# SUBOPTIMAL SECONDARY STRUCTURES OF RNA

DIPLOMARBEIT
eingereicht von
**Stefan Wuchty**
zur Erlangung des akademischen Grades
Magister rerum naturalium
an der Formal- und Naturwissenschaftlichen Fakultät
der Universität Wien

March 16, 1998

Diese Arbeit wurde in der Zeit von März 1997 bis Februar 1998 im Institut für theoretische Chemie der Universität Wien durchgeführt. Zu allererst möchte ich Peter Schuster für die (kurzfristige) Aufnahme in seine Arbeitsgruppe danken. An dieser Stelle ist mir die Möglickeit gegeben, allen Freunden und Kollegen für deren Unterstützung und Hilfe zur Erreichung dieses Zieles zu danken.

Walter Fontana hat diese Arbeit bestens betreut und mich in wissenschaftliches Arbeiten eingeführt. Ivo Hofacker hat mit mir endlos nach Bugs und anderen Fehlern gesucht.

Last but not least bedanke ich mich bei allen anderen Mitgliedern der Arbeitsgruppe: Peter Stadler, Ronke Babajide, Jan Cupal, Martin Fekete, Christoph Flamm, Thomas Griesmacher, Christian Haslinger, Stephan Kopp, Bärbel Krakhofer, Stefan Müller, Susanne Rauscher, Alexander Renner, Norbert Tschulenk, Andreas Wernitznig und die heimliche Leitung des Instituts, Judith Jakubetz, sorgten für eine ausgezeichnete Athmosphäre.

Meinen Eltern danke ich für ihre Unterstützung während des Studiums..

# Zusammenfassung

RNA-Moleküle dienen nicht nur als Träger von Information, sondern auch als selbstständige funktionelle Einheiten. Ihre dreidimensionale Struktur spielt eine wichtige Rolle bei einer großen Anzahl von biologischen Prozessen. Sekundärstrukturen bieten die Möglichkeit, die Struktur von RNA-Molekülen in einer gröberen Auflösung zu untersuchen. Ihr Studium liefert wertvolle Information für die Vorhersage von 3D-Strukturen und für das Verständnis biochemischer Vorgänge .

RNA-Sekundärstrukturen können als planare Graphen beschrieben werden. Neben der thermodynamisch optimalen Sekundärstruktur existieren noch weitere suboptimale Sekundärstrukturen, die ebenfalls eine biologische Funktion haben können. Früher entwickelte Algorithmen zur Berechnung suboptimaler Sekundärstrukturen werden beschrieben und miteinander verglichen. Ein neuer Algorithmus zur Berechnung aller Sekundärstrukturen von RNA-Sequenzen innerhalb eines bestimmten Energiebandes oberhalb der minimalen freien Energie wurde entwickelt und implementiert. Der Algorithmus benutzt die Methode des *dynamic programming.*

Mit der Möglichkeit, sämtliche suboptimale Sekundärstrukturen berechnen zu können, wurde die Rolle der modifizierten Basen in natürlichen tRNA-Sequenzen von *E.coli* untersucht. Es zeigte sich, daß die modifizierten Basen einen signifikanten stabilisierenden Einfluß auf die Definiertheit der RNA-Struktur haben. Weiters wurde die “Lower Density of States” (*LoDoS*) einiger tRNA-Sequenzen untersucht. Die Ergebnisse zeigen, daß die natürlichen tRNA-Sequenzen im Vergleich zu anderen Sequenzen, welche dieselbe Sekundärstrukturen ausbilden, weniger Zustände in der Umgebung des Grundzustandes aufweisen, der Abstand vom Grundzustand zum ersten angeregten Zustand höher ist und diese Zustände strukturstabiler sind. Ebenfalls konnte gezeigt werden, daß ein direkter Zusammenhang zwischen Mutationsstabilität einer Sequenz, ihrer thermodynamischen Stabilität und Wohldefiniertheit ihrer Struktur besteht. Die Zustandssumme einer RNA-Sequenz kann aus den tiefer

liegenden Zuständen ohne Kenntnis aller Strukturen in guter Näherung berechnet werden.

# Abstract

RNA molecules do not serve as carriers of information only, but also as functionally active units. The three-dimensional shape of tRNA molecules plays a crucial role for a wide variety of biological processes. Secondary structures provide a convenient form of coarse graining, and their study yields information useful in the prediction of the full 3D structures and in the interpretation of the biochemical function of the molecules. Furthermore, secondary structures are discrete and therefore well suited for computational methods.

RNA secondary structures can be represented as planar, vertex-labeled graphs. Beside the thermodynamic optimal secondary structure there exist further suboptimal structures providing also a biological function. Algorithms for calculating suboptimal secondary structures derived previously were compiled and presented. A new algorithm for calculating all acceptable suboptimal structures of RNA sequences within a given energy range above the *minimum free energy* based on *dynamic programming* was developed and implemented.

With the possibility to calculate all acceptable suboptimal structures the role of modified bases occuring in natural tRNA sequences of *E.coli* was investigated. The results show that the modified bases have a significant stabilizing effect on the *well-definedness* of the structure. Also the Lower Density of States (*LoDoS*) of these tRNA sequences was investigated. The results show that original tRNA sequences in comparison to sequences providing the same secondary structure have less states in the vicinity of the ground state, the energy gap is usually larger and the structures contained by these states are better conserved. A direct correlation between stability against mutation of a sequence, their thermodynamical stability and *well-definedness* of their related structure was observed. Furthermore the partition function of a RNA sequence was calculateable in a good approximation using the lower states without the knowledge of all structures.

# Contents

**1 Motivation 1**

**2 RNA Secondary Structures 4**

**3 Secondary Structure Graphs 7**

3.1 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

3.2 Representation of Secondary Structures . . . . . . . . . . . . . . 11

**4 Previous Solutions and Approaches 14**

4.1 Combinatorial Algorithms . . . . . . . . . . . . . . . . . . . . . 14

4.1.1 Some basic facts . . . . . . . . . . . . . . . . . . . . . . 14

4.1.2 Combinatorial Approaches to Suboptimal Folding . . . . 15

4.2 Recursive Algorithms . . . . . . . . . . . . . . . . . . . . . . . . 17

4.2.1 Some basic facts . . . . . . . . . . . . . . . . . . . . . . 17

4.2.2 The Zuker Algorithm . . . . . . . . . . . . . . . . . . . . 18

**5 Maximum Matching as Illustration of Waterman's Concept 21**

5.1 The Idea of "Maximum Matching" . . . . . . . . . . . . . . . . . 21

5.2 Backtracking and Waterman's Concept . . . . . . . . . . . . . . 22

5.2.1 Waterman's Algorithm . . . . . . . . . . . . . . . . . . . 23

5.2.2 Application of Waterman's Concept to the Maximum Matching Problem . . . . . . . . . . . . . . . . . . . . . . 24

5.3 Check for Reliability of the new Algorithm . . . . . . . . . . . . 26

**6 Structure Prediction 27**

6.1 Thermodynamic Nearest Neighbor Parameters . . . . . . . . . . 27

6.2 Assigning Energy Parameters to Graphs . . . . . . . . . . . . . 29

6.3 Further Works . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

6.4 Dynamic Folding Algorithm . . . . . . . . . . . . . . . . . . . . 31

6.5 Backtracking . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

6.5.1 Common Features . . . . . . . . . . . . . . . . . . . . . . 37

6.5.2 Extension of Backtracking by Waterman's Concept . . . 38

6.6 Check for Reliability of the SUBOPT Algorithm . . . . . . . . . . 40

**7 Performance of the SUBOPT Algorithm 43**

7.1 CPU and Memory Requirements . . . . . . . . . . . . . . . . . . 43

7.2 Performance in Comparison to Zuker's Algorithm . . . . . . . . 44

**8 Results 46**

8.1 What is "Well-definedness" ? . . . . . . . . . . . . . . . . . . . . . 46

8.2 *E.coli* tRNA Sequences . . . . . . . . . . . . . . . . . . . . . . . . 47

8.3 Gapstatistics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

8.3.1 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

8.3.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

8.4 Structure Distances . . . . . . . . . . . . . . . . . . . . . . . . . . 58

8.4.1 Definitions . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

8.4.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

8.5 Partition Function Plots . . . . . . . . . . . . . . . . . . . . . . . 62

8.6 Coarse Grained Approaches . . . . . . . . . . . . . . . . . . . . . 65

8.7 Neutrality . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65

**9 Conclusions and Outlook 71**

**A EMBL tRNA Database 74**

**B SUBOPT man page 79**

**References 80**

# List of Figures

1 Folding of an RNA Sequence . . . . 4
2 RNA Secondary Structure . . . . 7
3 Components of RNA Secondary Structures . . . . 9
4 Secondary Structure Motifs . . . . 10
5 RNA Diagramm . . . . 11
6 Circular Representation . . . . 12
7 Mountain Representation . . . . 13
8 Schematic Representation of Maximum Matching . . . . 22
9 Schematic Representation of the Different Terms yielding $C_{i,j}$ and $F_{i,j}^{M1}$. . . . . 33
10 Schematic Representation of the Different Terms yielding the Energy of Remaining Sections of a Multi-loop Structure . . . . 34
11 Schematic Representation of the Different Terms yielding $F_j^5$ . . 35
12 Test Sequences . . . . 43
13 Structure and Sequences of one *E.Coli* tRNA . . . . 48
14 Crosshaped Structures . . . . 49
15 Distribution of $1^{st}$ and $2^{nd}$ Gap Energies of tRNAs . . . . 50
16 Distribution of Frequency $f_{mfe}$ of tRNAs . . . . 51
17 Distribution of $1^{st}$ Gap Energy ($\Delta G_1$) and Frequency ($f_{mfe}$) of tRNAs . . . . 52
18 Distribution of Frequency $f_{mfe}$ of Crosshaped Structures . . . . 54
19 Distribution of $1^{st}$ Gap Energies of Crosshaped Structures . . . 55
20 Distribution of $1^{st}$ Gap Energy ($\Delta G_1$) and Frequency ($f_{mfe}$) of Crosshaped Structures . . . . 56
21 Hairpin Structures . . . . 57
22 Distribution of Frequency $f_{mfe}$ of Hairpin Structures . . . . 57
23 Distribution of Base Pair Distances of tRNA Sequences . . . . 59
24 Mean Gap Energies and Mean Base Pair Distances of natural tRNA Sequences . . . . 60

25 Mean Gap Energy and Mean Base Pair Distance of Crosshaped Structures . . . . . 61
26 Approximation of $Z$ . . . . . 63
27 Approximated Matrices of Base Pair Probability . . . . . 64
28 *LoDoS* of a Single tRNA Sequence . . . . . 67
29 Neutrality of Inverse Folded tRNA Sequences . . . . . 68
30 Neutrality of Crosshaped Structure A . . . . . 69
31 Neutrality of Crosshaped Structure B . . . . . 70

# List of Tables

1 Pseudo Code for the Dynamic Programming of the Maximum Matching Problem . . . 22
2 Pseudo Code for Backtracking for the Maximum Matching Problem including Waterman's Algorithm . . . 25
3 Check for Reliability of the New Algorithm . . . 26
4 Recursion for the calculation of the minimum free energy . . . 36
5 Check for Reliability of the SUBOPT Algorithm . . . 41
6 Pseudo Code for the SUBOPT Algorithm . . . 42
7 Data of CPU and Memory Requirements of the SUBOPT Algorithm . . . 43
8 Performance of Zuker's in Comparison to the SUBOPT Algorithm 44

# 1 Motivation

RNA molecules serve not only as carriers of information, but also as functionally active units. The three dimensional shape of tRNA molecules plays a crucial role in the process of protein synthesis. RNA is known to exhibit catalytic activity [5, 11, 12, 23]. While the activity of these so called "ribozymes" is usually restricted to cleavage and splicing of RNA itself, recent evidence suggests that RNA also plays a predominant role in ribosomal translation. These discoveries have given much support to the idea that an *RNA World* [10, 20, 21, 22] stood at the origin of life, in which RNA served both as carrier of genetic information as well as catalytically active substance. RNA may not necessarily have been the first step in prebiotic evolution, but the idea that RNA preceded not only DNA, but also the invention of the translational system, seems widely accepted. Furthermore, RNA provides an ideal, currently the only, system to study genotype-phenotype relationships. Following Sol Spiegelman [40], the phenotype for an RNA molecule can be defined as its spatial structure.

Although RNA offers a limited repertoire of catalytic functions, ribozymes gain importance for biotechnological applications, since these molecules are suited for *evolutionary design*: Large scale synthesis of RNA molecules underlying mutation and selection experiments, in which the ribozymes are screened for positive catalytic functions, are spreading in use.

In many biologically evolved RNA molecules such as viral genomes and tRNA, the structure seems to be more conserved than the sequence. Viruses belonging to the same family show often little sequence similarity, yet exhibit strongly conserved structural motifs in terminal regions. The wide variety of tRNA sequences provided by databases fit into almost ident cloverleaf patterns.

RNA secondary structures can be represented as planar vertex-labeled graphs. Dynamic programming algorithms for calculation of the minimum free energy structure [47, 53] as well as combinatorial algorithms [7] both based on graph enumeration have been available now for some time. Naturally the algorithms yield only the ground state structure; there is of course an exponentially high number of other configurations, and even though the ground state is more probable than any other state, the probability within the whole ensemble of structures may be negligible. An elegant solution for this problem was suggested by John McCaskill [26], who proposed an algorithm to compute the partition function and the matrix of base pairing probabilities of an RNA molecule. The Vienna RNA Package [16] provides an efficient implementation of both the minimum free energy and the partition function algorithm, which makes calculations even for large sequences possible.

Paul Higgs [14, 15] presented thermodynamic studies on the stability of tRNA molecules, based on an algorithm for the density of states, *i.e.*, the distribution of energies of all possible secondary structure configurations. From the density of states all thermodynamic parameters can be derived. While the partition, too, yields the frequency of the ground state in the thermal equilibrium, specific information about suboptimal structures can only be obtained from the density of states. Higgs algorithm is based on compiling compatible stems of minimum length 3 and uses a rather simlified energy model [14].

In our research group Jan Cupal [6] introduced a dynamic programming algorithm for the computation of the distribution of states of RNA secondary structures in his diploma thesis. The algorithm implements the energy parameter set used within the Vienna RNA Package and is not restricted to any minimum stem length. He showed that the recursions underlying all dynamic folding algorithms are accessible from a single basic recursion for the enumeration of secundary structure graphs. This algorithm can be extended to yield the complete density of states. The observation that the density of states of a sequence can be obtained from the density of states of smaller subsequences is fundamental in this work. However the algorithm is quite demanding both in terms of memory and CPU time. Thus the possibilities of application are restricted to an upper bound sequence length.

For many purposes it is not necessary to get the whole information of the density of states, but of a certain energy range above the minimum free energy. Also there may exist several suboptimal structures providing a biological function. There exist already combinatorial [28, 50] and dynamic programming approaches [51] to find all secondary structures within this window of the density of states. However, these algorithms feature either a high amount of approximation or simply do not find all secondary structures within the considered range of energy.

In this work we introduce an algorithm capable to calculate all secondary structures of an RNA sequence within a desired energy range above the minimum free energy, which implements the energy parameter set used within the Vienna RNA Package. Finding near-optimal paths between a specified origin and destination in an acyclic network, which is based on a idea of Waterman [46] is firstly applied to the "maximum matching" problem originally implemented by Nussinov [30]. This part is a kind of test of the applicability of that idea. Afterwards this idea will be applied to the energy folding problem.

With the possibility to calculate all secondary structures of a RNA sequence within a certain energy range it is possible to gain new insights in the well-definedness of a RNA structure. With this tool we were able to investigate the stabilizing role of modified (*i.e.* non-pairing) bases contained in RNA sequences. As example we used the natural tRNA sequences of *E.coli.* An insight into the relation between neutrality of RNA sequences, their thermodynamical stability and well-definedness of the related structures is given. The ability to calculate the Lower Density of States (*LoDoS*) within an energy range enables us to characterise the different states of natural RNA sequences of *E.coli.* Also the partition function of an RNA sequence using the lower states without knowing all structure energies can be calculated in a good approximation.

# 2 RNA Secondary Structures

RNA molecules consist of ribonucleotides linked together by covalent chemical bonds. Each ribonucleotide contains one of the four bases adenine, cytosine, guanine or uracil. The specific sequence of bases along the chain is called the primary structure and determines the kind of the molecule.

In biological systems RNA chains bend and twine about themselves, and bases in close vicinity form weak chemical hydrogen bonds with a complementary base: A binds with U, G with C (*Watson-Crick* base pairs).

Much like DNA, RNA can form stable double helices of complementary

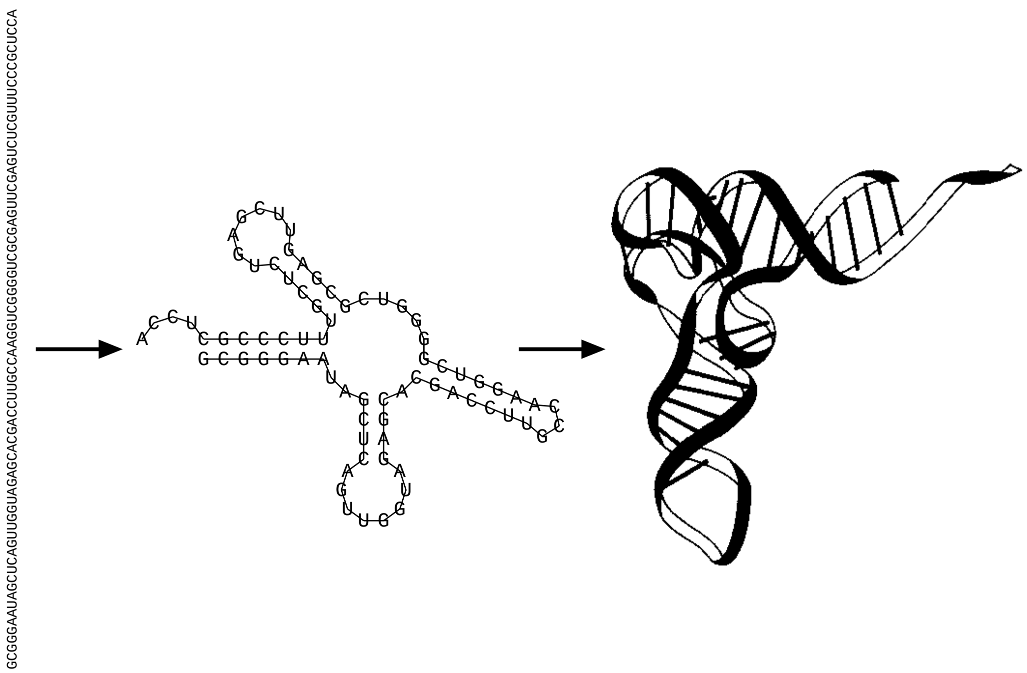

GCGGGAAUAGCUCAGUUGGUAGAGCACGACCUUGCCAAGGUCGGGGUCGCGAGUUCGAGUCUCGUUUCCCGCUCCA

**Figure 1**: Folding of an RNA sequence into its spatial structure. The process is partitioned into two phases: in the first phase only the Watson-Crick-type base pairs are formed which constitute the major fraction of the free energy, and in the second phase the actual spatial structure is built by folding the planar graph into a three dimensional object. The example shown here is phenylalanyl-transfer-RNA $\mathrm{tRNA}^{\mathrm{Phe}}$, whose spatial structure is known from X-ray crystallography.

strands. Since RNA usually occurs single stranded, formation of double helical regions is accomplished by the molecule folding back onto itself to form Watson-Crick G-C and A-U base pairs or the slightly less stable G-U pairs. Base stacking and pairing are the major driving forces for RNA structure formation. Other, usually weaker, intermolecular forces and the interaction with the aqueous solvent shape its spatial structure.

Since the number of degrees of freedom in the RNA chain is very high and exeeds that in polypeptides, the full structural prediction problem is hard to solve. However, for RNA it has seen to be possible to focus initially on an intermediate level representation of the folding. This secondary structure representation contains only information on what base pairs are formed and relegates more detailed and additional information to a later and subsequent stage of analysis. The resulting secondary structures are useful in the prediction of the full 3D structures and in the interpretation of the biochemical function of the molecules for several reasons:

(1) The conventional base pairing and the base pair stacking cover the major part of the free energy of folding.

(2) Secondary structures are used successfully in the interpretation of RNA function and reactivity.

(3) Secondary structures are conserved in evolutionary phylogeny.

At the same time the secondary structure representation is very convenient:

(1) Secondary structures are discrete and therefore easy to compare.

(2) They are easy to visualize since they are planar graphs.

(3) Efficient methods exist for the computation of secondary structures.

In the following section we will give a formal definition of secondary structures as graphs: RNA secondary structures can be represented as planar vertex-labeled graphs or as trees. Note, that our definition ranks pseudo-knots as a tertiary interaction. Although pseudo-knots seem to be important for biological function, their inclusion would complicate the mathematical and computational treatment unduly.

# 3 Secondary Structure Graphs

## 3.1 Definitions

**Definition 3.1.** [45, 47] A *secondary structure* is a vertex-labeled graph on $n$ vertices with an adjacency matrix $A$ fulfilling

(1) $a_{i,i+1} = 1$ for $1 \leq i < n$;

(2) For each $i$ there is at most a single $k \neq i-1, i+1$ such that $a_{ik} = 1$;

(3) If $a_{ij} = a_{kl} = 1$ and $i < k < j$ then $i < l < j$.

We will call an edge $(i, k)$, $|i-k| \neq 1$ a bond or base pair. A vertex $i$ connected only to $i-1$ and $i+1$ will be called unpaired. Condition (3) assures that the structure contains no pseudo-knots. A vertex $i$ is said to be *interior* to the base pair $(k, l)$ if $k < i < l$. If, in addition, there is no base pair $(p, q)$ such that $p < i < q$, we will say that $i$ is *immediately interior* to the base pair $(k, l)$. A base pair $(p, q)$ is said to be (immediately) interior, if $p$ and $q$ are (immediately) interior to $(k, l)$.

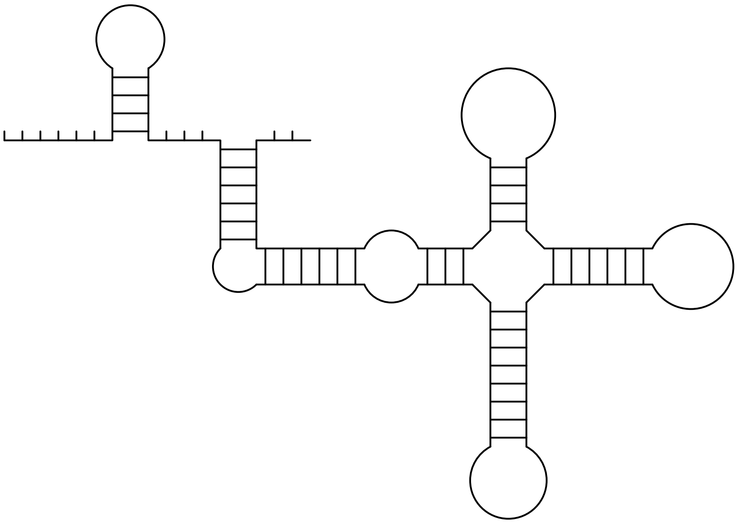


**Figure 2**: An example for an RNA secondary structure with free dangling ends, stems and loops.

**Definition 3.2.** A secondary structure consists of the following structure elements:

(1) A *stem* consists of subsequent base pairs $(p, q)$, $(p+1, q-1)$, …, $(p+h-1, q-h+1)$, $(p+h, q-h)$ such that neither $(p-1, q+1)$ nor $(p+h+1, q-h-1)$ is a base pair. $h+1$ is the *length* of the stem, $(p, q)$ is the terminal base pair of the stem.

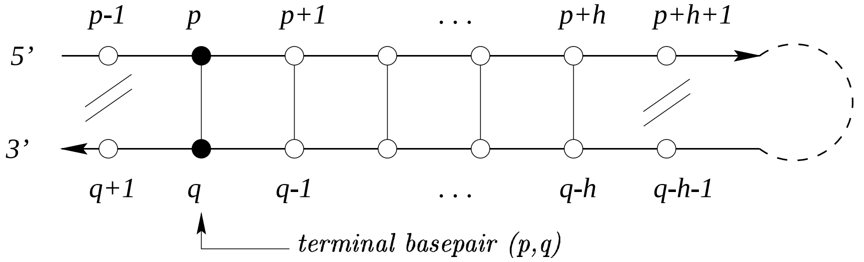
(2) A *loop* consists of all unpaired vertices which are immediately interior to some base pair $(p, q)$, the "closing" pair of the loop.

