGNMENT FOLDING

As the implementation of the algorithms for computing the partition function Q^{Ao} and the base pairing probabilities $P_{i,j}^{Ao}$, the source code of the program `RNAalifold` was modified.

A.4.1 *RNAalifold.c*

All modifications done in this file are similar to the appropriate ones in `RNAfold.c` that arized when modifying the calls of the partition function and the base pairing

probability algorithm to take circular RNAs into account.

A.4.2 *alipfold.c*

New variables were applied

```
int circ=0, *jindx; /* state control switch and index variable */
FLT_OR_DBL qo, qho, qio, qmo, *qm2, *qob; /* energy arrays */
```

to cover the energy contributions of the exterior loop types.

The previously existing function `float alipf_fold(char **sequences, char *structure, pair_info **pi)` was split into separate functions for forward and backward recursion similar to the modification of the partition function for single circular RNA sequences. Nevertheless, this function can be called as usually before as it internally executes forward and backward recursion similar to `pf_fold(char *sequence, char *structure)` of `part_func.c` again.

The resulting new functions

```
float alipf_linear(char **sequences, char *structure) /* fill energy arrays */
void  alipf_create_bppm(char **sequences, char *structure, pair_info **pi) /*
     calculate base pairing probabilities */
```

are self-explanatory and all modifications of previously existing source code blocks follow the principles used in the appropriate functions `pf_linear(char *sequence, char *structure)` and `pf_create_bppm(char *sequence, char *structure)` used in the single-sequence case.

Another resulting new function is

```
void alipf_circ(char **sequences, char *structure) /* do post-processing for circular
     RNAs */
```

that implements the complete post-processing step which has to be executed when computing the partition function Q^{Ao}. The underlying algorithmic principles are similar to those used in the single-sequence case, again.

Finally,

```
float alipf_circ_fold(char **sequences, char *structure, pair_info **pi)
```

summarizes all neccessary function calls for the calculation of the partition function and the base pairing probabilities for alignments of circular RNAs.

Regardless of that further changes were applied by modifying existing functions that mainly do memory management during the computations according to the extended requirements.

```
void  get_arrays(unsigned int length) /* allocate memory */
void  free_alipf_arrays(void) /* free memory */
short *encode_seq(const char *sequence) /* encode RNA sequence */
```

A.4.3 *alifold.h*

Function definition

```
extern float alipf_circ_fold(char **sequences, char *structure, pair_info **pi);
```

was added in the appropriate header file.

B

MANUAL PAGES

This chapter contains the modified man pages of the altered programs within the `ViennaRNAPackage`.

B.1 RNAFOLD MAN PAGES

B.1.1 *Name*

RNAfold - calculate secondary structures of RNAs

B.1.2 *Synopsis*

RNAfold [-p[*0*|*2*]] [-C] [-T *temp*] [-4] [-d[*0*|*1*|*2*|*3*]] [-noLP] [-noGU] [-noCloseGU] [-e *1*|*2*] [-P *paramfile*] [-nsp *pairs*] [-S *scale*] [-circ]

B.1.3 *Description*

RNAfold reads RNA sequences from stdin, calculates their minimum free energy (mfe) structure and prints to stdout the mfe structure in bracket notation and its free energy. If the -p option was given it also computes the partition function (pf) and base pairing probability matrix, and prints the free energy of the thermodynamic ensemble, the frequency of the mfe structure in the ensemble, and the ensemble diversity to stdout. It also produces PostScript files with plots of the resulting secondary structure graph and a "dot plot" of the base pairing matrix. The dot plot shows a matrix of squares with area proportional to the pairing probability in the upper right half, and one square for each pair in the minimum free energy structure in the lower left half. For each pair i-j with probability

p>10E-6 there is a line of the form

i j sqrt(p) ubox

in the PostScript file, so that the pair probabilities can be easily extracted. Sequences are read in a simple text format where each sequence occupies a single line. Each sequence may be preceded by a line of the form

> name

to assign a name to the sequence. If a name is given in the input PostScript files "name_ss.ps" and "name_dp.ps" are produced for

the structure and dot plot, respectively. Otherwise the file names default to rna.ps and dot.ps. Existing files of the same name will be overwritten.

The input format is similar to fasta except that even long sequences may not be interrupted by line breaks, and the header lines are optional. The program will continue to read new sequences until a line consisting of the single character @ or an end of file condition is encountered.

