The Density of States of RNA Secondary Structures

DIPLOMARBEIT

eingereicht von

Jan Cupal

zur Erlangung des akademischen Grades

Magister rerum naturalium

an der Formal- und Naturwissenschaftlichen Fakultät

der Universität Wien

June 9, 1997

Ich danke allen recht herzlich!

```
\newpage
```

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 für die Vorhersage von 3D-Strukturen und für das Verstüdnis biochemischer Vorgänge wertvolle Information.

RNA Sekundärstrukturen können als planare Graphen beschrieben werden. Eine Reihe schon früher entwickelter Algorithmen zur Berechnung der Grundzustandsenergie und der Zustandssumme, die auf der Abzählung von alternativen Graphen beruhen, wurden zusammengestellt und in konsistenter Notation beschrieben. Ein neuer Algorithmus zur Berechnung der Zustandsdichte von RNA Sekundärstrukturen, basierend auf *dynamic programming*, wurde entwickelt und in ein Programm umgesetzt.

Eine Anzahl von Berechnungen wurde durchgeführt, um die sich aus der Zustandsdichte ergebende Menge an Information zu verdeutlichen. Die vollständige Zustandsdichte der Phenylalanin-tRNA von Hefe wurde sowohl bei einer Energieauflösung von 0,1 kcal/mol, als auch – für den Bereich von 5 kcal/mol über der Grundzustandsenergie – mit einer Auflösung von 0,01 kcal/mol berechnet. Die Zustandsdichten von 30 E. Coli tRNAs wurden mit den Ergebnissen für Zufallssequenzen mit gleicher Basenzusammensetzung und gleicher Länge verglichen. Die Ergebnisse zeigen, daßdie ursprünglichen tRNA-Sequenzen im Vergleich weniger Zustände in der Umgebung des Grundzustandes aufweisen und der Abstand vom Grundzustand zum ersten angeregten Zustand höher ist.

Abstract

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 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. A variety of dynamic programming algorithms based on graph enumeration derived previously were compiled and presented in a consistent notation. A new dynamic programming algorithm for the density of states of RNA secondary structures was developed and implemented for the first time.

A number of sample calculations were performed in order to highlight the amount of information yielded from the density of states. The complete density of Yeast $tRNA^{Phe}$ was computed at a resolution of 0.1 kcal/mol, and, within a region of 5 kcal/mol above the ground state, at an energy resolution of 0.01 kcal/mol. A number of 30 E. Coli tRNAs were analyzed and compared with random sequences of same base composition and length. The results show that original tRNA sequences have less states in the vicinity of the ground state and the energy gap is usually larger.

Contents

List of Figures

List of Tables

1 Introduction

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 (Cech 1986; Guerrier-Takada *et al.* 1983; Guerrier-Takada & Altman 1984; Joyce 1989a). 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* (Gilbert 1986; Joyce 1988; 1989b; 1991) 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 (Spiegelman 1971), 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 *irrational 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 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 based on graph enumeration have been available now for some time (Waterman & Smith 1978; Zuker & Stiegler 1981). Naturally the

algorithm yields 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 negligble. An elegant solution for this problem was suggested by McCaskill (McCaskill 1990), who proposed an algorithm to compute the partition function and the matrix of base pairing probabilities of an RNA molecule. The Vienna RNA Package (Hofacker *et al.* 1994a) 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 (Higgs 1993; 1995) 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 (Higgs 1993).

In this work we introduce a dynamic programming algorithm for the computation of the distribution of states of RNA secondary structures, which implements the energy parameter set used within the Vienna RNA Package and is not restricted to any minimum stem length. It will be shown 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 key observation is that the density of states of a sequence can be obtained from the density of states of smaller subsequences.

The algorithm, however, is quite demanding both in terms of memory and CPU time: A total of $\mathcal{O}(n^3m^2)$ operations, where n and m give the sequence length and number of energy bins, respectively, are required to compute $\mathcal{O}(n^2m)$ entries, which have to be stored throughout the execution time. With a constant energy resolution the number of energy bins used to store the num-

ber of states, becomes proportional to the number of bases n, resulting in CPU time requirements of $\mathcal{O}(n^5)$.

In spite of this unfavorable scaling, it will be shown that the computation of the distribution of states of biologically important molecules is feasible at a sufficient energy resolution. This is due to implementation variants of the algorithm, which reduce the scaling of the CPU requirements and the prefactors. The study of large samples of (small) RNA molecules on statistical basis to gain thermodynamic information is possible.

A few examples were studied to elucidate the facilities offered by the algorithm. A variety of tRNA molecules from E. Coli were compared with random sequences of same base composition and length. The results show that biologically evolved sequences are far from equilibrium. It seems probable that a stable ground-state structure is an important criterion in natural selection.

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

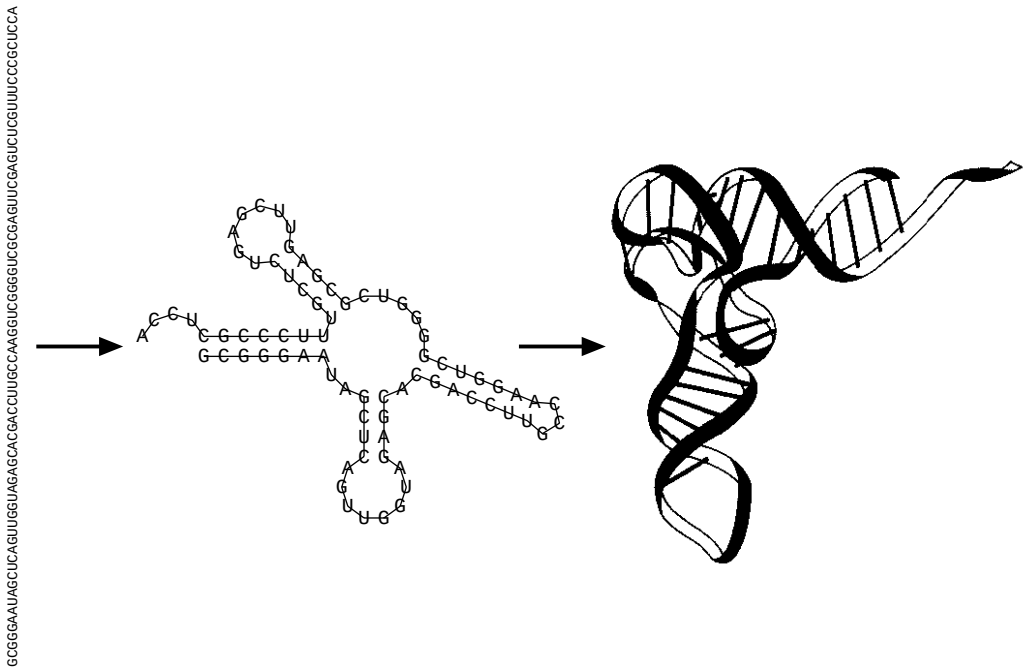


Figure 1: Folding of an RNA sequence into its spatial structure. The process is partitioned into two phases: in the first phase only the Watson-Crick-type base pairs are formed which constitute the major fraction of the free energy, and in the second phase the actual spatial structure is built by folding the planar graph into a three-dimensional object. The example shown here is phenylalanyl-transfer-RNA $tRNA^{Phe}$, 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, see section 5. 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 re 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. There is by now no satisfying secondary structure prediction algorithm dealing with pseudo-knots.

3 Secondary Structure Graphs

3.1 Definitions

Definition 3.1. (Waterman 1978; Waterman & Smith 1978) 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

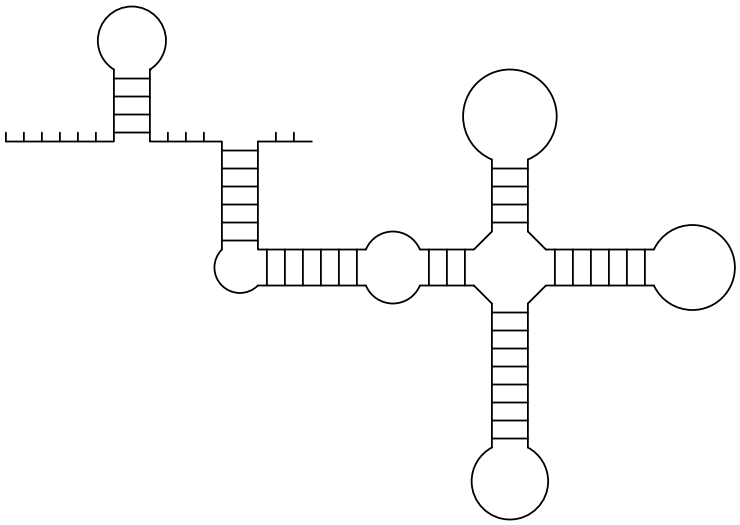


Figure 2: An example for an RNA secondary structure, with free dangling ends, stems and loops.

(k, l). A base pair (p, q) is said to be (immediately) interior, if p and q are (immediately) interior to (k, l).

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

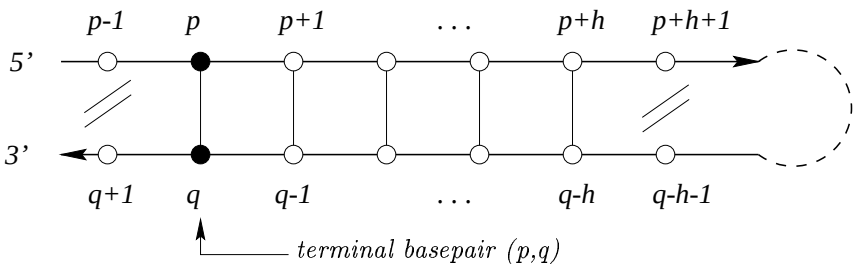
(1) A *stem* consists of subsequent base pairs (p, q), $(p+1, q-1)$, ..., $(p+h-1, q-h+1)$, $(p+h, q-h)$ such that neither $(p-1, q+1)$ nor $(p+h+1, q-h-1)$ is a base pair. $h+1$ is the *length* of the stem, (p, q) is the terminal base pair of the stem.

(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 Φ 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.



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 $k = 2$.

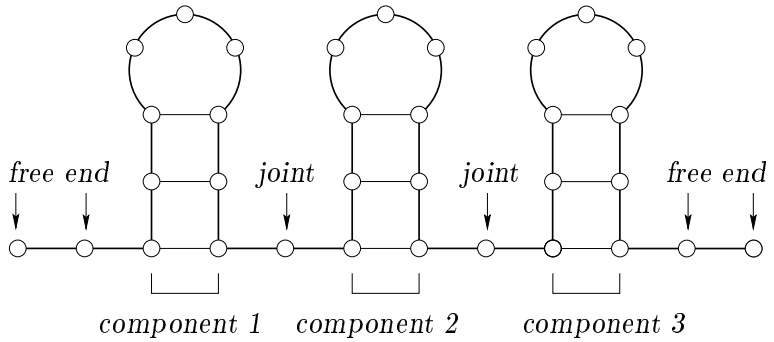


Figure 3: An example for an RNA secondary structure consisting of three components and six external vertices (2 joints and 4 free ends).

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 pair. The sub-structure enclosed by the terminal base pair (p,q) of a terminal stem will be called a *component* of Φ. 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).

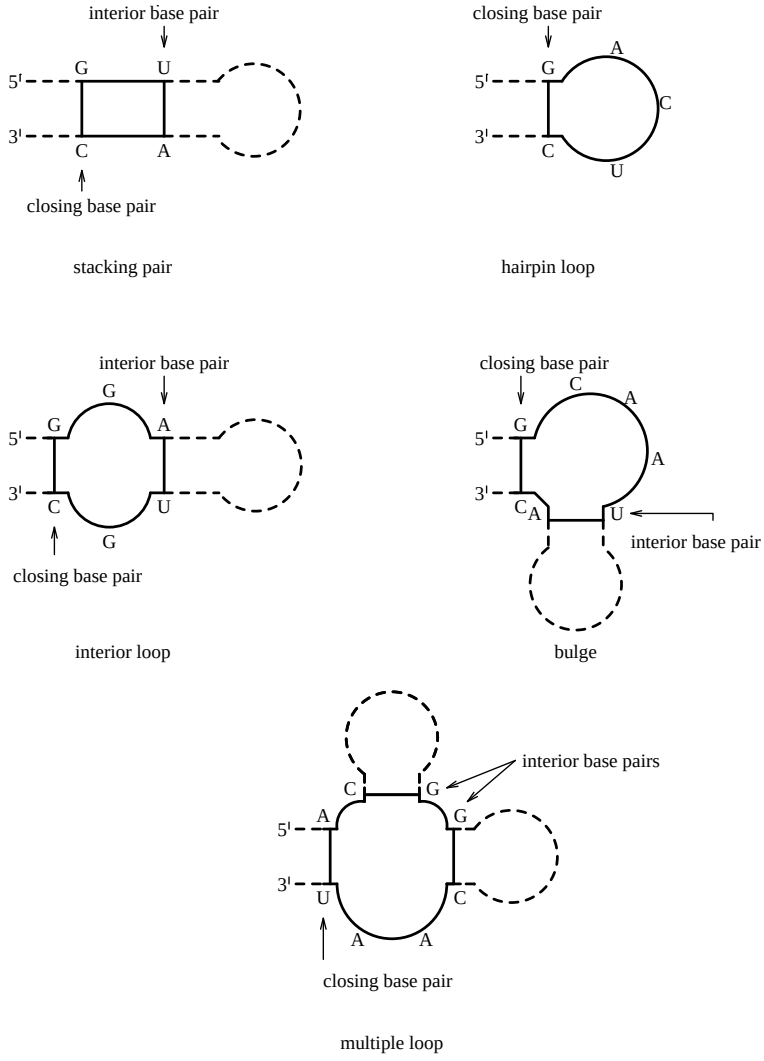


Figure 4: The classification of loops for the decomposition of RNA secondary structure.

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

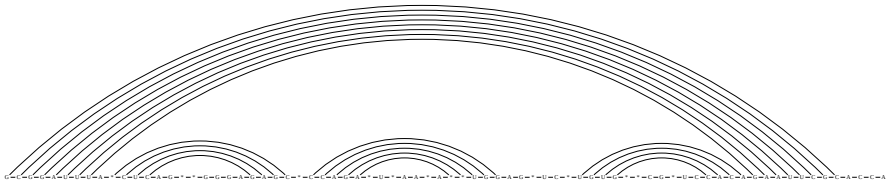


Figure 5: The secondary structure of $tRNA^{Phe}$ in *linked graph representation*.

A particularly easy way to draw secondary structure graphs was suggested by Ruth Nussinov (Nussinov *et al.* 1978). 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 un-knotted by definition, no two chords will intersect. See Figure 6 for circular representation of $tRNA^{Phe}$.

Paulien Hogeweg and Danielle Konings conceived a related graphical method for the comparison of RNA secondary structures called *mountain representation* (Hogeweg & Hesper 1984; Konings & Hogeweg 1989; Konings 1989) 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.

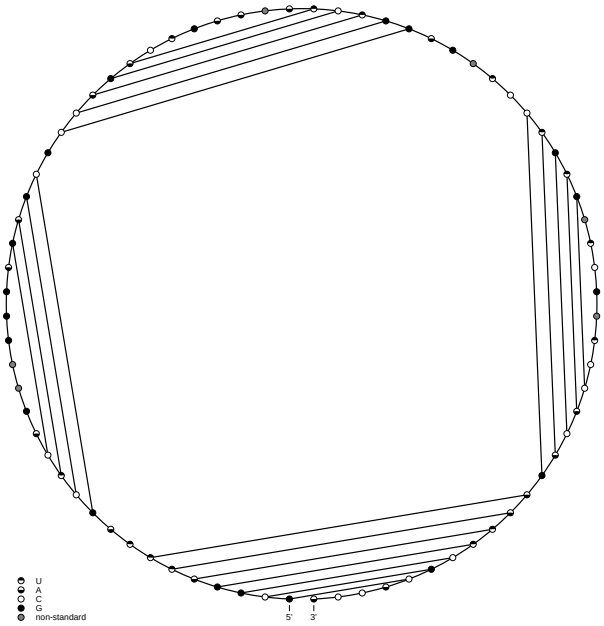


Figure 6: The secondary structure of $\mathrm{tRNA}^{\mathrm{Phe}}$ in *Circular representation.*

- *Plateaus* represent unpaired bases. When interrupting sloped regions they indicate bulges or interior loops, depending on whether they occur 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 for straightforward comparison of secondary structures and inspired a convenient algorithm for alignment of secondary structures (Konings & Hogeweg 1989).

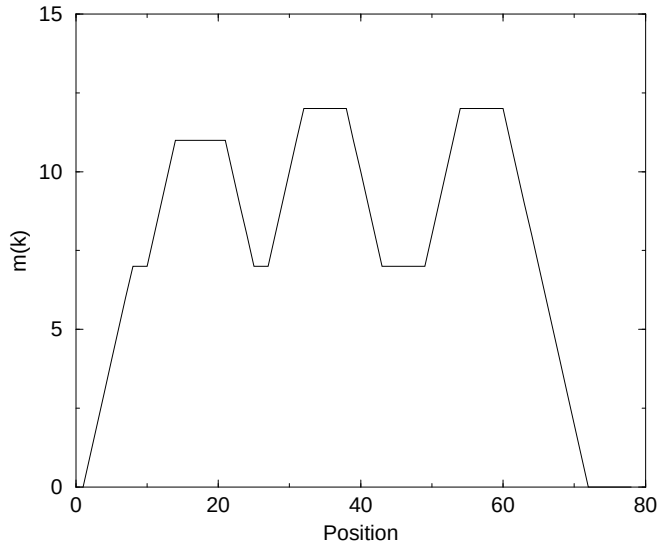


Figure 7: The secondary structure of $tRNA^{Phe}$ from Yeast (see Figure 1) in *mountain representation*. The same structure in string representation is
(((((((..((((........)))).(((((.......)))))).....(((((.......)))))))))))....

4 Enumeration of Secondary Structure Graphs

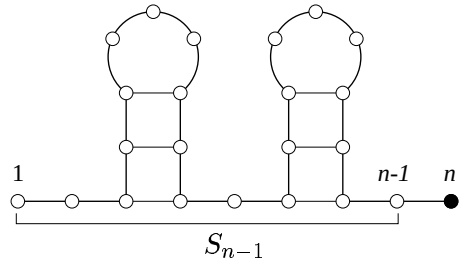
4.1 The Basic Recursion

A secondary structure on n digits may be obtained from a structure on $n-1$ digits by adding a base at the right hand. This base n may constitute a free end or form a base pair (k,n) with any other base k. In the first case the number of structures on n digits is equal to the number on $n-1$ digits. In the second case the substructure enclosed by the new pair is an arbitrary structure on $n-k-1$ digits, and the remaining part of length $k-1$ is also an arbitrary valid secondary structure. The total number of structures is the product of the number of substructures S_{k-1} and S_{n-k-1} on the substrings. Therefore, we obtain the following recursion formula for the number S_n of secondary structures:

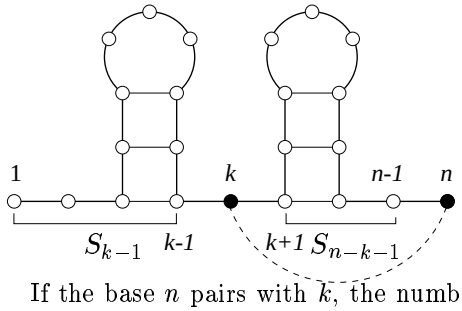
Theorem 4.1. *Let S_n be the number of secondary structures on n vertices. If m is minimal loop size and $S_0 = 1$ then S_n satisfies*

$$S_n = S_{n-1} + \sum_{k=1}^{n-m-1} S_{k-1}\, S_{n-k+1}, \qquad n \geq m+1 \tag{1}$$

$$S_0 = S_1 = \ldots = S_{m+1} = 1 \tag{2}$$



If the newly added base does not pair, the number of structures on string n equals the number of structures in string $n-1$.



If the base n pairs with k, the number of structures is the produkt of all structures in the remaining part S_{k-1} and the newly formed component S_{n-k-1}.

Proof. It is easy to see from Definition 3.1 that for $n \leq m+1$ the only structure is the open chain and therefore

$$S_0 = S_1 = \ldots = S_{m+1} = 1$$

For $n > m+1$ see above.