(3) An *external vertex* is an unpaired vertex which does not belong to a loop. A collection of adjacent external vertices is called an external element. If it contains the vertex 1 or $n$ it is a free end, otherwise it is called joint.

**Lemma 3.3.** Any secondary structure $\Phi$ can be uniquely decomposed into stems, loops, and external elements.

**Proof.** Each vertex which is contained in a base pair belongs to a unique stem. Since an unpaired vertex is either external or immediately interior to a unique base pair, the decomposition is unique: Each loop is characterized uniquely by its "closing" base pair.

**Definition 3.4.** A stem $[(p, q), \ldots, (p+k, q-k)]$ is called *terminal*, if $p-1 = 0$ or $q+1 = n+1$, or if the two vertices $p-1$ and $q+1$ are not interior to any base



Subsequent base pairs $(p, q)$, *(p+1,q−1)*, …, *(p+h,q−h)* form a *stem* such that neither *(p+h+1,q−h−1)* nor *(p−1,q+1)* is a base pair. $h+1$ is the *length* of the stem, *(p,q)* is the terminal base pair. *(p+h,q−h)* is the closing pair of a loop. Base pairs *(p,q)* to *(p+h−1,q−h+1)* can be seen as closing base pairs of *minimal loops* of size $z = 0$ and degree $k = 2$.

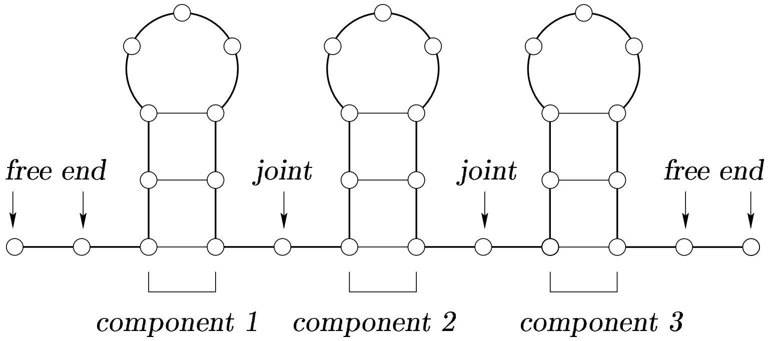


**Figure 3**: An example for an RNA secondary structure consisting of three components and six external vertices (2 joints and 4 free ends).

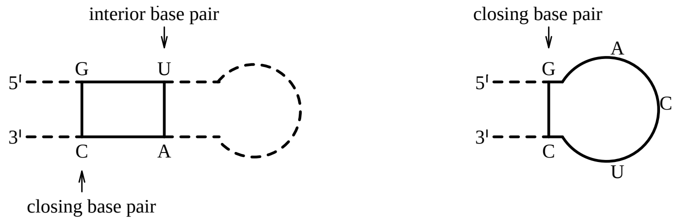
pair. The substructure enclosed by the terminal base pair $(p, q)$ of a terminal stem will be called a *component* of $\Phi$. We will say that a structure on $n$ vertices has a terminal base pair, if $(1, n)$ is a base pair.

**Lemma 3.5.** A secondary structure may be uniquely decomposed into components and external vertices. Each loop is contained in a component.
The proof is trivial. Note that by definition the open structure has 0 components.

**Definition 3.6.** The *degree* $k$ of a loop is given by 1 plus the number of terminal base pairs of stems which are interior to the closing bond of the loop. A loop of degree 1 is called *hairpin (loop)*, a loop of a degree larger than 2 is called *multi-loop.* A loop of degree 2 is called *bulge* if the closing pair of the loop and the unique base pair immediately interior to it are adjacent; otherwise a loop of degree 2 is termed *interior loop.*

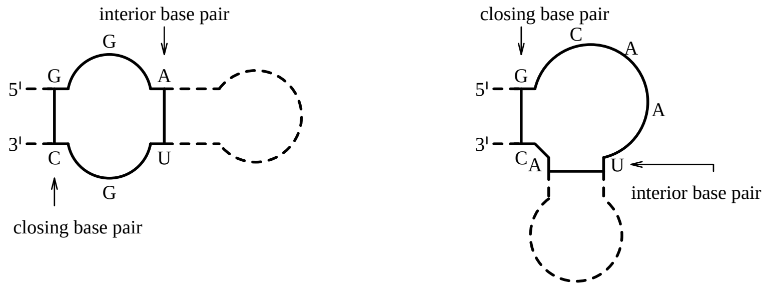
**Definition 3.7.** The *size* $z$ of a loop is given by the number of unpaired vertices *immediatly interior* to the closing base pair $(p, q)$ of the loop. If a stem ends in a base pair $(p, q)$ with no unpaired vertices immediately interior to it, we speak of a loop with size zero. $m$ denotes the minimum number of unpaired digits in a hairpin loop (minimal loop size).

It is often useful to lump loops of all degrees together into one class and to



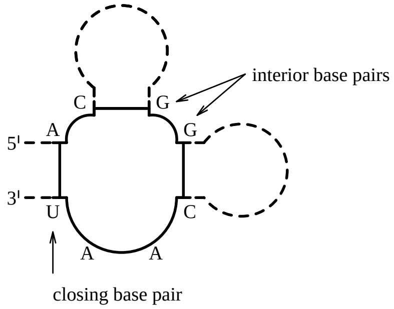
stacking pair

hairpin loop



interior loop

bulge



multiple loop

**Figure 4**: The classification of loops for the decomposition of RNA secondary structure.

consider, for example, the total number of loops

$$n_{\mathrm{L}} \;=\; n_{\mathrm{H}} + n_{\mathrm{B}} + n_{\mathrm{I}} + n_{\mathrm{M}}$$

which must be identical to the number of stems, $n_{\mathrm{L}} = n_{\mathrm{S}}$.

## 3.2 Representation of Secondary Structures

A string representation **S** can by obtained by the following rules:

(1) If vertex $i$ is unpaired, then $\mathbf{S}_i =$`'.'`

(2) If $(p, q)$ is a base pair and $p < q$, then $\mathbf{S}_p =$`'('` and $\mathbf{S}_q =$`')'`

These rules yield a sequence of matching brackets and dots called *bracket notation.*

Secondary structure graphs as defined above can be drawn by placing the bases of a sequence equidistant to one another on a line. Pairing bases are connected by arcs.

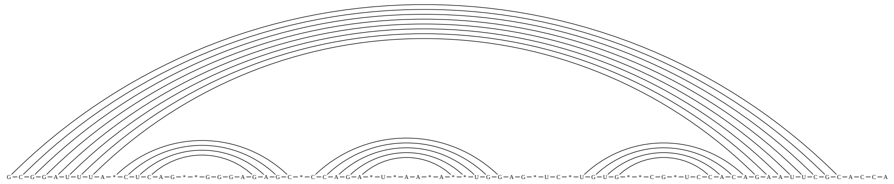

G—C—G—G—A—U—U—U—A—*—C—U—C—A—G—*—*—G—G—G—A—G—A—G—C—*—C—C—A—G—A—*—U—*—A—A—*—A—*—*—U—G—G—A—G—*—U—C—*—U—G—U—G—*—*—C—G—*—U—C—C—A—C—A—G—A—A—U—U—C—G—C—A—C—C—A

**Figure 5**: The secondary structure of tRNA$^{\mathrm{Phe}}$ in *linked graph representation.*

A particularly easy way to draw secondary structure graphs was suggested by Ruth Nussinov [30]. The bases of the sequence are placed equidistant to one another on a circle and for each base pair a chord is drawn between the two bonded bases. Since the structures are unknotted by definition, no two chords will intersect. See Figure 6 for circular representation of tRNA$^{\mathrm{Phe}}$.

Paulien Hogeweg and Danielle Konings conceived a related graphical method for the comparison of RNA secondary structures called *mountain representation* [17, 24, 25] by identifying `'('`, `')'`, and `'.'`, with "up", "down", and "horizontal", respectively. See Figure 7 for mountain representation.

- *Peaks* correspond to hairpins. The symmetric slopes represent the stems enclosing the unpaired bases in the hairpin loop, which appear as a plateau.
- *Plateaus* represent unpaired bases. When interrupting sloped regions they indicate bulges or interior loops, depending on whether they occur

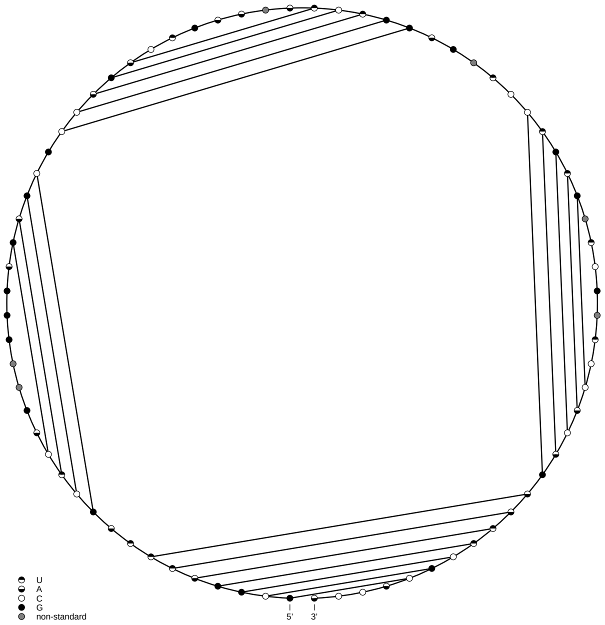


**Figure 6**: The secondary structure of tRNA$^{\mathrm{Phe}}$ in *Circular representation.*

alone or paired with another plateau on the other side of the mountain at the same height respectively.

- *Valleys* indicate the unpaired regions between the branches of a multi-stem loop or, when their height is zero, they indicate unpaired regions separating the components of secondary structures.

The height of the mountain at sequence position $k$ is simply the number of base pairs that enclose position $k$; *i.e.*, the number of all base pairs $(i, j)$ for which $i < k$ and $j > k$. The mountain representation allows straightforward comparison of secondary structures and inspired a convenient algorithm for alignment of secondary structures [25].

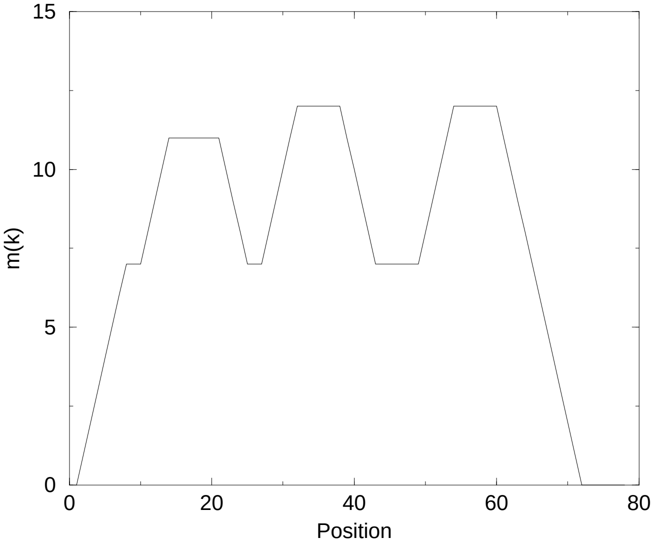


**Figure 7**: The secondary structure of tRNA$^{\mathrm{Phe}}$ in *mountain representation.* The same structure in string representation is (((((((..((((........)))).(((((.......))))).....(((((.......))))))))))))....

# 4 Previous Solutions and Approaches

## 4.1 Combinatorial Algorithms

Before an outline of combinatorial approaches to suboptimal folding is given, the basic ideas of those combinatorial folding algorithms will be described in this subsection.

### 4.1.1 Some basic facts

Combinatorial algortithms first develop a list of all helices that can be formed from a sequence, then determine the combination of these helices that gives the lowest free energy [3]. Two advantages of these helices are that, in principle, they can include knotted structures and they can include non-nearest-neighbour interactions. However, the number of possible helix combinations grows as $2^L$, where $L$ is the number of helices. In turn, $L$ increases approximately as $N^2$, where $N$ is the length of the considered sequence. Since it is usually necessary to include helices as short as 2 or 3 base pairs, the number of combinations quickly becomes enormous. Thus it has been necessary to devise algoithms that do not have to compute every possible combination. The most successful such algorithm assigns to each helix a free energy and a list of other helices that are compatible with it. A "tree search" procedure is used to generate combinations of compatible helices. As each helix is added to a combination, the number of other helices compatible with the combination decreases. To avoid computation of every compatible combination, the program also maintains lists of mutually incompatible helices. These lists are called *incompatibility islets* introduced by Dumas *et al.* [7]. Only one helix from a given islet can occur in a secondary structure. At each branch point in the development of a combination the most favourable helix free energy associated with each remaining islet is added to the free energy of the growing combination. If this sum is not as favorable as that already calculated for a complete secondary structure, then there is no reason to explore this branch of combinations further. In practice, combinations within a certain increment of the current best structure are explored to account for approximations. Thus it is possible to find suboptimal structures within any desired window of free energy. The algorithms are practical for sequences up to about 200 nucleotides. This limit is not likely to increase greatly with advances in computer speed because of the $2^L$ dependence of the number of combinations.

### 4.1.2 Combinatorial Approaches to Suboptimal Folding

Yamamoto *et al.* proposed a computer program for prediction of optimal and suboptimal secondary structures in 1985 [50]. Based on a work of Yamamoto *et al.* [49] all the possible folding structures are generated. Thus for each stem-loop its occurence, which is weighted by the free energy of the stem, in all the possible structures is computed. The frequency of occurence will reflect the probability of stem-loop formation. The probability is converted to the information value, which is merely the logarithm of the probability. The algorithm can be applied to RNA sequences of any length. However, it is an approximation. Quite apart from that the algorithm calculates only such suboptimal solutions which have more or less the same free energy but a different structure as the optimal structure.

In 1995 Nakaya *et al.* introduced a computer program for prediction of optimal and suboptimal secondary structures using highly parallel computers [28]. Their goal was to get all suboptimal structures fulfilling the follwing condition:

$$|E_i - E_{optimal}| < \Delta K. \tag{1}$$

Here $E_i$ is the free energy of a suboptimal structure and $E_{optimal}$ is the free energy of the optimal structure. $\Delta K$ is a given positive number. Their work is widely based on the work of Dumas *et al.* [7] mentioned above. However, the size of problems solved by Dumas' method is still restricted. So, exploitation of parallelism can promote further pruning of the tree. As thermodynamic parameters of stacking regions Salser's energy matrix is employed [35].

The algorithm consists of 2 phases:

- *Phase 1*: Starting point is finding all pairs of nucleotide sequences that could possibly form stacking regions of a given RNA sequence. These pairs are ordered in their free energy order. At the same time *incompatibility islets* as mentioned above are constructed. All stacking regions can not co-exist in a single feasible secondary structure, due to their structural restrictions. Hence these regions will be called *stacking region candidates*. These candidates are the actual units of computation.

- *Phase 2*: The algorithm checks, if candidates obtained in *phase 1* can coexist in a secondary structure by generating a search tree. The number of its leafs is $O(2^L)$ with $L$ being the number of candidates. Thus this tree is generated gradually by adding nodes to the interim tree. The nodes of the $n$th level of the tree correspond to the candidates with the $n$th free energy. Following the left branch of the $n$th level node means that the $n$th branch is selected and vice versa.

Prior to generating branches two decision must be made:

- Can the new left branch be generated? If a candidate is not compatible with all the candidates already selected as part of a secondary structure this branch can not be generated.
- Can secondary structures more stable than a threshold value exist under the non-leaf node? If they do not, that branch is not generated.

The following values are used for pruning:

- A good lower bound value of the free energies of secondary structures under a node.
- A threshold value that is compared with the lower bound calculated before.

The lower bound of the free energies of secondary structures under a node is calculated using the incompatibility islets. Under a node $x$ with depth $n$ from the root of a search tree the selectable candidates must be compatible with the already selected candidates on the path from the root to node $x$. The already selected candidates are the left branches of the tree on this path. In order to calculate the lower bound of the free energies firstly the free energy of all already found candidates is sumed up ($p$ kcal). Due to finding islets, where no candidates are selected, the most stable and compatible candidates are picked in these islets, and their free energy is also sumed up ($q$ kcal). The sumation of $p$ and $q$ can be used as the lower bound of the free energies of the secondary structures under node $x$.

At every leaf node firstly the lower bounds of a secondary structure at the left and right branch are calculated as outlined above. Only the branch whose lower bound is smaller than $threshold + \Delta K$ is generated, otherwise pruned. According to the first case this threshold will be denoted as $E_{best}$. In this algorithm the initial value of $E_{best}$ is calculated by iterating selection of candidate $i$ as long as it is compatible with all the already selected candidates of the secondary structure. The free energy of this secondary structure is denoted as the initial value of $E_{best}$.

This threshold value $E_{best}$ is updated at every leaf node by evaluating the free energy of the secondary structure at this node as long as it is smaller than $E_{best}$.

The stacking region candidates as outlined in this section are the units of computation. If base pairs were treated as a basic computation unit, this algorithm could eventually find all the feasible secondary structures. Computation time, however, grows prohibitively due to combinatorial explosion. Obviously the motivation of the authors was obtaining a good number of suboptimal secondary structures in a reasonable time, even if they are highly approximated.

## 4.2 Recursive Algorithms

### 4.2.1 Some basic facts

Minimum energy foldings can also be computed with recursive or *dynamic programming* algorithms. Such programs work in two stages. The first part, called fill algorithm, computes and stores minimum folding energies for all segments of the sequence. The process begins with all pentanucleotides and builds up to larger fragments in a recursive fashion. The second part, called *backtracking*, computes a minimum energy structure by searching systematically through the matrix of stored energies. The main advantages over combinatorial algorithms are speed and the ability to fold relatively large RNA sequences. By examining possible base pairs in the context of what neighboring base pairs might be, the algorithm escapes an exponentially growing number of structures. The mean weakness of the recursive folding algorithms is that by design they yield only a single solution. More details about that kind of folding algorithms will be given in the following sections.

### 4.2.2 The Zuker Algorithm

Zuker and Stiegler described a recursive folding algorithm in 1981 [53] computing and storing the minimum folding energy for each subsequence of the given RNA sequence. Also, for each subsequence, they calculated the minimum folding energy for the fragment with the ends constrained to form a base pair with each other if possible. For a fragment $[i, j]$ this number is denoted by $V(i, j)$ and is needed for proper function of the algorithm.

In 1989 Zuker introduced a recursive algorithm for finding all suboptimal foldings of an RNA molecule [51]. The key observation is that in a circular molecule composed of ribonuleotides $r_1, r_2, \ldots, r_n$ a base pair linking $r_i$ and $r_j$ divides the secondary structure into two parts. There is a folding of the "included fragment" from $r_i$ to $r_j$, and another folding of the "excluded fragment" from $r_j$ through the origin to $r_i$. The additivity assumption characteristic of recursive algorithms implies that the total folding energy is the sum of the energies of the two foldings.

The procedure for circular RNA generalizes to linear RNA. The linear molecule is handled as if it were circular, provided that the first and last bases, now regarded as adjacent, be allowed to pair with each other if necessary. The recursive algorithm is now extended by computing additional numbers $V(j, i)$, analogous to $V(i, j)$, but referring to the "excluded fragments" instead. These numbers can also be computed recursively. The observation was that

$V(i,j)+V(j,i)$ is the minimum free energy of a structure containing the base pair $(i,j)$ and that the minimum value of $V(i,j)+V(j,i)$ over all possible base pairs is the minimum folding energy $E_{min}$.

Instead of merely identifying a base pair $(i,j)$ that gives $E_{min}$ and computing an optimal folding, the strategy is to identify all base pairs, for which the sum $V(i,j)+V(j,i)$ is "close" to $E_{min}$. If $P$ is a number between 0 and 100, then a "$P$-optimal" base pair is a base pair $(i,j)$ for which

$$V(i,j) \ + \ V(j,i) \ \geq \ (1-\frac{P}{100})\ E_{min}. \tag{2}$$

Thus a $P$-optimal base pair is contained in at least one folding within $P$ percent of the minimum free energy. Such a folding is defined as a $P$-optimal folding. Recursive summation over all feasible $(i,j)$ pairs gives all the $P$-optimal secondary structures. However, we will see, that this algorithm does not compute really all $P$-optimal secondary structures.

Energy rules used are those set by Freier *et al.* [9]. This rules add single-base stacking energies for dangling bases adjacent to helices as well as for mismatched pairs adjacent to closing pairs of interior and hairpin loops. Also Ninio's correction for loopsided interior loops is used [31].

Nevertheless the authors wanted to present a typical set of $P$-suboptimal secondary structures of a given RNA sequence by introducing a distance measure. The procedure may generate a large number of foldings within 5 or 10 percent of the minimum free energy. Many of them will be very similar to each other. For this reason, a distance function was developed as a way of measuring topological differences between two structures. The distance between two foldings is the smallest whole number $d$ such that for every base pair $(i,j)$ of one, there is a base pair $(h,k)$ of the other satisfying

$$|i\ -\ h|\ \leq\ d\ \ and\ \ |j\ -\ k|\ \leq\ d. \tag{3}$$

This dimensionless quantity is zero, if and only if the two structures are identical. With this measure, in order to gain a set of typical $P$-suboptimal structures of a RNA sequence, the deficit regarding the total number of secondary structures within $P$ percent of the minimum free energy has not such a big effect. However, there is no guarantee, to find all typical secondary structures using this method.

# 5 Maximum Matching as Illustration of Waterman's Concept

## 5.1 The Idea of "Maximum Matching"

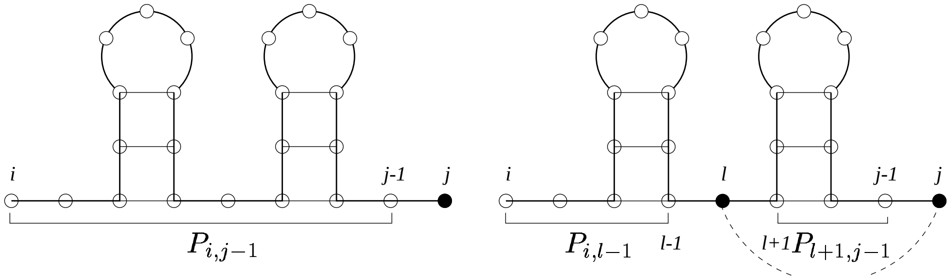
Before a description of the energy suboptimal folding is given in this subsection the basic ideas regarding the algorithmic implementation are shown with the "maximum matching" problem. "Maximum matching" is concerned with finding the structure providing the maximum number of basepairs. The first algorithmic solution was given by Nussinov [30] and was based on a dynamic programming consideration of Waterman [47] providing the assumption that the number of base pairs can be written as a sum of base pairs of noninteracting parts. The algorithm works by calculating optimal structures for all subsequences of the sequence $I$ to be folded, proceeding from smaller to larger fragments. Let $P_{i,j}$ be the maximum number of base pairs possible on the substructure $I_{i,j}$, then

$$P_{i,j} = \max\left\{ P_{i,j-1}, \max_{i \le l \le j-1} \left\{ \left[ P_{i,l-1} + 1 + P_{l-1,j-1} \right] \rho(a_l, a_j) \right\} \right\} \tag{4}$$

where $a_i$,$a_j \in \{A, U, G, C\}$ and

$$\rho(a_i, a_j) = \begin{cases} 1 & : \text{ when } a_i \text{ and } a_j \text{ can pair,} \\ 0 & : \text{ otherwise.} \end{cases}$$

Figure 8 gives a schematic representation of this procedure. Following these equations, the $P_{i,j}$-matrix is filled up. In table 1 a pseudo code of this procedure is given. The algorithm outlined so far calculates only the maximum number of basepairs, but no related structures. Typical dynamic programming implementations first calculate all entries in the $P$ array starting with the smallest subsequences and then construct the structure in a second pass proceeding from the largest to the smallest substructures. This technique, typical for dynamic programming, is called backtracking.



If the newly added base does not pair, the number of base pairs in part $i, j$ equals the number of base pairs in part $i, j-1$.

If the base $l$ pairs with $j$, the number of basepairs is the sum of all basepairs in the remaining part $i, l-1$ and the newly formed component $l-1, j-1$.

**Figure 8**: Schematic representation of the "maximum matching" problem.

```
for(i = 1...length)              \\ i:  [1 <-- length]
  for(j = i...length)            \\ j:  [i --> length]
    for(l = i...j)               \\ l,j is the
                                    considered pair
        temp = MAX(i <= l <= j: P[i,l-1]+1+P[l+1,j]))
    P[i,j] = MAX(P[i,j-1], temp)
max_number = P[1,length]
```

**Table 1**: **Pseudocode for the dynamic programming of the maximum matching problem:** `P[i,j]` denotes the maximum number of basepairs for the subsequence consisting of bases `i` through `j`. `length` denotes the length of the given sequence.