B.1.4 *Options*

-p Calculate the partition function and base pairing probability matrix in addition to the mfe structure. Default is calculation of mfe structure only. Prints a coarse representation of the pair probabilities in form of a pseudo bracket notation, the ensemble free energy, the frequency of the mfe structure, and the structural diversity. See the description of pf_fold() and mean_bp_dist() in the RNAlib documentation for details.

Note that unless you also specify -d2 or -do, the partition function and mfe calculations will use a slightly different energy model. See the discussion of dangling end options below.

-p0 Calculate the partition function but not the pair probabilities, saving about 50% in runtime. Prints the ensemble free energy -kT ln(Z).

-p2 In addition to pair probabilities compute stack probabilities, i.e. the probability that a pair (i,j) and the immediately interior pair (i+1,j-1) are formed simultaneously. A second postscript dot plot called "name_dp2.ps", or "dot2.ps" (if the sequence does not have a name), is produced that contains

pair probabilities in the upper right half and stack probabilities in the lower left.

-C Calculate structures subject to constraints. The program reads first the sequence, then a string containing constraints on the structure encoded with the symbols: | (the corresponding base has to be paired x (the base is unpaired) < (base i is paired with a base j>i) > (base i is paired with a base j<i) and matching brackets ( ) (base i pairs base j) With the exception of "|", constraints will disallow all pairs conflicting with the constraint. This is usually sufficient to enforce the constraint, but occasionally a base may stay unpaired in spite of constraints. PF folding ignores constraints of type "|".

-T ***temp*** Rescale energy parameters to a temperature of *temp* C. Default is 37C.

-4 Do not include special stabilizing energies for certain tetra-loops. Mostly for testing.

-d[0|1|2|3] How to treat "dangling end" energies for bases adjacent to helices in free ends and multi-loops: With (-d1) only unpaired bases can participate in at most one dangling end, this is the default for mfe folding but unsupported for the partition function folding. With **-d2** this check is ignored, dangling energies will be added for the bases adjacent to a helix on both sides in any case; this is the default for partition function folding (-p). **-d** or **-d0** ignores dangling ends altogether (mostly for debugging).

With **-d3** mfe folding will allow coaxial stacking of adjacent helices in multi-loops. At the moment the implementation will not allow coaxial stacking of the two interior pairs in a loop of degree 3 and works only for mfe folding.

Note that by default (as well as with -d1 and -d3) pf and mfe folding treat dangling ends differently. Use **-d2** in addition to **-p** to ensure that both algorithms use the same energy model.

-noLP Produce structures without lonely pairs (helices of length 1). For partition function folding this only disallows pairs that can **only** occur isolated. Other pairs may still occasionally occur as helices of length 1.

-noGU Do not allow GU pairs.

-noCloseGU Do not allow GU pairs at the end of helices.

-e *1*|*2* Rarely used option to fold sequences from the artificial ABCD... alphabet, where A pairs B, C-D etc. Use the energy parameters for GC (-e 1) or AU (-e 2) pairs.

-P <***paramfile***> Read energy parameters from *paramfile*, instead of using the default parameter set. A sample parameter file should accompany your distribution. See the RNAlib documentation for details on the file format.

-nsp ***pairs*** Allow other pairs in addition to the usual AU,GC,and GU pairs. *pairs* is a comma separated list of additionally allowed pairs. If a the first character is a "-" then AB will imply that AB and BA are allowed pairs. e.g. RNAfold -nsp -GA will allow GA and AG pairs. Nonstandard pairs are given 0 stacking energy.

-S ***scale*** In the calculation of the pf use *scale**mfe as an estimate for the ensemble free energy (used to avoid overflows). The default is 1.07, useful values are 1.0 to 1.2. Occasionally needed for long sequences. You can also recompile the program to use double precision (see the README file).

-circ Assume a circular (instead of linear) RNA molecule.

-noPS Do not produce postscript drawing of the mfe structure.