Theorem 4.1. was derived by Waterman (Waterman 1978), see also (Hofacker 1994) and (Waterman 1995). Note that our definition of S_n differs from (Waterman 1995) where $S_0 = 0$.

4.2 Structures with Given Numbers of Components

Let $J_n(b)$ denote the number of structures on n vertices with exactly b components. The derivation of the recursion relations parallels the argument leading to equ.(1):

$$\begin{aligned} J_n(b) &= J_{n-1}(b) + \sum_{k=1}^{n-m-1} S_{k-1} J_{n-k+1}(b-1), \qquad b > 0,\ n \geq m+1 \quad (3) \\ & J_n(b) = 0,\ b > 0, n \leq m+1, \qquad J_n(0) = 1, n \geq 0 \end{aligned}$$

because adding an unpaired digit to a structure on n digits does not change the number of components, while introducing an additional bracket makes the bracketed part of length k a single component and does not affect the remainder of the sequence.

4.3 Structures with Given Numbers of Base Pairs

Let $H_n(b)$ denote the number of structures with exactly b base pairs (bonds) on n vertices. The recursion

$$\begin{aligned} H_n(b) &= H_{n-1}(b) + \sum_{k=1}^{n-m-1} \sum_{\ell=0}^{b-1} H_{k-1}(\ell)\, H_{n-k+1}(b-\ell-1) \qquad (4) \\ & b > 0,\ n \geq m+1 \\ & H_n(b) = 0,\ b > 0, n \leq m+1 \\ & H_n(0) = 1, n \geq 0 \end{aligned}$$

is immediate. It is only necessary to introduce an additional sum over the number of unpaired digits in the newly bracketed part of the structure.

4.4 Structures with prescribed loop energies

Definition 4.2. If we treat stacked base pairs as loops of minimal size (*size zero* and *degree 2*) and assign each loop L a distinct loop energy $\mathcal{H}$, the total energy of a structure Φ is

$$E(\Phi) = \sum_{L \in \Phi} \mathcal{H}(L) \tag{5}$$

We then obtain for the number of structures $N_n(\epsilon)$ with exactly energy ϵ on n vertices

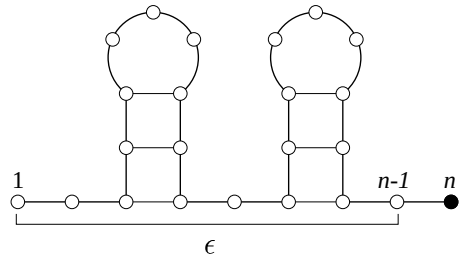
$$N_n(\epsilon) = N_{n-1}(\epsilon) + \sum_{k=1}^{n-m-1} \sum_{\epsilon'=0}^{\epsilon} N_{k-1}(\epsilon')\, N_{n-k-1}(\epsilon - \epsilon' - \mathcal{H}_C) \tag{6}$$

$$n \geq m+1$$

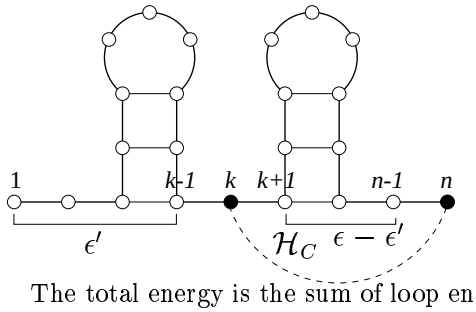
$$N_n(\epsilon = 0) = 1,\ N_n(\epsilon \neq 0) = 0, \tag{7}$$

$$n \leq m+1$$

$\mathcal{H}_C$ is the energy of the loop closed by the newly added base pair (k, n). See Section 5 for a detailed discussion of the energy model.



If the newly added base is not contained in a loop, the total energy of a structure is equal to the energy of the same structure without the free end.



The total energy is the sum of loop energies: $\epsilon - \epsilon'$ of the substructure (k+1,n-1), ϵ of the remaining structure (1,k-1) and $\mathcal{H}_C$ of the newly formed loop.

4.5 Number of Structures on a String of Bases

Up to now we have neglected the fact that secondary structures are built on sequences. Not all secondary structures can be formed by a given biological sequence, since not all combinations of nucleotides form base pairs. The results of the previous sections will be generalized to this situation in the following.

Definition 4.3. Let $\mathcal{A}$ be some finite alphabet of size κ, let Π be a symmetric Boolean $\kappa \times \kappa$-matrix and let $\Sigma = [\sigma_1 \dots \sigma_N]$ be a string of length N over $\mathcal{A}$. A secondary structure is *compatible* with the sequence Σ if for all base pairs (p, q) holds $\Pi_{\sigma_p, \sigma_q} = 1$.

The number of secondary structures N compatible with some string can be enumerated as follows:

Definition 4.4. Let $N_{i,j}$ denote the number of structures compatible with the substring $[\sigma_i \dots \sigma_j]$. The number of structures $N^B_{i,j}$ on a substring $[\sigma_i \dots \sigma_j]$ under the condition the σ_i and σ_j *form a base pair* then is

$$N^B_{i,j} = N_{i+1,j-1} \Pi_{\sigma_i, \sigma_j} \tag{8}$$

The total number of structres on a substring $[\sigma_i \dots \sigma_j]$ satisfies the recursion:

$$N_{i,j} = N_{i,j-1} + \sum_{k=i}^{j-m-1} N_{i,k-1} N_{k+1,j-1} \Pi_{\sigma_k, \sigma_j} \tag{9}$$

$$= N_{i,j-1} + \sum_{k=i}^{j-m-1} N_{i,k-1} N^B_{k,j} \tag{10}$$

Remark: (Hofacker 1994) For a random sequence, the expected number $\bar{S}_n$ of compatible structures is then

$$\bar{S}_{l,n} = \bar{S}_{l,n-1} + p \sum_{k=l}^{n-m-1} \bar{S}_{l,k-1} \bar{S}_{k+1,n-k} \tag{11}$$

where

$$p = \frac{1}{\kappa^2} \sum_{i,j=1}^{\kappa} \Pi_{ij} \tag{12}$$

is called the *stickiness* of the alphabeth $\mathcal{A}$.

For short substrings, $j - i \leq m$, where m is the minimal size of a hairpin loop, we find

$$N_{i,i} = \underbrace{N_{i,i-1}}_{1} + \sum_{k=i}^{i-m-1} \ldots = 1 \qquad (\text{see } S_0 = 1) \tag{13}$$

$$N_{i,i+1} = N_{i,i} + \sum_{k=i}^{i+1-m-1} \ldots = 1 \tag{14}$$

$$N_{i,i+2} = N_{i,i+1} + \sum_{k=i}^{i+2-m-1} \ldots = 1 \tag{15}$$

$$N_{i,i+m} = N_{i,i+m} + \sum_{k=i}^{i+m-m-1} \ldots = 1 \tag{16}$$

$$N_{i,i+m+1} = N_{i,i+m+1} + \sum_{k=i}^{i+m+1-m-1=i} \underbrace{N_{i,k-1}}_{1} N^B_{k=i,i+m+1} \tag{17}$$

The first $m+1$ sums are empty because $N^B_{k,j} = 0$ for $j - k \leq m$. This corresponds to condition (2) in equ. (10).

We are now recursivly substituting the first term in equ. (10).

$$\begin{aligned}
N_{i,j} &= N_{i,j-1} + \sum_{k=i}^{j-m-1} N_{i,k-1} N^B_{k,j} \\
&= N_{i,j-2} + \sum_{k=i}^{j-1-m-1} N_{i,k-1} N^B_{k,j-1} + \sum_{k=i}^{j-m-1} N_{i,k-1} N^B_{k,j} \\
&\vdots \\
N_{i,j} &= \underbrace{1}_{N_{i,i-1}} + \sum_{l=i+m+1}^{j} \sum_{k=i}^{l-m-1} N_{i,k-1} N^B_{k,l}
\end{aligned} \tag{18}$$

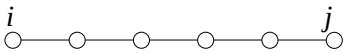
This expression will be useful in the following section, where we will show that an equal recursion scheme can be derived by decomposition of structures.

4.6 Decomposition of Structures

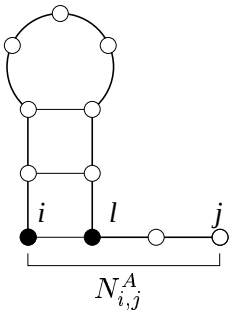
An equivalent recursive expression for the number of structures can be derived by the following approach.

Definition 4.5. Let $N^A_{i,j}$ denote the number of structures compatible with a string of bases $[\sigma_i \dots \sigma_j]$ consisting of a single component enclosed by base pair (i, l) with $(i+m+1 < l \leq j)$ and an arbitrary number of free ends at the right (3') side.

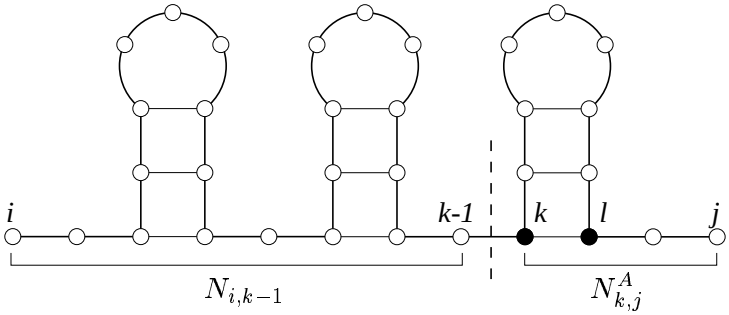
Each structure compatible with a string of bases can be attributed to one of the following three cases:



The open chain is a valid secondary structure compatible to a string of bases.



$N^A_{i,j}$ counts all structures consisting of a single component enclosed by a base pair (i, l) and an arbitrary number of free ends at the right (3') side.



For a vertex k $(i < k < j - m)$ the string $[\sigma_i ... \sigma_j]$ is diveded into two substrings $[\sigma_k ... \sigma_j]$ and $[\sigma_i ... \sigma_{k-1}]$. $N^A_{k,j}$ is the number of all structures compatible to $[\sigma_k ... \sigma_j]$ with k binding to a base l satisfying $k+m < l \leq j$, thus forming the terminal base pair of a terminal stack. $N_{i,k-1}$ is the number of all structures compatible to $[\sigma_i ... \sigma_{k-1}]$.

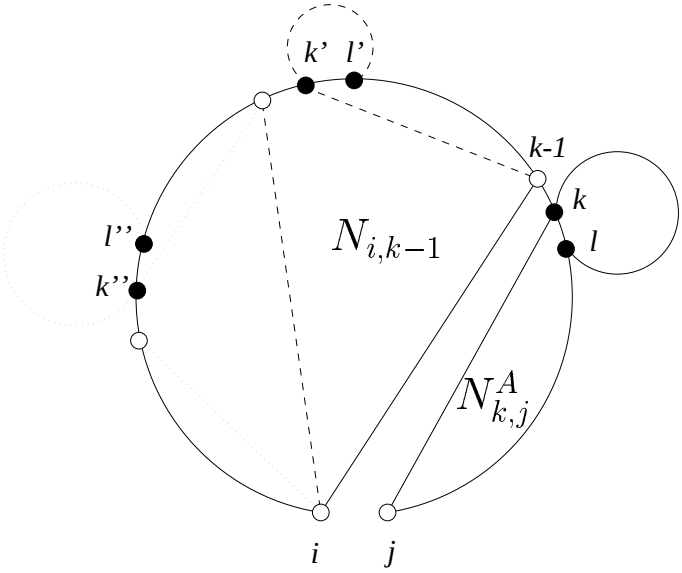


Figure 8: Recursive decomposition of multicomponent secondary structures: The total number of structures is always derived by the multiplication of the number of structures on two substrings $[\sigma_i \dots \sigma_{k-1}]$ and $[\sigma_k \dots \sigma_j]$. $N^A_{k,j}$ sums up all structures which consist of a single component with base k forming a base pair with any other base l $(k+m < l \leq j)$, and all bases $> l$ being unpaired. The unconstrained number of structures $N_{i,k-1}$ compatible to the remaining string is derived by (recursively) decomposeing the remaining string into a substring containing the rightmost component and the remaining string containing the other components or – at the end of the recursion – an arbitrary number of unpaired bases.

(1) The single open structure, containing no base pairs.

$$N^{(1)}_{i,j} = 1$$

(2) All structures consisting of a single component with no free ends at the left (5') side and and an arbitrary number of unpaired bases at the right side adjacent to the terminal base pair of the single component. The number of these structures on a string $[\sigma_i \dots \sigma_j]$ is denoted $N^A_{i,j}$.

$$N^{(2)}_{i,j} = N^A_{i,j}$$

(3) All structures consisting of a single component and unpaired bases at the left (5') side or consisting of more than one component. The number

of these structures on a string $[\sigma_i \dots \sigma_j]$ is derived recursively from the number of structures $N^A_{k,j}$ with k forming the terminal base pair of the rightmost component on $[\sigma_i \dots \sigma_j]$ and $N_{i,k-1}$ denoting the total number of structures on the remaining sequence, $[\sigma_i \dots \sigma_{k-1}]$.

$$N^{(3)}_{i,j} = \sum_{k=i+1}^{j-m-1} N_{i,k-1}\, N^A_{k,j}$$

The total number of structures on the string $[\sigma_i \dots \sigma_j]$ then is

$$\begin{aligned} N_{i,j} &= N^{(1)}_{i,j} + N^{(2)}_{i,j} + N^{(3)}_{i,j} \\ &= 1 + N^A_{i,j} + \sum_{k=i+1}^{j-m-1} N_{i,k-1}\, N^A_{k,j} \end{aligned} \tag{19}$$

By definition $N^A_{i,j}$ is the number of all structures with i pairing to any base l satisfying $(i+m+1 < l \leq j)$ and $l+1 \dots j$ being unpaired. Thus (i,l) is the terminal base pair of the single component. According to equ. (8), $N^B_{i,j}$ denotes

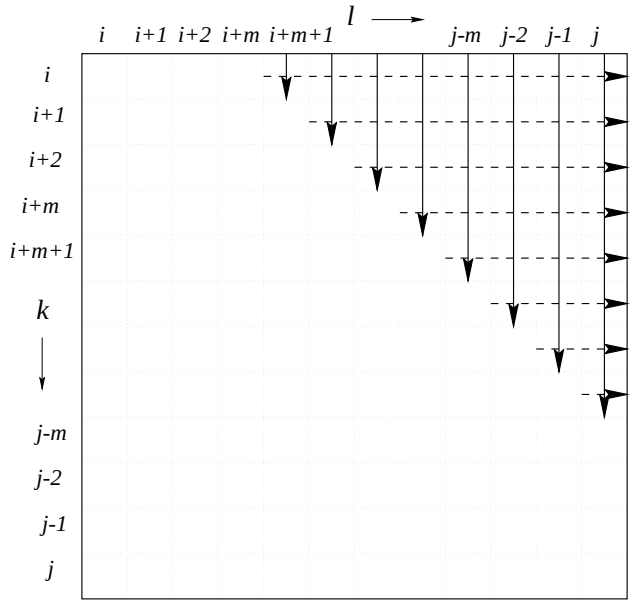


Figure 9: The array elements $N^B_{k,l}$ added up under the double sums in equ. (18) and (22) are identical. Equ. (18) sums up rows, equ. (22) sums up columns.

the number of structures on a string $[\sigma_i \dots \sigma_j]$ under the condition that σ_i and σ_j form a base pair. The number $N^A_{i,j}$ of all structures consisting of a single component enclosed by (i,l) therefore is

$$N^A_{i,j} = \sum_{l=i+m+1}^{j} N^B_{i,l} \tag{20}$$

Substituting this to equ. (19) yields

$$N_{i,j} = 1 + \sum_{l=i+m+1}^{j} N^B_{i,l} + \sum_{k=i+1}^{j-m-1} \left[N_{i,k-1} \sum_{l=k+m+1}^{j} N^B_{k,l} \right]$$

$$N_{i,j} = 1 + \sum_{k=i+m+1}^{j} N^B_{i,l} + \sum_{k=i+1}^{j-m-1} \sum_{l=k+m+1}^{j} N_{i,k-1} N^B_{k,l} \tag{21}$$

$$= 1 + \sum_{k=i}^{j-m-1} \sum_{l=k+m+1}^{j} N_{i,k-1} N^B_{k,l}. \tag{22}$$

The argument explained in Figure 9 shows that the last expression can be arranged in such a way that equ. (18) is recovered.

4.7 Loop Decomposition

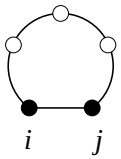
In this section we derive a recursion for the number $N^B_{i,j}$ of secondary structures in which i and j form a base pair. It involves three distinct possibilities (McCaskill 1990):

(1) Base pair (i,j) closes a hairpin loop; there are no base pairs interior to (i,j). For a given i and j there is only one structure forming a hairpin.

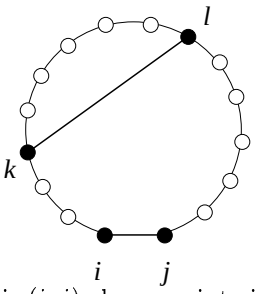
$$N^{B(1)}_{i,j} = \Pi_{\sigma_i,\sigma_j}$$

(2) (i,j) closes an interior loop or a stem; there is a single base pair (k,l) immediately interior to (i,j). The number of structures satifying this restriction is

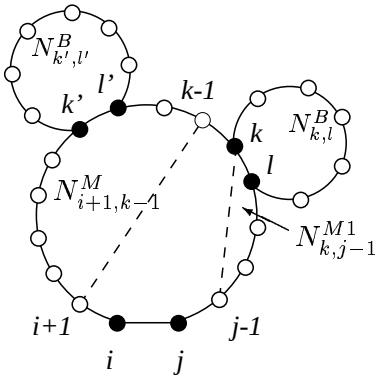
$$N^{B(2)}_{i,j} = \Pi_{\sigma_i,\sigma_j} \cdot \left[\sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} N^B_{k,l} \right]$$



For given vertices i and 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 (k,l) is immediatly interior to (i,j). The number of structures for all possible values of k and l are considered.



Base pair (i,j) closes a multiloop, base pairs (k,l), (k',l') ... are immediatly interior to (i,j). Multiloop structures are divided in substructures containing the rightmost stem and the remaining structure. The actual number of multiloop structures on (i+1,j-1) is given by the product of the number of structures on the two parts. The number $N^M_{i+1,k-1}$ of arbitrary structures on the 5' part is again determined from smaller fragments.