## 5.2 Backtracking and Waterman's Concept

When the procedure of dynamic programming is finished, $P_{1,n}$ gives the maximum number of basepairs, if $n$ denotes the length of the considered sequence. Starting with the segment $[1, n]$ the combination of segments $[1, l-1]$ and $[l+1, n-1]$ yielding the value of $P_{1,n}$ is found. The same procedure is performed for each new segment thus obtained.

### 5.2.1 Waterman's Algorithm

Waterman and Byers suggested a dynamic programming algorithm to find all solutions in the neighbourhood of an optimum [46]: The object of the "shortest path problem" is to locate the shortest path from node 1 to node $N$ in an acyclic network of $N$ nodes and $A$ arcs. Each arc $(i, j)$ has an associated weight $t(i, j)$. Nodes $i$ are labeled with $f(i)$, the length of the shortest path from node $i$ to node $N$. Provided Bellman's insight of optimality [2] "subpaths of optimal paths are themselves optimal" the recursion

$$f(i) = min\{\, t(i, j) + f(j) : (i, j)\ an\ arc\,\} \tag{5}$$

follows. The idea is, that to reach $i$ from $N$, the last step is from some node $j$. The node $j$ must be reached in an optimal manner, if $j$ is an optimal path from $N$ to $i$. Note, that $f(N) = 0$ is required to start the recursion. So far nothing else than the procedure of dynamic programming and backtracking was outlined. The new algorithm requires an interval $e$ above the optimal length $f(1)$ from the user. All paths less than or equal to the quantity $f(1)+e$ should then be found by the algorithm. The node labels $f(j)$ are found by working backwards from node $N$ until node 1 is labeled. The new algorithm then performs a depth-first search with stacking, starting at node 1 and continuing until all near-optimal paths are found. Consider a node $x$ not equal to the destination. Some path $P$ with cumulative distance $d$ led to node $x$ from node 1. The test for entry of the arc $(x, y)$ and distance $d$ onto the stack now takes the general form for all $(x, y) \in A$

$$d + t(x, y) + f(y) \leq f(1) + e, \tag{6}$$

where $d$ is the cumulative distance to node $x$ from node 1 by path $P$ (not necessarily the shortest path!), $t(x, y)$ is the distance from node $x$ to node $y$, and $f(y)$ is the optimal remaining distance to node $N$ from node $y$. The algorithm constructs a path $P$ of length $d$ from node 1 to node $N$. Then $P$ and $d$ are output and the stack is examined to see, if other near-optimal paths exist. Hence the algorithm performs a last in, first out or depth first search.

### 5.2.2 Application of Waterman's Concept to the Maximum Matching Problem

The model of the algorithm outlined above can be applied to RNA maximum matching to find all structures , whose number of base pairs lies within $P_{1,n}$ and $P_{1,n} - \delta$. Starting with the segment $[1, n]$ the combination of segments $P_{1,l-1}$ and $P_{l+1,n-1}$, which fulfill this condition is found. The found combination of intervals $[1, l-1]$ and $[l+1, n-1]$ is written to an interval stack. Also the found base pair $(l, n)$ is written to a separate base pair stack. Both stacks are contents of a separate state stack. In the next round of the algorithm the last state is taken from the state stack and the last interval, in general $[i, j]$, from the interval stack within that state (*last-in, first-out*). Again within the interval $[i, j]$ those combinations of $P_{i,l-1}$ and $P_{l+1,j-1}$ are traced, whose number of base pairs lies within $P_{1,n}$ and $P_{1,n} - \delta$. However, this time also the already found basepair and the best possible number of basepairs of the intervals remaining on the stack denoted by $P_{p,q}$ must be taken into account, so that the condition reads as

$$N_{bp} + P_{i,l-1} + 1 + P_{l+1,j-1} + \sum_{p,q} P_{p,q} \;\geq\; P_{1,n} - \delta. \tag{7}$$

in analogy to

$$d + t(x, y) + f(y) \;\leq\; f(1) + e. \tag{8}$$

$N_{bp}$ denotes the number of all already found basepairs and $p, q$ the several yet unconsidered intervals remaining on the interval stack. The state containing the interval stack the interval $[i, j]$ was taken from is copied and the new found basepair and intervals are written to the related stack within the state. The state is pushed back to the state stack. That happens to every new found combination of segmentations, which accomplish the condition outlined above. If no basepair, which accomplish the condition, can be found, the remaining state is pushed back to the state stack. The iteration goes on by taking out the first state of the state stack and following the first interval of the interval stack. If the interval stack is empty, a solution *i.e.* a structure, is found, and the state is skipped. The iteration continues until no state remains on the

```
while (pop STATE)
  pop INTERVAL [i,j]
  rest = sum_of_pairs [STATE->BASEPAIRS]

  if (P[i,j-1] + rest >= P[1,n] + delta) \\ j unpaired
    copy STATE
    push INTERVAL [i,j-1]
    push STATE

  for (l = i...j)
    if (P[i,l-1] + 1 + P[l+1,j-1] + rest >= P[1,n] + delta)
           \\ if positive, l,j are a basepair, otherwise not
      copy STATE
      push BASEPAIR [l,j]
      push INTERVAL [i,l-1]
      push INTERVAL [l+1,j-1]

  if (nothing_is_pushed)
    push STATE
  else
    free STATE

  free INTERVAL
```

**Table 2**: **Pseudo code for backtracking of the maximum matching problem including Waterman's algorithm:** `P[i,j]` denotes the maximum number of basepair for the subsequence consisting of bases `i` through `j`. `rest` is the optimal number of basepairs the remaining intervals on the interval stack contain within a state. `STATE` denotes the last entry in the state stack. One `STATE` consists of a interval stack and a basepair stack. `INTERVAL` and `BASEPAIR` denote the last entries in the interval and basepair stack respectively. `push` and `pop` denote the subroutines writing and taking away the last entry from the various stacks.

stack. A summary of this procedure is given with the pseudo code of this kind of backtracking in table 2.

## 5.3 Check for Reliability of the new Algorithm

With Cupal's density of states there exists a tool to check the results of the new algorithm [6]. With his algorithm it is possible to calculate the number of all possible structures of a given sequence. Hence the results of the new algorithm with $\delta$ equal to the maximum number of basepairs can be compared to the results of a density of states calculation. The Density of States algorithm and the new algorithm yield identical results for the number of structures of sequences shown in table 3 as far we were able to check.

**Table 3**: Check for reliability of the new algorithm: The Density of States algorithm and the new algorithm yield identical results for the number of structures of the shown sequences.

| Sequence | Number of Base pairs | Number of Structures |
|---|---|---|
| `ACUGAUCGUAGUCAC` | 4 | 142 |
| `AAGGCGAAAACCGCACCCCAAAAGGGAAC` | 7 | 7232 |
| `GGGGACCCUUUGGGAGGGAAACCCACCCC` | 10 | 1201833 |
| `GGGGGGACCCUUUGGGAGGGAAACCCACCCCCC` | 12 | 11208028 |

# 6 Structure Prediction

Before an idea of the computational implementation is given the underlying energy model is illustrated in the upcoming section.

## 6.1 Thermodynamic Nearest Neighbor Parameters

Base-base interactions in nucleic acids are of three kinds: (a) base pairing in the plane of the bases (horizontal) due to hydrogen bonding, (b) base stacking perpendicular to the plane of bases stabilized by London dispersion forces and hydrophobic effects [33, 34] and (c) entropic contributions, which get lost by closing a multi-loop. While hydrogen bonding is fundamental to the genetic code, all kinds of interactions play a significant role in determining the spatial structure and energy state of an RNA molecule.

The results of both quantum chemical calculations and thermodynamic measurements suggest that base pairing contributions to the total energy depend exclusively on the base pair composition, whereas base stacking contributions depend on base pair composition *and* base sequence *i.e.* the upstream and downstream neighbors along the chain [34]. The *nearest neighbor model* introduced by Borer *et al.* [4] makes the assumption that the stability of a base pair or any other structural element of an RNA depends only on the identity of the adjacent bases and/or base pairs. The model is justified by the major contribution of short-range interactions (hydrogen bonding, base stacking) to the overall stabilizing energy of nucleic acid structures. In addition, it is natural to assign loop entropies to entire loops instead of individual bases. Treating stacks as a type of loops of degree 2 and size 0, one assumes therefore that the energy of an RNA secondary structure $\Phi$ is given by the sum of energy contributions $\epsilon$ of it's loops $L$.

$$E(\Phi) = \sum_{L \in \Phi} \epsilon(L) + \epsilon(L_{ext}), \tag{9}$$

where $L_{ext}$ is the contribution of the "exterior" loop containing the free ends. In the following we shall discuss the individual contributions in some detail.

The current work uses the compilation of [9, 13, 42], who performed measurements of melting curves of oligonucleotides at 37°C in 1 M NaCl.

**Stacked pairs, Watson-Crick and G-U pairs** contribute the major part of the energy stabilizing a structure. Surprisingly, in aqueous solution parallel stacking of base pairs is more important than hydrogen bonding of the complementary bases. By now all 21 possible combinations of A-U, G-C and G-U pairs have been measured in several oligonucleotide sequences with an accuracy of a few percent. The parameters involving G-U mismatches were measured more recently in Douglas Turner's group [13] and brought the first notable violation of the nearest-neighbor model: while all other combinations could be fitted reasonably well to the model, the energy of the $\begin{smallmatrix} 5'\mathrm{G\text{-}U}\,3' \\ 3'\mathrm{U\text{-}G}\,5' \end{smallmatrix}$ stacked pair seems to vary from $+1.5\,\mathrm{kcal/mol}$ to $-1.0$ kcal/mol depending on its context.

**Unpaired terminal nucleotides and terminal mismatches**: Unpaired bases adjacent to a helix may also lower the energy of the structure through parallel stacking. In the case of free ends, the bases dangling on the $5'$ and $3'$ ends of the helix are evaluated separately, and unpaired nucleotides in multiloops are treated in the same way. For interior and hairpin loops the so called *terminal mismatch* energy depends on the last pair of the helix and both neighboring unpaired bases. While stacking of an unpaired base at the $3'$ end can be as stabilizing as some stacked pairs, $5'$ dangling ends usually contribute little stability. Terminal mismatch energies are often similar to the sum of the two corresponding dangling ends. Typically, terminal mismatch energies are not assigned to hairpins of size three. Few measurements are available for the stacking of unpaired nucleotides on G-U pairs, and for this reason they have to be estimated from the data for G-C and A-U pairs.

**Loop energies** are destabilizing and modeled as purely entropic. Few experimental data are available for loops, most of these for hairpins. The parameters for loop energies are therefore particularly unreliable. Data in the newer compilation by Jaeger *et al.* [19] differ widely from the values given previously [9]. Energies depend only on the size and type (hairpin, interior or bulge) of the the loop. Hairpins must have a minimal size of 3. Turner *et al.*

[42] extrapolated values for large loops ($k > 30$) logarithmically:

$$\mathcal{H}(k) = \mathcal{H}(30) + \text{const.} * \log(k/30) \tag{10}$$

Asymmetric interior loops are furthermore penalized [31] using an empirical formula depending on the difference $|u_1 - u_2|$ of unpaired bases on each side of the loop

$$\Delta F_{\text{ninio}} = \min\Big\{\Delta F_{\max}, |u_1 - u_2| * \Delta F_{\text{ninio}}\left[\min\{4.0, u_1, u_2\}\right]\Big\}. \tag{11}$$

For bulge loops of size 1 a stacking energy for the stacking of the closing and the interior pair is usually added, while larger loops are assumed to prohibit stacking. Finally, a set of eight hairpin loops of size 4 are given a bonus energy of 2 kcal/mol. These tetraloops have been found to be especially frequent in rRNA structures determined from phylogenetic analysis. Melting experiments on several tetraloops [1] show a strong sequence dependence that is not yet well reflected in the energy parameters.

No measured parameters are available for multi-loops, their contribution (apart from dangling ends within the loop) being usually approximated by the linear ansatz

$$\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot \text{degree} + \mathcal{M}_B \cdot \text{unpaired}, \tag{12}$$

where $\mathcal{M}_C$ denotes the multi-loop closing energy, $\mathcal{M}_I$ denotes the energy contribution related to the number of stems (= loop *degree*), and $\mathcal{M}_B$ the destabilizing energy per *unpaired* base (size of the loop). Good results have been achieved using $\mathcal{M}_C = 4.6$, $\mathcal{M}_I = 0.4$ and $\mathcal{M}_B = 0.1$ kcal/mol by Jaeger *et al.* [19]. While a logarithmic size dependency of loop energies would be more realistic following the Jacobson-Stockmayer theory, the linear ansatz allows faster prediction algorithms. Since all energies are measured relative to the unfolded chain, free ends do not contribute to the energy.

## 6.2 Assigning Energy Parameters to Graphs

The energy contributions described above result in nearest neighbor parameters for the individual types of loops. Assigning energy values to secondary structure graphs depending on the degree $k$ and size $z$ of each loop, we distinguish the following cases:

(1) *Stacking Pairs* ($k = 2$, $z = 0$): The energy $\mathcal{I}(i,i{+}1,j{-}1,j)$ depends on the identity of the bases $i$, $i{+}1$, $j{-}1$, $j$

(2) *Interior Loops and Bulges* ($k = 2$): The energy $\mathcal{I}(i,k,l,j)$ depends on the identity of the bases $i$, $k$, $l$, $j$ and on the size $z$ of the loop with $z = k - (i+1) + j - (l+1)$.

(3) *Hairpin Loops* ($k = 1$): The loop energy $\mathcal{H}(z)$ depends on the size $z$ of the loop with $z = j - i$. $m$ is the minimal loop size with $m = 3$.

(4) *Multi-loops* ($k \geq 2$): Multi-loop energies $\mathcal{M}$ are modeled by the linear ansatz

$$\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot \text{degree} + \mathcal{M}_B \cdot \text{unpaired}, \tag{13}$$

where $\mathcal{M}_C$ denotes the multi-loop closing energy, $\mathcal{M}_I$ denotes the energy contribution related to the number of stems (= loop-degree) and $\mathcal{M}_B$ the destabilizing energy per unpaired base (size of the loop).

(5) *Terminal Mismatches*: The mismatch energy $mm(i,i{+}1,j{-}1,j)$ depends on the identity of stacking first unpaired bases $i+1, j-1$ adjacent to the last base pair $(i,j)$ of a helix within *interior loops* and *hairpins*.

(6) *Dangling Ends*: The dangling end energies $d^5$ $(i,j,i{-}1)$ and $d^3$ $(i,j,j{+}1)$ respectively, depend on the identity of stacking last unpaired bases $i-1$ and $j+1$ adjacent to the first base pair $(i,j)$ of a helix in *multi-loops* and *hairpins*.

## 6.3 Further Works

Energy parameters for the contributions described above have been derived mostly from melting experiments on small oligonucleotides. The first compilation of such parameters was done by Salser [35]. The parameters most widely in use today are based on work of Turner and coworkers. More recently the differences between symmetric and asymmetric loops have been reported to be only half the magnitude suggested by Papanicolau *et. al.* [31] and of higher sequence dependence [32]. Thus equation 11 was simplified to

$$\Delta F = \min \Big\{ 3.0,\ 0.3 * |u_1 - u_2| \Big\} \tag{14}$$

depending on the difference $|u_1 - u_2|$ of unpaired bases on each side of the loop. Serra *et al.* found a dependence of hairpin loop energies on the closing base pair [39] and presented a model to predict the stability of hairpin loops [38]. Walter and coworkers suggested a model system for the coaxial stacking of helices [43]. SantaLucia *et al.* [36] investigated the influence of hydrogen bonding between GA mismatches in interior loops by substitution of functional groups. The work was extended to consecutive A-C, C-C, G-G, U-C and U-U mismatches finding evidence for stable hydrogen bonded U-U and C-C pairs [37]. Wu and Walter studied the stability of tandem GA mismatches and found them to depend upon both sequence and adjacent base pairs [44, 48]. Ebel and coworkers measured the thermodynamic stability of RNA duplexes containing tandem G-A mismatches [8]. Morse and Draper presented thermodynamic parameters for RNA duplexes containing several mismatches flanked by C-G pairs. Mismatches are reported to have a wide range of effects on duplex stability. The nearest neighbor model is considered not to be valid for G-A mismatches [27]. These results are, however, not yet included into the parameter set used in this work.

## 6.4 Dynamic Folding Algorithm

The additive form of the energy model in eqation 9 allows for an elegant solution of the minimum free energy problem through dynamic programming first realized and exploited by Waterman [45], [47]. The algorithm outlined below is based on an implementation by Zuker and Stiegler [53], [52]. Analogously to the maximum matching problem the algorithm works by calculating optimal structures for all subsequences of the sequence $I$ to be folded proceeding from smaller to larger fragments. An additional feature is the formal construction of multicomponent structures from smaller fragments. Let $C_{i,j}$ be the minimum energy possible on the substructure $I_{ij}$ provided that $i$ and $j$ pair. Since the energy of some substructure $\mathcal{S}_{i,j}$ with $i$ and $j$ paired is given by the energy of the loop closed by $(i,j)$ plus the energy of any loops directly interior to it,

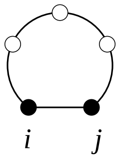
$$C_{i,j} = \min_{\substack{loops\ L \\ closed\ by\ i,j}} \{E(L) + \sum_{\substack{interior\ pairs \\ (p,q)\ \in\ L}} C_{p,q}\} \tag{15}$$

and $C_{i,i} = \infty$. Three subsets are contributing to this set of structures, depending on the number of base pairs immediately interior to $(i,j)$. The minimum energies of these three subsets are (recursively) obtained from smaller fragments:

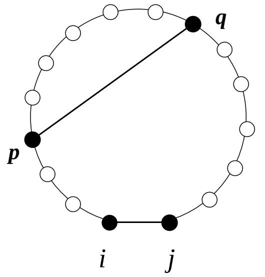
$$\begin{aligned} C_{i,j} &= \min\Big\{\mathcal{H}(i,j), \min_{\substack{p\in[i+1,j-m-2] \\ q\in[p+m+1,j-1]}} \{C_{p,q} + \mathcal{I}(i,j,p,q)\ \}, \\ &\qquad \min_{k\in[i+1,j-m-2]} \{F^{M}_{i+1,k-1} + F^{M1}_{k,j-1} + d^{5}_{i,j,j-1} + d^{3}_{i,j,i+1} + \mathcal{M}_C\}\Big\}. \end{aligned} \tag{16}$$

$\mathcal{H}(i,j)$ denotes the free energy of a hairpin loop closed by $(i,j)$. The second element is the minimum energy of all structures, where $(i,j)$ close an interior loop; their minimum energy equals the sum of the minimum energy of the smaller fragment, $C_{p,q}$, and the energy of the closing loop, $\mathcal{I}(i,j,p,q)$. Multiloop structures enclosed by $(i,j)$ are obtained by constructing the multi-loop from two parts, $F^{M}_{i+1,k-1}$ and $F^{M1}_{k,j-1}$, plus the multi-loop closing energy $\mathcal{M}_C$ and the contributions of dangling ends on the 5' and the 3' side of the pair $(i,j)$. $d^{5,(3)}_{i,j,j-1,(i+1)}$ denotes the energy contribution of the 5' (3') dangling end indicating the $(i,j)$-pair and the $i-1$ $(j+1)$ unpaired end. Note, that the dangling end contributions are added in every case even though the adjacent bases are paired. $F^{M1}_{i,j}$ denotes the minimum free energy of the rightmost stem plus an arbitry number of unpaired bases at the right side. $F^{M1}_{i,j}$ is obtained from the sum of the minimum energy of the stem, $C_{i,l}$, the multi-loop base energy, $\mathcal{M}_B(j-l)$, which is added for each unpaired base, and the multi-loop internal energy, $\mathcal{M}_I$.

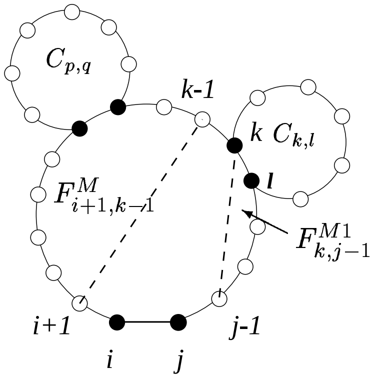
$$F^{M1}_{i,j} = \min_{l\in[i+m+1,j]} \Big\{\ C_{i,l} + \mathcal{M}_B(j-l) + d^{5}_{i,l,i-1} + d^{3}_{i,l,l+1} + \mathcal{M}_I\ \Big\} \tag{17}$$



For a given base pair $(i, j)$ there is only a single possibility to form a hairpin loop. Minimal loop size is 3.



Base pair $(i, j)$ closes an interior loop, base pair $(p, q)$ is immediatly interior to $(i, j)$. The number of structures for all possible values of $p$ and $q$ are considered.



Base pair $(i, j)$ closes a multi-loop, base pairs $(p, q)$, $(p', q')$ ... are immediatly interior to $(i, j)$. Multi-loop structures are divided into substructures containing the rightmost stem and the remaining structure. The energy $F^M_{i+1,k-1}$ of arbitrary structures on the 5' part is again determined from smaller fragments. Dangling end contributions $d^{5,(3)}_{i,j,j-1,(i+1)}$ and multi-loop energy contributions $\mathcal{M}_C$, $\mathcal{M}_I$ and $\mathcal{M}_B$ are not detailed for simplicity.

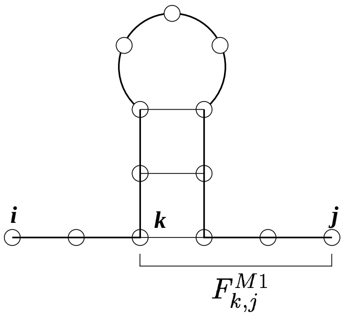
**Figure 9**: Schematic representation of the different terms yielding $C_{i,j}$ and $F^{M1}_{i,j}$.

$F^M_{i+1,k-1}$, equ. (16), denotes the minimum free energy of the remaining section of a multi-loop structure. This section may contain one ore more stems. Figure 9 gives a schematic representation of the different terms yielding $C_{i,j}$ and $F^{M1}_{i,j}$ in equation 16 and 17. We derive for the minimum free energy

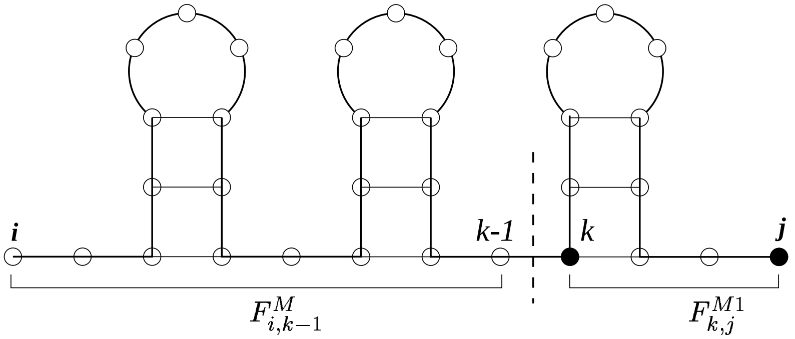
$$F^M_{i,j} = \min\Big\{ \min_{k\in[i+m+1,j-m-1]} \{F^M_{i,k-1} + F^{M1}_{k,j}\}, \tag{18}$$

$$\min_{k\in[i,j-m-1]} \{F^{M1}_{k,j} + \mathcal{M}_{\mathcal{B}}(k-i)\} \Big\}. \tag{19}$$

The first element yields the minimum energy of all multi-loop sections containing at least 2 stems. The second element treats all substructures, which contain only a single stem. The energy of such a structure is obtained from the sum of the minimum energy of the stem plus the bases at the right side, *i.e.* $F^{M1}_{k,j}$, see equ. (17), plus the energy of the unpaired bases at the left side of the



The energy of such remaining sections of a multi-loop structure can either be obtained from $F^{M1}_{k,j}$ plus the unpaired bases at the 5' end of the stem, if there exists one stem,



or from the sum of $F^{M}_{i,k-1}$, if the section contains at least 2 stems, plus the energy of the stem and the unpaired bases at the 3' end of the stem, i.e. $F^{M1}_{k,j}$.