B.1.5 *References*

The calculation of mfe structures is based on dynamic programming algorithm originally developed by M. Zuker and P. Stiegler. The partition function algorithm is based on work by J.S. McCaskill. The energy parameters are taken from:

D.H. Mathews, J. Sabina, M. Zuker and H. Turner "Expanded Sequence Dependence of Thermodynamic Parameters Provides Robust Prediction of RNA Secondary Structure" JMB, 288, pp 911-940, 1999

A. Walter, D Turner, J Kim, M Lyttle, P M[:u]ller, D Mathews, M Zuker "Coaxial stacking of helices enhances binding of oligoribonucleotides.." PNAS, 91, pp 9218-9222, 1994

If you use this program in your work you might want to cite:

I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoeffer, M. Tacker, P. Schuster (1994)

Fast Folding and Comparison of RNA Secondary Structures. Monatshefte f. Chemie 125: 167-188

M. Zuker, P. Stiegler (1981) Optimal computer folding of large RNA sequences using thermodynamic and auxiliary information, Nucl Acid Res 9: 133-148

J.S. McCaskill (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structures, Biopolymers 29: 1105-1119

I.L. Hofacker & P.F. Stadler (2006) Memory Efficient Folding Algorithms for Circular RNA Secondary Structures, Bioinformatics (2006)

D. Adams (1979) The hitchhiker's guide to the galaxy, Pan Books, London

B.1.6 *Version*

This man page documents version 1.6.4 Vienna RNA Package.

B.1.7 *Authors*

Ivo L Hofacker, Walter Fontana, Sebastian Bonhoeffer, Peter F Stadler.

B.1.8 *Bugs*

If in doubt our program is right, nature is at fault. Comments should be sent to rna@tbi.univie.ac.at.

B.2 RNASUBOPT MAN PAGES

B.2.1 *Name*

RNAsubopt - calculate suboptimal secondary structures of RNAs

B.2.2 *Synopsis*

RNAsubopt [-e *range*] [-ep *prange*] [-s] [-p *n*] [-T *temp*] [-d[0|1|2|3]] [-4] [-noGU] [-noCloseGU] [-P paramfile] [-logML] [-nsp pairs] [-circ]

B.3 DESCRIPTION

RNAsubopt reads RNA sequences from stdin and (in the default -e mode) calculates all suboptimal secondary structures within a user defined energy range above the minimum free energy (mfe).It prints the suboptimal structures in bracket notation followed by the energy in kcal/mol to stdout. Be careful, the number of structures returned grows exponentially with both sequence length and energy range. Alternatively, when used with the -p option, RNAsubopt produces Boltzmann weighted samples of secondary structures. Sequences are read in the usual format, i.e. each sequence occupies a single line, possibly preceded by a fasta-style header line of the form

> name

B.3.1 *Options*

-e *range* Calculate suboptimal structures within *range* kcal/mol of the mfe. Default is 1.

-s Sort the structures by energy. Since the sort in is done in memory, this becomes impractical when the number of structures produced goes into millions. In such cases better pipe the output through 'sort +1n'.

-p *n* Instead of producing all suboptimals in an energy range, produce a random sample of *n* suboptimal structures, drawn with probabilities equal to their Boltzmann weights via stochastic backtracking in the partition function. The -e and -p options a mutually exclusive.

-d[0|1|2|3] Change treatment of dangling ends, as in RNAfold and RNAeval. The default is -d2 (as in partition function folding). If -d1 or -d3 are specified

the structures are generated as with -d2 but energies are re-evaluated before printing.

-logML re-calculate energies of structures using a logarithmic energy function for multi-loops before output. This option does not effect structure generation, only the energies that is printed out. Since logML lowers energies somewhat, some structures may be missing.

-ep *prange* Only print structures with energy within *prange* of the mfe. Useful in conjunction with -logML, -d1 or -d3: while the -e option specifies the range before energies are re-evaluated, -ep specifies the maximum energy after re-evaluation.

-noLP Only produce structures without lonely pairs (helices of length 1). This reduces the number of structures drastically and should therefore be used for longer sequences and larger energy ranges.

-circ Assume a circular (instead of linear) RNA molecule.

The **-T**, **-4**, **-noGU**, **-noCloseGU**, **-P**, **-nsp**, options work as in RNAfold.

B.3.2 *References*

Please cite:

S. Wuchty, W. Fontana, I. L. Hofacker and P. Schuster "Complete Suboptimal Folding of RNA and the Stability of Secondary Structures", Biopolymers, 49, 145-165 (1999)

B.3.3 *Version*

This man page documents version 1.6.4 Vienna RNA Package.

B.3.4 *Authors*

Ivo L Hofacker, Stefan Wuchty, Walter Fontana.