(3) (i,j) closes a multiloop; there are at least two pairs immediately interior to (i,j). The string forming the multiloop is divided in two parts, one containing the rightmost stem, the other containing the remaining string. This decomposition is essentially the same as the one discussed in the previous section, i. e., the total number of structures is the product of two contributions derived from a substructure containing a single component, N^{M1}, and an arbitrary remaining structure, N^{M}, which may consist of one or more stems plus joining or tailing unpaired bases, see Figure 10.

$$N_{i,j}^{B(3)} = \Pi_{\sigma_i,\sigma_j} \cdot N_{i+1,j-1}^{M} = \Pi_{\sigma_i,\sigma_j} \cdot \left[\sum_{k=i+1}^{j-m-2} N_{i+1,k-1}^{M} N_{k,j-1}^{M1} \right]$$

The total number of structures $N_{i,j}^{B}$ on the string $[\sigma_i \ldots \sigma_j]$ then is

$$\begin{aligned} N_{i,j}^{B} &= \Pi_{\sigma_i,\sigma_j} \cdot \left[N_{i,j}^{B(1)} + N_{i,j}^{B(2)} + N_{i,j}^{B(3)} \right] \\ &= \Pi_{\sigma_i,\sigma_j} \cdot \left[1 + \sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} N_{k,l}^{B} + \sum_{k=i+1}^{j-m-2} N_{i+1,k-1}^{M} N_{k,j-1}^{M1} \right] \end{aligned} \quad (23)$$

It remains to derive the recursion for the multiloop-related contributions. By definition $N_{i,j}^{M1}$ counts all structures consisting of a single component enclosed by the base pair (i,l). In complete analogy with equ. (20) we obtain

$$N_{i,j}^{M1} = \sum_{l=i+m+1}^{j} N_{i,l}^{B} \quad (24)$$

The number $N_{i,j}^{M}$ of (arbitrary) structures on the remaining substring can be obtained recursively:

$$N_{i,j}^{M} = \sum_{k=i+m+1}^{j-m-1} N_{i,k-1}^{M} N_{k,j}^{M1} + \sum_{k=i}^{j-m-1} N_{k,j}^{M1} \quad (25)$$

$$N_{i,i-1}^{M} = N_{i,i}^{M} = N_{i,i+1}^{M} = \ldots = N_{i,i+m}^{M} = 0 \quad (26)$$

The first term contributing to $N_{i,j}^{M}$ takes into account all substructures containing two or more than two stems, the second term counts all structures with

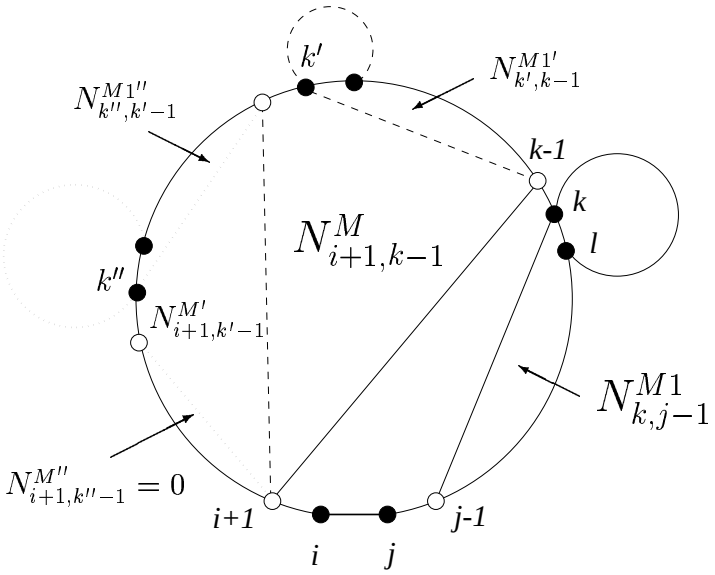


Figure 10: Recursive decomposition of multiloops: The total number of structures is derived by the multiplication of the number of structures on two substrings $[\sigma_{i+1}...\sigma_{k-1}]$ and $[\sigma_k...\sigma_{j-1}]$ with k running over all possible values. Multiloop structures are thus decomposed into substructures consisting of a single (arbitrary) rightmost component enclosed by (k,l) plus free ends and an arbitrary remaining structure. According to equ. (24), $N^{M1}_{k,j-1}$ depends on the number of structures $N^{B}_{k,l}$, $k+m<l\leq j-1$. An expression for N^{M} has to take into account that the remaining substructure may again consist of two or more components $N^{M}_{i+1,k-1}$, which are recursively decomposed, or might consist of a single component $N^{M'}_{i+1,k'-1}$, thus constituting the end of the recursion (equ. 25).

one stem. The substructure enclosed by the closing pair (i,j) of a multiloop is thus recursively decomposed into a substructure containing the rightmost stem and the arbitrary remaining structure. Remaining structures consisting only of a single stem are taken into account by the second term, see Figure 10.

Table 1 summarizes the loop-decomposed version of the the basic recursion (equ. 1), which forms the basis of all thermodynamic based folding algorithms.

In the following section the energy model will be introduced and discussed in detail.

$$
\begin{aligned}
N^B_{i,j} &= \Pi_{\sigma_i,\sigma_j} \cdot \left[1 + \sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} N^B_{k,l} \right. \\
&\qquad \left. + \sum_{k=i+1}^{j-m-2} N^M_{i+1,k-1} \, N^{M1}_{k,j-1} \right] \\
N^{M1}_{i,j} &= \sum_{l=i+m+1}^{j} N^B_{il} \\
N^M_{i,j} &= \sum_{k=i+m+1}^{j-m-1} N^M_{i,k-1} \, N^{M1}_{k,j} + \sum_{k=i}^{j-m-1} N^{M1}_{k,j} \\
N^A_{i,j} &= \sum_{l=i+m+1}^{j} N^B_{il} \\
N_{i,j} &= 1 + N^A_{i,j} + \sum_{k=i+1}^{j-m-1} N_{i,k-1} \, N^A_{k,j}
\end{aligned}
$$

Table 1: **Recursion for the enumeration of secondary structure graphs:** The number N^B_{ij} 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 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 N^M and N^{M1} are necessary for handling the multi-loops (McCaskill 1990), N^A helps reducing the CPU requirements. The unconstrained number of structures of the substring $[i, j]$ is stored in N_{ij}. The first term accounts for the unpaired structure. The second term collects all structures that consist of a single component, possibly with an unpaired "tail" at the 3' end. The final term arises from the formal construction of multi-component structures from a 1-component part at the 3' side and an arbitrary structure at the 5' side.

5 The Energy Model

5.1 Base-Base Interactions in Nucleic Acids

Base-base interactions in nucleic acids are of two kinds: (a) base pairing in the plane of the bases (horizontal) due to hydrogen bonding and (b) base stacking perpendicular to the plane of bases stabilized by London dispersion forces and hydrophobic effects (Saenger 1984; Poerschke 1977). Whilst hydrogen bonding is fundamental to the genetic code, both kinds of interactions play a significant role in determining the spatial structure and energy state of an RNA molecule.

5.1.1 Hydrogen Bonding

Hydrogen bonds (Schuster 1987) are mainly electrostatic in character. A hydrogen bond X–H$\cdots$Y is formed when a hydrogen atom H is situated between two atoms X, Y of higher electronegativity. The strength of the hydrogen bond is determined by the partial charges located on X and Y. In the case of base-base interactions, the hydrogen bonding involved is of type N–H$\cdots$O and N–H$\cdots$N, with the donor N–H group of either the amino or imino type.

Compared with covalent bonds, hydrogen bonds are weaker and do not show well-defined length and orientation. Modification of the charges on the involved atoms in a hydrogen bond due to polarisation lead to additivity and cooperativity of the bond forming process: H becomes more electropositive, X,Y more negative. The thus increased affinity of X,Y for accepting further hydrogen bonds facilitates the forming of (at least) a second hydrogen bond. With bases A,C,G and U ten combinations of purine-pyrimidine base pairs involving at least two hydrogen bonds are possible, see Figure 11. *Watson-Crick*, *Reverse Watson-Crick*, *Hoogssteen* and *Reverse Hoogsteen* A-U pairs differ in relative orientation of the bases and in selection of the binding sites. In apolar solvents, a mixture of Watson-Crick and Hoogsteen base pairs are formed with at least two hydrogen bonds, involving all potential binding sites. Association constants depend greatly on the chemical nature of the two bases: Modification of bases leads to different association constants. Thermodynamic

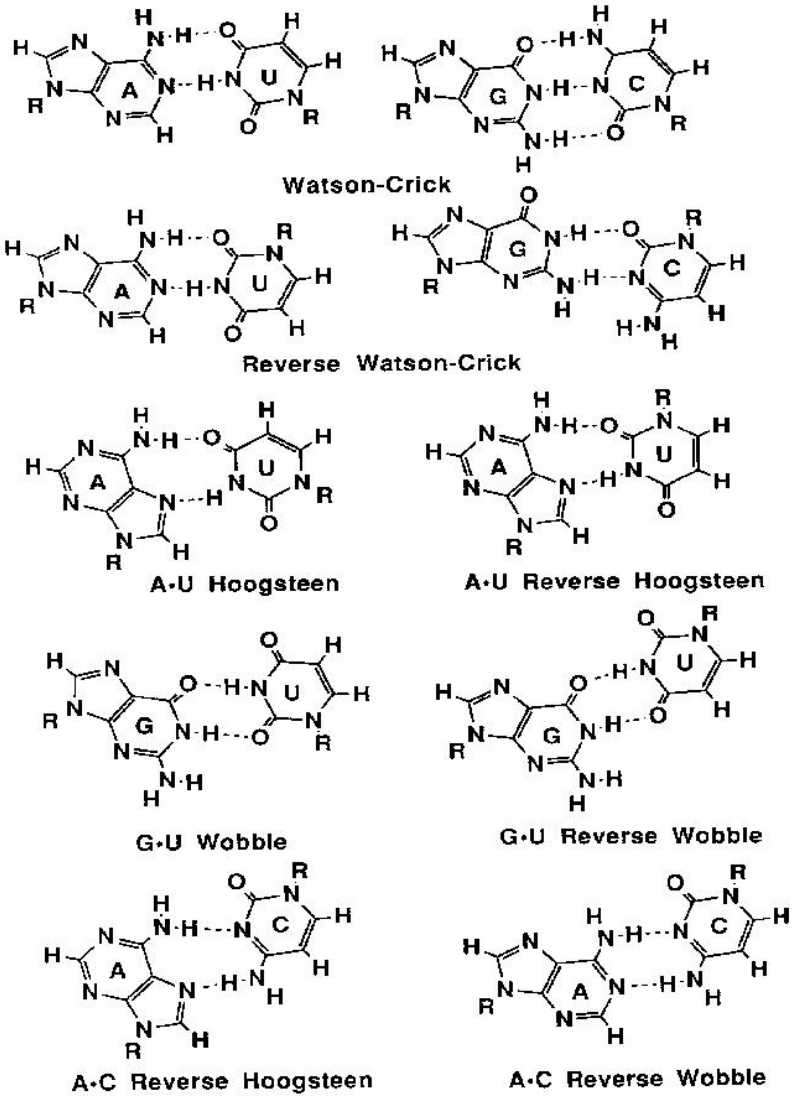


Figure 11: The ten possible purine-pyrimidine base pairs (Saenger 1984; Tinoco 1993).

investigations have shown that complementary A-U and G-C bases are more stable than self-associates. Watson-Crick, Reversed Watson-Crick, Hoogsteen and Reversed Hoogsteen base pairs cannot be differentiated, so that all thermodynamic data for A-U and G-C refer to a combination of base pair types (Saenger 1984). Quantum chemical studies have demonstrated that *electronic complementarity* is most important for the stability of base pairs, a term referring to the intrinsic electronic structures of associating bases and not merely

to the number of hydrogen bonds (Saenger 1984): Relative energy values for different base pairs suggest that complementary pairs in the Watson-Crick sense are more stable than the self-associates of the individual components. All non-complementary base pairs (such as A-G, G-U) are less stable than the corresponding self-associated pairs.

5.1.2 Vertical Base-Base Stacking

In addition to the horizontal base-base interactions due to hydrogen bonding described above, vertical stacking of bases such that one base plane is at the *van der Waals* distance ($\sim 3.4\,\text{Å}$) and parallel to the adjacent base plane, is observed in aqueous solution and in the solid state (Saenger 1984). This interaction strongly influences the stability of nucleic acid secondary structure (Poerschke 1977). Association and stacking of bases in aqueous solution goes beyond the dimeric state and follows *isodesmic* behavior: The addition of one base to another or to an existing stack is reversible with a constant free energy increment for each step and thus additive; each addition step is independent and displays the same thermodynamic and kinetic parameters, see Figure 12. Thermodynamic parameters for the self-association (stacking) of nucleosides

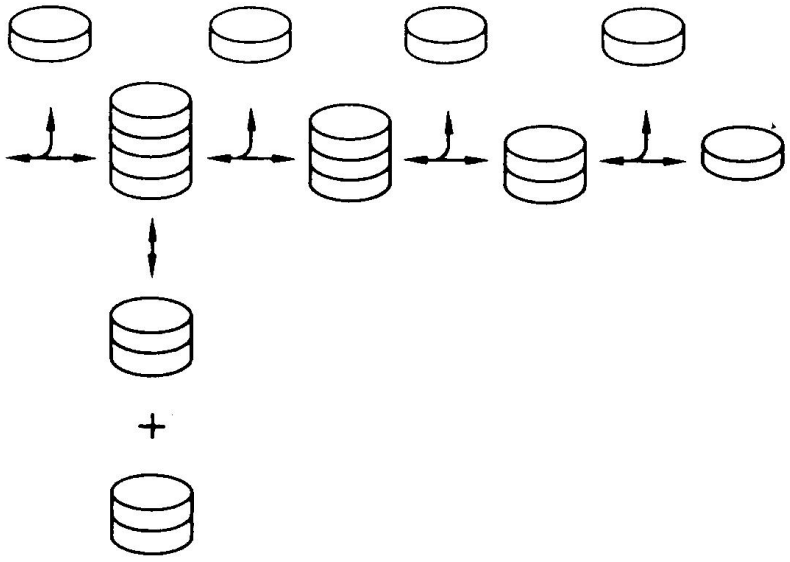


Figure 12: Reaction Scheme of base stacking (Saenger 1984; Poerschke 1977).

and bases in aqueous solution indicate that (a) association (reaction) constants K are characteristic for weak interactions, (b) both enthalpies ΔH and ΔS are negative, (c) Gibbs Free Energy change ΔG is negative in the order of thermal energy $kT = 0.6\,\text{kcal/mol}$. Methylation of bases in general leads to a moderate increase of stacking. Stability of stacks greatly depends on the chemical nature of the bases; purine-purine stacks are most stable, followed by pyrimidine-purine and pyrimidine-pyrimidine stacks.

Bases linked together to oligonucleotides or polynucleotides in aqueous solution form single-stranded, helical structures due to stacking interactions between adjacent bases, see Figure 13. Their stabilities exhibit the same dependence on the character of the stacking bases with *poly*A chains forming stable helices and *poly*U forming random coils at room temperature. Again methylation gives rise to increased stability, indicated by higher melting temperature T_m at higher degree of methylation. Investigations on oligomers of different chain length suggest that the formation of the single stranded structure is again noncooperative (Poerschke 1977).

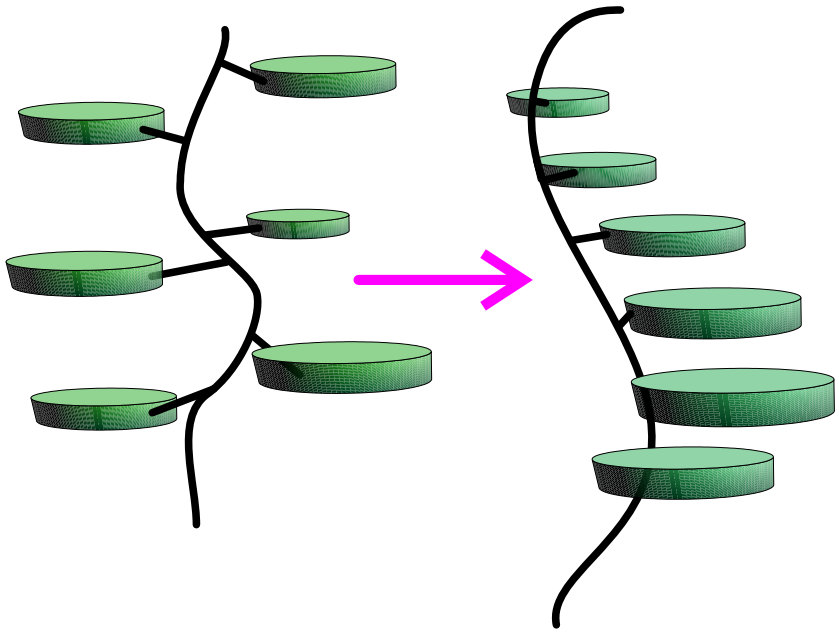


Figure 13: Base stacking to *poly*A single stranded helices (Saenger 1984; Poerschke 1977).

Forces mainly contributing to the stabilisation of base stacking in aqueous solution are dipolar and *London dispersion* forces in combination with hydrophobic forces due to an overall gain in entropy during the association process: Bases dissolved in water adopt a *hydration sphere* with the distribution of water structures within this sphere shifted into a state with more-ordered H_2O molecules. Association of bases results in the reduction of their surface exposed to water and thus in the reduction of the higher-order hydration sphere (and increase of entropy). Albeit, hydrophobic interactions cannot explain the stacking specificity, see above. These sequence determined properties are due to dipolar and London dispersion forces, which depend mainly on permanent dipoles and polarizability of the interacting molecules. Both effects are more pronounced in purine than in pyrimidine bases.

Quantum chemical calculations were employed to estimate the total stabilizing energy of base paired stacking dimers as $^{5'}$C-G$^{3'}$ $^{3'}$G-C$^{5'}$. Due to the restrictions of the model (molecules *in vacuo*), the base pairing components of the total energy appear to be larger than the stacking components. In aqueous solutions, however, hydrophobic interactions have to be taken into account. Melting experiments on *oligo*A-*oligo*U double helices show that with increasing chain length (a) T_m increases and (b) the slope at the point of inflection (T_m) becomes steeper due to enhanced cooperativity, thus suggesting a *two-state* model (helix - coil). Melting temperatures of double-helical nucleic acids increase also with the G-C/A-U ratio of the polynucleotide. Because of this dependence of melting behaviour on nucleotide composition, in a double helical nucleic acid with random base sequence, A-U rich regions should melt at lower temperatures than G-C rich regions. The resulting *local breakdown* of the helical order leads to broader spectra of the relaxation process. Analysis of melting profiles yields different melting points for individual regions of distinct base composition. From these melting information, stability parameters for individual base pairs can be derived.

5.2 Thermodynamic Nearest Neighbor Parameters

The results of both quantum chemical calculations and thermodynamic measurements suggest that horizontal (base pairing) contributions to the total energy depend exclusively on the base pair composition, whereas vertical (base stacking) contributions depend on base pair composition *and* base sequence i.e. the upstream and downstream neighbors along the chain (Saenger 1984). The *nearest neighbor model* introduces the assumption that the stabiltity of a base pair, or any other structural element of an RNA, is dependent 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 special types of loops, one assumes therefore that the energy of an RNA secondary structure Φ is given by the sum of energy contributions ϵ of it's loops L.

$$E(\Phi) = \sum_{L \in \Phi} \epsilon(L) + \epsilon(L_{ext}), \tag{27}$$

where L_{ext} is the contribution of the "exterior" loop containing the free ends. Note that here stacked pairs are treated as minimal loops of degree 2 and size 0. In the following we shall discuss the individual contributions in some detail.

In particular, the energy model contains the following contributions (Turner, Sugimoto, & Freier 1988):

Stacked pairs and G-U mismatches 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 (He *et al.* 1991) 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 $\begin{smallmatrix}5'\text{G-U}\,3'\\3'\text{U-G}\,5'\end{smallmatrix}$ stacked pair seems to vary form +1.5 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. (Jaeger, Turner, & Zuker 1989) differ widely from the values given previously (Freier *et al.* 1986). Energies depend only on the size and type (hairpin, interior or bulge) of the the loop. Hairpins must have a minimal size of 3, and values for large loops ($k > 30$) are extrapolated logarithmically:

$$\mathcal{H}(k) = \mathcal{H}(30) + \text{const.} * \log(k/30) \tag{28}$$

Asymmetric interior loops are furthermore penalized (Papanicolau, Gouy, & Ninio 1984), using an empirical formula depending on the difference $|u_1 - u_2|$ of unpaired bases on each side of the loop.

$$\Delta F_{\text{ninio}} = \min\left\{\Delta F_{\max}, |u_1 - u_2| * \Delta F_{\text{ninio}}\left[\min\{4.0, u_1, u_2\}\right]\right\} \tag{29}$$

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 (Antao & Tinoco 1992) 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

$$\Delta G = a + bu + cm, \tag{30}$$

where u is the size of the loop and m is the number of base pairs interior to the loop, i.e. its degree-1. Good results have been achieved using $a = 4.6$, $b = 0.4$ and $c = 0.1$ kcal/mol. While a logarithmic size dependency of loop energies would be more realistic, 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.