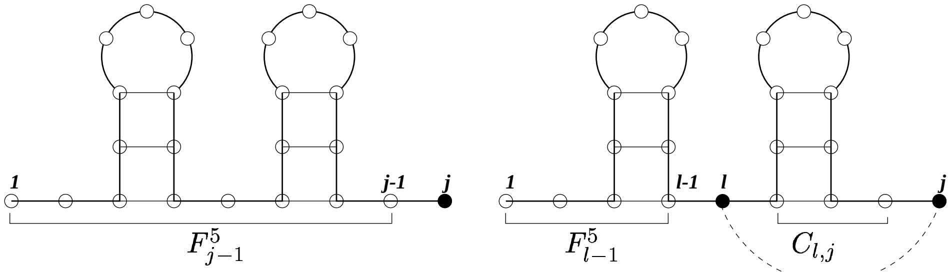
**Figure 10**: Schematic representation of the different terms yielding the energy of remaining sections of a multi-loop structure.

stem, $\mathcal{M}_B(k-i)$. Note, that the distinction of $F^{M}_{i,j}$ and $F^{M1}_{i,j}$ ensures, that there is only one decomposition of a structure into substructures. It avoids identical multi-loop decompositions while backtracking. A schematic illustration of the terms yielding the energy of multi-loop sections gives figure 10.

Let $F^5_j$ denote the minimum free energy on the segment*[1,j]*. $F^5_j$ can be constructed recursively using

$$F^5_j = \min_{l \in [1, j-m-1]} \left\{ F^5_{j-1}, F^5_{l-1} + C_{l,j} + d^5_{l,j,l-1} + d^3_{l,j,j+1} \right\} \tag{20}$$

with being $C_{l,j}$ defined in equation 16. The first term represents the case, that base $j$ is unpaired. When $l$ is paired with some base $j$, $F^5_j$ is given by the second term. A schematic representation of these terms is given in figure 11.



If the base $j$ does not pair, the energy of section $[1, j]$, $F^5_j$, equals $F^5_{j-1}$.

If the base $l$ pairs with $j$, the energy of section $[1, j]$ is the sum of section [1,l-1], $F^5_{l-1}$, and the section closed by the base pair $(l, j)$, $C_{l,j}$.

**Figure 11**: Schematic representation of the different terms yielding $F^5_j$. Dangling end contributions have been neglected for simplicity.

Table 4 summarizes the algorithm for the computation of the minimum free energy.

$$
\begin{aligned}
C_{i,j} &= \min\Big\{\mathcal{H}(i,j),\ \min_{\substack{k\in[i+1,j-m-2]\\ l\in[k+m+1,j-1]}}\left\{C_{k,l}+\mathcal{I}(i,j,k,l)\right\},\\
&\qquad \min_{k\in[i+1,j-m-2]}\left\{F^{M}_{i+1,k-1}+F^{M1}_{k,j-1}+\mathcal{M}_C\right\}\Big\}\\[2ex]
F^{M1}_{i,j} &= \min_{l\in[i+m+1,j]}\Big\{\,C_{i,l}+d^5_{i,l,i-1}+d^3_{i,l,l+1}+\mathcal{M}_B(j-l)+\mathcal{M}_{\mathcal{I}}\Big\}\\[2ex]
F^{M}_{i,j} &= \min\Big\{\min_{k\in[i+m+1,j-m-1]}\left\{F^{M}_{i,k-1}+F^{M1}_{k,j}\right\},\\
&\qquad \min_{k\in[i,j-m-1]}\left\{F^{M1}_{k,j}+\mathcal{M}_B(k-i)\right\}\Big\}\\[2ex]
F^5_j &= \min_{l\in[1,j-m-1]}\Big\{F^5_{j-1},F^5_{l-1}+C_{l,j}+d^5_{l,j,l-1}+d^3_{l,j,j+1}\Big\}
\end{aligned}
$$

**Table 4**: **Recursion for the calculation of the minimum free energy:** Calligraphic symbols denote energy parameters for different loop types: hairpin loops $\mathcal{H}(i,j)$, interior loops, bulges and stacks $\mathcal{I}(i,j,k,l)$; the multi-loop energy is modeled by the linear ansatz $\mathcal{M}=\mathcal{M}_C+\mathcal{M}_I\cdot\text{degree}+\mathcal{M}_{\mathrm{B}}\cdot\text{unpaired}$. The minimum free energy $C_{i,j}$ of substructures on the substring $[i,j]$, subject to the condition that $i$ and $j$ form a base pair, is determined recursively from smaller fragments. The contributions depend on the type of the secondary structure element as a consequence of the energy model. The base pair $(i,j)$ can be the closing pair of a hairpin, it may close an interior loop (or extend a stack) or it might close a multi-loop. The auxiliary variables $F^M$ and $F^{M1}$ are necessary for handling the multi-loops. The minimum free energy of the substring $[1,j]$ is stored in $F^5_j$.

## 6.5 Backtracking

After the procedure of dynamic programming is finished, $F^5_n$ gives the minimum free energy, denoted as *mfe*, since $n$ is the length of the considered sequence. However, we still don't know, which structure gives rise to this optimal energy. Now we try to construct the optimal structure by using a so called *backtracking* strategy. The mechanism is roughly the same as of the maximum matching problem. In the following setion we give an outline of the backtracking procedure for calculating the *mfe* structure. In the second section this concept will be extended by Waterman's concept.

### 6.5.1 Common Features

While calculation of the *mfe* proceeds from smaller to larger segments, backtracking progresses from the full length fragment $[1, n]$ to smaller ones. For any segment $[i, j]$ the task is to find all base pairs immediately interior to $i, j$. The same procedure is performed for each new segment thus obtained. This strategy indicates a *last in, first out* or *depth first* search and will hold until there is no segment left to investigate.

Given a segment $[1, j]$, *i.e.* a segment not enclosed by any base pair, we have to find the outermost base pair at the 3' end. The base $j$ is either unpaired, in which case

$$F^5_j = F^5_{j-1} \tag{21}$$

holds, yielding a new segment $[1, j-1]$. Otherwise, we have to find a $k$, such that

$$F^5_j = F^5_{k-1} + C_{k,j} + d^5_{k,j,k-1} + d^3_{k,j,j+1} \tag{22}$$

yielding a base pair $[k, j]$, which defines a new segment and the exterior segment $[1, k-1]$.

Backtracking in the $F^M$ array starts with the condition

$$F^M_{i,j} = F^M_{i,j-1} + \mathcal{M}_{\mathcal{B}}. \tag{23}$$

If this condition is fulfilled, no base pair is found and the 3' end is nibbled creating the segment $[i, j-1]$. The condition

$$F^M_{i,j} = C_{i,j} + d^5_{i,j,i-1} + d^3_{i,j,j+1} + \mathcal{M}_I \tag{24}$$

explores the base pair $(i, j)$, which delimits a multi-loop and the segment $[i+1, j-1]$. Using the $F^M$ array multi-loop decompositions can be performed accomplishing the condition

$$F^M_{i,j} = F^M_{i,k} + C_{k+1,j} + d^5_{k+1,j,k} + d^3_{k+1,j,j+1} + \mathcal{M}_I, \tag{25}$$

if the segment $[i,j]$ contains more than one stack. This condition finds a base pair $(k+1,j)$ and generates the segments $[i,k]$ and $[k+2,j-1]$. If the considered segment contains only one stack, the condition turns to

$$F^M_{i,j} = C_{k+1,j} + d^5_{k+1,j,k} + d^3_{k+1,j,j+1} + \mathcal{M}_I + (k-i+1) \cdot \mathcal{M}_B \tag{26}$$

finding the base pair $(k+1,j)$ and the segment $[k+2,j-1]$. Both conditions hold for $i<k<j$.

If the current segment is formed by a base pair $(i,j)$, then the pair $(i,j)$ could be either part of a stack, bulge, internal loop or multi-loop. If $(i,j)$ closes a loop of degree 2,

$$C_{i,j} = C_{p,q} + \mathcal{H}(i,j,p,q) \tag{27}$$

and $i<p<q<j$. This condition finds the base pair $(p,q)$ and therefore a new segment $[p,q]$. Alternately $(i,j)$ might close a hairpin. As a last possibility $(i,j)$ might close a multi-loop, in which case we have to find a $k$ accomplishing

$$C_{i,j} = F^M_{i+1,k} + F^M_{k+1,j-1} + d^5_{i,j,i+1} + d^3_{i,j,j-1} + \mathcal{M}_C, \tag{28}$$

and $i<k<j$. This condition finds two new segments $[i+1,k]$ and $[k+1,j-1]$. Note, that in this part there is no use of the $F^{M1}$ array, which will be just taken into account in the upcoming section. $F^{M1}_{i,j}$ ensures that there is only one decomposition of a structure into substructures. It avoids identical multi-loop decompositions while backtracking.

With these conditions we have the necessary tools to trace back through the filled arrays in order to get the *mfe* structure.

### 6.5.2 Extension of Backtracking by Waterman's Concept

The model of Waterman's algorithm can be applied to the RNA energy folding in order to find all suboptimal structures within a given energy range above the *mfe*. Subsequently, we will call this the "SUBOPT" algorithm. The conditions for tracing back through the various arrays are extended analogously to equation 6. That is the conditions of the previous section modified to something like

$$E_f + E_{i,j} + \sum_{k,l} E_{k,l} \ \leq \ E_{1,n} + \delta, \tag{29}$$

where $E_f$ is the sumed energy of all already found substructures. $E_{i,j}$ denotes the energy of the considered segment $[i, j]$ and $\sum E_{k,l}$ is the best possible energy of all remaining uninvestigated segments. $E_{1,n}$ is the optimal energy, whereas $\delta$ is the given energy range. In order to handle this amount of information, the data management used in the maximum matching algorithm was implemented. All found base pairs and segments are written to seperate base pair and interval stacks, which are contained by states. States are written to a state stack. This strategy will hold until there is no state left on the state stack to investigate.

Given a segment $[i, j]$ the SUBOPT algorithm starts with the condition of finding the outermost base pair at the 3' end with

$$E_f + F^5_{j-1} + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta \tag{30}$$

yielding a new segment $[1, j-1]$. Otherwise we have to find a $k$, such that

$$E_f + F^5_{k-1} + C_{k,j} + d^5_{k,j,k-1} + d^3_{k,j,j+1} + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta \tag{31}$$

yielding a base pair $(k, j)$, which defines also a new segment and the exterior segment $[1, k-1]$. Backtracking in the $F^M$ and $F^{M1}$ arrays starts with the condition

$$E_f + F^M_{i,j-1} + \mathcal{M}_B + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta \tag{32}$$

and

$$E_f + F^{M1}_{i,j-1} + \mathcal{M}_B + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta. \tag{33}$$

If these conditions are fulfilled, no base pair is found and the 3' end is nibbled creating the segment $[i, j-1]$. The condition

$$E_f + C_{i,j} + d^5_{i,j,i-1} + d^3_{i,j,j+1} + \mathcal{M}_I + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta \tag{34}$$

explores the base pair $(i, j)$, which delimits a multi-loop, and the segment $[i+1, j-1]$. Using the $F^M$ array multi-loop decompositions can be performed accomplishing the conditions

$$E_f + F^M_{i,k} + C_{k+1,j} + d^5_{k+1,j,k} + d^3_{k+1,j,j+1} + \mathcal{M}_I + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta \tag{35}$$

if the segment $[i, j]$ contains more than one stack. This condition finds a base pair $(k + 1, j)$ and generates the segments $[i, k]$ and $[k + 2, j - 1]$. If the considered segment contains only one stack, the condition turns to

$$E_f + C_{k+1,j} + d^5_{k+1,j,k} + d^3_{k+1,j,j+1} + \mathcal{M}_I + (k-i+1) \cdot \mathcal{M}_B + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta, \tag{36}$$

finding the base pair $(k + 1, j)$ and the segment $[k + 2, j - 1]$. Both conditions hold for $i < k < j$.

If the current segment $[i, j]$ is formed by a base pair $(i, j)$, then the pair could be either part of a stack, bulge, internal loop or multi-loop. If $(i, j)$ closes a loop of degree 2, we have

$$E_f + C_{p,q} + \mathcal{I}(i, j, p, q) + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta \tag{37}$$

provided $i < p < q < j$. This condition finds the base pair $(p, q)$ and therefore the new segment $[p, q]$. Alternately the base pair $(i, j)$ might close a hairpin or a multi-loop, in which case we have to find a $k$ such that

$$E_f + F^M_{i+1,k} + F^{M1}_{k+1,j-1} + d^5_{i,j,i+1} + d^3_{i,j,j-1} + \mathcal{M}_C + \sum_{k,l} E_{k,l} \ \leq F^5_n + \delta \tag{38}$$

with $i < k < j$. This condition finds two new segments $[i+1, k]$ and $[k+1, j-1]$ such that there is only one decomposition of a structure into substructures. It avoids identical multi-loop decompositions. A detailed summary of this procedure is given with the pseudo code of the SUBOPT algorithm in table 6.

## 6.6 Check for Reliability of the SUBOPT Algorithm

Taking the results of Cupal's density of states in the previous subsection we checked the results of the SUBOPT algorithm. Desireable is the same number of structures of a given sequence received by the SUBOPT algorithm as well as by the Density of States algorithm. Both algorithms yield identical results for the number of structures of sequences shown in table 5 as far we were able to check.

**Table 5**: Check for reliability of the SUBOPT algorithm: The Density of States algorithm and the SUBOPT algorithm yield identical results for the number of structures of the shown sequences.

| Sequence | Number of Structures |
|---|---|
| `ACUGAUCGUAGUCAC` | 142 |
| `AAGGCGAAAACCGCACCCCAAAAGGGAAC` | 7232 |
| `GGGGACCCUUUGGGAGGGAAACCCACCCC` | 1201833 |
| `GGGGGGACCCUUUGGGAGGGAAACCCACCCCC` | 11208028 |

```
while
  pop STATE [Basepairs, Intervals, partial_energy)
    pop INTERVAL [i,j, array_flag]
     if (array_flag = external)
           if (F5[j-1] + best_energy <= threshold)
              push INTERVAL [i,j-1,external], STATE
           for (k = 1...j)
              if (F5[k-1] + C[k,j] + d5 + d3 + best_energy <= threshold)
                 partial_energy += d5 + d3
                 push INTERVAL [k,j,2loop], STATE
           if (C[1,j] + d3 + best_energy <= threshold)
              partial_energy += d3
              push INTERVAL [1,j,2loop], STATE
     if (array_flag = 2loop)
           if (hairpin [unpaired] + best_energy <= threshold)
              partial_energy += hairpin [unpaired]
              push PAIR [i,j], STATE
           for (p = 1...i)
              for (q = 1...j)
                 if (I(i,j,p,q) + C[p,q] + best_energy <= threshold)
                    partial_energy += I(i,j,p,q)
                    push PAIR [i,j],[p,q],INTERVAL [p,q,2loop], STATE
           for (k = i...j)
               if (FM[i+1,k] + FM1[k+1,j-1] + d3 + d5 + best_energy <= threshold)
                  partial_energy += d3 + d5
                  push INTERVAL[i+1,k,mloop],[k+1,j-1,mloop],PAIR[i,j],STATE
     if (array_flag = FM1loop)
           if (FM1 [i,j-1] + Mb + best_energy <= threshold)
              partial_energy += Mb
              push INTERVAL [i,j-1,FM1loop], STATE
     if (array_flag = mloop)
           if (FM[i,j-1] + Mb + best_energy <= threshold)
              partial_energy += Mb
              push INTERVAL [i,j-1,mloop], STATE
           for (k = i...j)
              if (FM[i,k] + C[k+1,j] + Mi + d5 + d3 + best_energy <= threshold)
                 partial_energy += Mi + d5 + d3
                 push INTERVAL [i,k,mloop],[k+1,j,2loop], STATE
           for (k = i...j)
              if (C[k,j] + Mi + (k-i)*Mb + d5 + d3 + best_energy <= threshold)
                 partial_energy += Mi + (k-i)*Mb + d5 + d3
                 push INTERVAL [k,j,2loop], STATE
     if (nothing_is_pushed)
        push STATE
     else
        free STATE, INTERVAL
```

**Table 6**: **Pseudocode for the SUBOPT algorithm:** The arrays, contributions of multi-loops and dangling ends are denoted as in the text. `I(i,j,p,q)` denotes the energy contents of either stacks, bulges or interior loops. `best_energy` is the sum of the optimal energy of the remaining intervals on the interval stack and the already found substructures (`partial_energy`). `STATE` is the last entry in the state stack. One `STATE` consists of an interval stack, a base pair stack and the `partial_energy` entry. `INTERVAL` and `BASEPAIR` denote the last entries in the interval and base pair stack. Every `INTERVAL` wears an `array_flag` directing the considered interval to the relevant conditions. `push` and `pop` denote writing and taking away the last entries from the stacks. `threshold` means *mfe* plus $\delta$.

# 7 Performance of the SUBOPT Algorithm

## 7.1 CPU and Memory Requirements

In the following section we will have a look on the CPU and memory requirements of the SUBOPT algorithm considering calculations of suboptimal structures of RNA sequences of variable length within various ranges of energy. As test sequences we took 4 sequences with 25, 50, 75 and 100 bases length. Energy ranges were taken in multiples of $kT$ ($\sim$ 0,61 kcal/mol). The test sequences are shown in figure 12. Table 7 shows the results of the CPU and memory requirements of the SUBOPT algorithm. Also the number of calculated suboptimal structures of the considered sequences found within the

```
GGACCCUUUGGGAGGGAAACCCACC
AGGGGGGGAAAGGGGGAAAACCCCCAAACCCCAAAAGGGAAAACCCCCC
GCGGAUUUAGCUCAGDDGGGAGAGCGCCAGACUGAAYAUCUGGAGGUCCUGUGTPCGAUCCACAGAAUUCGCACC
CCCCACCCAAAGGGAGGGUUUCCGCGGAUUUAGCUCAGDDGGGAGAGCGCCAGACUGAAYAUCUGGAGGUCCUGUGTPCGAUCCACAGAAUUCGCACCAC
```

**Figure 12**: Test sequences used for providing performance data regarding CPU and memory requirements of the SUBOPT algorithm.

**Table 7**: Data of CPU and memory requirements of the SUBOPT algorithm using sequences and energy ranges of variable size. Also the number of calculated suboptimal structures are shown.

| sequence length | range of energy/$kT$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | 5 | 10 | 12 | 15 | 17 | 20 | |
| 25 | 17 | 187 | 441 | 1299 | 2569 | 6048 | structures |
| | 0.01 | 0.02 | 0.05 | 0.12 | 0.22 | 0.49 | MByte |
| | 0 | 0 | 0 | 0 | 0 | 0 | CPU secs |
| 50 | 9 | 108 | 254 | 900 | 2178 | 6477 | structures |
| | 0.02 | 0.03 | 0.05 | 0.11 | 0.23 | 0.65 | MByte |
| | 0 | 0 | 0 | 0 | 0 | 2 | CPU secs |
| 75 | 86 | 1664 | 5056 | 24299 | 67601 | 295722 | structures |
| | 0.06 | 0.25 | 0.69 | 3.16 | 8.70 | 34.11 | MByte |
| | 0 | 1 | 2 | 10 | 34 | 201 | CPU secs |
| 100 | 121 | 4439 | 16567 | 103935 | 341054 | 1864633 | structures |
| | 0.09 | 0.78 | 2.75 | 16.70 | 54.21 | 295.92 | MByte |
| | 1 | 6 | 10 | 54 | 169 | 1815 | CPU secs |

various ranges of energy are also shown. The calculations were performed on the Alpha-Cluster at the Rechenzentrum of the University of Vienna. The cluster consists of sixteen DEC AlphaServers 2100 5/375 with four CPUs each. Interestingly the number of structures, CPU and memory requirements increase with growing length of the considered sequence and range of energy as well. The data show a highly exponential behaviour. However, such a growth of the memory requirement with increasing length of sequence and range of energy exhausts the memory recources quickly. Having a glimpse on the correlations of memory and CPU requirements one might expect both scaling approximately linearly with the number of calculated suboptimal structures.

## 7.2 Performance in Comparison to Zuker's Algorithm

In section 4.2.2 we maintained that Zuker's algorithm is not able to find all suboptimal secondary structures of a RNA sequence within a given range of energy. In this subsection we want to compare the results of Zuker's algorithm with those of the SUBOPT algorithm.

**Table 8**: Performance of Zuker's algorithm in comparison to the SUBOPT algorithm. Within 10 percent of the *mfe* all suboptimal structures were calculated with both algorithms. Percentage denotes the part of the number of suboptimal structures calculated by Zuker's algorithm to this found by the SUBOPT algorithm.

| Sequence | Number of Structures (ZUKER) | Number of Structures (SUBOPT) | percentage |
|---|---|---|---|
| `RH1660(modified)` | 2 | 7 | 40.0 |
| `RH1660(unmodified)` | 9 | 19 | 47.4 |
| `RI1660(modified)` | 4 | 7 | 57.1 |
| `RI1660(unmodified)` | 17 | 79 | 21.5 |
| `RV1660(modified)` | 6 | 11 | 54.6 |
| `RV1660(unmodified)` | 24 | 131 | 19.4 |
| `RS1661(modified)` | 17 | 73 | 23.3 |
| `RS1661(unmodified)` | 15 | 91 | 16.5 |
| `RE1660(modified)` | 7 | 30 | 23.3 |
| `RE1660(unmodified)` | 10 | 40 | 25.0 |

Within 10 percent of the minimum free energy the number of suboptimal structures of 10 sequences were calculated using both algorithms. The result obtained is the percentage of the number of suboptimal structures calculated by Zuker's algorithm to this found by the SUBOPT algorithm. As sequences we chose 5 natural tRNA sequences of *E.coli*. The other 5 sequences were the same unmodified by translating the modified nonpairing bases to the natural pairing ones (see section 8.2 and Appendix A for further information).

In Table 8 the results are shown. Obviously the SUBOPT algorithm finds much more suboptimal structures within a given range of energy than Zuker's algorithm does.

# 8 Results

## 8.1 What is “Well-definedness” ?

In this section we want to find measures of the “well- definedness” of a secondary structure. Previous definitions [18] of the well-defindness are restricted to a certain region of the structure with

$$d(k) = \max\Big\{\max_i \{P_{i,k}, P_{k,i}\}, 1 - \sum_i P_{i,k}\Big\}. \tag{39}$$

$P_{i,k}$ is the base pair probability of base pair $(i,k)$, whereas $d(k)$ is the probability of the most probable base pair involving $k$ or the probability that $k$ is unpaired, whichever is larger. The base pair probability is defined as the probability of a base pair $(i,j)$ in a Boltzmann weighted ensemble of structures:

$$P_{i,j} = \sum_{\substack{\Phi \\ i,j\,\in\,\Phi}} P(\Phi) = \sum_{\substack{\Phi \\ i,j\,\in\,\Phi}} \frac{e^{-\frac{E(\Phi)}{kT}}}{Z} \tag{40}$$

with $P(\Phi)$ being the probability of a structure $\Phi$ with energy $E(\Phi)$ and $Z$ being the partition function. This equation implicates a measure of well-definedness for the whole structure. If $P(\Phi)$ denotes the probability of the structure $\Phi$ in a Boltzmann weighted ensemble, we can define the *mfe* structure as the most probable and best “defined” one. Subsequently the fraction of the *mfe* structure in the Boltzmann ensemble will be denoted as $f_{mfe}$ with

$$f_{mfe} = \frac{e^{-\frac{E_{mfe}}{kT}}}{Z}. \tag{41}$$

Using energetical terms the equation turns to

$$kT \ln f_{mfe} = F - E_{mfe} \tag{42}$$

with $F$ being the free energy of the ensemble. In the following sections we try to find additional measures of well-definedness.