Send comments and bug reports to <rna@tbi.univie.ac.at> LocalWords: RNA-SUBOPT RNAsubopt suboptimal RNAs fBRNAsubopt fIrange fP ep

LocalWords: prange lodos fItemp noGU noCloseGU paramfile logML nsp stdin

LocalWords: mfe kcal mol stdout TP RNAfold RNAeval multi fIprange noLP fB

LocalWords: br Wuchty Fontana Hofacker Schuster Biopolymers Ivo

B.4 RNAALIFOLD MAN PAGES

B.4.1 *Name*

RNAalifold - calculate secondary structures for a set of aligned RNAs

B.4.2 *Synopsis*

RNAalifold [-cv *weight*] [-nc *weight*] [-E] [-p[*o*]] [-C] [-T *temp*] [-4] [-d] [-noLP] [-noGU] [-noCloseGU] [-e *1*|*2*] [-P *paramfile*] [-nsp *pairs*] [-S *scale*] [-circ] [<*file.aln*>]

B.4.3 *Description*

RNAalifold reads aligned RNA sequences from stdin or *file.aln* and calculates their minimum free energy (mfe) structure, partition function (pf) and base pairing probability matrix. Currently, the input alignment has to be in CLUSTAL format. It returns the mfe structure in bracket notation, its energy, the free energy of the thermodynamic ensemble and the frequency of the mfe structure in the ensemble to stdout. It also produces Postscript files with plots of the resulting secondary structure graph ("alirna.ps") and a "dot plot" of the base pairing matrix ("alidot.ps"). The file "alifold.out" will contain a list of likely pairs sorted by credibility, suitable for viewing with "AliDot.pl". Be warned that output file will overwrite any existing files of the same name.

B.4.4 *Options*

-cv ***factor*** Set the weight of the covariance term in the energy function to *factor*. Default is 1.

-nc ***factor*** Set the penalty for non-compatible sequences in the covariance term of the energy function to *factor*. Default is 1.

-E Score pairs with endgaps same as gap-gap pairs.

-mis Output "most informative sequence" instead of simple consensus: For each column of the alignment output the set of nucleotides with frequence greater than average in IUPAC notation.

-p Calculate the partition function and base pairing probability matrix in addition to the mfe structure. Default is calculation of mfe structure only.

-noLP Avoid structures without lonely pairs (helices of length 1). In the mfe case structures with lonely pairs are strictly forbidden. For partition function folding this disallows pairs that can **only** occur isolated. Setting this option provides a significant speedup.

-circ Assume circular (instead of linear) RNA molecules.

-color Produce a colored version of the consensus strcture plot "alirna.ps" (default b&w only).

-aln Produce a colored and structure annotated alignment in PostScript format in the file "aln.ps" in the current directory.

The **-T**, **-d**, **-4**, **-noGU**, **-noCloseGU**, **-e**, **-P**, **-nsp**, options should work as in RNAfold If using **-C** constraints will be read from stdin, the alignment has to given as a filename on the command line.

B.4.5 *Caveats*

Since gaps are not removed for the evaluation of energies, it may be of advantage to remove any columns with more than, say, 75% gaps from the alignment before

folding with RNAalifold. Sequences are not weighted. If possible, do not mix very similar and dissimilar sequences. Duplicate sequences, for example, can distort the prediction.

B.4.6 *See Also*

The ALIDOT package http://www.tbi.univie.ac.at/RNA/ALIDOT/

B.4.7 *References*

The algorithm is a variant of the dynamic programming algorithms of M. Zuker and P. Stiegler (mfe) and J.S. McCaskill (pf) adapted for sets of aligned sequences with covariance information. The energy parameters are taken from:

D.H. Mathews, J. Sabina, M. Zuker and H. Turner "Expanded Sequence Dependence of Thermodynamic Parameters Provides Robust Prediction of RNA Secondary Structure" JMB, 288, pp 911-940, 1999

If you use this program in your work you might want to cite:

Ivo L. Hofacker, Martin Fekete, and Peter F. Stadler "Secondary Structure Prediction for Aligned RNA Sequences" J.Mol.Biol. 319: 1059-1066 (2002).

B.4.8 *Version*

This man page documents version 1.6.4 of the Vienna RNA Package.