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 (Salser 1977). The parameters most widely in use today are based on work of D. Turner and coworkers . The current work uses the compilation of (Freier *et al.* 1986; Turner, Sugimoto, & Freier 1988; He *et al.* 1991), who performed measurements at 37°C in 1 M NaCl. More recently the differences between symmetric and asymmetric loops have been reported to be only half the magnitude suggested by Papanicolau *et. al.* (Papanicolau, Gouy, & Ninio 1984) and of higher sequence dependence (Peritz *et al.* 1991). Serra *et. al.* found a dependence of hairpin loop energies on the closing base pair (Serra *et al.* 1993) and presented a model to predict the stability of hairpin loops (Serra, Axenson, & Turner 1994). Walter and coworkers suggested a model system for the coaxial stacking of helices (Walter *et al.* 1994). Wu and Walter studied the stability of tandem GA mismatches and found them to depend upon both sequence and adjacent base pairs (Walter, Wu, & Turner 1994;

Wu, McDowell, & Turner 1995). Ebel and coworkers measured the thermodynamic stability of RNA duplexes containing tandem G-A mismatches (Ebel, Brown, & Lane 1994). 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 (Morse & Draper 1995). These results are, however, not yet included into the parameter set used in this work.

The energy contributions described above result in nearest neighbor parameters for the individual types of loops, thus constituting the energy model used in the present work. 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) *Multiloops* ($k \geq 2$): Multiloop energies $\mathcal{M}$ are modeled by the linear ansatz

$$\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot \text{degree} + \mathcal{M}_B \cdot \text{unpaired}, \tag{31}$$

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

Since the concept of dangling ends is not compatible with the definition of RNA secondary structure, energy paramenters reflecting the contributions of unpaired terminal nucleotides to the stability of an RNA are not passed to the energy model used in this work.

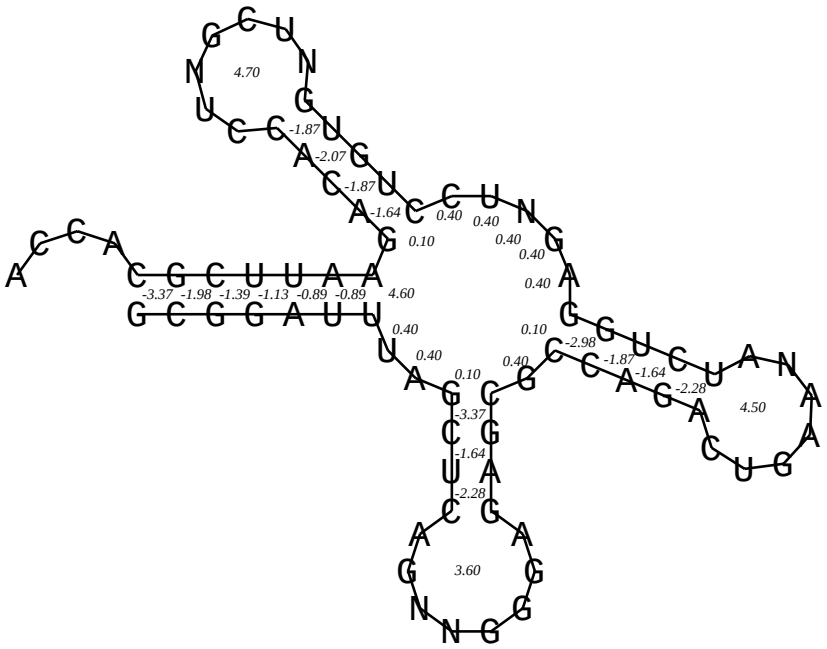


Figure 14: The secondary structure of Yeast tRNA$^{\mathrm{Phe}}$. The sequence ($n = 76$) is taken from the EBI database (Steegborn *et al.* 1995): `GCGGAUUUALCUCAGDDGGGAGA-GCRCCAGABU#AAYAP?UGGAG7UC?UGUGTPCG"UCCACAGAAUUCGCACCA.`

The Free Energy of the structure according to the energy model used in this work is -12.26 kcal/mole. Multiloop energies are $\mathcal{M}_C = 4.60$ kcal/mole, $\mathcal{M}_B = 0.40$ kcal/mole and $\mathcal{M}_I = 0.10$ kcal/mole. See the appendix for the abbreviation and translation of modified bases.

6 Dynamic Programming Scheme for the Density of States

In this section we will show that an algorithm for the computation of the complete density of states of an RNA (in terms of secondary structure) can be derived from the recursion scheme outlined in Section 4, see Table 1.

Within the energy model for RNA secondary structure graphs used in this work, the total energy of an RNA molecule is given by the sum of the *loop energies* of its structural elements, see Section 5. An expression for the number of structures compatible to a string $[\sigma_i \dots \sigma_j]$ with exactly energy ϵ can be derived as follows.

Remember the underlying decomposition of structures: The finite set $\mathcal{S}$ of all structures compatible to a string $[\sigma_i \dots \sigma_j]$ is split into subsets $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ such that $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 = \mathcal{S}$ and $\mathcal{S}_i \cap \mathcal{S}_j = 0$, see Figure 15. $\mathcal{S}_1$ contains the open chain; there is only one unpaired structure, with the number of structures in this subset being always 1. $\mathcal{S}_2$ is the collection of all single-component structures where the leftmost base σ_i forms the closing base pair with another base σ_l and all bases right of σ_l are unpaired. The number of structures in this subset is denoted $N^A_{i,j}$. $\mathcal{S}_3$ is the subset of all multi-component structures, consisting of at least two components, and of all single component structures with a tailing end at the left (5') side. This set is further split into subsets.

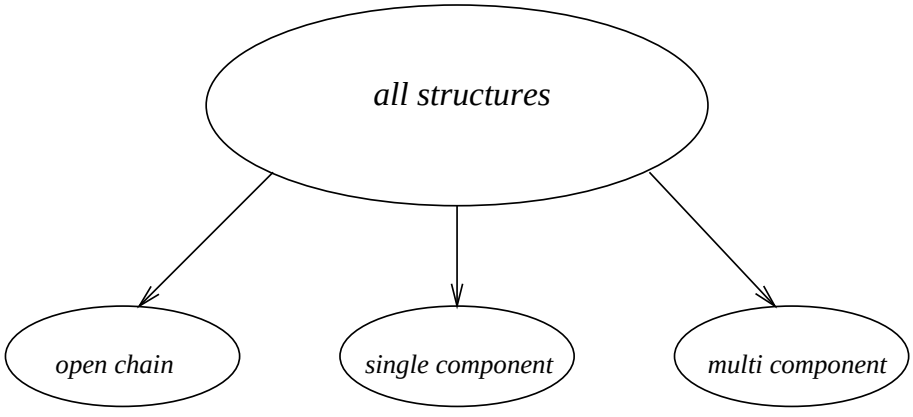


Figure 15: The complete set $\mathcal{S}$ of all secondary structures Φ compatible to string $[\sigma_i ... \sigma_j]$ is split into subsets $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$.

Each subset contains those structures which can be formally constructed from an arbitrary – even unpaired – structure at the left (5') side on a substring $[\sigma_i ... \sigma_{k-1}]$ and an single-component structure at the right side on a substring $[\sigma_k ... \sigma_j]$, where σ_k forms the closing base pair (k, l) of the component and all bases $> l$ are unpaired, see Figure 16. There is a subset for each value of k, with k running from $i+1$ to $j-m-1$. The number of structures in a set is equal to the product of the number of structures on the two substrings.

The reason for dividing the complete set of structures into distinct subsets by applying this relation is the opportunity to construct the number of structures on a larger string from the – earlier computed – number of structures on smaller strings, which is only possible for structures consisting of two or more independent components. This idea underlying the recursion scheme implies that, when dealing with energies, the total energy of the complete structure has to be given by the sum of the energies of the components. The number of structures with a certain energy is therefore derived by the number of structures on a substring $[\sigma_i \dots \sigma_{k-1}]$ with energy ϵ' and the number of structures on a substring $[\sigma_k \dots \sigma_j]$ with energy $(\epsilon - \epsilon')$. For a given vertex k this means

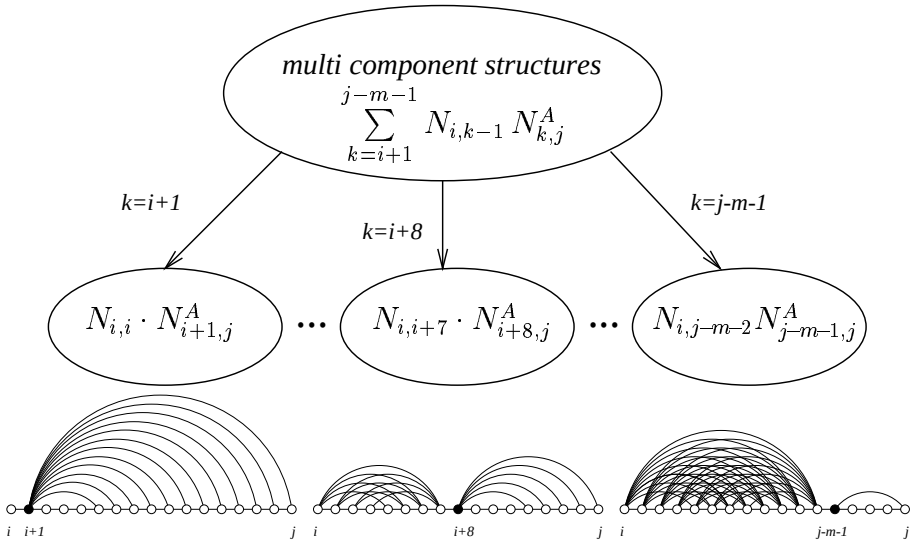


Figure 16: The set of all multi-component structures is split into subsets consisting of all secondary structures formed by an arbitrary structure on substring $[\sigma_i ... \sigma_{k-1}]$ and a single component on substring $[\sigma_k ... \sigma_j]$ with (k, l) enclosing the component.

that there is only a structure with energy ϵ on $[\sigma_i \ldots \sigma_j]$ consisting of two or more components, if there is a structure on $[\sigma_i \ldots \sigma_{k-1}]$ with exactly energy ϵ', and if there is at least one single-component structure on $[\sigma_k \ldots \sigma_j]$ with exactly energy (ϵ - ϵ'), see equ. (6).

Following the argument given above, let $N_{i,j}(\epsilon)$ be the number of structures compatible to a string $[\sigma_i \ldots \sigma_j]$ with exactly energy ϵ. $N_{i,j}(\epsilon)$ now can be written as

$$\begin{aligned} N_{ij}(\epsilon) &= \delta(0,\epsilon) \\ &\quad + N^A_{ij}(\epsilon) \\ &\quad + \sum_{k=i+1}^{j-m-1} \left[\sum_{\epsilon'=0}^{\epsilon} N_{i,k-1}(\epsilon')\, N^A_{k,j}(\epsilon - \epsilon') \right]. \end{aligned} \tag{32}$$

The first term referring to the open structure equals 1, if $\epsilon = 0$, and is 0 otherwise, because there is only a single open chain structure; its energy is 0 by definition. The second term counts all structures consisting of a single component. The energy of that components equals the energy of the total structure, for tailing ends do not contribute to the energy. The last term counts all structures consisting of at least two components; these structures are constructed from their components, see Figure 8. We have to take into account that the total energy is the sum of the energy of the components, and, therefore, and we have to consider all possible energy distributions between the individual components. Therefore an additional sum over the component energy ϵ' is introduced.

The number of all structures on substring $[\sigma_i \ldots \sigma_j]$ consisting of a single component is derived by

$$N^A_{i,j}(\epsilon) = \sum_{l=i+m+1}^{j} N^B_{i,l}(\epsilon). \tag{33}$$

$N^B_{i,j}(\epsilon)$ denotes the number of structures on substring $[\sigma_i \ldots \sigma_j]$ under the condition that i and j form a base pair. $N_{i,j}(\epsilon)$ can again be obtained recursively from smaller fragments:

$$N^B_{i,j}(\epsilon) = \delta(\mathcal{H}(i,j),\epsilon)$$

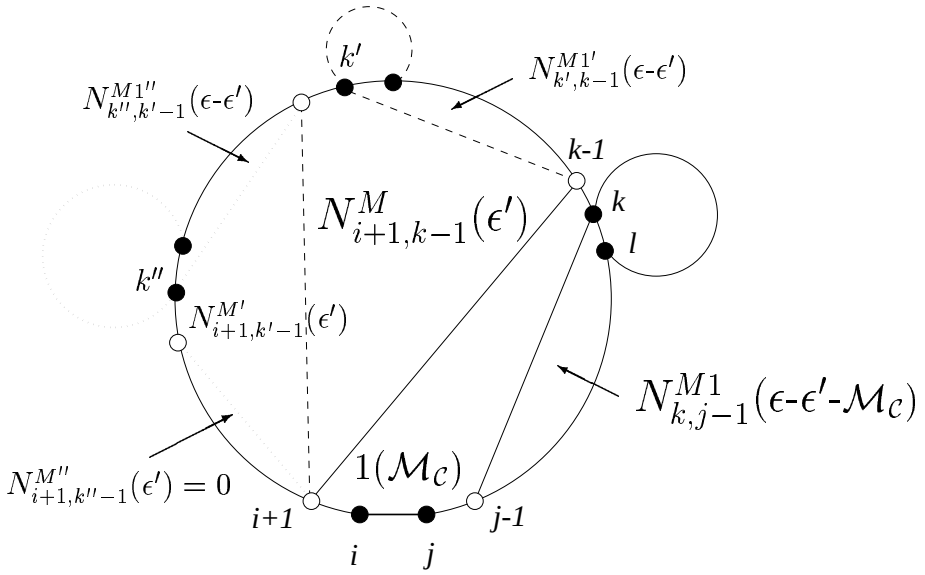


Figure 17: Recursive decomposition of multiloops and multiloop energies: Multiloop structures of energy ϵ are constructed from the closing base pair (i, j) with *multiloop closing energy* $\mathcal{M}_C$, a region running from k to $j-1$ with energy $(\epsilon\text{-}\epsilon'\text{-}\mathcal{M}_C)$ containing a single component with a possible tailing end at the right side, and a region running from $i+1$ to $k-1$ containing an arbitrary structure of energy ϵ'. Multiloop energy contributions are attributed to individual vertices or base pairs and are additive, see equ. (31), Figure 14.

$$+ \sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} N^B_{k,l}(\epsilon - \mathcal{I}(i,j,k,l))$$

$$+ \sum_{k=i+1}^{j-m-2} \left[\sum_{\epsilon'=0}^{\epsilon-\mathcal{M}_C} N^M_{i+1,k-1}(\epsilon')\, N^{M1}_{k,j-1}(\epsilon - \epsilon' - \mathcal{M}_C) \right] \qquad (34)$$

The first term represents the hairpin case; its contribution is 1, if the energy of the loop is exactly ϵ, and otherwise 0. If bases σ_i and σ_j do not pair, the hairpin energy $\mathcal{H}$ is infinite by definition; therefore, we do not have to weigh the contributions to N^B by Π_{σ_i,σ_j}. The second term counts all structures where base pair (i, j) closes an internal loop, a bulge or a stack. The number of these structures with energy ϵ is given by the number of structures enclosed by (k, l) with energy ϵ minus the energy of the loop $\mathcal{I}(i, j, k, l)$).

Following the scheme discussed in Section 4.7, the multiloop structures are constructed from three parts: The first part consists only of the base pair (i, j) closing the multiloop. The second part contains a region at the right side running from base k satisfying $i < k < j - m$ to base $(j - m - 1)$, where k is forming an interior base pair with another base l, satisfying $k+m+1 \leq l \leq j+1$ and $l + 1, ..., j - 1$ being unpaired. This region thus consists of a substructure enclosed by $(k.l)$ and, if any, unpaired bases between l and j. The number of structures deriving from this region is denoted $N_{ij}^{M1}(\epsilon)$. The third region running from $(i + 1)$ to $(k - 1)$ contains at least 1 base pair immediately interior to the closing pair. The number of structures contained by this region is denoted $N_{ij}^{M}(\epsilon)$. Since the multiloop structures are again constructed from independent parts, the total energy of a multiloop structure equals the sum of the energy of these parts. This is only possible, if the energy model for a multiloop follows a linear ansatz similar to equ. (31). An additional sum over energy ϵ' is introduced.

If the rightmost multiloop region contains more than one stem, it is further decomposed into independent components $M1$ and M. The total number of structures is the product of the number of structures N^{M1} and N^{M} on the two substrings, see Figure 17. The total energy ϵ is the sum of the energies ϵ' and $\epsilon - \epsilon'$ of the two substructures.

$$N_{ij}^{M1}(\epsilon) = \sum_{l=i+m+1}^{j} N_{il}^{B}(\epsilon - \mathcal{M}_{\mathcal{B}} \cdot (j - l) - \mathcal{M}_{\mathcal{I}}) \tag{35}$$

$$\begin{aligned} N_{ij}^{M}(\epsilon) &= + \sum_{k=i+m+1}^{j-m-1} \left[\sum_{\epsilon'} N_{i,k-1}^{M}(\epsilon')\, N_{k,j}^{M1}(\epsilon - \epsilon') \right] \\ &\quad + \sum_{k=i}^{j-m-1} N_{kj}^{M1}(\epsilon - \mathcal{M}_{\mathcal{B}} \cdot (k - i)) \end{aligned} \tag{36}$$

The second term in equ. (36) is nonzero when the leftmost region N^{M} does not contain any stem and is thus 0: The number of multiloop substructures N^{M} formed of unpaired bases is 0 by definition. The energy contribution of that

region, $(\mathcal{M}_{\mathcal{B}} \cdot fk - i))$, depends on the number of unpaired bases $(k - i)$ and is constant for a given substring.

Table 2 summarizes the recursion scheme for the density of states. The next section will extend the recursion scheme to the computation of the partition function.

$$
\begin{aligned}
N^B_{ij}(\epsilon) &= \delta(\mathcal{H}(i,j),\epsilon) + \sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} N^B_{kl}(\epsilon - \mathcal{I}(i,j,k,l)) \\
&\quad + \sum_{k=i+1}^{j-m-2} \left[\sum_{\epsilon'}^{\epsilon-\mathcal{M}_C} N^M_{i+1,k-1}(\epsilon')\, N^{M1}_{k,j-1}(\epsilon - \epsilon' - \mathcal{M}_C) \right] \\
N^{M1}_{ij}(\epsilon) &= \sum_{l=i+m+1}^{j} N^B_{il}(\epsilon - \mathcal{M}_\mathcal{B}(j-l) - \mathcal{M}_\mathcal{I}) \\
N^M_{ij}(\epsilon) &= \sum_{k=i+m+1}^{j-m-1} \left[\sum_{\epsilon'} N^M_{i,k-1}(\epsilon')\, N^{M1}_{k,j}(\epsilon - \epsilon') \right] \\
&\quad + \sum_{k=i}^{j-m-1} N^{M1}_{kj}(\epsilon - \mathcal{M}_\mathcal{B}(k-i)) \\
N^A_{ij}(\epsilon) &= \sum_{l=i+m+1}^{j} N^B_{il}(\epsilon) \\
N_{ij}(\epsilon) &= \delta(0,\epsilon) + N^A_{ij}(\epsilon) \\
&\quad + \sum_{k=i+1}^{j-m-1} \left[\sum_{\epsilon'} N_{i,k-1}(\epsilon') N^A_{k,j}(\epsilon - \epsilon') \right]
\end{aligned}
$$

Table 2: **Recursion for the calculation of the density of states:** 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 $\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot \text{degree} + \mathcal{M}_\text{B} \cdot \text{unpaired}$, e.g. (Zuker & Sankoff 1984). The number $N^B_{ij}(\epsilon)$ of substructures on the substring $[i,j]$ with energy ϵ 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 N^M and N^{M1} are necessary for handling the multi-loops (McCaskill 1990), N^A helps reducing the CPU requirements. The unconstrained d.o.s. of the substring $[i,j]$ is stored in $N_{ij}(\epsilon)$. The first term accounts for the unpaired structure. The second term collects all structures that consist of a single component, possibly with an unpaired "tail" at the 3' end. The final term arises from the formal construction of multi-component structures from a 1-component part at the 3' side and an arbitrary structure at the 5' side.

7 Partition Function

In this section we will present the algorithm for the computation of the partition function of an RNA molecule first derived by McCaskill (McCaskill 1990). It will be shown that the algorithm follows the same general recursion scheme as described in the previeous sections.

The Free Energy F is related to the *partition function* Q by

$$F = -kT \ln Q, \tag{37}$$

where Q is the partition function, T is the temperature and k is the Boltzmann factor.