## 8.2 *E.coli* tRNA Sequences

A number of tRNA sequences from EMBL tRNA Database, based on a compilation of Steegborn [41], were analyzed in the following section. In this subsection a short introduction of the treatment of tRNA sequences is given (See Appendix A for the sequences and sequence numbers referred to in the text). tRNAs differ to some extent from other types of RNA: tRNAs contain a large variety of modified bases in addition to the four standard bases A, C, G, and U. There are, however, no experimentally measured parameters available for non-standard bases. It is therefore necessary to develop a consistent method for dealing with these bases. Since it seems plausible, that some of these bases are modified to prevent bonding, a class of non-bonding bases is introduced. This method was first suggested by Ninio [29]. In 1993 Higgs, following Ninio, treated the following bases as non-bonding: Dihydrouridine (D), 7-methyl guanosine (7), N2-methyl guanosin (L), 1-methyl guanosine (K), queuosine (Q), wybutosine (Y) and 3-methyl cytidine ('). All other bases were treated as the standard base to which they most resemble [14]. A slightly different method was described by Higgs in 1995 [15]: Since the majority of all tRNA sequences fit the well-known cloverleaf folding pattern, it is possible to identify a class of modified bases, which never occur in a paired position. These bases are treated as non-bonding. All other bases are translated to their standard base analogue. This leads to the following assignments:

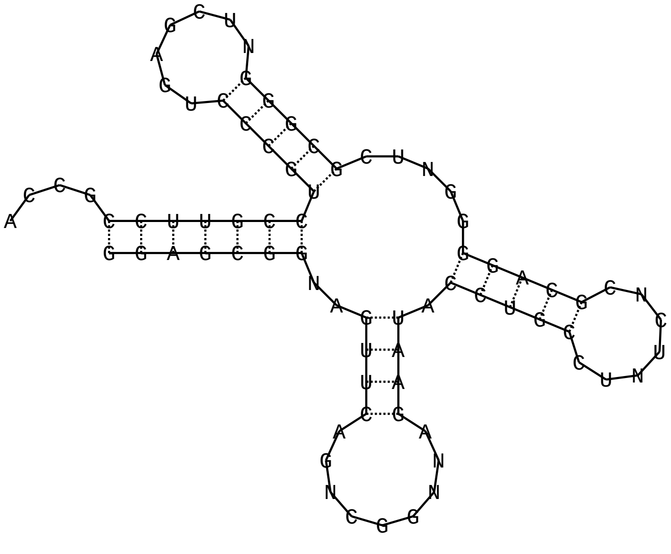
`H ^` → A
`< B M ?` → C
`; L # R` → G
`N J P ] Z` → U
all other symbols → N

For the purpose of comparison, a second set of "natural" tRNA sequences was obtained by translating all modified bases to the corresponding unmodified ones (See Appendix A for further details of translation).

## 8.3 Gapstatistics

### 8.3.1 Definitions

A pool of 2000 inverse folded sequences whose *mfe* structure is identical to those of selected natural tRNA sequences were generated. In addition a pool of inverse folded sequences of these selected tRNA structures were generated with nonpairing bases (N) at sites of the sequence, where they occur in the natural tRNA sequences. Figure 13 shows the typical tRNA structure and their related natural *E.coli* tRNA sequences used for generating those pools. The algorithm yields all secondary structures within a given interval of energy above the *mfe* of a given sequence. Hence it was possible to calculate the structures and energies of the first and the second best structures; *i.e.* the energy gap



```
RI1660 AGGCUUGUAGCUCAGGDGGDDAGAGCGCACCCCUGAU6AGGGUGAG7XCGGUGGTPCAAGUCCACPCAGGCCUACCA
RI1661 AGGCUUGUAGCUCAGGUGGDDAGAGCGCACCCCUGAU6AGGGUGAG7XCGGUGGTPCAAGUCCACPCAGGCCUACCA
RR1661 GCAUCCG4AGCUCAGCDGGADAGAGUACUCGGCUICG/ACCGAGCG7XCGGAGGTPCGAAUCCUCCCGGAUGCACCA
RV1660 GCGUCCG4AGCUCAGDDGGDDAGAGCACCACCUUGACAUGGUGGGG7XCGGUGGTPCGAGUCCACUCGGACGCACCA
RV1661 GCGUUCA4AGCUCAGDDGGDDAGAGCACCACCUUGACAUGGUGGGG7XCGUUGGTPCGAGUCCAAUUGAACGCACCA
RD1660 GGAGCGG4AGUUCAGDCGGDDAGAAUACCUGCCUQUC/CGCAGGGG7UCGCGGGTPCGAGUCCCGPCCGUUCCGCCA
```

**Figure 13**: Structure and related natural sequences of a *E.coli* tRNA. This structure was used to generate pools of inverse folded sequences.

between the ground state and the two first "excited" states. Subsequently the first and second gap of energy will be denoted as $\Delta G_1$ and $\Delta G_2$. The fraction of $\Delta G_1$ and $\Delta G_2$ in the *mfe* can be plotted against their frequency in the pools. In order to gain an idea of how "well defined" a structure is, the fraction of the *mfe* structure in the partition function, denoted as $f_{mfe}$, is plotted against the frequency. Subsequently we try to find a connection between $f_{mfe}$ and $\Delta G_1$. Additionally pools of 2000 inverse folded sequences, whose *mfe* structure is identical to those cross shaped structures shown in figure 14, were generated and investigated in the same manner.

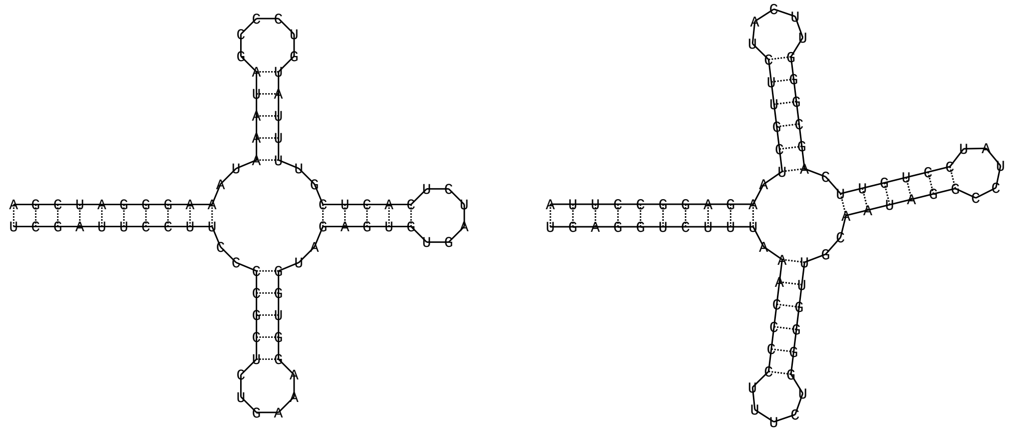


**Figure 14**: Cross shaped structures A (left) and B (right) similar to the cloverleaf structure of tRNAs. These structures were used to generate pools of 2000 inverse folded sequences.

### 8.3.2 Results

In figure 15 we show the comparison of the distributions of $\Delta G_1$ and $\Delta G_2$ regarding the pools of tRNA sequences. The distributions pertaining the modified sequences are much broader than those of the unmodified sequences. It is interesting to note, that these distributions of modified sequences are shifted

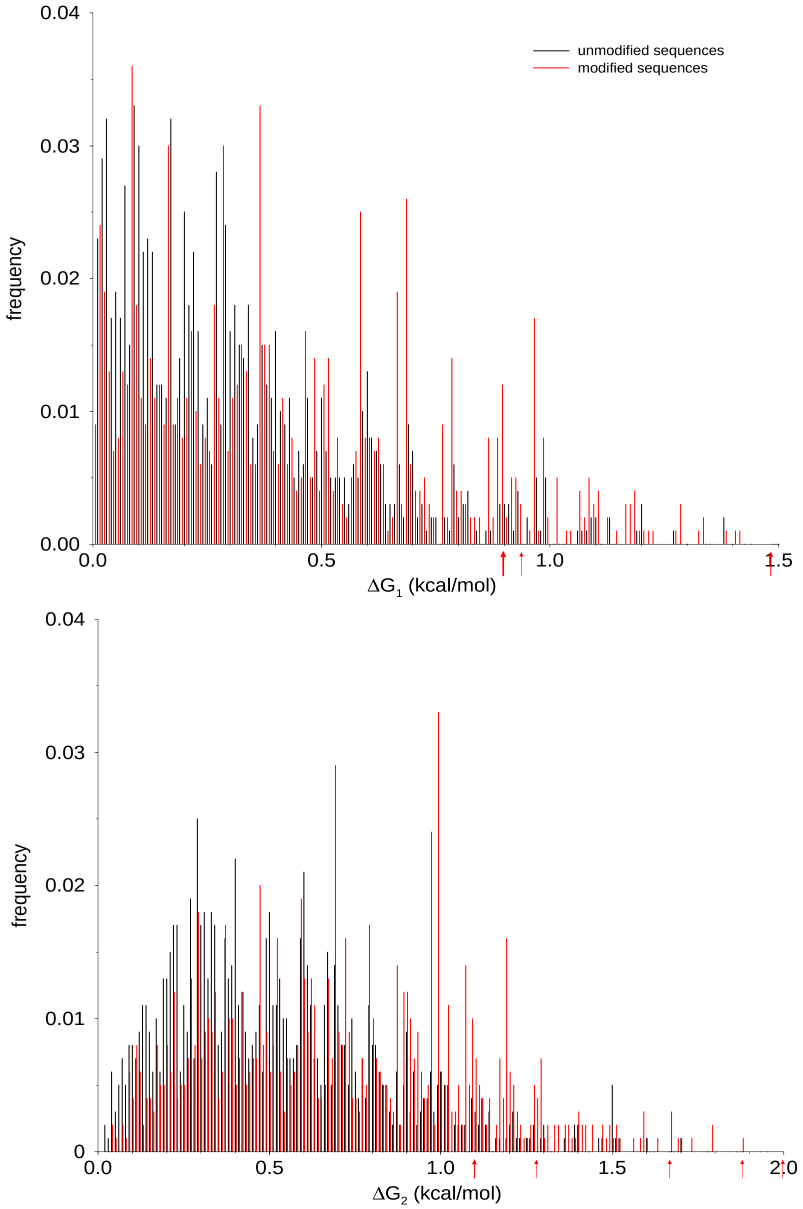


**Figure 15**: Comparison of the energydistributions of $1^{st}$ (up) and $2^{nd}$ (down) gap energies regarding pools of 2000 inverse folded modified and unmodified tRNA sequences. Arrows indicate the position of the natural 6 tRNA sequences.

to higher values of gap energies. Obviously the nonpairing modified bases have a strong influence on the distribution of gap energies. In figure 16 we show the fraction of *mfe* in the Boltzmann ensemble, $f_{mfe}$, plotted against its frequency in the pool of natural tRNA sequences. Interestingly these distributions are similar to the distributions of first gap energies. There seems to be a correlation of the energy gap between the ground state and the $1^{st}$ "excited" state ($\Delta G_1$) and the fraction of the *mfe* in the Boltzmann ensemble ($f_{mfe}$). Strictly speaking a higher $\Delta G_1$ is related to a higher $f_{mfe}$, which means a better defined structure. In order to show this fact, $\Delta G_1$ is plotted against $f_{mfe}$ regarding the pools of tRNA sequences in figure 17. Note, that the natural logarithm of $f_{mfe}$ in multiples of $kT$ ($\sim$ 0.61 kcal/mol) is the difference of the free energy of the ensemble and the *mfe* as outlined in equation 42. Compared to the modified

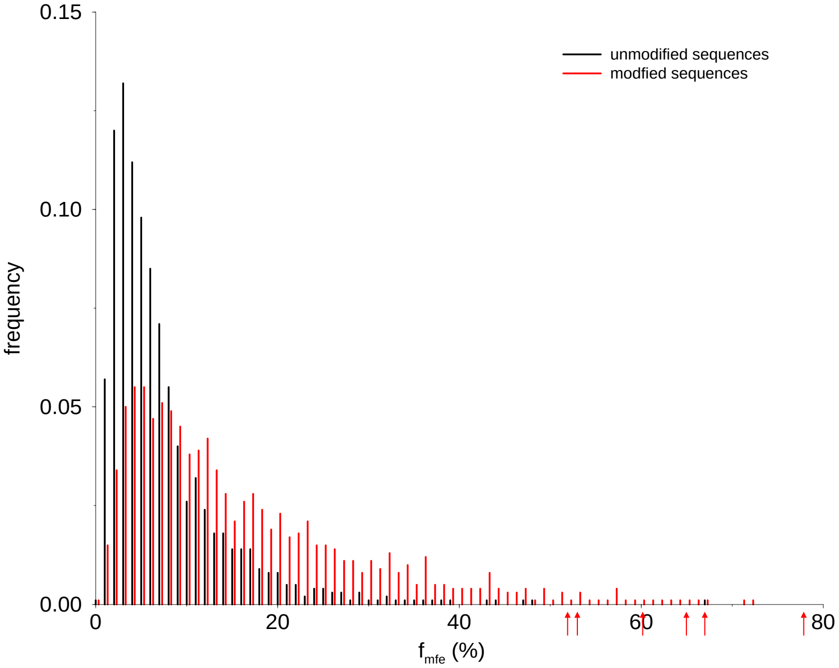


**Figure 16**: The abscisse shows the fraction of the *mfe* structure in the Boltzmann ensemble ($f_{mfe}$). The ordinate shows the frequency with which that fraction was realized in the modified and unmodified tRNA pools. Arrows indicate the position of the 6 natural tRNA sequences.

tRNA sample the data points belonging to the unmodified tRNA sequences

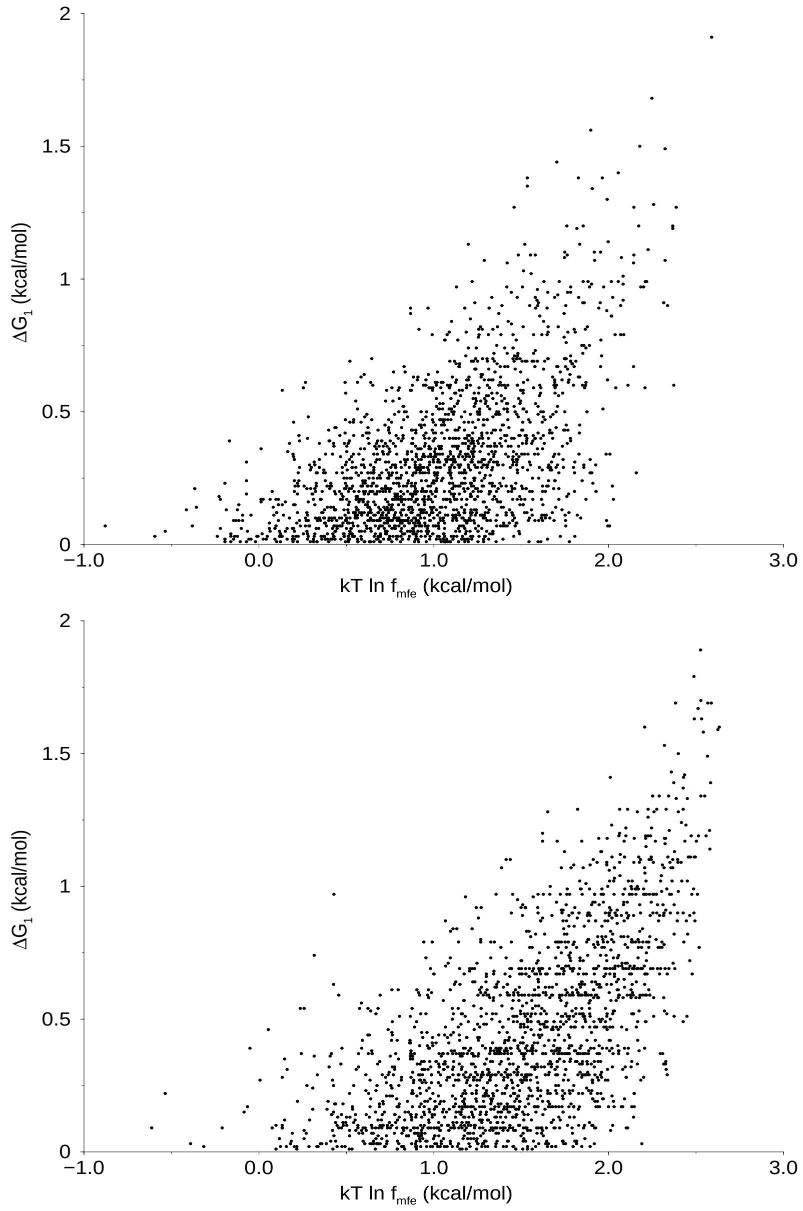


**Figure 17**: Distribution of $1^{st}$ gap energy ($\Delta G_1$) and frequency of *mfe* in the Boltzmann ensemble ($kT \ln f_{mfe}$) regarding the pools of unmodified (up) and modified (down) tRNA sequences.

are more grouped at smaller first gap energies and are shifted accordingly to smaller frequencies of *mfe*. Obviously the nonpairing bases have a strong influence on the distributions of $\Delta G_1$ and $f_{mfe}$. As mentioned before there is a correlation between $\Delta G_1$ and $f_{mfe}$. Hence the modified bases strength the "definition" of the structure.

In order to distinguish between "well defined" und "less well defined" structures, another two examples will be considered. In Figure 14 we showed two cross shaped structures similar to the tRNA cloverleaf structure. Although both structures show a very similar structure, the plots of $f_{mfe}$ distributions of inverse folded samples look completely different. This is shown in figure 18. As before there is a correlation between $f_{mfe}$ and $\Delta G_1$. Figure 19 shows the distributions of the $1^{st}$ gapenergies of both structures. The distributions pertaining the B structure are broader than those of the A structure. Again the distribution of the B structure is shifted to higher values of first gap energies. In order to confirm our assumption of a correlation between $f_{mfe}$ and $\Delta G_1$, figure 20 shows $\Delta G_1$ plotted against $kT \ln f_{mfe}$ regarding the pools of inverse folded sequences of both structures. Again the data points belonging to the pool of structure A sequences are grouped at smaller first gap energies and accordingly at smaller frequencies of *mfe* in contrast to the pool of structure B. Hence structure B shows the (expected) better defined structure than structure A. Although both structures show a very similar structure, the plots of $f_{mfe}$ and $\Delta G_1$ distributions of inverse folded samples look completely different as shown in figures 18, 19 and 20. Given the similarity of the two structures A and B (Fig. 14) the difference in the $f_{mfe}$ distribution is rather surprising. It is explained as follows. Both pools of first suboptimal structures have some common structural features. In most cases the first suboptimal structure lacked the first base pair of the multi-loop stem, or contained an additional base pair in the other stacks. The occurence of a completely different structure is rather rare. For the first suboptimal stacks of structures A and B this means that one of the hairpin loops often shrinks to size 4 and 3, respectively. In the currently used energy model the unpaired bases adjacent to a stack contribute some stabilizing mismatch energy, which is not the case, if the loopsize is 3. This effect is shown by generating pools of inverse folded sequences with hairpin structures containing loops of size 6 and 5. In Figure 21 these two hairpin

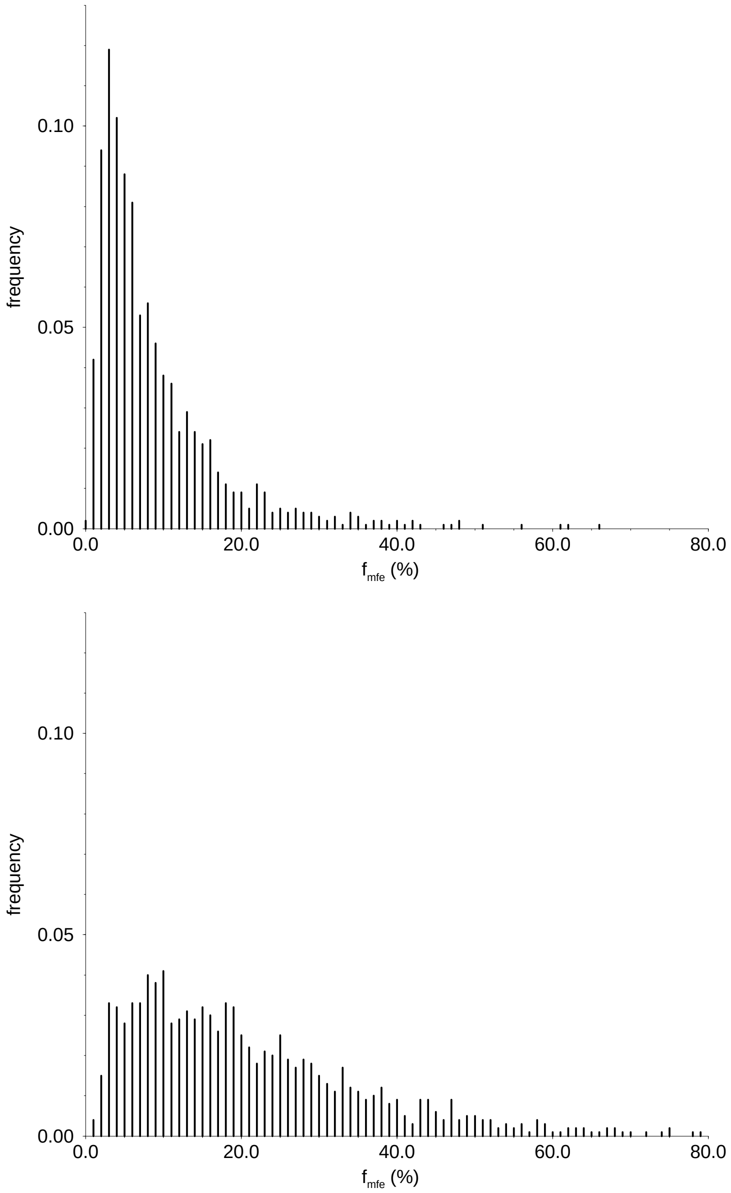


**Figure 18**: Distributions of $f_{mfe}$ for pools of 2000 inverse folded sequences with structure A (up) and B (down).

structures are shown. The influence of dangling-end contributions is larger than the mismatch energies. Hence, it is neccessary to exclude dangling-end

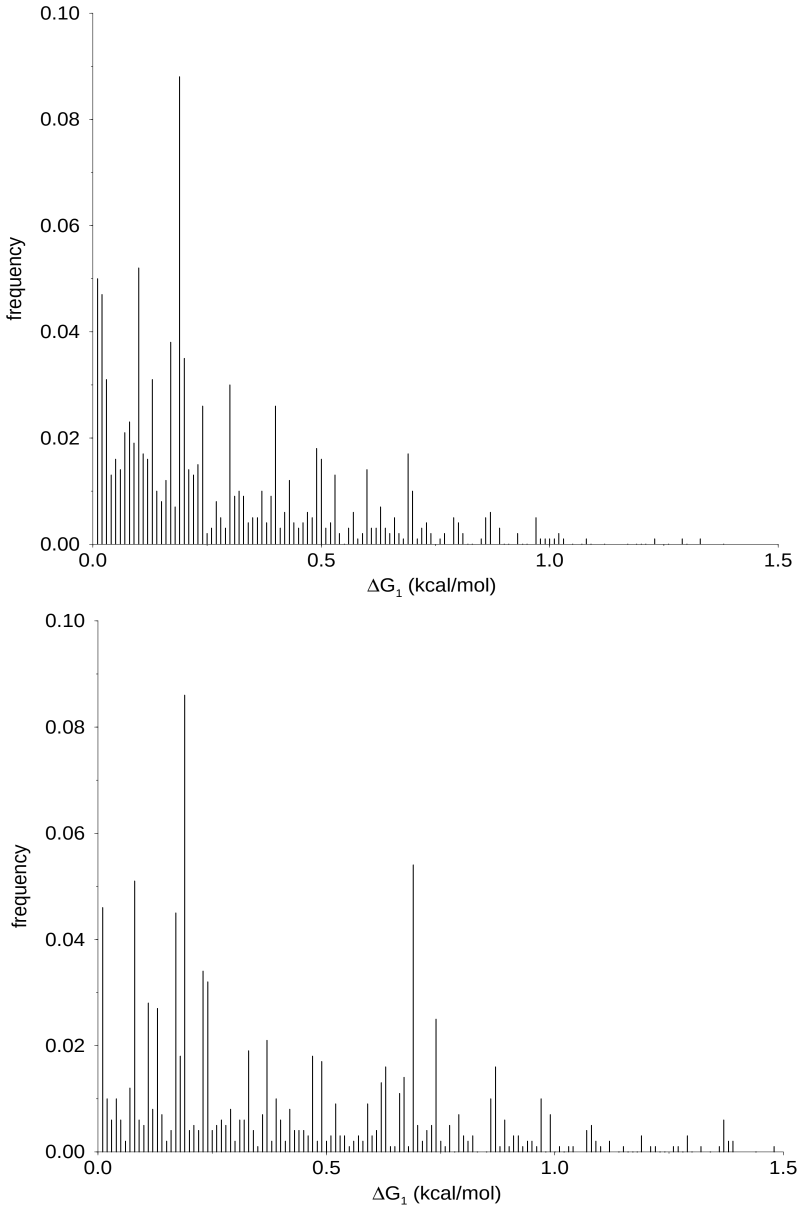


**Figure 19**: Comparison of the distributions of $1^{st}$ gap energies ($\Delta G_1$) regarding pools of crosshaped structures A (up) and B (down).

contributions. Figure 22 confirms our expectation. There exists a difference between the $f_{mfe}$ distribution of these two structures. The $f_{mfe}$ distribution