B.5 AUTHORS

Ivo L Hofacker <ivo@tbi.univie.ac.at>

B.5.1 *Bugs*

If in doubt our program is right, nature is at fault. Comments should be sent to rna@tbi.univie.ac.at. LocalWords: RNAalifold ViennaRNA RNAs fBRNAalifold fP fI fItemp noLP noGU

LocalWords: noCloseGU fIparamfile nsp fIpairs fIscale fIfile aln stdin mfe

LocalWords: alirna ps alidot alifold AliDot TP cv fIfactor nc fBonly fB br

LocalWords: fUshould RNAfold Stiegler McCaskill JMB Fekete CLUSTAL stdout

BIBLIOGRAPHY

[1] Douglas Adams. *The Restaurant at the End of the Universe*. Pan Macmillan, 1980. ISBN 0-345-39181-0. (Cited on page v.)

[2] E. Biondi, S. Branciamore, L. Fusi, S. Gago, and E. Gallori. Catalytic activity of hammerhead ribozymes in a clay mineral environment: Implications for the RNA world. *Gene*, 389:10–18, 2007. (Cited on page 9.)

[3] T.R Cech. A model for the RNA-catalyzed replication of RNA. *Proc. Natl. Acad. Sci. USA*, 83, 1986. (Cited on page 9.)

[4] M. Chaffai, P. Serra, M. Gandia, C. Hernandez, and N. Duran-Vila. Molecular characterization of CEVd strains that induce different phenotypes in gynura aurantiaca: structure-pathogenicity relationships. *Archives of Virology*, 152, 2007. (Cited on page 1.)

[5] E. Chargaff. Chemical specificity of nucleic acids and mechanism of their enzymatic degradation. *Experientia*, 6:201–240, 1950. (Cited on page 7.)

[6] F.M. Codoner, J.-A. Daros, R.V. Sole, and S.F. Elena. The fittest versus the flattest: Experimental confirmation of the quasispecies effect with subviral pathogens. *PLoS Pathogens*, 2(12), December 2006. (Cited on page 9.)

[7] J.-A. Daros, S.F. Elena, and R. Flores. Viroids: an ariadnes's thread into the RNA labyrinth. *EMBO reports*, 7(6), 2006. (Cited on page 1.)

[8] J. Demongeot and A. Moreira. A possible circular RNA at the origin of life. *Journal of Theoretical Biology*, 2007. doi: 10.1016/j.jtbi.2007.07.010. in press. (Cited on page 9.)

[9] F. Di Serio, J.A. Daros, A. Ragozzino, and R. Flores. Close structural relationship between two hammerhead viroid-like RNAs associated with cherry chlorotic rusty spot disease. *Archives of Virology*, 151, 2006. (Cited on page 2.)

[10] T.O. Diener. Circular RNAs: Relics of precellular evolution? *Proc. Natl. Acad. Sci. USA*, 86, 1989. (Cited on pages 2 and 71.)

[11] B. Ding, A. Itaya, and X. Zhong. Viroid trafficking: a small RNA makes a big move. *Current Opinion in Plant Biology*, 8:606–612, 2005. (Cited on page 1.)

[12] Y. Ding and C. E. Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acid. Res.*, 31(24):7280–7301, December 2003. ISSN 1362-4962. URL http://nar.oxfordjournals.org/cgi/content/abstract/31/24/7280. (Cited on page 38.)

[13] Y. Ding and C. E. Lawrence. A bayesian statistical algorithm for RNA secondary structure prediction. *Computers and Chemistry*, 23:387–400, 1999. (Cited on page 38.)

[14] S.F. Elena, J. Dopazo, R. Flores, T.O. Diener, and A. Moya. Phylogenies of viroids, viroidlike satellite RNAs, and the viroidlike domain of hepatitis delta virus RNA. *Proc. Natl. Acad. Sci. USA*, 88:5631–5634, 1991. (Cited on page 43.)

[15] C. Flamm, I.L. Hofacker, S. Maurer-Stroh, F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7, 2001. (Cited on page 57.)

[16] R. Flores, J.W. Randles, M. Bar-Joseph, and T.O. Diener. Subviral agents: viroids. In M. H. V. van Regenmortel et al., editor, *Virus taxonomy, 7th Report of the International Committee on Taxonomy of Viruses*. (Cited on page 1.)

[17] J.R. Fresco, B.M. Alberts, and P. Doty. Some molecular details of the secondary structure of ribonucleic acid. *Nature*, 188, 1960. (Cited on page 12.)

[18] W. Gilbert. Origin of life: The RNA world. *Nature*, 319:618, 1986. (Cited on page 9.)