The partition function of a given RNA molecule is

$$Q = \sum_{\Phi \in \mathcal{M}} e^{-F(\Phi)/kT}, \tag{38}$$

where $\mathcal{M}$ is the set of all secondary structures Φ compatible to the nucleotide sequence.

The additivity of free energy contribution of the various loops L of a structure Φ, see equ. (27), implies a multiplicativity in the partition function Q.

$$Q = \sum_{\Phi \in \mathcal{M}} e^{-\left[\sum_{L \in \Phi} F_L\right]/kT} \tag{39}$$

$$= \sum_{\Phi \in \mathcal{M}} \prod_{L \in \Phi} e^{-F_L/kT} \tag{40}$$

Decomposing an individual secondary structure Φ into its components, $S_1 \ldots S_n$, leads to an expression emphasizing that every loop is contained in one of the components and that the contribution of the structure to the partition function can be derived from the product of the contributions of its components.

$$Q = \sum_{\Phi \in \mathcal{M}} \prod_{\mathcal{S} \in \Phi} \prod_{L \in \mathcal{S}} e^{-F_L/kT} \tag{41}$$

$$= \sum_{\Phi \in \mathcal{M}} \prod_{\mathcal{S} \in \Phi} e^{-F_S/kT} \tag{42}$$

This multiplicativity of the partition function contributions in terms of components (and loops) parallels the multiplicativity of the number of structures, see previous sections. Therefore, the complete partition function of an RNA molecule can be derived by following the recursion scheme presented for the density of states.

In complete analogy with section 6 we split the set $\mathcal{M}$ of all structures into three subsets. The first subset contains the open structure, the second all structures consisting of a single component with an arbitrary number of unpaired bases at the right side. The third subset contains all structures consisting of more than one component, see Figure 15. The complete partition function $Q_{i,j}$ on the string $[\sigma_i ... \sigma_j]$ is the sum of the contributions of the three subsets:

$$Q_{i,j} = Q_{i,j}(\mathcal{S}_1) + Q_{i,j}(\mathcal{S}_2) + Q_{i,j}(\mathcal{S}_3) \tag{43}$$

The first term is always 1, because the energy of the open structure is 0 by definition and $e^0 = 1$. The second term is the sum of the contributions of all structures in subset $\mathcal{S}_2$. Their contribution is denoted $Q^A_{i,j}$. Again in

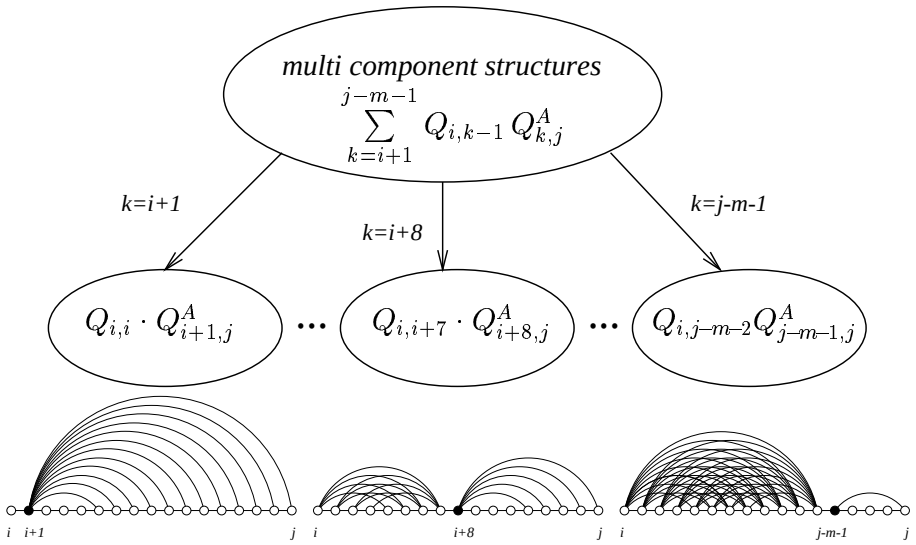


Figure 18: The contribution of each set is derived by the multiplication of the unconstrained partition function on the left substring times the contribution of all single-component structures on the right string. Summing up yields the total contribution of all multi-component structures.

analogy with section 6, the set of multicomponent structures is recursively split into subsets consisting of all structures formed by an arbitrary structure on substring $[\sigma_i ... \sigma_{k-1}]$ and a single component on substring $[\sigma_k ... \sigma_j]$. The contribution of all structures contained in a certain subset is derived by the product of the contributions $Q_{i,k-1}$ of *all* structures on substring $[\sigma_i ... \sigma_{k-1}]$ and the contributions $Q^A_{k,j}$ of all single-component structures on $[\sigma_k ... \sigma_j]$. The contribution of all multicomponent structures is the sum of the contributions of all subsets, see Figure 18. Therefore, we receive for the complete partition function

$$Q_{i,j} = 1.0 + Q^A_{i,j} + \sum_{k=i+1}^{j-m-1} Q_{i,k-1} Q^A_{k,j}. \tag{44}$$

The contribution to the partition function of all single-component structures, $Q^A_{i,j}$, is received by summing up all contributions $Q^B_{i,j}$ of all structures which contain a base pair (i, j).

$$Q^A_{i,j} = \sum_{l=i+m+1}^{j} Q^B_{i,l} \tag{45}$$

Hence Q^B_{ij} is the partition function of the segment S_{ij}, given that σ_i and σ_j pair, i. e. that $(i, j) \in \Phi_{i,j}$. Q^B_{ij} can be written as a recursive formula

$$Q^B_{ij} = \sum_L e^{-F_L/kT} \prod_{\substack{(h,l)\in L \\ i<h<l<j}} Q^B_{hl} \tag{46}$$

where the sum runs over all possible loops closed by (i, j). If L is a hairpin loop, there is no pair $(h, l) \in L$; if L is an interior loop or a bulge, there is exactly one pair $(h, l) \in L$. But if L is multiloop, then there are n pairs $(h, l) \in l$ with $i < h_1 < l_1 < \cdots < h_n < l_n < j$. Clearly no base can pair with itself, therefore the initial condition of the above recursion formula is $Q^B_{ii} = 0$.

Dividing Q^B_{ij} into the contribution coming from the different loop forms, equation (46) can be rewritten as

$$Q^B_{i,j} = e^{-\mathcal{H}(i,j)/kT} + \sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} Q^B_{k,l}\, e^{-\mathcal{I}(i,j,k,l)]/kT} \tag{47}$$

$$+ \sum_{k=i+1}^{j-m-2} Q^M_{i+1,k-1} Q^{M1}_{k,j-1}\, e^{-\mathcal{M}_C/kT}. \tag{48}$$

Calligraphic symbols $\mathcal{H}, \mathcal{I}, \mathcal{M}$ refer to the classification of loops described in the previous sections according to the value of k ($k = 0 \rightarrow$ hairpin loop, $k = 1 \rightarrow$ stack, interior loop, bulge). The third term in equation (48) represents the multiple loop contribution, derived in analogy to equations (36) and (36), see Figure 17. We obtain for the multiloop contributions

$$Q^M_{ij} = \sum_{k=i+m+1}^{j-m-1} Q^M_{i,k-1}\, Q^{M1}_{k,j} + \sum_{k=i}^{j-m-1} Q^{M1}_{k,j}\, e^{-\mathcal{M}_{\mathcal{B}}(k-i)/kT} \tag{49}$$

with $Q^M_{ii} = 0$ and $Q^M_{i+1,i} = 0$. The contribution of all structures forming a single rightmost stem, $Q^{M1}_{k,j}$, is obtained to

$$Q^{M1}_{i,j} = \sum_{l=i+m+1}^{j} Q^B_{i,l}\, e^{-[\mathcal{M}_{\mathcal{I}} + \mathcal{M}_{\mathcal{B}}(j-l)]/kT} \tag{50}$$

Table 3 summarizes the recursion scheme for the partition function. The next section will extend the recursion scheme to the computation of the minimum free energy.

$$
\begin{aligned}
Q^B_{i,j} &= e^{-\mathcal{H}(i,j)/kT} + \sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} Q^B_{k,l}\, e^{-\mathcal{I}(i,j,k,l)]/kT} \\
&\quad + \sum_{k=i+1}^{j-m-2} Q^M_{i+1,k-1} Q^{M1}_{k,j-1}\, e^{-\mathcal{M}_C/kT} \\
Q^{M1}_{i,j} &= \sum_{l=i+m+1}^{j} Q^B_{i,l}\, e^{-[\mathcal{M}_I + \mathcal{M}_B(j-l)]/kT} \\
Q^M_{i,j} &= \sum_{k=i+m+1}^{j-m-1} Q^M_{i,k-1}\, Q^{M1}_{k,j} \\
&\quad + \sum_{k=i}^{j-m-1} Q^{M1}_{k,j}\, e^{-\mathcal{M}_B(k-i)/kT} \\
Q^A_{i,j} &= \sum_{l=i+m+1}^{j} Q^B_{i,l} \\
Q_{i,j} &= 1.0 + Q^A_{i,j} + \sum_{k=i+1}^{j-m-1} Q_{i,k-1} Q^A_{k,j}
\end{aligned}
$$

Table 3: **Recursion for the calculation of the partition function:** 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 $\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot \text{degree} + \mathcal{M}_\text{B} \cdot \text{unpaired}$, e.g. (Zuker & Sankoff 1984). The partition function Q^B_{ij} 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 Q^M and Q^{M1} are necessary for handling the multi-loops (McCaskill 1990), Q^A helps reducing the CPU requirements. The unconstrained partition function of the substring $[i,j]$ is stored in Q_{ij}. The first term accounts for the unpaired structure. The second term collects all structures that consist of a single component, possibly with an unpaired "tail" at the 3' end. The final term arises from the formal construction of multi-component structures from a 1-component part at the 3' side and an arbitrary structure at the 5' side.

8 Minimum Free Energy

The algorithm used in the previous sections to compute the partition function and the density of states of an RNA molecule can also be used to obtain the *minimum free energy* of the RNA, i.e. the free energy of the ground state secondary structure (Zuker & Stiegler 1981; Zuker & Sankoff 1984; Hofacker *et al.* 1994b). The minimum free energy algorithm relys on the same mechanisms and displays the same CPU requirements: (a) The complete set of all structures is (recursively) split into subsets of single-component and multi-component structures and (b) multicomponent structures are formally constructed from smaller fragments. Therefore, the algorithm implements dynamic programming; earlyer computed values for substrings yield values for larger strings, thus reducing CPU requirements to $\mathcal{O}(n^2)$.

Let $F^A_{i,j}$ denote the minimum free energy of all *single-component* structures on string $[\sigma_i ... \sigma_j]$ with (i,l) forming the closing pair and bases $> l$ being unpaired. The minimum free energy $F_{i,j}$ of *all* structures on string $[\sigma_i ... \sigma_j]$ then is

$$F_{i,j} = \min_{k \in [i+1, j-m-1]} \Big\{ 0,\ F^A_{i,j},\ \big[F_{i,k-1} + F_{k,j} \big] \Big\}. \tag{51}$$

The first element, 0, is the free energy of the open chain. The second element is the minimum energy of all single-component structures, see above. All following elements, $\{F_{i,k-1} + F_{k,j} \mid i+1 \le k \le j-m-1\}$, are the minimum free energies of a distinct subset of all multi-component structures, see Figures 16 and 18. Multi-component structures are constructed from smaller fragments, i. e. from a arbitrary structure on substring $[\sigma_i ... \sigma_{k-1}]$ and a single-component structure on substring $[\sigma_k ... \sigma_j]$, thus the minimum free energy of the complete structure equals the sum of the minimum energies of it's components.

In analogy to equ. (20), $F^A_{i,j}$ is obtained from the minimum of all minimum energies of all structures on $[\sigma_i ... \sigma_j]$, which have a closing pair (i,l):

$$F^A_{i,j} = \min_{l \in [i+m+1, j]} \Big\{ F^B_{i,l} \Big\} \tag{52}$$

$F^B_{i,j}$ then is the minimum free energy of all structures on $[\sigma_i ... \sigma_j]$, which are enclosed by (i,j), i. e. $(i,j) \in \Phi_{i,j}$. Three subsets are contributing to this set

of structures, depending on the number of base pairs immediatly interior to (i,j), see equ. (23). The minimum energies of these three subsets are again (recursively) obtained from smaller fragments:

$$F^B_{i,j} = \min\Big\{\mathcal{H}(i,j), \min_{\substack{k\in[i+1,j-m-2]\\ l\in[k+m+1,j-1]}}\big\{F^B_{k,l}+\mathcal{I}(i,j,k,l)\big\}, \min_{k\in[i+1,j-m-2]}\big\{F^M_{i+1,k-1}+F^{M1}_{k,j-1}+\mathcal{M}_{\mathcal{C}}\big\}\Big\} \quad (53)$$

$\mathcal{H}(i,j)$ 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) closes an interior loop; their minimum energy equals the sum of the minimum energy of the smaller fragment, $F^B_{k,l}$, and the energy of the closing loop, $\mathcal{I}(i,j,k,l)$. Multiloop structures enclosed by (i,j) are obtained by constructing the multiloop from two sections, see Figure 10. The minimum free energy is thus the sum of the minimum energy of the two parts, $F^M_{i+1,k-1}$ and $F^{M1}_{k,j-1}$, plus the multiloop closing energy $\mathcal{M}_{\mathcal{C}}$. $F^{M1}_{i,j}$ denotes the minimum free energy of the rightmost stem plus an arbitry number of unpaired bases at the right side and is obtained from the sum of the minimum energy of the stem, $F^B_{i,l}$, the multiloop base energy, $\mathcal{M}_{\mathcal{B}}(j-l)$, which is added for each unpaired base, and the multiloop internal energy, $\mathcal{M}_{\mathcal{I}}$.

$$F^{M1}_{i,j} = \min_{l\in[i+m+1,j]}\Big\{F^B_{i,l}+\mathcal{M}_{\mathcal{B}}(j-l)+\mathcal{M}_{\mathcal{I}}\Big\} \quad (54)$$

$F^M_{i+1,k-1}$, equ. (53), denotes the minimum free energy of the remaining section of a multiloop structure, see Figure 10. This section may contain one ore more stems. In analogy with equ. (25), we derive for the minimum free energy

$$F^M_{i,j} = \min\Big\{\min_{k\in[i+m+1,j-m-1]}\big\{F^M_{i,k-1}+F^{M1}_{k,j}\big\}, \quad (55)$$

$$\min_{k\in[i,j-m-1]}\big\{F^{M1}_{k,j}+\mathcal{M}_{\mathcal{B}}(k-i)\big\}\Big\}. \quad (56)$$

The first element yields the minimum energy of all multiloop sections, which can themselves be constructed from one part containing the rightmost stem a remaining part consisting of at least one stem at the left side. The energy is

the sum of the energy of the two components. The second element yields the minimum free energy of multiloop substructres, which consist only of a single remaining stem. These structures are constructed only from the stem plus unpaired bases at both sides. The energy of the structure is abtained from the sum of the minimum energy of the stem plus the bases at the right side, $F_{k,j}^{M1}$, see equ. (54), plus the energy of the unpaired bases at the left side of the stem, $\mathcal{M}_{\mathcal{B}}(k-i)$.

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

$$
\begin{aligned}
F^B_{i,j} &= \min\Big\{\mathcal{H}(i,j),\ \min_{\substack{k\in[i+1,j-m-2]\\ l\in[k+m+1,j-1]}}\big\{F^B_{k,l}+\mathcal{I}(i,j,k,l)\big\},\\
&\qquad \min_{k\in[i+1,j-m-2]}\big\{F^M_{i+1,k-1}+F^{M1}_{k,j-1}+\mathcal{M}_{\mathcal{C}}\big\}\Big\}\\
F^{M1}_{i,j} &= \min_{l\in[i+m+1,j]}\Big\{F^B_{i,l}+\mathcal{M}_{\mathcal{B}}(j-l)+\mathcal{M}_{\mathcal{I}}\Big\}\\
F^M_{i,j} &= \min\Big\{\min_{k\in[i+m+1,j-m-1]}\big\{F^M_{i,k-1}+F^{M1}_{k,j}\big\},\\
&\qquad \min_{k\in[i,j-m-1]}\big\{F^{M1}_{k,j}+\mathcal{M}_{\mathcal{B}}(k-i)\big\}\Big\}\\
F^A_{i,j} &= \min_{l\in[i+m+1,j]}\Big\{F^B_{i,l}\Big\}\\
F_{i,j} &= \min_{k\in[i+1,j-m-1]}\Big\{0,\ F^A_{i,j},\ \big[F_{i,k-1}+F_{k,j}\big]\Big\}
\end{aligned}
$$

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 $\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot \text{degree} + \mathcal{M}_{\text{B}} \cdot \text{unpaired}$, e.g. (Zuker & Sankoff 1984). The minimum free energy F^B_{ij} 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 (McCaskill 1990), F^A helps reducing the CPU requirements. The unconstrained minimum free energy of the substring $[i,j]$ is stored in F_{ij}. The first term accounts for the unpaired structure. The second term collects all structures that consist of a single component, possibly with an unpaired "tail" at the 3' end. The final term arises from the formal construction of multi-component structures from a 1-component part at the 3' side and an arbitrary structure at the 5' side.

9 Implementation of the Algorithms

9.1 The Vienna RNA Package

Implementations of the algorithms described in sections 7 and 8 are available within the Vienna RNA Package (Hofacker *et al.* 1994b; Hofacker 1994). The package provides both stand-alone programs for folding and comparing of secondary structures as well as a library to link with other C programs. It can be obtained via anonymous ftp from `www.tbi.univie.ac.at`, (Hofacker *et al.* 1994a).

9.2 Density of States

The algorithm for the computation of the density of states of RNA molecules presented in section 6 was implemented as an interactive programm running on UNIX workstations. All code was written in ANSI C. Table 5 shows an interactive example run of RNAdos, and table 6 provides a pseudocode.

In complete analogy to the dynamic programming solution of the minimum free energy problem (Waterman 1978; Waterman & Smith 1978; Nussinov *et al.* 1978; Nussinov & Jacobson 1980; Hofacker *et al.* 1994b), the additive form of the energy model, Section 5, allows for an dynamic programming algorithm for the density of states of secondary structures, Section 6. The algorithm described in this work is essentially an extension of the algorithm for the computation of the partition function by McCaskill (McCaskill 1990). The algorithm works by calculating the density of states of all $(j-i)^2$ substrings, proceeding from smaller to larger fragments: Values of $N_{i,j}(\epsilon)$, $N^B_{i,j}(\epsilon)$, $N^M_{i,j}(\epsilon)$, $N^{M1}_{i,j}(\epsilon)$, and $N^A_{i,j}(\epsilon)$, see Table 2, are calculated from values computed before, see Figure 19. The algorithm uses integer arithmetic, since integral numbers of states are computed. The triangular matrices are stored in columns, each entry containing a vector of length m, where m is the number of energy bins used for storing the number of states in an energy interval. Memory requirements, therefore, are generally high and scale as $\mathcal{O}(n^2 m)$.

The algorithm is rather demanding both in terms of memory and CPU

time: While it is possible to reduce the CPU requirements of the minimum free energy algorithm, see Table 4, from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$ by restricting the size of interior loops to a constant maximum value (Hofacker 1994), execution time of the density of states algorithm remains high: For each of the $\mathcal{O}(n^2)$ subsequences one needs to compute $\mathcal{O}(nm)$ convolutions which in turn require $\mathcal{O}(m)$ operations. Thus a total of $\mathcal{O}(n^3m^2)$ operations is required to compute the $\mathcal{O}(n^2m)$ entries that need to be stored throughout the calculation. With a fixed energy resolution the vector length m becomes proportional to the chain length n resulting in $\mathcal{O}(n^5)$ operations and a memory requirement of $\mathcal{O}(n^3)$.