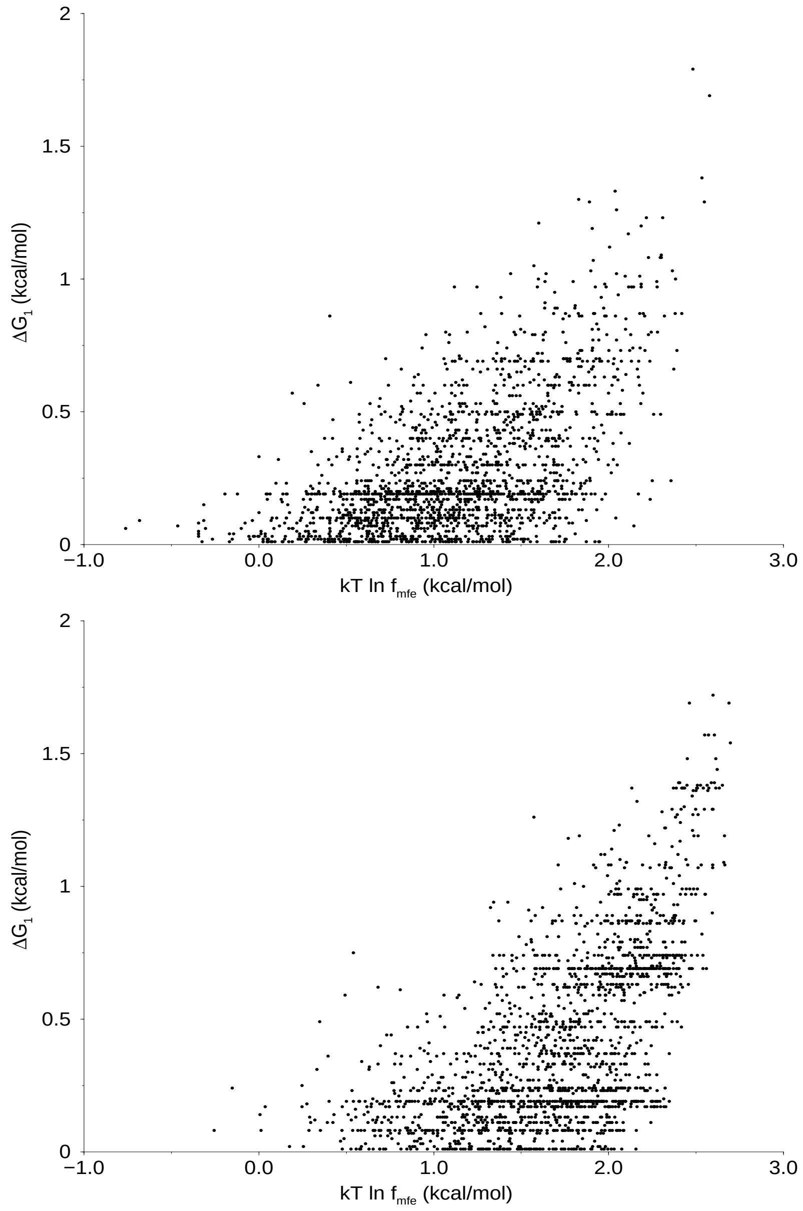


**Figure 20**: Distribution of $1^{st}$ gap energy ($\Delta G_1$) and frequency of *mfe* in the Boltzmann ensemble ($kT \ln f_{mfe}$) regarding the pools of crosshaped structure A (up) and B (down).

of the A'-sample (loopsize 5) is shifted to higher values than the one of the B'-sample (loopsize 6). Obviously the contributions of mismatch energies have

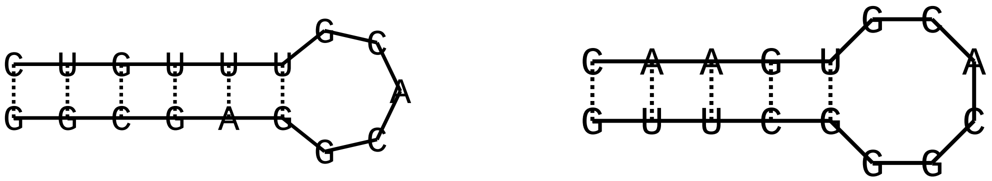


**Figure 21**: Hairpin structures A' and B'. This structures were used to generate pools of 2000 inverse folded sequences, each without contributions of dangling-end energies in order to show the influence of mismatch-energies on the $f_{mfe}$ plots.

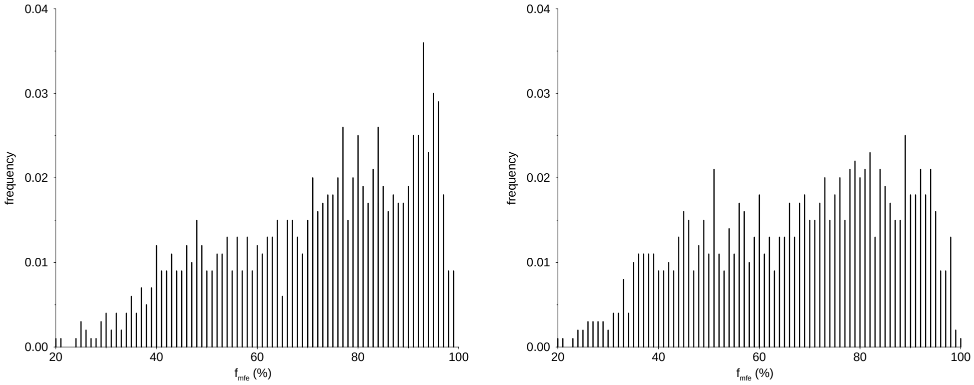


**Figure 22**: Distributions of $f_{mfe}$ for pools of 2000 inverse folded sequences with structure A' (left) and B' (right).

a significant influence on the definition of the structure A'. This observation explains the differences in the $f_{mfe}$ distributions of cross shaped structures A and B.

## 8.4 Structure Distances

### 8.4.1 Definitions

Here we consider the diversity of structures in the *LoDoS* by means of various measures of structural distance. One such distance is the so-called base pair distance. The base pair distance counts the number of basepairs, in which two structures are different. In figure 23 we show distributions of base pair distances between first and second gap structures respectively and the *mfe* structure regarding the pools of tRNA sequences. The influence of the modified bases is easily to see. As expected the distributions of the samples of modified sequences are shifted to lower values of base pair distance. We will consider the *mean base pair distance*, given as

$$< d_{bp} > = \sum_i d_{bp}(0,i)\; p_i, \tag{43}$$

where $d_{bp}$ denotes the base pair distance between the *mfe* structure (0) and the $i$th suboptimal structure. $p_i$ denotes the probability of the $i$th suboptimal structure in the ensemble

$$p_i = \frac{e^{-\frac{E_i}{kT}}}{Z}. \tag{44}$$

An energy distance is defined by the *mean gap energy*

$$< d_G > = \sum_i (E_i \; - \; E_{mfe})\; p_i. \tag{45}$$

### 8.4.2 Results

All structures and their energies of modified and unmodified natural tRNA sequences were calculated within an interval of 10 $kT$. As ground state of the tRNA sequences the well known cloverleaf structure was assumed. In the considered energy model almost all modified tRNA sequences have the cloverleaf as *mfe* structure. On contrast the unmodified tRNA sequences often have a different structure. We take, therefore, as our reference state the lowest lying suboptimal state providing the cloverleaf structure. Additionally a pool

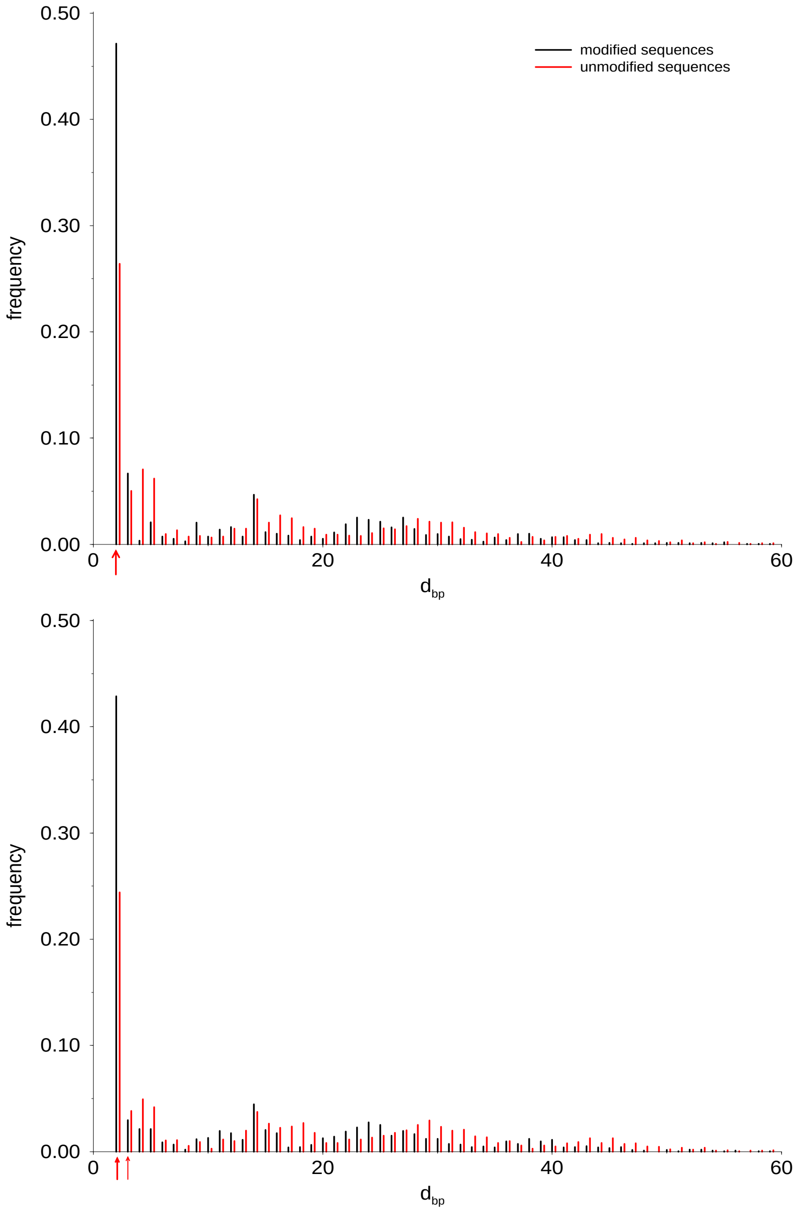


**Figure 23**: Distributions of base pair distances, $d_{bp}$, between $1^{st}$ gap structures (up) and $2^{nd}$ gap structures (down) respectively and the *mfe* structure regarding the pools of tRNA sequences. Arrows indicate the position of the 6 natural tRNA sequences.

of randomly modified tRNA sequences was generated: Firstly the modified bases of the natural tRNA sequences were translated into the corresponding unmodified ones (see Appendix A). Next the same number of modified bases (non-pairing) was inserted at random positions, such that the cloverleaf structure was retained. Figure 24 shows the results. Note, that $< d_G >$ and $< d_{bp} >$ were calculated within 10 $kT$ and are therefore only good approximations. Again the influence of nonbonding modified bases is quite strong. While the data points of the unmodified sequences seem to be spread widely, those of the modified bases are largely restricted to a specific area corresponding to higher mean gap energies and smaller base pair distances. These two features are indicative for the well-definedness of a structure as we suggested

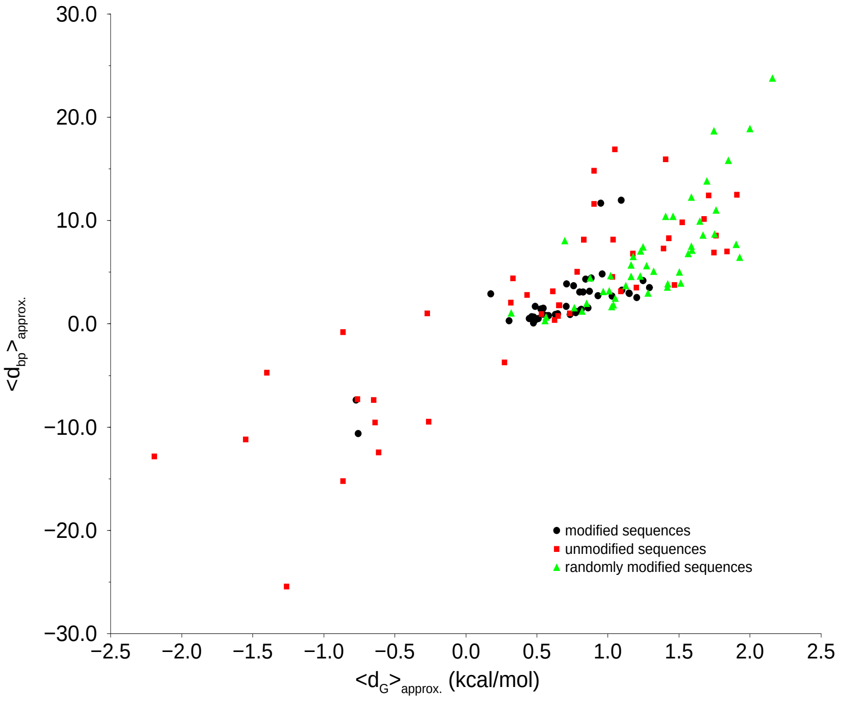


**Figure 24**: Plot of the mean gap of energy $< d_G >$ vs. mean distance of basepairs $< d_{bp} >$ regarding the pools of modified, unmodified and randomly modified *E.coli* tRNA sequences.

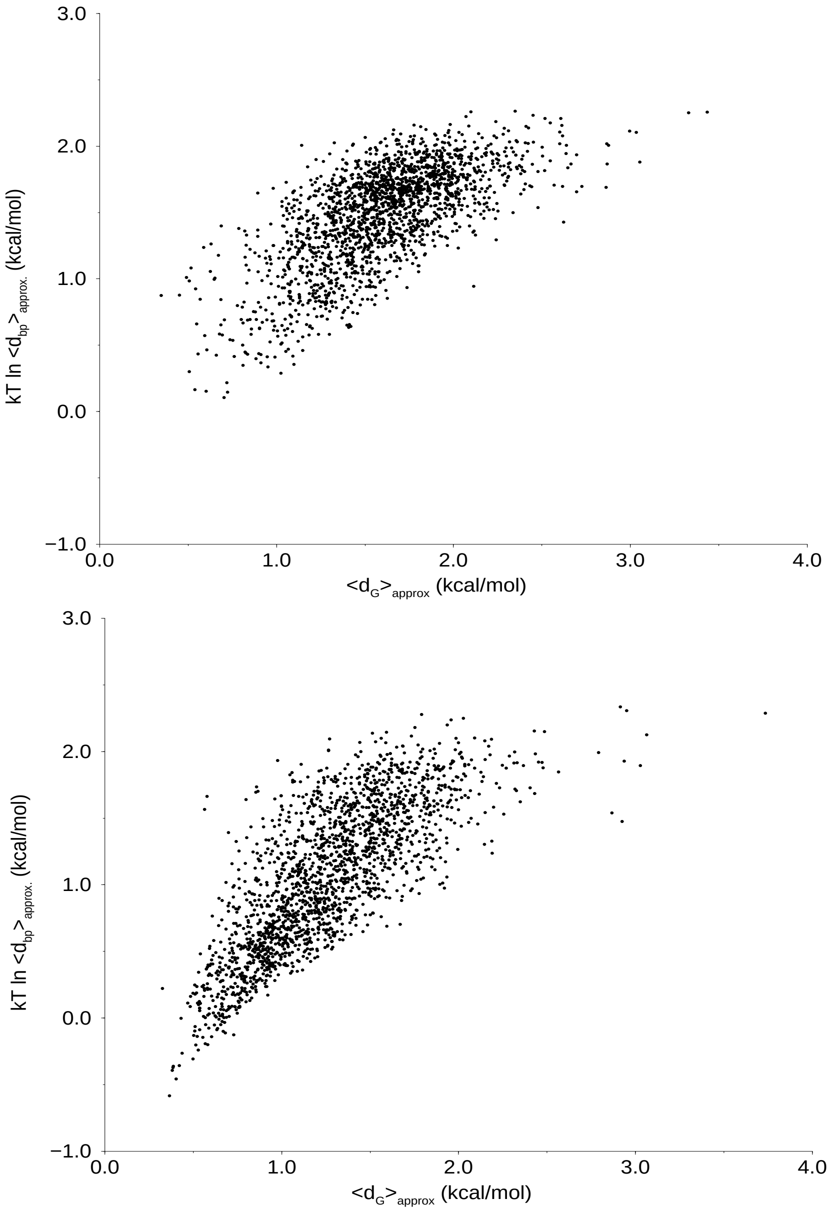


**Figure 25**: Plot of the mean gap energy $< d_G >$ vs. mean base pair distance $kT\ ln\ < d_{bp} >$ regarding the pools of inverse folded sequences of crosshaped structures A (up) and B (down).

before. Some data points relating to unmodified sequences lie in the negative quadrant. That means that the *mfe* structure of these sequences is not the cloverleaf structure. For these one of the suboptimal structures was chosen to be the reference state of our calculations (see above).

The data points of randomly modified sequences are in the neighbourhood of modified sequences. However, they are shifted to higher mean gap energies and mean base pair distances. Hence, the definition of the tRNA structure is dependent on the position of modification.

Pools of inverse folded sequences of the cross shaped structures A and B (Figure 14) were analyzed in a similar fashion. In figure 25 the plots of the pools are compared. Note, that the mean base pair distance, $< d_{bp} >$, is given as its natural logarithm in multiples of $kT$ in order to show a better correlation to $< d_G >$. Again, compared to the A-sample the points belonging to the better defined structure B are more grouped at smaller base pair distances than computed mean gap energies. Similar results are obtained with samples of inverse folded modified and unmodified sequences of tRNA structures. We conclude that $< d_{bp} >$ is a more reliable indicator of well-defindness than $< d_G >$.

## 8.5 Partition Function Plots

The partition function is defined as

$$Z = \sum_i e^{-\frac{E_i}{kT}} \tag{46}$$

with $E_i$ being the energy of the $i$th suboptimal structure in the Boltzmann ensemble. An approximation to $Z$ will be calculated with *LoDoS* ensembles of various sizes, $\Delta G' = -kT \ln Z'$. As a value of reference $\Delta G$ is also calculated by McCaskill's partition function algorithm [26]. This algorithm calculates $Z$. With the possibility to calculate all structures and their related energies strictly within a given energy band, $Z'$, *i.e.* $\Delta G'$, will be obtained by suming over the Boltzmann factors of all structures in the *LoDoS*. The sumation over the Boltzmann factors stops, when 99,9 percent of $\Delta G$ $(Z)$ are obtained. Figure 26 shows the course of such calculations. Interestingly the number of needed structures and related energies increases, the more "undefined" the *mfe* structure of a given sequence is.

The base pair probability is defined as the probability of a base pair $(i, j)$ in a Boltzmann weighted ensemble of structures following equation 40. The calculation of the base pairing probabilities $P_{i,j}$ leads to the construction of a

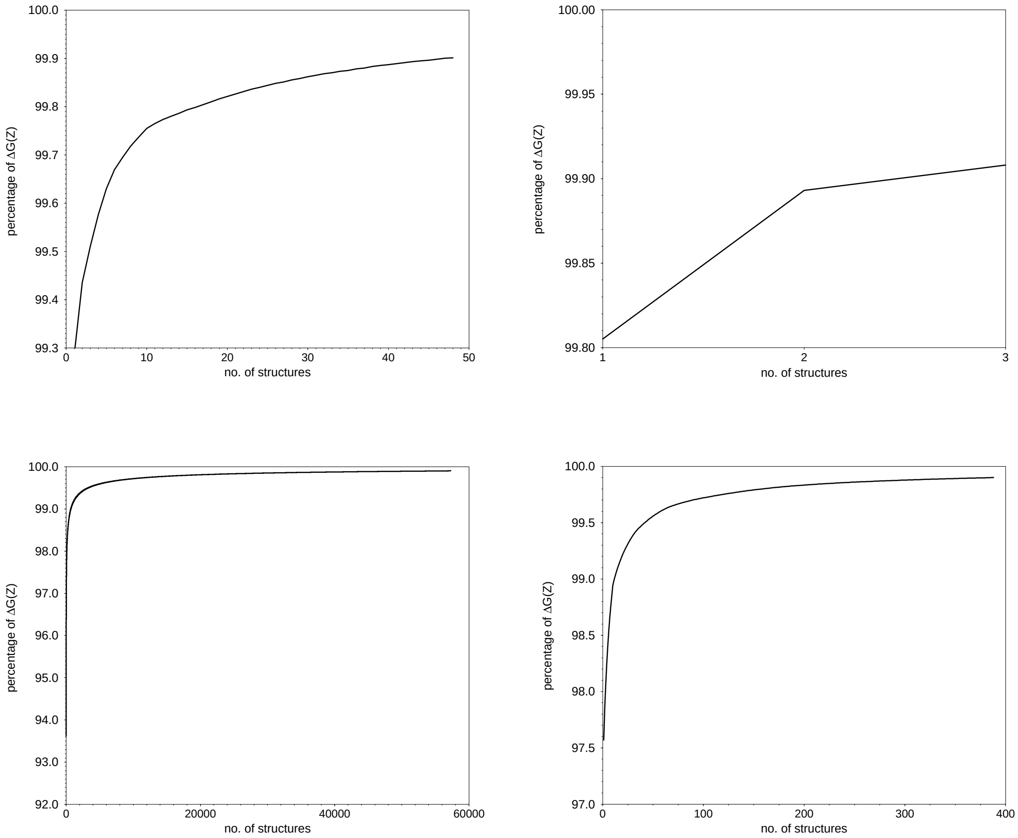


**Figure 26**: Course of the approximation of $\Delta G'$ $(Z')$ to 99,9 percent of $\Delta G$ $(Z)$ related to the number of required states of energy. As contrasting examples the modified and unmodified tRNA-sequences RI1662 and RW1660 were chosen.
top left: RI1662/unmodified GGCCCCUUAGCUCAGUGGUUAGAGCAGGCGACUNAUAAUCGCUUGGUCGCUGGUUCAAGUCCAGCAGGGGCCACCA
$\Delta G(Z)$ = -33,366 kcal/mol, number of structures: 48
top right: RI1662/modified GGCCCCUNAGCUCAGUGGNNAGAGCAGGCGACUNAUNAUCGCUUGNNCGCUGGNUCAAGUCCAGCAGGGGCCACCA
$\Delta G(Z)$ = -31,482 kcal/mol, number of structures: 3
bottom left: RW1660/unmodified AGGGGCGUAGUUCAAUUGGUAGAGCACCGGUCUCCAAAACCGGGUGUUGGGAGUUCGAGUCUCUCCGCCCCUGCCA
$\Delta G(Z)$ = -25,077 kcal/mol, number of structures: 57285
bottom right: RW1660/modified AGGGGCGNAGUUCAANNGGNAGAGCACCGGUCUCCANAACCGGGUNUUGGGAGNUCGAGUCUCUCCGCCCCUGCCA
$\Delta G(Z)$ = -23,657 kcal/mol, number of structures: 388

base pairing matrix. The different base pair probabilities $P_{i,j}$ obtained from the *LoDoS* are compared with those from the full partition function $Z$. This is shown in figures 27 for the tRNA sequences used in Figure 26. The dots corresponding to base pairs occuring with a probability of less than $10^{-5}$ are suppressed. The plot is divided into two triangles. While the upper left triangle contains the base pairing probability matrix obtained with the full $Z$ the right

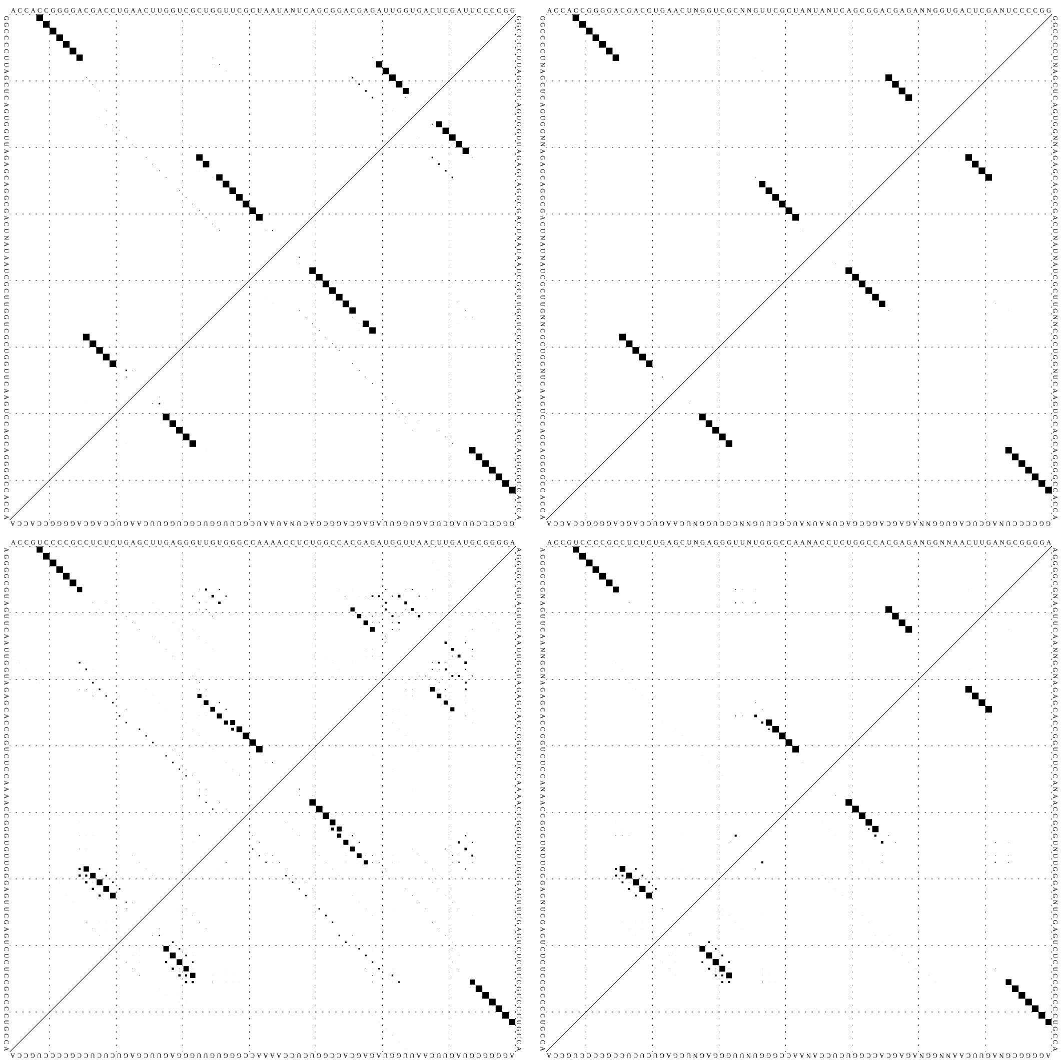


**Figure 27**: Dot Plots of selected tRNA sequences. Upper left triangle contains base pairing probabilities ($P_{i,j}$) obtained with the full $Z$. Lower right triangle displays the base pairing probabilities yielded with $Z'$.top left: RI1662/unmodified top right: RI1662/modified bottom left: RW1660/unmodified bottom right: RW1660/modified

triangle displays the matrix obtained with the $Z'$ from the *LoDoS* up to 10 $kT$ above the *mfe*.