[19] G. Gomez and V. Pallas. Mature monomeric forms of hop stunt viroid resist RNA silencing in transgenic plants. *The Plant Journal*, 51, 2007. (Cited on page 1.)

[20] A. Gora-Stochacka. Viroids: unusaual small pathogenic rnas. *Acta Biochemica Polonica*, 51(3), 2004. (Cited on page 1.)

[21] S.O. Gudima, J. Chang, and J.M. Taylor. Restoration in vivo of defective hepatitis delta virus RNA genomes. *RNA*, 12, 2006. (Cited on page 1.)

[22] R.W. Hammond and R.A. Owens. Mutational analysis of potato spindle tuber viroid reveals complex relationships between structure and infectivity. *roc. Natl. Acad. Sci. USA*, 84:3967–3971, 1987. (Cited on pages 66, 67, 68, and 69.)

[23] I.F. Hassen, S. Massart, J. Motard, S. Roussel, O. Parisi, J. Kummert, M. Fakhfakh, H. Marrakchi, J.P. Perreault, and M.H. Jijakli. Molecular features of new peach latent mosaic viroid variants suggest that recombination may have contributed to the evolution of this infectious RNA. *Virology*, 360:50–57, 2007. (Cited on page 9.)

[24] R.M. Hazen, P.L. Griffin, J.M. Carothers, and J.W. Szostak. Functional information and the emergence of biocomplexity. *PNAS*, 104, 2007. (Cited on page 9.)

[25] C. Hernandez and R. Flores. Plus and minus RNAs of peach latent mosaic viroid self-cleave in vitro via hammerhead structures. *Proc. Natl. Acad. Sci. USA*, 89, 1992. (Cited on page 2.)

[26] I. L. Hofacker, M. Fekete, and P. F. Stadler. Secondary structure prediction for aligned RNA sequences. *J Mol Biol*, 319(5):1059–1066, June 2002. URL http://dx.doi.org/10.1016/S0022-2836(02)00308-X. (Cited on pages 43 and 46.)

[27] I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoeffer, M Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, 125, 1994. (Cited on pages 2, 8, 12, 15, and 38.)

[28] Ivo L. Hofacker and Peter F. Stadler. Memory efficient folding algorithms for circular RNA secondary structures. *Bioinformatics*, 22:1172–1176, 2006. Earlier version in: *German Conference on Bioinformatics 2005*, Torda, Andrew and Kurtz, Stefan and Rarey, Matthias (eds.), Lecture Notes in Informatics **P-71**, pp 3-13, Gesellschaft f. Informatik, Bonn 2005. (Cited on pages 2, 50, and 57.)

[29] Ivo L. Hofacker, Peter Schuster, and Peter F. Stadler. Combinatorics of RNA secondary structures. *Discrete Appl. Math.*, 88(1-3):207–237, 1998. URL http://portal.acm.org/citation.cfm?id=301750. (Cited on page 38.)

[30] P. Hogeweg and B. Hesper. The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *Journal of Molecular Evolution*, 20:175–186, 1984. (Cited on page 15.)

[31] John a. Jaeger, Douglas H. Turner, and Michael Zuker. Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci., USA, Biochemistry*, 86:7706–7710, 1989. (Cited on pages 17 and 22.)

[32] G.F. Joyce. RNA evolution and the origins of life. *Nature*, 338:217–224, 1989. (Cited on page 9.)

[33] D.A.M. Konings and P. Hogeweg. Statistical prediction of single-stranded regions in RNA secondary structure and application to predicting effective antisense target sites and beyond. *Journal of Molecular Biology*, 207(3):597–614, 1989. (Cited on pages 15 and 16.)

[34] Y.-J. Li, T. Macnaughton, L. Gao, and M.M.C. Lai. RNA-templated replication of hepatitis delta virus: Genomic and antigenomic RNAs associate with different nuclear bodies. *Journal of Virology*, 80(13), 2006. (Cited on page 1.)

[35] S.D. Linnstaedt, W. K. Kasprzak, B.A. Shapiro, and J.L. Casey. The role of a metastable RNA secondary structure in hepatitis delta virus genotype iii RNA editing. *RNA*, 12, 2006. (Cited on page 2.)

[36] S. Makino, T. Sawasaki, Y. Tozawa, Y. Endo, and K. Takai. Covalent circularization of exogenous RNA during incubation with wheat embryo cell extract. *Biochemical and Biophysical Research Communications*, 347:1080–1087, 2006. (Cited on page 1.)