The performance data compiled in Table 7 shows that only the calculation of the density of states (d.o.s.) of small molecules is feasible within a few hours. To allow the computation of the d.o.s. of larger molecules, it is necessary to reduce the energy resolution. Since execution time scales as $\mathcal{O}(n^3m^2)$, a reduction of the number of energy bins yields a significant acceleration of the calculations. The energy parameters used within the Vienna RNA Package are

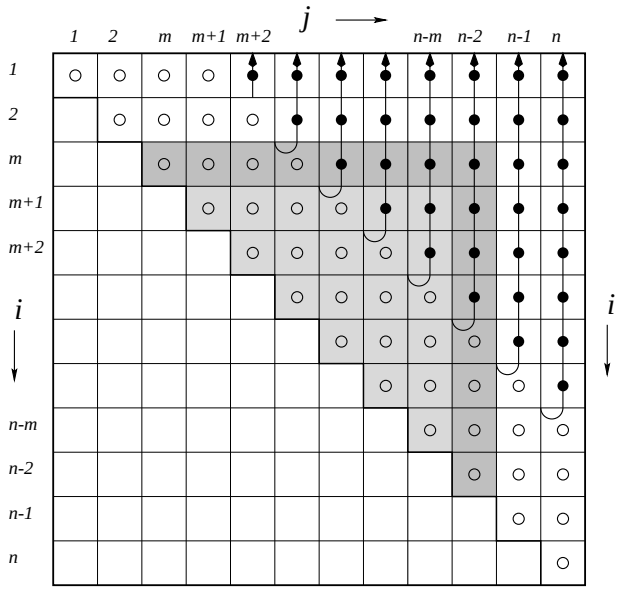


Figure 19: Filled circles denote entries in N, N^B, ..., that are computed as indicated. Unfilled circles denote entries set by initial conditions. Computation of an entry `N[m,n-2,e]` requires entries `N[i,j,e]` left and below, shown as shaded bars. Calculation of `NB[m,n-2,e]` requires all entries within the more slightly shaded triangle.

implemented with an accuracy of 0.01 kcal/mol, thereby limiting the energy resolution that can be achieved. On the other hand, a resolution coarser than thermal energy ($RT \approx 0.6$ kcal/mol) will hide the most interesting information.

Oftentimes one is not interested in the complete density of states but only in the vicinity of the ground state. It is sufficient in this case to consider only a limited energy range above the most stable state for each subsequence; this technique should lead to a significant reduction of the CPU requirements.

To implement both possibilities, optional arguments can be supplied to RNAdos. Option `'-s 10'` forces a re-calculation of the energy parameter set. Only 1/10 of the original number of energy bins are used; the resolution is thus coarser. This option yields a reduction of execution time that makes the calculation of the d.o.s. of tRNA molecules feasible, see Table 7, Figure 21. However, a resolution of 0.01 kcal/mol at least within a limited range above

```
turner ~> RNAdos
'RNAdos', ver <97/01/14 16:36:25 >
Input string (upper or lower case); @ to quit
....,....1....,....2....,....3....,....4....,....5....,....6
ACGAUCGUAGUCACGAUG
...(((((....)))))).
Fold: minimum free energy =  -2.52 kcal/mol
Fold: partition function = 87.715393
MinEn: minimum free energy =  -2.52 kcal/mol (scale=1)
number of bins: 3457
Dfold: Number of Structures = 1265
Results: N[ij] =  1265
         NB[i,j] =   0
         NM[i,j] = 1264
```

Table 5: **Interactive example run of RNAdos:** RNAdos calls routines contained in the Vienna RNA package to compute the minimum free energy and the partition function of the sequence (Fold). Depending on the string length, the energy parameters are rescaled and the minimum free energy recalculated by an own routine similar to Fold (MinEn). For small string length the scaling factor is 1; this yields an energy resolution of 0.01 kcal/mol. The number of energy bins is calculated from the minimum energy. A modified version of the partition function routine, where all loops are assigned 0 energy, is called to obtain the total number of structures.

```
// data arrays
int N[i,j,e],NB[i,j,e],NA[i,j,e],NM[i,j,e],NM1[i,j,e]

for(j=1...n)
   for(i=j..1)
      NB[i,j,H(j-i)] + 1     // H = hairpin energy
      for(k=i+1...j-1)
         for(l=k...l)
            for(e=mfe...max)    // I = internal energy
               NB[i,j,e] + NB[k,l,e-I(i,j,k,l)]
      for(k=i...j)
         for(e=mfe...max)
            for(e'=mfe...max)
               NB[i,j,e] +  NM[i+1,k-1,e']*NM1[k,j-1,e-e'-Mc]
      for(l=i...j)
         for(e=mfe...max)
            NM1[i,j,e] + NB[i,l,e-Mb(j-l)-Mi]
      for(k=i...j)
         for(e=mfe...max)
            for(e'=mfe...max)
               NM[i,j,e] + NM[i,k-1,e']*[NM1[k,j,e-e']
            NM[i,j,e] + NM1[k,j,e-Mb(k-i)
      for(l=1...j)
         for(e=mfe...max)
            NA[i,j,e] + NB[i,l,e]
      N[i,j,0] +1          // open chain
      for(e=mfe...max)
         N[i,j,e] + NA[i,j,e]  // single component
         for(k=i...j)
            for(e'=mfe...max)  // multi component
               N[i,j,e] + N[i,k-1,e']*NA[k,j,e-e']
for(e=mfe...max)
   DensityOfStates[e] = N[1,n,e]
```

Table 6: **Pseudocode for the calculation of the density of states:** Calligraphic symbols denote energy parameters for different loop types: hairpin loops $\mathcal{H}(i,j)$, interior loops, bulges, and stacks $\mathcal{I}(i,j,k,l)$, multiloops $\mathcal{M}$.

ground state would be interesting. By restricting the number of energy bins to a certain fraction of the normal energy span, the computation time is reduced. Albeit, CPU requirements still scale with m^2, see Figures 22, 24. The cutoff

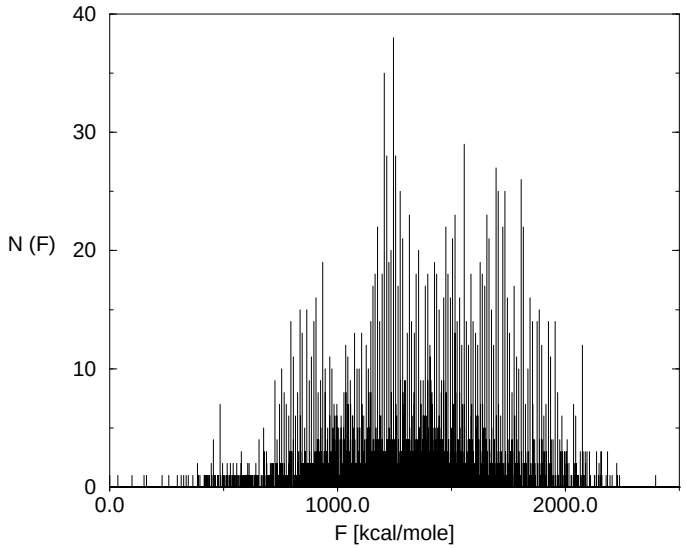


Figure 20: Example for a density of states plot. Height of lines indicates the number of structures with a certain free energy. The programm uses an energy scale relative to the ground state.

option is invoked by `'-K 1000'` (for a cutoff of 1000 bins). Care has to be taken that the cutoff is chosen high enough, so that no structures at low energies are lost, see Figure 23.

Table 7: Performance Data for the Density of States. CPU times are measured on an `SGI Power Challange R8000` with 1 GB memory. All times are in seconds.

n	Sequence	Number of Structures	Energy Resolution 0.01	0.1
8	$(ACGU)_2$	5	8	< 1
12	$(ACGU)_3$	35	139	2
16	$(ACGU)_4$	$2.7{\cdot}10^2$	1254	14
20	$(ACGU)_5$	$2.2{\cdot}10^3$	5049	51
24	$(ACGU)_6$	$2.0{\cdot}10^4$	16926	143
28	$(ACGU)_7$	$1.8{\cdot}10^5$	41089	329
32	$(ACGU)_8$	$1.7{\cdot}10^6$	*	691
35	random	$2.0{\cdot}10^7$	*	804
40	$(ACGU)_{10}$	$1.6{\cdot}10^8$	*	1791
76	tRNA-phe	$1.5{\cdot}10^{16}$	*	28678

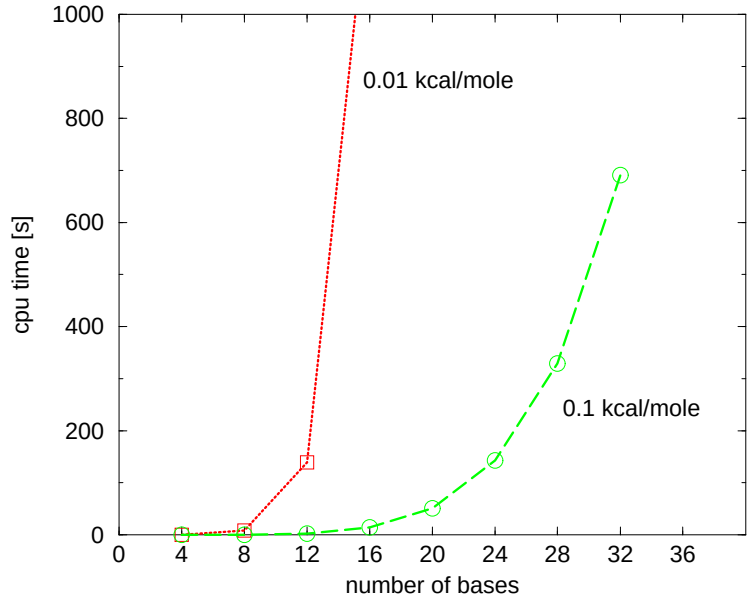


Figure 21: CPU requirements of RNAdos at energy resolutions of 0.01 kcal/mol (squares) and 0.1 kcal/mol (circles) for the computation of the d.o.s. of n(ACGU).

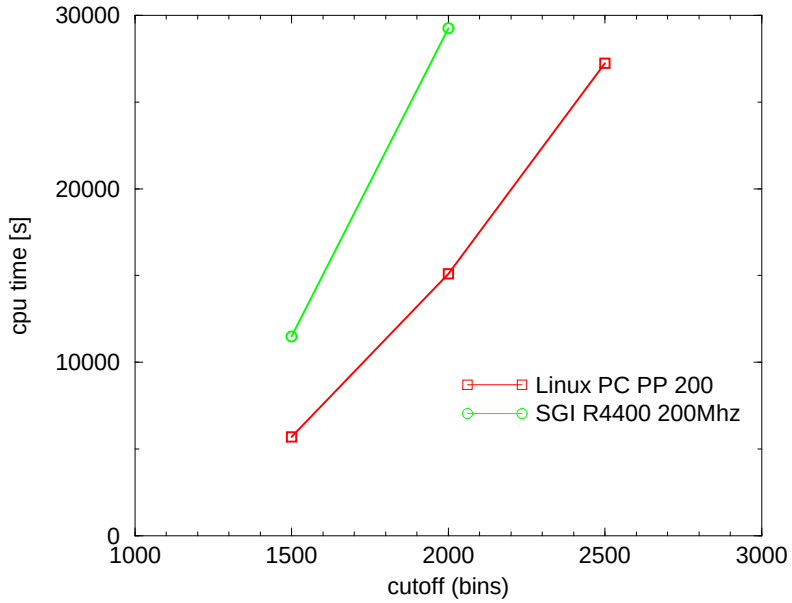


Figure 22: CPU requirements of RNAdos for the computation of the d.o.s. of a random sequence, $n = 60$, at cutoff values of 1500, 200 and 2500 energy bins.

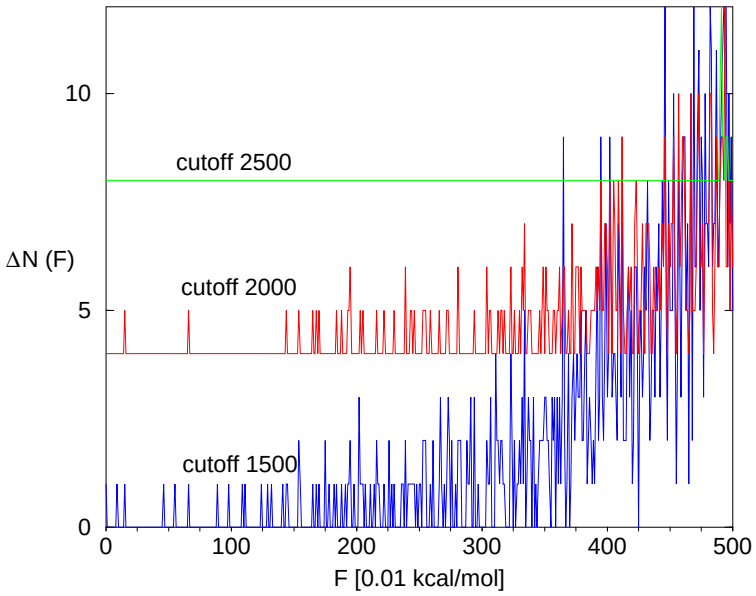


Figure 23: Computation of the d.o.s. of $tRNA^{Phe}$ with different cutoff bins. The figure shows the missing states: While the first missing state at a cutoff of 2500 is as high as 5 kcal/mol above the ground state, at lower cutoff values many states are not found.

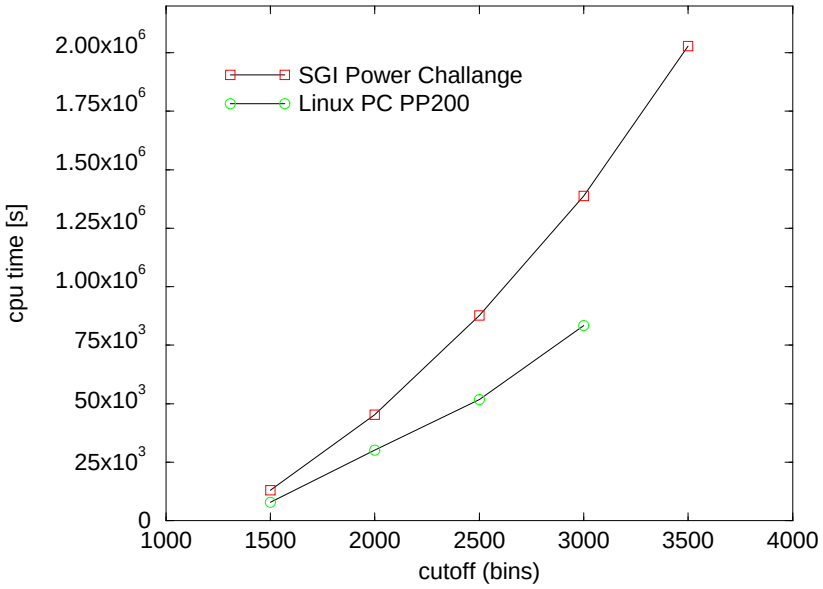


Figure 24: CPU requirements for the computation of the d.o.s. of $tRNA^{Phe}$ with different cutoff bins. Due to the integer arithmetics dominant within the programm, the PP 200 Linux PC with 128 MB proved to be faster than the SGI Power Challange (MIPS R8000, 75MHZ) with 1 GB main memory.

10 Examples of Applications

10.1 Random Sequences

A set of 100 random sequences of equal base composition, $n = 30$, was analyzed. Figure 25 shows an example of the density of states. The minimum free energy, the partition function, and the denisty of states were calculated for each sequence. The density of states yielded the gap energy, *i.e.* the energy gap between the ground state and the first "exited" state. The structural entropies of the molecules were calculated from the d.o.s. The partition function yields the frequency of the minimum free energy structure in the ensemble. Table 8 provides some example results. Note that the free energy of the ensemble is derived from $\Delta G = -kT \ln Z$ and includes entropic contributions from structural entropy. From the density of states the *geometric entropy* S was obtained by

$$S = -k_B \sum_{i=1}^{N} p_i \ln p_i, \tag{57}$$

with p_i being the probability of state i in the ensemble in Z the partition function of the molecule:

$$p_i = \frac{e^{-\frac{E_i}{kT}}}{Z} \tag{58}$$

The overall shape of $N(F)$ is Gaussian, see Figure 25. This is not surprising, since F is composed of a large number of additive contributions. The overwhelming majority of structures has positive energy, hence only a small subset of all possible structures is physically important. The ground state of all sequences was unique both at a resolution of 0.1kcal/mol and 0.01kcal/mol. However, in general there is a substantial number of structures within a few RT above the ground state. It is also worth noting that there is a strong correlation between the size of the energy gap between the ground state and the first "excited state" and the fraction p_0 of ground state structure in thermodynamic equilibrium, see Figure 26. The latter quantity can be obtained directly from the partition function (McCaskill 1990; Hofacker *et al.* 1994b).

Table 8: Example results for the the calculated values: MFE = minimum free energy, F(Ens) = free energy of ensemble, CPU = execution time, Z = partition function, gap = gap energy [kcal/mol], N = number of structures, ν = frequency of minimum free energy structure in ensemble.

MFE	F(Ens)	CPU	Z	gap	S/k_B	N	ν
-2.53	-3.87	17318	537.61	0.10	4.11506	757277	0.112801
-5.94	-6.20	25132	23266.7	1.60	1.48706	1014238	0.659061
-3.17	-3.86	18709	522.74	0.35	2.24817	784080	0.327705
-1.37	-2.13	15109	31.7853	0.69	3.30730	755248	0.290505
-4.82	-5.28	22555	5248.38	0.87	1.93574	718227	0.474701
-3.49	-4.00	19735	659.428	0.85	2.50307	830975	0.4366
-3.66	-3.82	20045	488.285	1.83	1.24879	731060	0.776906
-3.54	-3.95	19371	608.39	1.40	2.52384	671276	0.513219
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

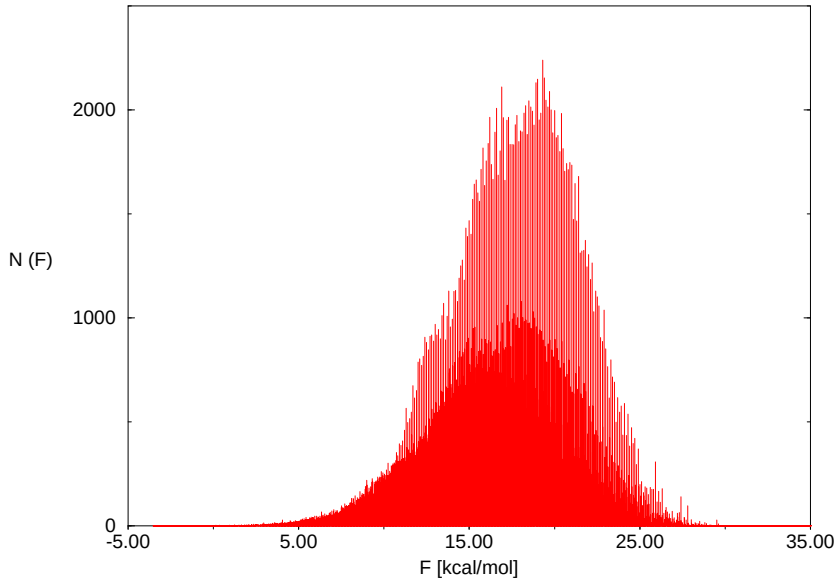


Figure 25: Example for the density of states of a random sequence of 30 bases. minimum free energy = −3.54 kcal/mol, Total number of structures = 671,276, sequence: ACUAGUCGCGGGGAAUACCUUGGUUCCAAC.

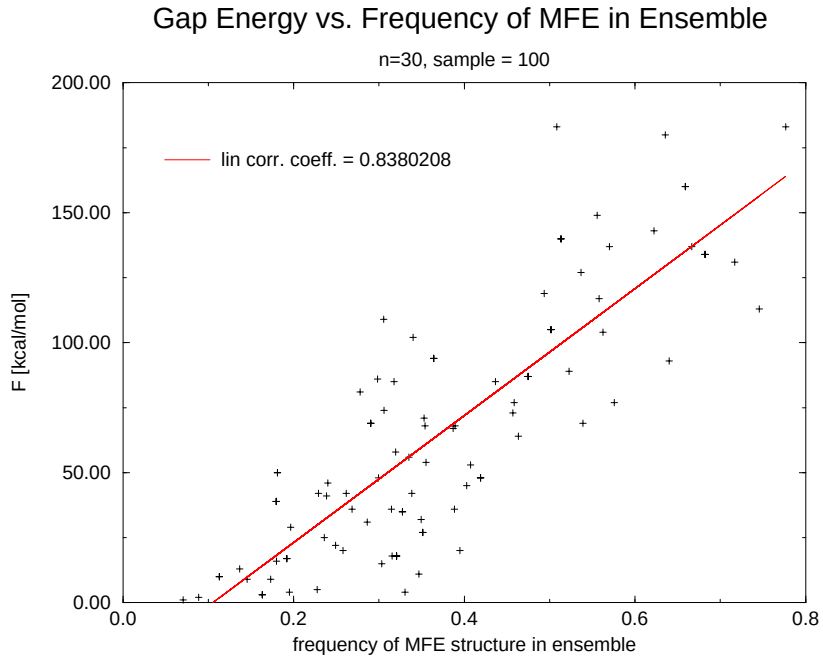


Figure 26: Frequency of the minimum free energy structure in the ensemble versus energy gap between ground state and first suboptimal state. Data shows a linear correlation.