## 8.6 Coarse Grained Approaches

Within a certain energy width all suboptimal structures and related energies of a tRNA sequence were calculated. The energies were discretized in increments of $0,1$ and binned accordingly. For each energy interval several features were investigated. For the case of the modified and unmodified tRNA sequence *RK1660* the *LoDoS* of the unmodified sequence was calculated up to 15 $kT$ while that of the modified sequence had to be calculated up to 30 $kT$. For each interval the arithmetic mean of the base pair distance of the structures contained in it to the *mfe* structure was calculated. In addition every suboptimal structure contained in the interval was "coarse grained". A coarse grained structure is derived from the known secondary structure by ignoring size and length of loops and stacks. The number of different coarse grained structures was counted. The results are shown in figure 28. The diversity, the number of different coarse grained structures, increases faster in the unmodified case. Strikingly, the arithmetic mean base pair distance in the unmodified case is much higher than that of the modified case. The same holds for any other tRNA sequence (data not shown).

## 8.7 Neutrality

Neutrality is defined as the percentage of neutral mutations among all 1-error-mutants of a given sequence. In Figure 29 the mean base pair distances $< d_{bp} >$, frequencies $f_{mfe}$ and mean gap energies $< d_G >$ are plotted against the neutrality. As sample the pools of inverse folded modified and unmodified sequences of the tRNA structure are used. As we saw in previous subsections, the modification of some bases increases the mean basepairdistance $< d_{bp} >$, the frequency $f_{mfe}$ and the mean gap energy $< d_G >$. Modified sequences are shifted to higher amounts of neutrality.

The same calculations were performed using the pools of inverse folded sequences of the crosshaped structures A and B. The results are shown in figures 30 and 31. In agreement with the tRNA case the data points of structure B, the well defined structure, are shifted to higher values of neutrality, to higher values of frequency $f_{mfe}$, to higher values of mean gap energy $< d_G >$ and to lower values of mean base pair distance $< d_{bp} >$. It is interesting to note, that using the natural logarithms of the mean base pair distances $< d_{bp} >$, frequencies $f_{mfe}$ and mean gap energies $< d_G >$ in the diagrams yield a good correlation to the neutrality.

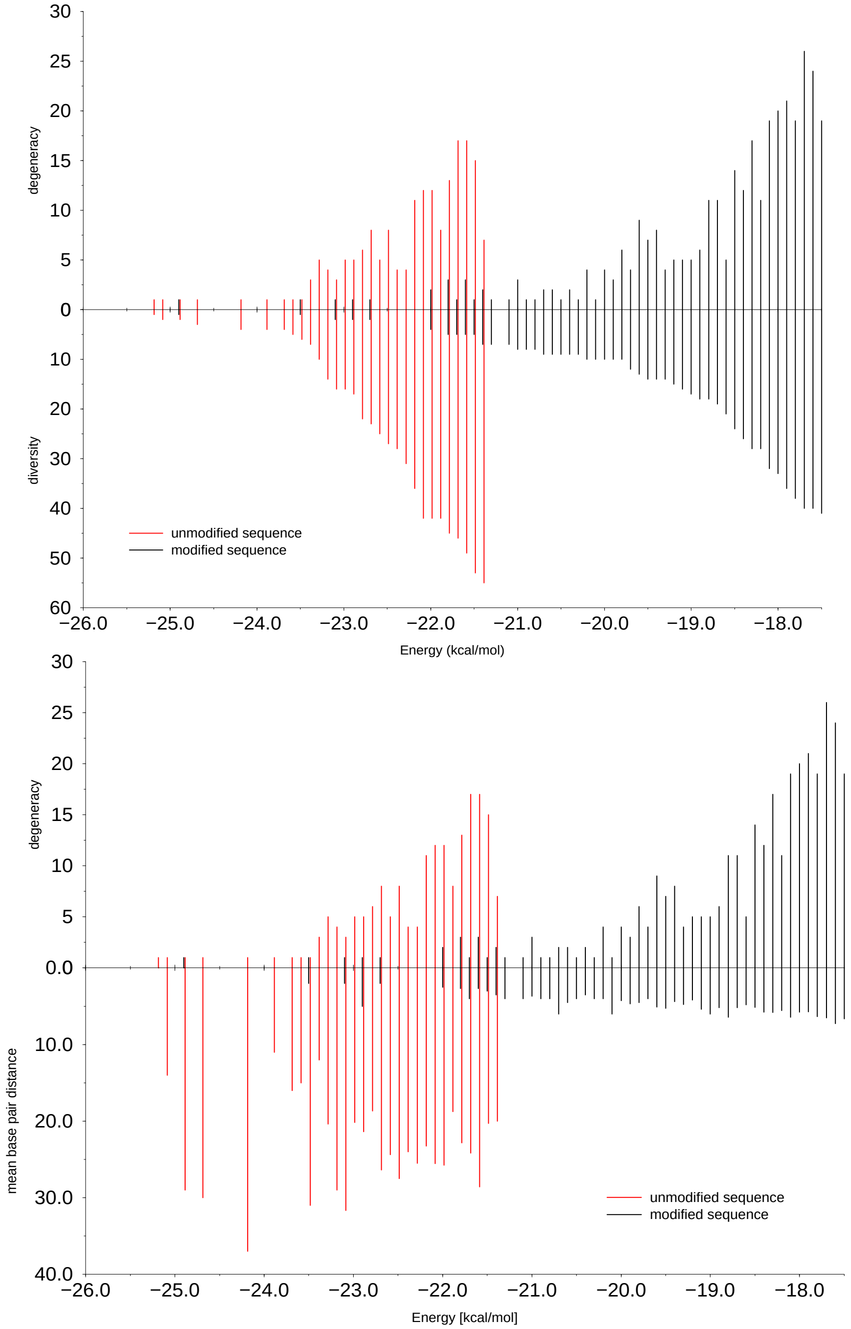


**Figure 28**: Comparison of *LoDoS* to diversity, *i.e.* number of different coarse grained structures, and arithmetic mean of the base pair distance regarding the modified and unmodified sequence of tRNA *RK1660*. *LoDoS* of the unmodified sequence was calculated up to 15 $kT$ while the *LoDoS* of the modified sequence had to be calculated up to 30 $kT$.

`RW1660/modified:  GGGUCGUUAGCUCAGNNGGNAGAGCAGUUGACUNUUNAUCAAUUGNNCGCAGGNUCGAAUCCUGCACGACCCACCA`
`RW1660/unmodified:  GGGUCGUUAGCUCAGUUGGUAGAGCAGUUGACUUUUAAUCAAUUGGUCGCAGGUUCGAAUCCUGCACGACCCACCA`

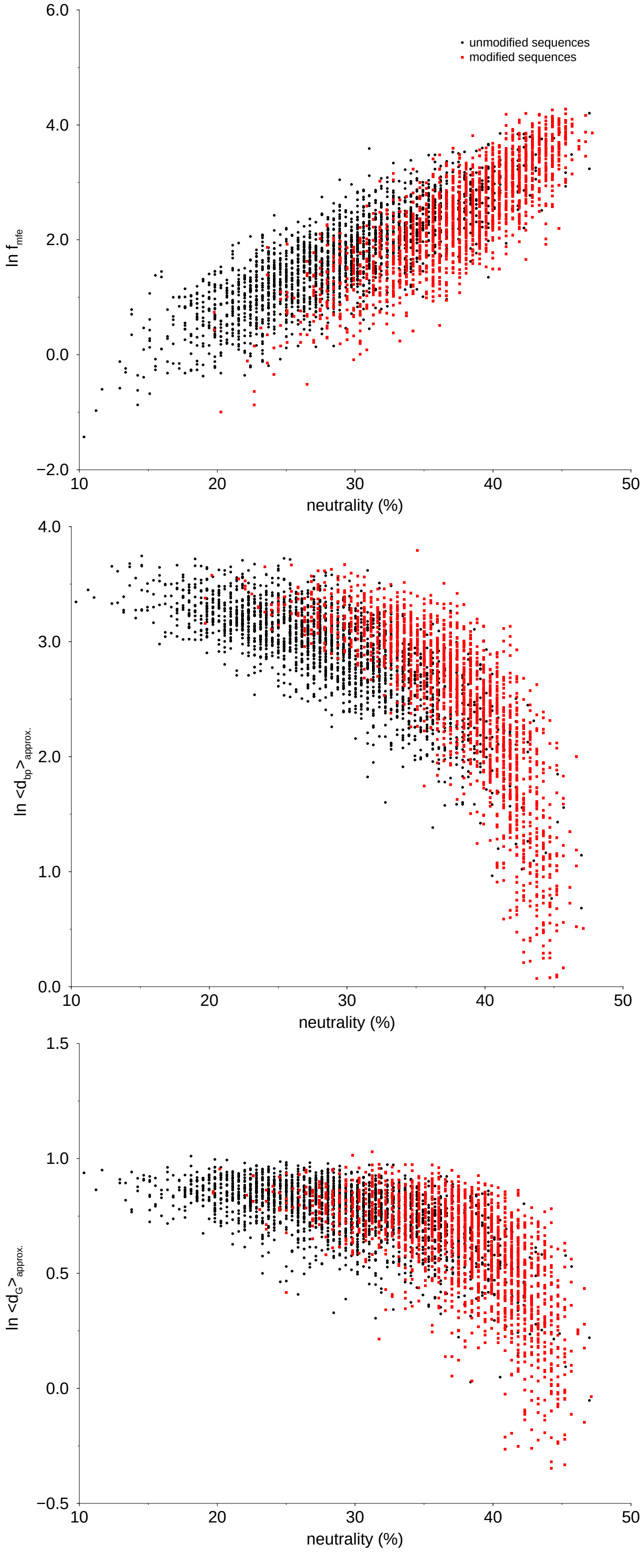


**Figure 29**: Distributions of the frequency $f_{mfe}$, mean base pair distance $< d_{bp} >$ and mean gap energy $< d_G >$ against neutrality regarding the pools of inverse folded modified and unmodified tRNA sequences.

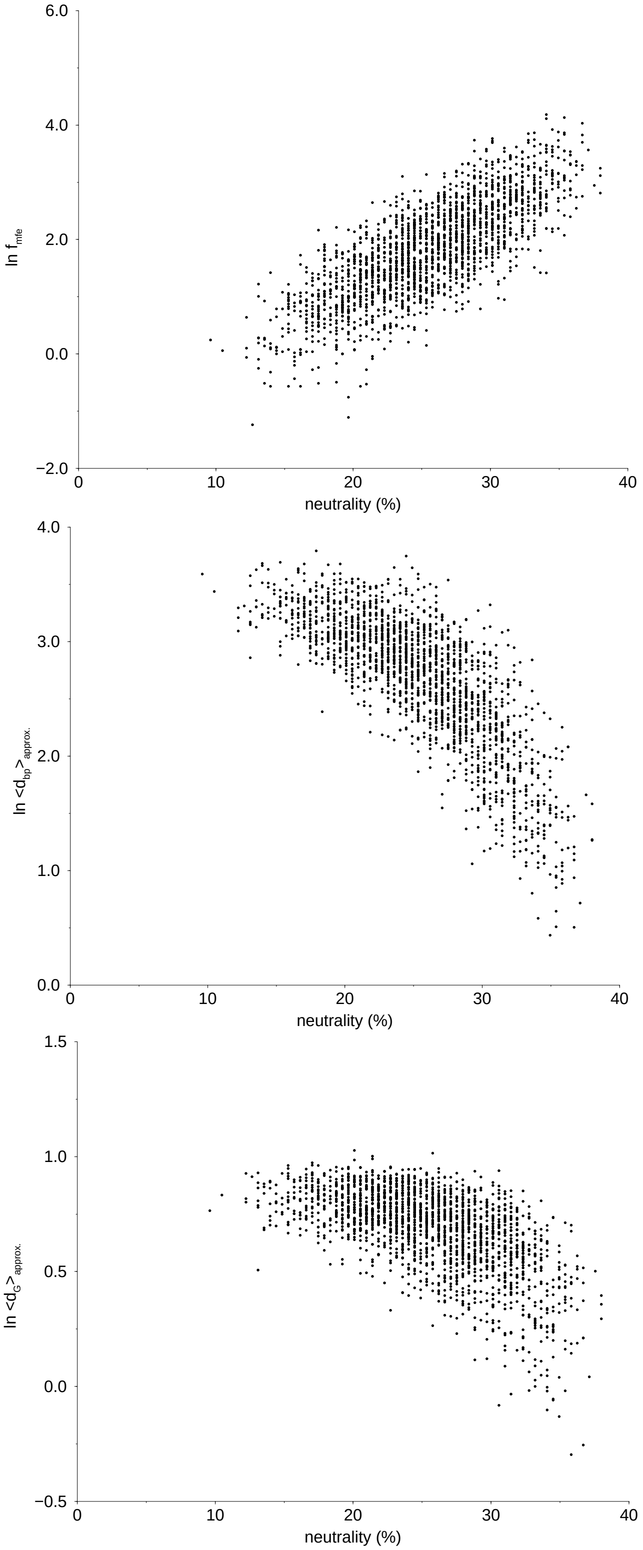


**Figure 30**: Distributions of the frequency $f_{mfe}$, mean base pair distance $< d_{bp} >$ and mean gap energy $< d_G >$ against neutrality regarding the pools of inverse folded sequences of crosshaped structure A.

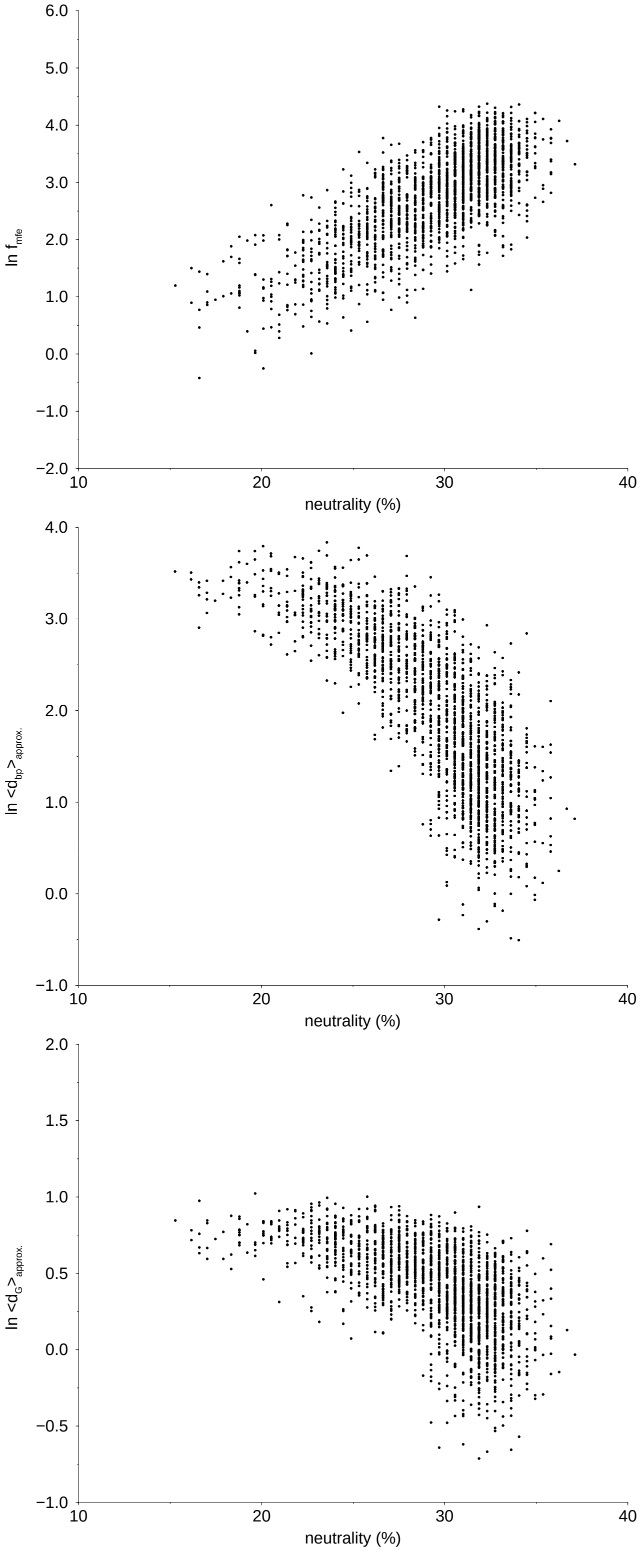


**Figure 31**: Distributions of the frequency $f_{mfe}$, mean base pair distance $< d_{bp} >$ and mean gap energy $< d_G >$ against neutrality regarding the pools of inverse folded sequences of crosshaped structure B.

# 9 Conclusions and Outlook

RNA structures play a significant role in a great variety of different problems. Secondary structures provide a convenient form of coarse graining, and their study yields information useful in the prediction of the full 3D structures and in the interpretation of the biochemical function of the molecules. Furthermore, secondary structures are discrete and therefore well suited for computational methods.

To understand the biological role of an RNA molecule, it is not necessary to know the complete density of states. For many purposes it is sufficient to know the *LoDoS* accounting for all states within a certain energy range above the minimum free energy. There may exist several suboptimal structures providing the sam or a different biological function.

The representation of RNA secondary structures as vertex-labeled, planar graphs are discussed. There exist already combinatorial and dynamic programming approaches to find all secondary structures within this window of the density of states. However, these algorithms feature either a high amount of approximation or simply do not find all secondary structures within the considered range of energy. Such algorithms derived previously were compiled and presented.

In this work we introduced an algorithm capable to calculate all secondary structures of a RNA sequence within a desired energy range above the minimum free energy, which implements the energy parameter set used within the Vienna RNA Package. Finding near-optimal paths between a specified origin and destination in an acyclic network was firstly applied to the maximum matching problem intended as a kind of test of the applicability of that concept. Afterwards this idea was applied to the energy folding problem.

With the possibility to calculate all secondary structures of a RNA sequence within a certain energy range it is possible to gain new insights in the well-definedness of a RNA structure. This investigations lead to the discovery of the size of the gap between the energies of the minimum free energy structure and the first suboptimal structure as new measure of well-definedness in comparison to the frequency of the minimum free energy within the partition function. With this new tool we were also able to investigate the structure stabilizing role of modified (*i.e.* non-pairing) bases. As example we used the natural tRNA sequences of *E.coli* showing higher first gaps than the unmodified tRNA sequences.

An investigation of structural distances of suboptimal structures to the minimum free energy structure of well defined structures showed also lower base pair distances and higher gap energies in contrast to less well defined structures.

The partition function of an RNA sequence without knowing all free energies can be calculated in a good approximation using the lower states. It appeared, that due to the size of the first gaps 99,9 percent of the partition function of modified tRNA sequences, *i.e.* well defined structures, can be calculated with much less suboptimal structures than unmodified ones.

The ability to calculate the *LoDoS* within an desired energy range enabled us to characterise the different states of natural RNA sequences of *E.coli.* The results show that original tRNA sequences have less states in the vicinity of the ground state and the energy gap is usually larger. Also the mean base pair distances and the number of coarse grained structures within the states are much smaller than within the states of the unmodified tRNA sequences.

An insight into the relation of neutrality of RNA sequences and well-definedness of the related structures is given. It turned out that the features of well-definedness are related to a higher amount of neutrality.

A problem arising with long sequences and high ranges of energy is the exponential growth of memory requirements of the algorithm. As a future consideration the programming of a more efficient memory management of the various stacks must be taken into account.

With the possibility to calculate all acceptable suboptimal structures within a desired energy range an insight into transition states of secondary structures can be probably obtained. In combination with kinetic folding algorithms available in our research group interesting perspectives on folding dynamics of RNA secondary structures must be receivable. Mentioned above we found a naturally occuring example of increasing well-definedness of a structure by inserting nonpairing bases in tRNA sequences. With this algorithm we have a tool for showing the effects of modified bases contained in further natural sequences. The good correlation between stability against mutation and thermodynamic stability should be investigated in that sense to point out to which extent thermodynamic stability of a RNA sequence and RNA structure implies stability against mutation and vice versa. This feature can be strengthened by searching for natural occuring examples either stabilized against mutation or thermodynamically stabilized.

# A EMBL tRNA Database

All tRNA sequences are from the compilation of Steegborn [41], which can be obtained via anonymous ftp from EMBL Heidelberg, `ftp.embl-heidelberg.de`, in directory `/pub/databases/trna/`.