[37] S.C. Manrubia and C. Briones. Modular evolution and increase of functional complexity in replicating RNA molecules. *RNA*, 13, 2007. (Cited on page 9.)

[38] A.E. Martinez de Alba, R. Flores, and C. Hernandez. Two chloroplastic viroids induce the accumulation of small RNAs associated with posttranscriptional gene silencing. *Journal of Virology*, 76(24):13094–13096, 2002. (Cited on page 1.)

[39] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependance of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288:911–940, 1999. (Cited on page 17.)

[40] Y. Matsumoto, R. Fishel, and R. B. Wickner. Circular single-stranded rna replicon in saccharomyces cerevisiae. *Proc. Natl. Acad. Sci. USA*, 87, 1990. (Cited on page 1.)

[41] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990. ISSN 0006-3525. doi: 10.1002/bip.360290621. URL http://dx.doi.org/10.1002/bip.360290621. (Cited on pages 29 and 31.)

[42] D. Molina-Serrano, L. Suay, M.L. Salvador, R. Flores, and J.-S. Daros. Processing of RNAs of the family avsunviroidae in chlamydomonas reinhardtii chloroplasts. *Journal of Virology*, 81(8), 2007. (Cited on page 2.)

[43] A Mosig, I.L. Hofacker, and P.F. Stadler. Comparative analysis of cyclic sequences: Viroids and other small circular RNAs. *Lecture Notes in Informatics*, P-83:93–102, 2006. (Cited on page 62.)

[44] H. Nielsen, T. Fiskaa, A. B. Birgisdottir, P. Haugen, and C. Einvik. The ability to form full-length intron rna circles is a general property of nuclear group i introns. *RNA*, 9, 2003. (Cited on page 1.)

[45] H. F. Noller, v. Hoffarth, and L Zimniak. Unusual resistance of peptidyl transferase to protein extraction procedures. *Science*, 256:1416–1419, 1992. (Cited on page 9.)

[46] R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded rna. *Proc Natl Acad Sci U S A*, 77(11):6309–6313, November 1980. ISSN 0027-8424. URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=81101160. (Cited on pages 17 and 21.)

[47] Ruth Nussinov, George Pieczenik, Jerrold R. Griggs, and Daniel J. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35(1): 68–82, 1978. doi: 10.1137/0135006. URL http://link.aip.org/link/?SMM/35/68/1. (Cited on pages 13 and 18.)

[48] A. Rich and J.D. Watson. Some relations between DNA and RNA. *Proc. Natl. Acad. Sci. USA*, 40(8):759–764, 1954. (Cited on page 7.)

[49] L. Rocheleau and M. Pelchat. The Subviral RNA Database: a toolbox for viroids, the hepatitis delta virus and satellite RNAs research. *BMC Microbiol.*, 6(24), 2006. URL http://subviral.med.uottawa.ca. (Cited on pages 1, 66, and 67.)

[50] S.R. Salgia, S.K. Singh, P. Gurha, and R. Gupta. Two reactions of haloferax volcanii rna splicing enzymes: Joining of exons and circularization of introns. *RNA*, 9, 2003. (Cited on page 1.)

[51] R. Sanjuan, J. Forment, and S.F. Elena. In silico predicted robustness of viroids RNA secondary structures. i. the effect of single mutations. *Mol. Biol. Evol.*, 23(7), . (Cited on page 2.)

[52] R. Sanjuan, J. Forment, and S.F. Elena. In silico predicted robustness of viroids RNA secondary structures. ii. interactions between mutation pairs. *Mol. Biol. Evol.*, 23(11), . (Cited on page 2.)

[53] N.G. Starostina, S. Marshburn, L.S. Johnson, S.R. Eddy, R. M. Terns, and M. P. Terns. Circular box c/d rnas in pyrococcus furiosus. *PNAS*, 101, 2004. (Cited on page 1.)

[54] G. Steger, H. Hofmann, J. Foertsch, H.J. Gross, J.W. Randles, H.L. Saenger, and D. Riesner. Conformational transitions in viroids and virusoids: comparison of results from energy minimization algorithm and from experimental data. *J. Biomol. Struct. Dyn.*, 2(3):543–571, 1984. (Cited on pages 49 and 66.)