10.2 Yeast tRNA$^{\mathrm{Phe}}$

While it is not possible to calculate the density of states of tRNA at full energy resolution, the d.o.s. at reduced energy resolution or within a certain energy range is computationally feasible. The density of states of Yeast Phenylalanine tRNA at different energy resolutions is given as an example.

The total number of structures is $14,995,224,405,213,184$; again only a minimal fraction of $1.77 \cdot 10^6$ structures have negative energy. The reference state is the open structure. The minimum free energy is $E = -12.26$ kcal/mol. The full density of states can be calculated at an energy resolution of 0.1 kcal/mol, see Figure 27. The overall shape of $N(F)$ is again Gaussian. An enlargement of the left side of the figure shows that a number of suboptimal states can be distinguished even at low resolution, see Figure 28.

Calculations performed with cutoff of 3500 bins yield the density of state within a few kcal/mol above ground state. It can be seen that there is a

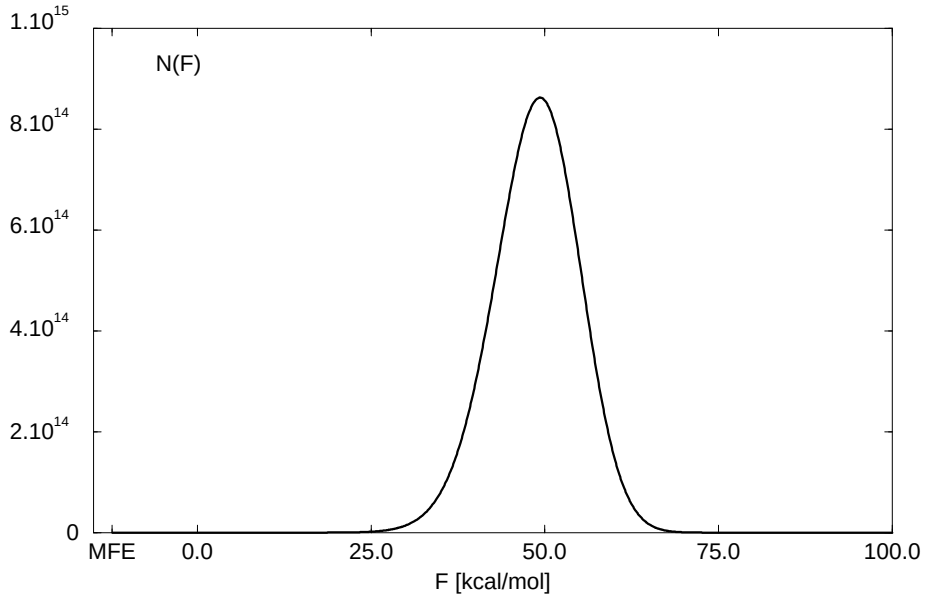


Figure 27: Density of states of Yeast tRNA$^{\mathrm{Phe}}$, (n=76). Energy resolution is 0.1 kcal/mol.The total number of structures, $14,995,224,405,213,184$ emphasizes the need for a recursive approach. Less than $1.77 \cdot 10^6$ structures have negative energy, the reference state being the open structure. The minimum energy structure is the familiar *cloverleaf* with $E = -12.26$ kcal/mol.

considerable energy gap between the minimum free energy and the energy of the first suboptimal structure. The ground state has proven to be unique both at a resolution of 0.1kcal/mol and 0.01 kcal/mol. However, in general, there is a substantial number of structures within a few RT above the ground state.

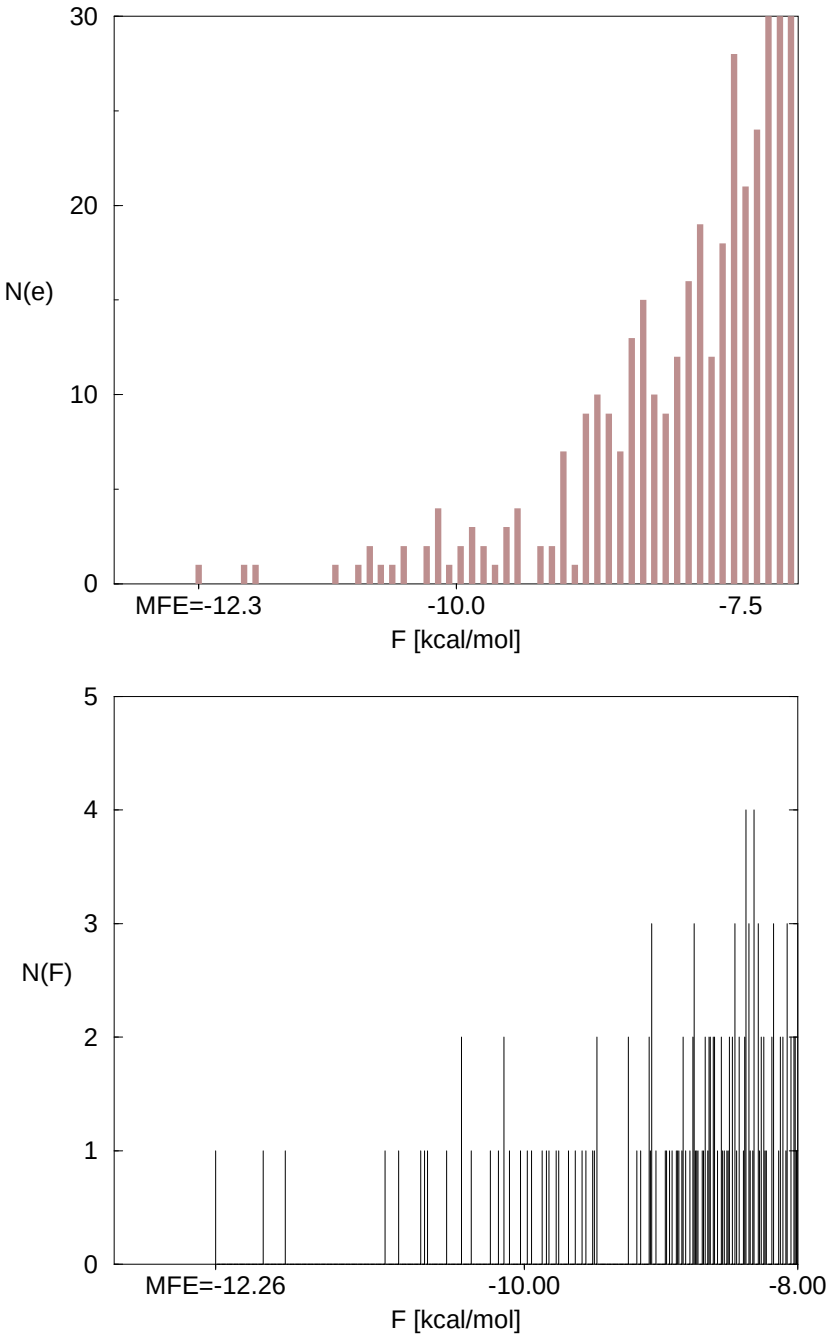


Figure 28: Low region of the density of states of Yeast $\mathrm{tRNA}^{\mathrm{Phe}}$, ($n$=76). Energy resolution is 0.01 kcal/mol at the lower figure and 0.1 kcal/mol at the upper figure. The ground state is unique both at a resolution of 0.1 kcal/mol and 0.01 kcal/mol. There are, only 2 suboptimal structures within 1 kcal/mol above the ground state.

10.3 E. Coli tRNA

Higgs (Higgs 1993; 1995) found that the density of states of natural (evolved) sequences such as tRNAs differs significantly from random RNA sequences. His studies were based on a non-recursive algorithm using a drastically simplified energy model (Higgs 1993; 1995). Our own computations support his

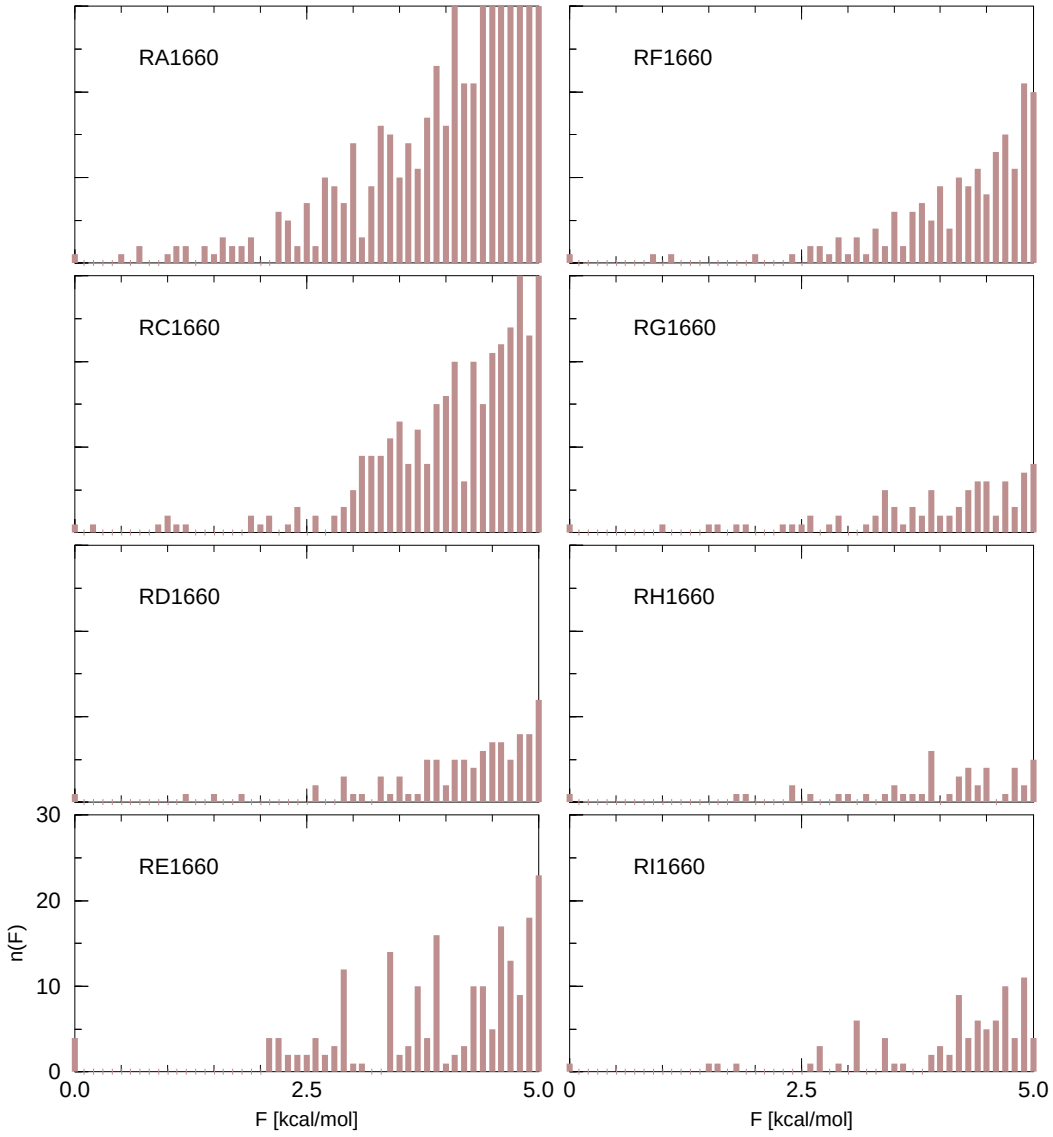


Figure 29: The complete density of states with an energy resolution of 0.1 kcal/mol was computed for a variety of E. Coli tRNA sequences. The enlargement shows all states within 5 kcal/mol above ground state. Each calculation was done on a SGI Power Challenge and took 9h cpu time and 150 MB mail memory.

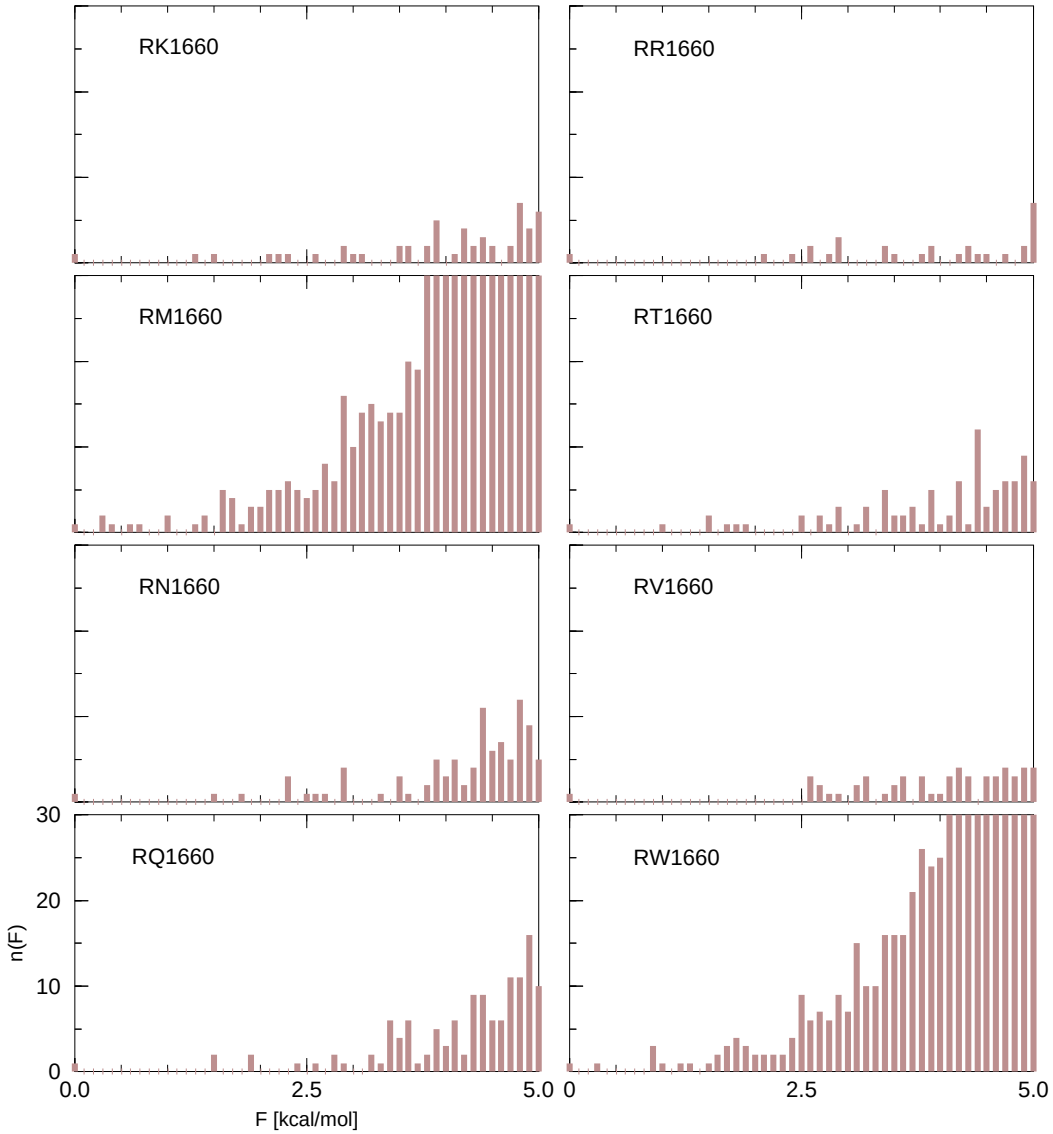


Figure 30: Density of states of E. Coli tRNA sequences. The enlargement shows all states within 5 kcal/mol above ground state. See the appendix A for the sequence numbers and the text for the translation of modified bases.

conclusions:

A number of tRNA sequences from EMBL tRNA Database, which is based on a compilation of Steegborn (Steegborn *et al.* 1995), were analyzed. See Appendix A for the sequences and sequence numbers referred to in the text. Figures 29, 30 provide enlargements of the regions containing all states within

5 kcal/mol above the ground state. Reference state is the minimum free energy.

tRNAs differ in some extend 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, so it is necessary to develope a consistent method of dealing with these bases. Since it seems obvious that some of these bases are modified to prevent bonding, a class of non-bonding bases ('N') has to be introduced. This method was first suggested by Ninio (Ninio 1979). Higgs 1993, following Ninio, treated the following bases as non-bonding: Dihydro uridine (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 (Higgs 1993). A slightly different method was described by Higgs 1995 (Higgs 1995): Since the majority of all tRNA sequences fit the familiar cloverleaf folding pattern, it is possible to construct a class of all modified bases which never occur in a paired position in the cloverleaf. These bases were treated as non-bonding. All other bases were translated to their standard base analogue. Following this method we worked with the following assignments:

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

All calculations were performed at an energy resolution of 0.1 kcal/mol. The mean execution time on a SGI Power Challenge was 9h cpu. Each calculation required at maximum 150 MB main memory.

Figure 31 shows the mean distribution of 30 tRNA sequences. The mean energy gap of these sequences between ground state and first suboptimal state is 1.1$\dot{\text{k}}$cal/mol. There are only 1.3 structures within the first kcal/mol above ground state and 6.7 structures within 2 kcal/mol. It would be interesting to compare this values with mean data for other classes of tRNA and with

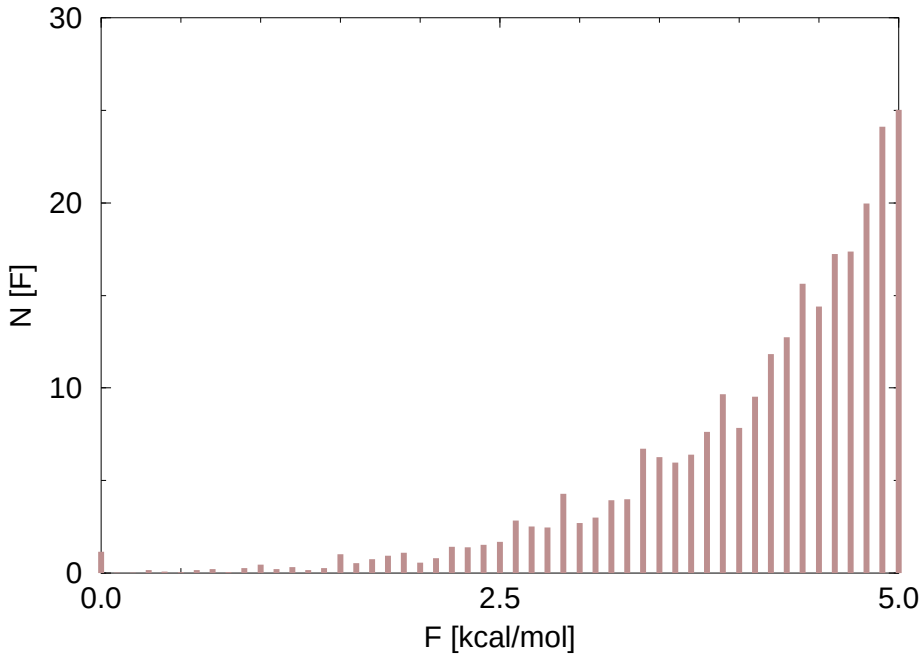
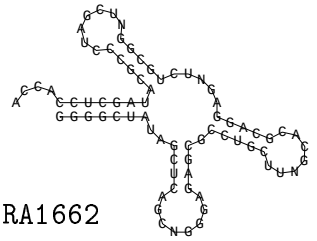


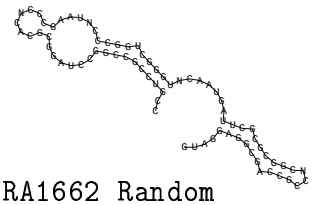
Figure 31: Mean distribution of states of 30 E. Coli tRNA sequences at an energy resolution of 0.1 kcal/mol. The mean energy gap is 1.1 kcal/mol.

data from random sequences: Higgs 1995 computed the density of states for a number of sequences, using his own program, which implemented a rather simplified energy model. His figures, however, show consistent differences between random and biological evolved sequences. To follow his calculations with our program and as an example of application, we computed the distribution of states both of a number of tRNA sequences from E. Coli and of random sequences of same length and same base composition. Figure 32 shows some example plots. It is clearly visible that (a) original tRNA sequences have less states in the vicinity of the ground state and (b) the energy gap is usually larger. Table 9 compiles similar data, showing lower minimum free energy values and larger energy gaps for tRNA sequences than for random sequences of the same base composition. Note that the number and the position of non-bonding bases have not been changed. The results of Higgs (Higgs 1995) are thus supported by our calculations.