## Abbreviation of Modified Bases

The one-letter code and the abbreviation for all modified bases in the tRNA database:

| | | |
|---|---|---|
| U | (U) | uridine |
| C | (C) | cytidine |
| A | (A) | adenosine |
| G | (G) | guanosine |
| T | (T) | thymine (for sequences of tRNA genes only) |

| | | |
|---|---|---|
| H | (?A) | unknown modified adenosine |
| " | (m1A) | 1-methyladenosine |
| / | (m2A) | 2-methyladenosine |
| + | (i6A) | N6-isopentenyladenosine |
| * | (ms2i6A) | 2-methylthio-N6-isopentenyladenosine |
| = | (m6A) | N6-methyladenosine |
| 6 | (t6A) | N6-threonylcarbamoyladenosine |
| E | (m6t6A) | N6-methyl-N6-threonylcarbamoyladenosine |
| [ | (ms2t6A) | 2-methylthio-N6-threonylcarbamoyladenosine |
| : | (Am) | 2’-O-methyladenosine |
| I | (I) | inosine |
| O | (m1I) | 1-methylinosine |
| ^ | (Ar(p)) | 2’-O-ribosyladenosine (phosphat) |
| ‘ | (io6A) | N6-(cis-hydroxyisopentenyl)adenosine |

| | | |
|---|---|---|
| < | (?C) | unknown modified cytidine |
| % | (s2C) | 2-thiocytidine |
| B | (Cm) | 2’-O-methylcytidine |
| M | (ac4C) | N4-acetylcytidine |
| ? | (m5C) | 5-methylcytidine |
| ’ | (m3C) | 3-methylcytidine |
| > | (f5C) | 5-formylcytidin |

| | | |
|---|---|---|
| ; | (G) | unknown modified guanosine |
| K | (m1G) | 1-methylguanosine |
| L | (m2G) | N2-methylguanosine |
| # | (Gm) | 2’-O-methylguanosine |
| R | (m22G) | N2,N2-dimethylguanosine |
| \| | (m22Gm) | N2,N2,2’-O-trimethylguanosine |
| 7 | (m7G) | 7-methylguanosine |
| ( | (fa7d7G) | archaeosine |
| Q | (Q) | queuosine |
| 8 | (manQ) | mannosyl-queuosine |
| 9 | (galQ) | galactosyl-queuosine |
| Y | (yW) | wybutosine |
| W | (o2yW) | peroxywybutosine |

| | | |
|---|---|---|
| N | (?U) | unknown modified uridine |
| { | (mnm5U) | 5-methylaminomethyluridine |
| 2 | (s2U) | 2-thiouridine |
| J | (Um) | 2’-O-methyluridine |
| 4 | (s4U) | 4-thiouridine |
| & | (ncm5U) | 5-carbamoylmethyluridine |
| 1 | (mcm5U) | 5-methoxycarbonylmethyluridine |
| S | (mnm5s2U) | 5-methylaminomethyl-2-thiouridine |
| 3 | (mcm5s2U) | 5-methoxycarbonylmethyl-2-thiouridine |
| V | (cmo5U) | uridine 5-oxyacetic acid |

| | | |
|---|---|---|
| 5 | (mo5U) | 5-methoxyuridine |
| ! | (cmnm5U) | 5-carboxymethylaminomethyluridine |
| $ | (cmnm5s2U) | 5-carboxymethylaminomethyl-2-thiouridine |
| X | (acp3U) | 3-(3-amino-3-carboxypropyl)uridine |
| , | (mchm5U) | 5-(carboxyhydroxymethyl)uridinemethyl ester |
| ) | (cmnm5Um) | 5-carboxymethylaminomethyl-2'-O-methyluridine |
| ~ | (ncm5Um) | 5-carbamoylmethyl-2'-O-methyluridine |
| D | (D) | dihydrouridine |
| P | (psi) | pseudouridine |
| ] | (m1psi) | 1-methylpseudouridine |
| Z | (psi m) | 2'-O-methylpseudouridine |
| T | (m5U) | ribosylthymine |
| F | (m5s2U) | 5-methyl-2-thiouridine |
| \ | (m5Um) | 5, 2'-O-dimethyluridine |

Bases are translated as suggested by Higgs [15]: Modified bases in pairing regions were translated to their non-modified analogues; Bases exclusively found in loop regions were treated as non-bonding bases.

## E. Coli tRNA Sequences

All *E. Coli* tRNA sequences from the EMBL tRNA Database used in this work are given. The sequence number codes as follows: First letter is D or R for DNA or RNA respectively. Second letter gives the one-letter symbol of the amino acid. In addition to the commonly used one-letter amino acid code, Z means seleno cysteine and X stands for initiator tRNA. The four digit number codes for organism and isoacceptor (see `manual.txt` in the database).

```
Sequence      Anti-  Organism  Kingdom
Number        Codon

RA1660         GGC     E.COLI  EUBACT
GGGGCUANAGCUCAGCDGGGAGAGCGCUUGCAUGGCAUGCAAGAG7UCAGCGGTPCGAUCCCGCUUAGCUCCACCA
RA1661         VGC     E.COLI  EUBACT
```

```
GGGGGCA4AGCUCAGCDGGGAGAGCGCCUGCUUVGCACGCAGGAG7UCUGCGGTPCGAUCCCGCGCGCUCCCACCA
RA1662        VGC     E.COLI  EUBACT
GGGGCUAUAGCUCAGCDGGGAGAGCGCCUGCUUVGCACGCAGGAG7UCUGCGGTPCGAUCCCGCAUAGCUCCACCA
RC1660        GCA     E.COLI  EUBACT
GGCGCGU4AACAAAGCGGDDAUGUAGCGGAPUGCA*APCCGUCUAGUCCGGTPCGACUCCGGAACGCGCCUCCA
RD1660        QUC     E.COLI  EUBACT
GGAGCGG4AGUUCAGDCGGDDAGAAUACCUGCCUQUC/CGCAGGGG7UCGCGGGTPCGAGUCCCGPCCGUUCCGCCA
RE1660        SUC     E.COLI  EUBACT
GUCCCCUUCGUCPAGAGGCCCAGGACACCGCCCUSUC/CGGCGGUAACAGGGGTPCGAAUCCCCUGGGGGACGCCA
RE1661        SUC     E.COLI  EUBACT
GUCCCCUUCGUCPAGAGGCCCAGGACACCGCCCUSUC/CGGCGGUAACAGGGGTPCGAAUCCCCUAGGGGACGCCA
RE1662        SUC     E.COLI  EUBACT
GUCCCCUUCGUCPAGAGGCCAGGACACCGCCCUSUC/CGGCGGUAACAGGGGTPCGAAUCCCCUAGGGGACGCCA
RF1660        GAA     E.COLI  EUBACT
GCCCGGA4AGCUCAGDCGGDAGAGCAGGGGAPUGAA*APCCCCGU7XCCUUGGTPCGAUUCCGAGUCCGGGCACCA
RG1660        CCC     E.COLI  EUBACT
GCGGGCG4AGUUCAAUGGDAGAACGAGAGCUUCCCAAGCUCUAUACGAGGGTPCGAUUCCCUUCGCCCGCUCCA
RG1661        GCC     E.COLI  EUBACT
GCGGGAAUAGCUCAGDDGGDAGAGCACGACCUUGCCAAGGUCGGG7UCGCGAGTPCGAGUCUCGUUUCCCGCUCCA
RG1662        NCC     E.COLI  EUBACT
GCGGGCAUCGUAUAAUGGCUAUUACCUCAGCCUNCCAAGCUGAUGAUGCGGGTPCGAUUCCCGCUGCCCGCUCCA
RH1660        QUG     E.COLI  EUBACT
GGUGGCUA4AGCUCAGDDGGDAGAGCCCUGGAUUQUG/PPCCAGUU7UCGUGGGTPCGAAUCCCAUUAGCCACCCCA
RI1660        GAU     E.COLI  EUBACT
AGGCUUGUAGCUCAGGDGGDDAGAGCGCACCCCUGAU6AGGGUGAG7XCGGUGGTPCAAGUCCACPCAGGCCUACCA
RI1661        GAU     E.COLI  EUBACT
AGGCUUGUAGCUCAGGUGGDDAGAGCGCACCCCUGAU6AGGGUGAG7XCGGUGGTPCAAGUCCACPCAGGCCUACCA
RI1662        }AU     E.COLI  EUBACT
GGCCCCU4AGCUCAGU#GDDAGAGCAGGCGACU}AU6APCGCUUG7XCGCUGGTPCAAGUCCAGCAGGGGCCACCA
RK1660        SUU     E.COLI  EUBACT
GGGUCGUUAGCUCAGDDGGDAGAGCAGUUGACUSUU6APCAAUUG7XCGCAGGTPCGAAUCCUGCACGACCCACCA
RL1660       HAA     E.COLI  EUBACT
GCCCGGA4GGUGGAADC#GDAGACACAAGGGAPUHAA*APCCCUCGGCGUUCGCGCUGUGCGGGTPCAAGUCCCGCUCCGGGUACCA
RL1661       CAG     E.COLI  EUBACT
GCGAAGGUGGCGGAADD#GDAGACGCGCUAGCUUCAG;PGPUAGUGUCCUUACGGACGUGGGGGTPCAAGUCCCCCCCCUCGCACCA
RL1662       GAG     E.COLI  EUBACT
GCCGAGGUGGUGGAADD#GDAGACACGCUACCUUGAG;PGGUAGUGCCCAAUAGGGCUUACGGGTPCAAGUCCCGUCCUCGGUACCA
RM1660        MAU     E.COLI  EUBACT
GGCUACG4AGCUCAGDD#GDDAGAGCACAUCACUMAU6APGAUGGG7XCACAGGTPCGAAUCCCGUCGUAGCCACCA
RN1660        QUU     E.COLI  EUBACT
UCCUCUG4AGUUCAGDCGGDAGAACGGCGGACUQUU6APCCGUAU7UCACUGGTPCGAGUCCAGUCAGAGGAGCCA
RQ1660        CUG     E.COLI  EUBACT
UGGGGUA4CGCCAAGC#GDAAGGCACCGGAJUCUG/PPCCGGCAUUCCGAGGTPCGAAUCCUCGUACCCCAGCCA
RQ1661        NUG     E.COLI  EUBACT
UGGGGUA4CGCCAAGC#GDAAGGCACCGGUJUNUG/PACCGGCAUUCCCUGGTPCGAAUCCAGGUACCCCAGCCA
RR1660        ICG     E.COLI  EUBACT
GCAUCCG4AGCUCAGCDGGDAGAGUACUCGG%UICG/ACCGAGCG7XCGGAGGTPCGAAUCCUCCCGGAUGCACCA
RR1661        ICG     E.COLI  EUBACT
```

```
GCAUCCG4AGCUCAGCDGGADAGAGUACUCGGCUICG/ACCGAGCG7XCGGAGGTPCGAAUCCUCCCGGAUGCACCA
RR1662        {CU     E.COLI  EUBACT
GUCCUCUUAGUUAAAUGGADAUAACGAGCCC%U{CU6AGGGCUAAUUGCAGGTPCGAUUCCUGCAGGGGACACCA
RR1663        {CU     E.COLI  EUBACT
GCGCCCUUAGCUCAGUUGGAUAGAGCAACGAC%U{CU6AGPCGUGGGCCGCAGGTPCGAAUCCUGCAGGGCGCGCCA
RR1664        CCG     E.COLI  EUBACT
GCGCCCGUAGCUCAGCDGGADAGAGCGCUGCC%UCCGKAGGCAGAG7UCUCAGGTPCGAAUCCUGUCGGGCGCGCCA
RS1660       CGA     E.COLI  EUBACT
GGAGAGAUGCCGGAGC#GCDGAACGGACCGGUCUCGA*AACCGGAGUAGGGGCAACUCUACCGGGGGTPCAAAUCCCCCUCUCUCCGCCA
RS1661       GCU     E.COLI  EUBACT
GGUGAGG4GGCCGAGAGGCDGAAGGCGCUCCC%UGCU6AGGGAGUAUGCGGUCAAAAGCUGCAUCCGGGGTPCGAAUCCCCGCCUCACCGCCA
RS1662       GGA     E.COLI  EUBACT
GGUGAGGUGUCCGAGU#GCDGAAGGAGCACGCCUGGAAAGPGUGUAUACGGCAACGUAUCGGGGGTPCGAAUCCCCCCCUCACCGCCA
RS1663       GGA     E.COLI  EUBACT
GGUGAGG4GUCCGAGU#GDDGAAGGAGCACGCCUGGAAAGPGUGUAUACGGCAACGUAUCGGGGGTPCGAAUCCCCCCCUCACCGCCA
RS1664       VGA     E.COLI  EUBACT
GGAAGUG4GGCCGAGC#GDDGAAGGCACCGGUBUVGA*AACCGGCGACCCGAAAGGGUUCCAGAGTPCGAAUCUCUGCGCUUCCGCCA
RT1660        GGU     E.COLI  EUBACT
GCUGAUAUAGCUCAGDDGGDAGAGCGCACCCUUGGUEAGGGUGAG7UCGGCAGTPCGAAUCUGCCUAUCAGCACCA
RT1661        GGU     E.COLI  EUBACT
GCUGAUAUGGCUCAGDDGGDAGAGCGCACCCUUGGUEAGGGUGAG7UCCCAGTPCGACUCUGGGUAUCAGCACCA
RV1660        GAC     E.COLI  EUBACT
GCGUCCG4AGCUCAGDDGGDDAGAGCACCACCUUGACAUGGUGGGG7XCGGUGGTPCGAGUCCACUCGGACGCACCA
RV1661        GAC     E.COLI  EUBACT
GCGUUCA4AGCUCAGDDGGDDAGAGCACCACCUUGACAUGGUGGGG7XCGUUGGTPCGAGUCCAAUUGAACGCACCA
RV1662        VAC     E.COLI  EUBACT
GGGUGAU4AGCUCAGCDGGGAGAGCACCUCCCUVAC=AGGAGGGG7UCGGCGGTPCGAUCCCGUCAUCACCCACCA
RW1660        CCA     E.COLI  EUBACT
AGGGGCG4AGUUCAADDGGDAGAGCACCGGUBUCCA*AACCGGGU7UUGGGAGTPCGAGUCUCUCCGCCCCUGCCA
RX1660        CAU     E.COLI  EUBACT
CGCGGGG4GGAGCAGCCUGGDAGCUCGUCGGGBUCAUAACCCGAAGAUCGUCGGTPCAAAUCCGGCCCCCGCAACCA
RX1661        CAU     E.COLI  EUBACT
CGCGGGG4GGAGCAGCCUGGDAGCUCGUCGGGBUCAUAACCCGAAG7UCGUCGGTPCAAAUCCGGCCCCCGCAACCA
RY1660       QUA     E.COLI  EUBACT
GGUGGGG4UCCCGAGC#GCCAAAGGGAGCAGACUQUA*APCUGCCGUCAUCGACUUCGAAGGTPCGAAUCCUUCCCCCACCACCA
RY1661       QUA     E.COLI  EUBACT
GGUGGGG4UCCCGAGC#GCCAAAGGGAGCAGACUQUA*APCUGCCGUCACAGACUUCGAAGGTPCGAAUCCUUCCCCCACCACCA
RZ1665       UCA     E.COLI  EUBACT
_AAGAUCGUCGUCUCCGGDGAGGCGGCUGGACUUCA+AUCCAGUUGGGGCCGC_GCGGUCCCGGGCAGGTPCGACUCCUGUGAUCUU_GCCA
```

# B SUBOPT man page

SUBOPT(l) SUBOPT(l)

**NAME**

subopt − calculate suboptimal secondary structures of RNAs

**SYNOPSIS**

**subopt [−s] [-g] [-d] [−l** *number***] [−e** *range***]**

**DESCRIPTION**

*subopt* reads RNA sequences from stdin and calculates all suboptimal secondary structures within a user defined energy range above the minimum free energy (mfe). It returns the suboptimal structures in bracket notation, its energy, the degeneracy related to the free energy, the energy difference to the preceding structure and the energy difference to the mfe structure to stdout.

**OPTIONS**

**−s** Calculate the Lower Density of States. It returns the mfe, the degeneracy related to the free energy, the energy difference to the preceding structure and the energy difference to the mfe structure to stdout.

**−g** Calculate gapstatistics. Read a set of RNA sequences from stdin and generate files struc.1st and struc.2nd containing the first and second suboptimal structure, their corresponding mfe, the free energies of the suboptimal structures, the energy difference to the preceding structure and the energy difference to the mfe structure.

**−d** Don’t give stabilizing energies to single stacked bases in free ends and multiloops (dangling ends).

**−l** *number*

Input the number of desired energy levels in the output of the program. Default is 2.

**−e** *range*

Input the number of the desired energy range from the ground state in percent of the mfe. Default is 15%.

# References

[1] Vincent P. Antao and Jr. Ignacio Tinoco. Thermodynamic parameters for loop formation in RNA and DNA hairpin tetralooops. *Nucl.Acid.Res*, 20(4):819–824, 1992.

[2] R. Bellman and R. Kalaba. On Kth best policies. *J. SIAM*, 8:582–588, 1960.

[3] M. J. Bishop and C. J. Rawlings. *Nucleic Acid and Protein Sequence Analysis: An practical Approach.* IRL Press, Oxford, 1987.

[4] P.N. Borer, B. Dengler, I. Tinoco Jr., and O.C. Uhlenbeck. Travelling salesman approach to protein conformation. *J. Mol. Biol.*, 86:843–853, 1974.

[5] Tom Cech. RNA as an enzyme. *Scientific American*, 11:76–84, November 1986.

[6] J. Cupal. The density of states of RNA secondary structures, 1997. Master Thesis.

[7] J.P. Dumas and J. Ninio. Efficient algorithms for folding and computing nucleic acid sequences. *Nucl. Acids Res.*, 10:197–206, 1982.

[8] S. Ebel, T. Brown, and A. N. Lane. Thermodynamic stability and solution conformation of tandem GA mismatches in RNA and RNA.DNA hybrid duplexes. *Eur. J. Biochem.*, 220:703–15, 1994.

[9] Susan M. Freier, Ryszard Kierzek, John A. Jaeger, Naoki Sugimoto, Marvin H. Caruthers, Thomas Neilson, and Douglas H. Turner. Improved free-energy parameters for prediction of RNA duplex stability. *Proc.Natl.Acad.Sci.USA*, 83:9373–9377, 1986.

[10] W. Gilbert. The RNA world. *Nature*, 319:618, 1986.

[11] C. Guerrier-Takada and S. Altman. Catalytic activity of an RNA molecule prepared by transcription *in vitro*. *Science*, 223:285–286, 1984.

[12] C. Guerrier-Takada, K. Gardiner, T. Marsh, N. Pace, and S. Altman. The RNA moiety of ribonuclease P is the catalytic subunit of the enzyme. *Cell*, 35:849–857, 1983.

[13] L. He, R. Kierzek, J. SantaLucia, A.E. Walter, and D.H. Turner. Nearest-neighbour parameters for G-U mismatches. *Biochemistry*, 30:11124, 1991.

[14] P. G. Higgs. RNA secondary structure: a comparison of real and random sequences. *J.Phys.I (France)*, 3:43, 1993.

[15] P. G. Higgs. Thermodynamic properties of transfer RNA: A computational study. *J.Chem.Soc.Faraday Trans.*, 91(16):2531–2540, 1995.

[16] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. `Vienna RNA Package`. `pub/RNA/ViennaRNA-1.03 ftp.itc.univie.ac.at`, 1994. (Public Domain Software).

[17] Pauline Hogeweg and B. Hesper. Energy directed folding of RNA sequences. *Nucl. Acid. Res.*, 12:67–74, 1984.

[18] M.A. Huynen, A. Perelson, W.A. Vieira, and P.F. Stadler. Base pairing probabilities in a complete HIV-1 RNA. *Journal of Computational Biology*, 3(2):253–274, 1996.

[19] 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.

[20] G. F. Joyce. RNA evolution and the origins of life. *Nature*, 338:217–224, 1989.

[21] Gerald. F. Joyce. The rise and fall of the RNA world. *The New Biologist*, 3:399–407, 1991.

[22] G.F. Joyce. Building the RNA world: evolution of catalytic RNA in the laboratory. In T.R. Cech, editor, *Molecular Biology of RNA. UCLA Symposium on Molecular and Cellular Biology*, pages 361–371. New York: Alan R.Liss, 1988.

[23] G.F. Joyce. Amplification, mutation, and selection of catalytic RNA. *Gene*, 82:85–87, 1989.

[24] D.A.M. Konings. Pattern analysis of RNA secondary structures. *Proefschrift, Rijksuniversiteit te Utrecht*, 1989.

[25] D.A.M. Konings and P. Hogeweg. Pattern analysis of RNA secondary structure, similarity and consensus of minimal-energy folding. *J. Mol. Biol.*, 207:597–614, 1989.

[26] John S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.

[27] S. E. Morse and D. E. Draper. Purine-purine mismatches in RNA helices: evidence for protonated GA pairs and next nearest neighbors effects. *Nucl. Acids Res.*, 23:302–306, 1995.

[28] A. Nakaya, K. Yamamoto, and A. Yonezawa. RNA secondary structure prediction using highly parallel computers. *Comput. Applic. Biosci.*, 11(6):685–692, 1995.

[29] J. Ninio. Prediction of pairing schemes in RNA molecules - loop contributions and energy of wobble and non wobble pairs. *Biochimie*, 61:1133, 1979.

[30] Ruth Nussinov, George Piecznik, Jerrold R. Griggs, and Daniel J. Kleitman. Algorithms for loop matching. *SIAM J. Appl. Math.*, 35(1):68–82, 1978.

[31] C. Papanicolau, M. Gouy, and J. Ninio. An energy model that predicts the correct folding of the tRNA and the 5S RNA molecules. *Nucl. Acid. Res.*, 12:31–44, 1984.

[32] A.E. Peritz, R. Kierzek, N. Sugimoto, and D. Turner. Thermodynamic study of internal loops in oligoribonucleotides: Symmetric loops are more stable than asymmetric loops. *Biochemistry*, 30:6428–6436, 1991.

[33] D. Poerschke. Elementary steps of base recognition and helix-coil transitions in nucleic acids. In I. Pecht and R.Rigler, editors, *Chemical relaxation in molecular biology*, pages 191–218, Berlin, 1977. Springer-Verlag.

[34] Wolfram Saenger. *Principles of Nucleic Acid Structure*. Springer, 1984.

[35] W. Salser. Globin messenger RNA sequences - analysis of base-pairing and evolutionary implications. *Cold Spring Harbour Symp. Quant. Biol.*, 42:985, 1977.

[36] J. SantaLucia, R. Kierzek, and D.H. Turner. Functional group substitutions as probes of hydrogen bonding between GA mismatches in RNA internal loops. *J. Am. Chem. Soc*, 113:4313–4322, 1991.

[37] J. SantaLucia, R. Kierzek, and D.H. Turner. Stabilities of consecutive AC,CC,GG,UC, and UU mismatches in RNA internal loops: Evidence for stable hydrogen-bonded UU and CC+ pairs. *Biochemistry*, 30:8242–8251, 1991.

[38] M. J. Serra, T. J. Axenson, and D.H. Turner. A model for the stabilities of RNA hairpins based on a study on the sequence dependence of stability for hairpins of six nucleotides. *Biochemistry*, 33:14289–14296, 1994.

[39] Martin J. Serra, Matthew H. Lyttle, Theresa J. Axenson, Calvin A. Schadt, and Douglas H. Turner. RNA hairpin loop stability depends on closing base pair. *Nucl. Ac. Res.*, 21(16):3845 – 3849, 1993.

[40] S. Spiegelman. An approach to the experimental analysis of precellular evolution. *Quart. Rev. Biophys.*, 17:213, 1971.

[41] C. Steegborn, S. Steinberg, F. Huebel, and M. Sprinzl. Compilation of tRNA sequences and sequences of tRNA genes. *Nucl. Acids Res.*, 24(1), 1995.

[42] D. H. Turner, N. Sugimoto, and S. Freier. RNA structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17:167–192, 1988.

[43] A. E. Walter, D. H. Turner, J. Kim, M. H. Lyttle, P. Muller, D. H. Mathews, and M. Zuker. Coaxial stacking of helices enhances binding of oligoribonucleotides and improves prediction of RNA folding. *Proc. Natl. Acad. Sci.*, 91:9218 – 9222, 1994.

[44] A. E. Walter, M. Wu, and D.H. Turner. The stability and structure of tandem G-A mismatches in RNA depend on closing base pair. *Biochmistry*, 33:11349 – 54, 1994.

[45] 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.

[46] M. S. Waterman and T.H. Byers. A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Mathematical Biosciences*, 77:179–188, 1985.

[47] M. S. Waterman and T. F. Smith. RNA secondary structure: A complete mathematical analysis. *Mathematical Biosciences*, 42:257–266, 1978.

[48] M. Wu, J. A. McDowell, and D. H. Turner. A periodic table of symetric tandem mismatches in RNA. *Biochemistry*, 34:2304–11, 1995.

[49] K. Yamamoto, Y. Kitamura, and H. Yoshikura. Computation of statistical secondary structure of nucleic acids. *Nucl. Acids Res.*, 12:335–346, 1984.

[50] K. Yamamoto and H. Yoshikura. Computer program for prediction of the optimal and suboptimal secondary structures of long RNA molecules. *Comput. Applic. Biosci.*, 1(2):89–94, 1985.

[51] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[52] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bull.Math.Biol.*, 46(4):591–621, 1984.

[53] M. Zuker and P. Stiegler. Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148, 1981.

# Curriculum Vitae

## Stefan Wuchty

* 11.6.1972, Wien

| | |
|---|---|
| 1978 – 1980 | Volksschule Lingenau, 6951 Lingenau in Vorarlberg |
| 1980 – 1982 | Volksschule Kleistgasse, Kleistgasse 12, 1030 Wien |
| 1982 – 1990 | Bundesrealgymnasium BRG III, Radetzkystrasse 2a, 1030 Wien |
| 5.90 | Reifeprüfung |
| 1990 – 1991 | Präsenzdienst FMAR, Maria-Theresienkaserne Fasangartenstrasse 8, 1120 Wien |
| 1991 – 1997 | Studium der Chemie, Studienzweig Biochemie an der Universität Wien |
| 9.94 | 1. Diplomprüfung Chemie |
| 3.97 – 2.98 | Diplomarbeit am Institut für Theoretische Biochemie an der Universität Wien |
| 3.98 | 2. Diplomprüfung Chemie mit Auszeichnung |

# Publications