[55] M. Tabler and M. Tsagris. Viroids: petite RNA pathogens with distinguished talents. *Trends Plant Sci.*, 9, 2004. (Cited on page 1.)

[56] Manfred Tacker, Peter F. Stadler, Erich G. Bornberg-Bauer, Ivo L. Hofacker, and P. Schuster. Algorithm independent properties of RNA secondary structure predictions. *European Biophysics Journal*, 25(2):115–130, December 1996. URL http://dx.doi.org/10.1007/s002490050023. (Cited on page 38.)

[57] D.H. Turner, N Sugimoto, and S. Freier. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, 1988. (Cited on pages 17 and 19.)

[58] T. Tuschl, C. Gohlke, T.M. Jovin, E. Westhof, and F. Eckstein. A three-dimensional model for the hammerhead ribozyme based on fluorescence measurements. *Science*, 266, 1994. (Cited on page 8.)

[59] S. Umekage and Y. Kikuchi. Production of circular form of streptavidin rna aptamer in vitro. *Nucleic Acids Symposium*, 50:323–324, 2006. (Cited on page 1.)

[60] G. Varani and W. H. McClain. The g x u wobble base pair. a fundamental building block of rna structure crucial to rna function in diverse biological systems. *EMBO Rep*, 1(1):18–23, July 2000. ISSN 1469-221X. URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=11256617. (Cited on page 7.)

[61] D. Voet, J.G. Voet, and C.W. Pratt. *Lehrbuch der Biochemie*. WILEY-VCH, 2003. ISBN 3-527-30519-X. (Cited on page 9.)

[62] A.E. Walter, D.H. Turner, J. Kim, M.H. Lyttle, P. Muller, D.H. Mathews, and M. Zuker. Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proc. Natl. Acad. Sci. USA*, 91:9218–9222, 1994. (Cited on page 24.)

[63] M.-B. Wang, X.-Y. Bian, L.-M. Wu, L.-X. Liu, N.A. Smith, D. Isenegger, R.-M. Wu, C. Masuta, V.B. Vance, J.M. Watson, A. Rezaian, E.S. Dennis, and P.M. Waterhouse. On the role of RNA silencing in the pathogenicity and evolution of viroids and viral satellites. *PNAS*, 101(9):3275–3280, 2004. (Cited on page 1.)

[64] M. S. Waterman. Secondary structure of single - stranded nucleic acids. *Studies on foundations and combinatorics, Advances in mathematics supplementary studies, Academic Press N.Y.*, 1:167–212, 1978. (Cited on pages 10, 29, and 33.)

[65] M. S. Waterman and T. H. Byers. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Math. Biosci.*, 77:179–188, 1985. (Cited on page 33.)

[66] M. S. Waterman and T. F. Smith. RNA secondary structure: A complete mathematical analysis. *Math. Biosci.*, 42:257–266, 1978. (Cited on pages 10 and 18.)

[67] J. D. Watson and F. H. Crick. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, April 1953. URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=13054692. (Cited on page 7.)

[68] M. T. Wolfinger. The Energy Landscape of RNA Folding. Master's thesis, University Vienna, 2001. (Cited on page 57.)

[69] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–165, February 1999. ISSN 0006-3525. URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=99169417. (Cited on pages 33, 59, 65, 77, and 78.)

[70] X. Zhong, N. Leontis, S. Qian, A. Itaya, Y. Qi, K. Boris-Lawrie, and B. Ding. Tertiary structural and functional analyses of a viroid RNA motif by isosteric matrix and mutagenesis reveal its essential role in replication. *Journal of Virology*, 80(17), 2006. (Cited on page 1.)

[71] X. Zhong, X. Tao, J. Stombaugh, N. Leontis, and B. Ding. Tertiary structure and function of an RNA motif required for plant vascular entry to initiate systemic trafficking. *The EMBO Journal*, 26(16), 2007. (Cited on page 1.)

[72] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244(4900):48–52, April 1989. ISSN 0036-8075. URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=2468181. (Cited on pages 2 and 32.)

[73] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, July 1984. (Cited on pages 11, 20, and 49.)

[74] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res*, 9(1): 133–148, January 1981. ISSN 0305-1048. (Cited on pages 17, 18, 20, and 49.)

EIDESSTATTLICHE ERKLÄRUNG

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschliesslicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Leipzig, November, 19. 2007

Ronny Lorenz