The problem remains, that it might not be justified to compare tRNA sequences with random sequences, even at the same base composition, since

they fold into completely different minimum free energy structures:


RA1662


RA1662 Random

It is clear, that the structure of tRNA is highly functional, so that a biologically active tRNA has to exhibit increased stability. This gives rise to lower minimum free energies and larger gaps. If we want to show that of the large

Table 9: Minimum free energy and gap energy of E. Coli tRNA sequences (upper part) and random sequences of same base composition (lower part). Biologically evolved sequences have lower minimum free energies and exhibit larger energy gaps.

Sequence Number	Anti-Codon	MFE [kcal/mol]	Number of Structures	Energy gap [kcal/mol]
RA1660	GGC	−2.04	$64 \cdot 10^{15}$	0.5
RA1661	VGC	−2.15	$71 \cdot 10^{15}$	0.3
RA1662	VGC	−1.98	$63 \cdot 10^{15}$	0.1
RC1660	GCA	−1.89	$3.9 \cdot 10^{15}$	0.2
RD1660	QUC	−1.63	$35 \cdot 10^{15}$	1.2
RE1660	SUC	−2.57	$125 \cdot 10^{15}$	2.1
RE1661	SUC	−2.54	$82 \cdot 10^{15}$	2.0
RE1662	SUC	−2.57	$48 \cdot 10^{15}$	0.9
MUA1660		−1.60	$3.1 \cdot 10^{15}$	0.2
MUA1661		−1.88	$82 \cdot 10^{15}$	0.1
MUA1662		−1.92	$84 \cdot 10^{15}$	0.2
MUC1660		−0.88	$0.41 \cdot 10^{15}$	0.7
MUD1660		−1.25	$171 \cdot 10^{15}$	0.1
MUE1660		−1.31	$48 \cdot 10^{15}$	0.1
MUE1661		−0.89	$49 \cdot 10^{15}$	0.1
MUE1662		−0.77	$4.6 \cdot 10^{15}$	0.2

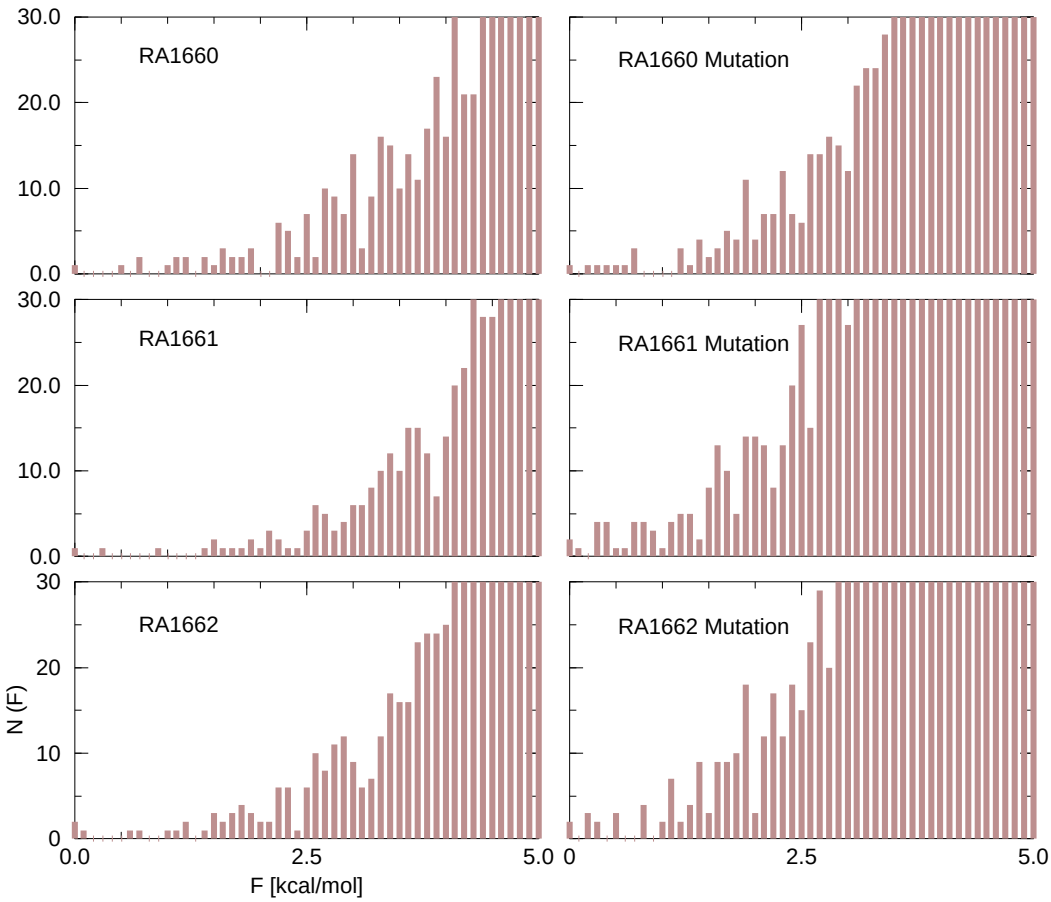


Figure 32: The distribution of states 5 kcal/mol above ground state are shown for three E. Coli tRNA molecules (on the left, see Appendix A), and for three random sequences with the same base composition. The energy gap between the ground state and the first suboptimal state is usually larger for tRNA than for random sequences, and there are less suboptimal structures within 1 kcal/mol than for random sequences. The random sequences do not fold into the cloverleaf structure,however. These calculations were performed with an energy resolution of 0.1 kcal/mol and an cutoff of 500 energy bins and took 1h CPU time on an SGI Power Challenge.

number of sequences, that fold into the cloverleaf, the most stable sequences have evolved, we have to compare tRNA sequences with neutral mutants, *i.e.* sequences that are one-point mutations and fold into the same structure. We have considered the neutral one-point mutations of RN1660 E. Coli tRNA as an example. 206 one-point mutations were produced by changing all standard bases. All non-bonding bases N remained unchanged in number and position.

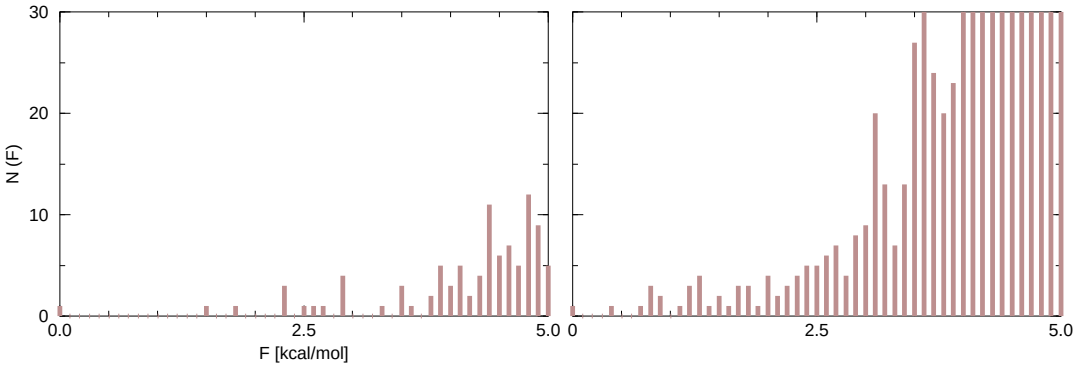


Figure 33: The distribution of states 5 kcal/mol above ground state are shown for RN1660 E. Coli tRNA (on the left, see Appendix A), and for a one-point mutation of the original sequence. The mutated sequence has the same ground state structure, *i.e.* sequences fold into the same minimum free energy structure. Only the biologically evolved sequence shows enhanced stability: the gap energy is larger (1.5 kcal/mol for the original and 0.4 kcal/mol for the evolved sequence), and there are generally more structures within 1 kcal/mol above ground state. These calculations were performed with an energy resolution of 0.1 kcal/mol and an cutoff of 500 energy bins and took 1h cpu time on an SGI Power Challenge.

94 of the neighbors fold into the clover leaf. Generally speaking, most of the one-point mutants are almost undistinguishable from the original string. The average energy gap is only slightly lower, due to the majority of sequences, which have exactly the same gap. In those sequences, where the mutation shows some effect, the gap energy is smaller. Figure 33 presents an example calculation for the original string and one mutated sequence. It is clearly visible, that the overall distribution has changed and that there are generally more accessible suboptimal structures. The gap energy is 1.5 kcal for the original string and only 0.4 kcal/mol for the mutated sequence. While we do not have sufficient data for a detailed statistical analysis, our results so far are consistent with the conjecture that biologically evolved sequences with functionally important structure are generally stabilized by larger energy gaps and a reduced number of suboptimal states.

11 Conclusion and Outlook

RNA structures play a significant role in a wide range of 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 sufficient, to know the ground state structure. Only the complete density of states, *i.e.* the distribution of energies of all possible configurations, can provide all information concerning the stability and structural flexibility of a structure and the suboptimal states.

The representation of RNA secondary structures as vertex-labeled, planar graphs are discussed in detail. A variety of dynamic programming algorithms derived previously were compiled and presented in a consistent notation. While the algorithms for the minimum free energy and the partition function have already been available for some time, the density of states algorithm was developed and implemented for the first time. CPU time requirements of the algorithm scale as $\mathcal{O}(n^3m^2)$, with n denoting the number of bases and m the number of energy bins used to store the number of states. The computation of the density of states of biologically significant molecules is feasible at sufficient energy resolutions. Variants of the implementation allow for a reduced energy resolution and for a restriction to a sufficient energy range above the ground state.

A number of sample calculations were performed in order to highlight the amount of information yielded from the density of states. The complete density of Yeast $\mathrm{tRNA}^{\mathrm{Phe}}$ was computed at a resolution of 0.1 kcal/mol, and, within a region of 5 kcal/mol above the ground state, at an energy resolution of 0.01 kcal/mol. A number of 30 E. Coli tRNAs were analyzed and compared with random sequences of same base composition and length. The results show that original tRNA sequences have less states in the vicinity of the ground state and the energy gap is usually larger. However, large investigations on a

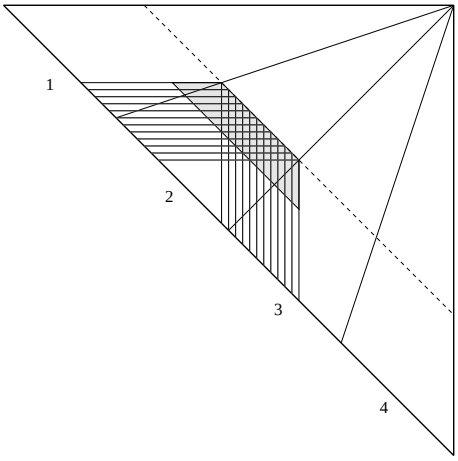


Figure 34: (Hofacker *et al.* 1996) Representation of the memory usage at the parallel folding algorithm. The triangular data matrices are divided into sectors with an equal number of diagonal elements. The computation proceeds from the main diagonal towards the upper right corner. The information needed by processor two in order to calculate the elements of the dashed diagonal are highlighted.

statistical base have not yet been performed.

An additional feature not yet included into the program, is a back-tracking mechanism. It would be of great interest, not only to know the mere number of states, but to gain knowlegde of the structures themselves. The implementation of such a mechanism will constitute a next step.

Hofacker (Hofacker *et al.* 1996) implemented a parallel version of the minimum free energy of very large chains. Since the data elements are stored in triangular matrices, the entries can be calculated by proceeding from the main diagonal towards the upper right corner. The matrices are divided into sectors with an equal number of diagonal. The implementation of a parallel version of the algorithm for the density of states seems particularly promising.

A EMBL tRNA Database

All tRNA sequences are from the compilation of Steegborn (Steegborn *et al.* 1995), 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:

```
D  (D)       dihydrouridine
B  (Cm)      2'-O-methylcytidine
Y  (yW)      wybutosine
?  (m5C)     5-methylcytidine
;  (G)       unknown modified guanosine
L  (m2G)     N2-methylguanosine
#  (Gm)      2'-O-methylguanosine
R  (m22G)    N2,N2-dimethylguanosine
7  (m7G)     7-methylguanosine
K  (m1G)     1-methylguanosine
'  (m3C)     3-methylcytidine
<  (?C)      unknown modified cytidine
M  (ac4C)    N4-acetylcytidine
?  (m5C)     5-methylcytidine
T  (T)       thymine
"  (m1A)     1-methyladenosine
*  (ms2i6A)  2-methylthio-N6-isopentenyladenosine
H  (?A)      unknown modified adenosine
^  (Ar(p))   2'-O-ribosyladenosine (phosphat)
N  (?U)      unknown modified uridine
J  (Um)      2'-O-methyluridine
P  (psi)     pseudouridine
```

```
]  (m1psi)    1-methylpseudouridine
Z  (psi m)    2'-O-methylpseudouridine
\  (m5Um)     5, 2'-O-dimethyluridine
{  (mnm5U)    5-methylaminomethyluridine
X  (acp3U)    3-(3-amino-3-carboxypropyl)uridine
S  (mnm5s2U)  5-methylaminomethyl-2-thiouridine
V  (cmo5U)    uridine 5-oxyacetic acid
Q  (Q)        queuosine
}  (k2C)      lysidine
```

Bases are translated as suggested by Higgs (Higgs 1995): 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
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
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
```

References

Antao, V. P., and Tinoco, Jr., I. 1992, Thermodynamic parameters for loop formation in RNA and DNA hairpin tetraloops. *Nucl. Acids Res.* 20(4):819–824.

Cech, T. 1986, RNA as an enzyme. *Scientific American* 11:76–84.

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

Freier, S. M.; Kierzek, R.; Jaeger, J. A.; Sugimoto, N.; Caruthers, M. H.; Neilson, T.; and Turner, D. H. 1986, Improved free-energy parameters for prediction of RNA duplex stability. *Proc. Natl. Acad. Sci. USA* 83:9373–9377.

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

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

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

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

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

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

Hofacker, I. L.; Fontana, W.; Stadler, P. F.; Bonhoeffer, L. S.; Tacker, M.; and Schuster, P. 1994a. `Vienna RNA Package`. `http://www.tbi.univie.ac.at`. (Public Domain Software).

Hofacker, I. L.; Fontana, W.; Stadler, P. F.; Bonhoeffer, S.; Tacker, M.; and Schuster, P. 1994b, Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.* 125(2):167–188.

Hofacker, I. L.; Huynen, M. A.; Stadler, P. F.; and Stolorz, P. E. 1996, Rna folding and parallel computers: The minimum free energy structures of complete hiv genomes. *Concurrency.* submitted, SFI preprint 95-10-089.

Hofacker, I. L. 1994. *The rules of the evolutionary game for RNA: A statistical characterization of the sequence to structure mapping in RNA.* Ph.D. Dissertation, University of Vienna.

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

Jaeger, J. A.; Turner, D. H.; and Zuker, M. 1989, Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci., USA, Biochemistry* 86:7706–7710.

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

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

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

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

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

Konings, D. 1989. Pattern analysis of RNA secondary structures. *Proefschrift, Rijksuniversiteit te Utrecht.*

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

Morse, S. E., and Draper, D. E. 1995, Purine-purine mismatches in RNA helices: evidence for protonated GA pairs and next-nearest neighbor effects. *Nucleic Acids Res.* 23:302–6.

Ninio, J. 1979. *Biochemie* 61:1133.

Nussinov, R., and Jacobson, A. B. 1980, Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. USA* 77(11):6309–6313.

Nussinov, R.; Piecznik, G.; Griggs, J. R.; and Kleitman, D. J. 1978, Algorithms for loop matching. *SIAM J. Appl. Math.* 35(1):68–82.

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

Peritz, A. E.; Kierzek, R.; Sugimoto, N.; and Turner, D. H. 1991, Thermodynamic study of internal loops in oligonucleotides: Symmetric loops are more stable than assymetric loops. *Biochemistry* 30:6428–36.

Poerschke, D. Elementary steps of base recognition and helix-coil transitions in nucleic acids. In Pecht, I., and Rigler, R., eds., *Molecular Biology, Biochemistry and Biophysics*, volume 24. Springer-Verlag, Berlin 1977. 191–218.

Saenger, W. *Principles of Nucleic Acid Structure.* Springer Advanced Texts in Chemistry. Springer-Verlag, New York 1984.

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

Schuster, P. K. Hydrogen bonds. In *Encyclopedia of Physical Science and Technology*, volume 6. Academic Press, Inc. 1987. 518–554.

Serra, M. J.; Axenson, T. J.; and Turner, D. H. 1994, A model for the

stabilities of RNA hairpins based on a study on the sequence dependence of stability for hairpins of six nucleotides. *Biochemistry* 33:14289–96.

Serra, M. J.; Lyttle, M. H.; Axenson, T. J.; Schadt, C. A.; and Turner, D. H. 1993, RNA hairpin loop stability depends on the closing base pair. *Nucleic Acids Res.* 21:3845–9.

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

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

Tinoco, Jr., I. Structures of base pairs involving at least two hydrogen bonds. In GesteLand, R. F., and Atkins, J. F., eds., *The RNA World.* CSHL Press 1993. 603–609.

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

Walter, A. E.; Turner, D. H.; Kim, J.; Lyttle, M. H.; Muller, P.; Mathews, D. H.; and Zuker, M. 1994, Coaxial stacking of helices enhances binding of oligoribonucleotides and improves prediction of RNA folding. *Proc. Natl. Acad. Sci.* 91:9218–22.

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

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

Waterman, M. S. 1978, Secondary structure of single-stranded nucleic acids. *Studies on foundations and combinatorics, Advances in mathematics supplementary studies, Academic Press N.Y.* 1:167 – 212.

Waterman, M. S. *Introduction to Computational Biology: Sequences, Maps and Genomes.* Chapman & Hall, London 1995.

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

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

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

Curriculum Vitae

Jan Cupal

* 9.5.1969, Wien

1975 – 1979	Volksschule Wien IX
1979 – 1987	Neusprachliches Gymnasium Wien IX
5/87	Reifeprüfung am Gymnasium Wien IX
1987 – 1997	Studium der Chemie, Hauptfach Chemie, an der Univiersität Wien
1996 – 1997	Diplomarbeit am Insititut für Theoretische Chemie an der Univiersität Wien

Publications

- Jan Cupal, Ivo L. Hofacker, and Peter F. Stadler.
  Dynamic Programming Algorithm for the Density of States of RNA Secondary Structures.
  In: R. Hofestädt, T. Lengauer, M. Löffler, and D. Schomburg, editors, *Computer Science and Biology 96 (Proceedings of the German Conference on Bioinformatics)*, pages 184–186, Leipzig (Germany), 1996. Univeristät Leipzig.

- Jan Cupal, Christoph Flamm, Alexander Renner, and Peter F. Stadler.
  Density of States, Metastable States, and Saddle Points – Exploring the Energy Landscape of an RNA Molecule.
  In: T. Gaasterland, P. Karp, K. Karplus, Ch. Ouzounis, Ch. Sander, and A. Valencia, editors, *Proceedings of the Fift International Conference on Intelligent Systems and Molecular Biology*, pages 88–91, Menlo Park, CA, 1997. AAAI Press.