Prediction of RNA Secondary Structures using Parallel Computers

DIPLOMARBEIT

eingereicht von

Martin Fekete

zur Erlangung des akademischen Grades

Magister rerum naturalium

an der Formal- und Naturwissenschaftlichen Fakultät

der Universität Wien

November 17, 1997

An dieser Stelle ist es mir eine Freude all jenen zu danken, die zum Gelingen dieser Arbeit beigetragen haben.
Peter Stadler hat diese Arbeit betreut und mich in wissenschaftliches Arbeiten eingeführt. Walter Fontana danke ich für die klare Darstellung auch noch so komplexer Zusammenhänge. Ivo Hofacker sei gedankt für die Einführung in die Geheimnisse der Parallelrechner, und für die stetige Unterstützung bei Problemen, Peter Schuster für die Aufnahme in seine Arbeitsgruppe.
Allen Freunden und Freundinnen am Institut danke ich für Unterstützung und Freude bei der Arbeit.
Besonders danken möchte ich Doris, die sich während meiner Diplomarbeit stets liebevoll um mich gekümmert hat und meinen Eltern, die mir das Studium ermöglicht haben.

Zusammenfassung

RNA-Moleküle sind nicht nur Informationträger, sondern auch selbständige funktionelle Einheiten. Bei einer großen Anzahl von biologischen Prozessen spielt ihre dreidimensionale Struktur daher eine wichtige Rolle. Das Studium der Sekundärstruktur von RNA-Molekülen bietet uns die Möglichkeit, diese in einer gröberen Auflösung zu untersuchen. Dadurch wird die Vorhersage von 3D-Strukturen unterstützt, und man erhält wertvolle Informationen über ihre biochemischen Funktionen. Sekundärstrukturen sind darüberhinaus diskret, und eignen sich daher gut für Computeralgorithmen.

RNA-Sekundärstrukturen können mit Hilfe von dynamischen Computeralgorithmen vorhergesagt werden. Dabei ist der Rechenaufwand für die Computeralgorithmen $O(n^3)$ und der Speicherbedarf $O(n^2)$, wobei n die Sequenzlänge bezeichnet. Lange RNA-Moleküle, wie etwa ganze Virusgenome, sind außerhalb der Reichweite von Workstations. Für lange RNA-Sequenzen haben wir deshalb einen Parallelalgorithmus entwickelt, der es erlaubt, die Vorhersage der minimalen freien Energie, die Zustandssumme und die Basen-Paarungs-Wahrscheinlichkeiten auf "distributed memory" Parallelrechner durchzuführen. Auf einem Intel iPSC Hypercube und einem Intel DELTA Supercomputer wurde der Algorithmus bereits erfolgreich implementiert. Selbst auf einigen hundert Prozessoren erhielten wir dabei eine gute Effizienz des Algorithmus. Als eine erste Anwendung haben wir die Sekundärstruktur eines ganzen HIV1 Virus ($n = 9229$) vorhergesagt und analysiert.

Mit Hilfe der heute verfügbaren Parallelcomputern wird die Sekundärstruktur-Vorhersage von langen RNA-Molekülen zu einer Routinemethode werden. Ein umfassender Vergleich aller heutzutage verfügbaren Virus Genome ist möglich geworden und könnte wichtige Neuerungen im Verständnis der funktionalen Bedeutung der Virus-Struktur, sowie in der Frage der Evolution von RNA-Viren bringen.

Abstract

RNA molecules serve not only as carriers of information, but also as functionally active units. The three dimensional shape of RNA molecules plays a crucial role in 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 also in the interpretation of the biochemical abilities of the molecules. Furthermore, secondary structures are discrete and therefore well suited for computational methods.

RNA secondary structure can be predicted by dynamic programming algorithms. For these algoritms the computational effort is $O(n^3)$ and needs $O(n^2)$ memory, where n denotes the sequence length. Long RNA molecules, such as the genomes of RNA viruses, are beyond the capabilities of typical workstations. We have therefore developed a parallel algorithm for the prediction of minimum free energy, partition function and base pair probabilities of large RNA sequences on distributed memory machines. The algorithm was successfully implemented and tested on an Intel iPSC hypercube and the Intel DELTA supercomputer. Our algorithm achieves good efficiencies even on hundreds of processors. As a first applications we have predicted and analysed the secondary structure of a complete HIV1 genome ($n = 9229$).

With the help of massively parallel computers the secondary structure prediction of long RNA molecules will become a routine method. A complete comparison and structure prediction of all presently available RNA virus genomes is now within reach, and may cause a better understanding of functional RNA structures in viruses, as well as their evolutionary relationships.

Contents

1 Introduction **1**

2 RNA Secondary Structures **4**

3 Secondary Structure **7**

3.1 Definitions 7

3.2 Representation of Secondary Structure 11

4 The Energy Model **16**

4.1 Base-Base Interactions in Nucleic Acids 16

4.1.1 Hydrogen Bonding 16

4.1.2 Vertical Base-Base Stacking 18

4.2 Thermodynamic Nearest Neighbor Parameters 21

5 Folding Algorithms **26**

5.1 Computing the Partition Function 26

5.2 Calculating the Base Pair Probability: Backtracking 32

5.3 The Problem of Large Numbers 35

5.4 Computing the Minimum Free Energy 38

6 Hardware: Parallel Computers **41**

6.1 The Intel iPSC Hypercube Parallel Computer 41

6.2 The Intel Delta Parallel Computer 42

6.2.1 The Mesh Interconnect 43

6.2.2 System Description 43

6.2.3 Types of Nodes 44

6.2.4 Message Passing 45

6.2.5 The NX/M Operating System 45

6.2.6 The concurrent File System 46

7 Implementation of the Parallel Partition Function Algorithm **47**

7.1 Parallel MFE fold 47

7.2 Parallel Partition Function 50
7.3 Calculating Base Pair Probabilities: Backtracking 52

8 Performance of the Parallel Algorithm **56**

9 Base Pair Probabilities in $HIV1_{LAI}$ **61**

10 Conclusion and Outlook **70**

References **72**

List of Figures

1 Folding of an RNA Sequence 4
2 RNA Secondary Structure 7
3 Components of RNA Secondary Structures 9
4 Secondary Structure Motifs 10
5 RNA Diagram 11
6 Circular Representation 12
7 RNA Dot Plot 13
8 Mountain Representation 14
9 Generalized Mountain Representation 15
10 Energy Contributions 17
11 Stacking of Nucleic Bases 18
12 Single Stranded Helices 20
13 Energy Contributions 25
14 Secondary Structure Sets and Subsets 27
15 Partition Function of Multicomponent Structures 28
16 Multi-loop Energies and Decomposition 30
17 Probability of P_{hl} Closing Component 33
18 Probability of P_{hl} in Interior Loop 34
19 Probability of P_{hl} in Multi-loop 34
20 The Intel iPSC Parallel Computer 41
21 The Touchstone DELTA System Mesh Arrangement 43
22 The Mesh Interconnection Hardware 44
23 Memory Requirement for the MFE 48
24 Calculation of the `F5` Array 49
25 Communication between Processors. 50
26 Memory for the Parallel Partition Function 51
27 Calculating of Base Pair Probabilities 52
28 Message Passing for the Backtracking 53
29 Single Node Time for $Q\beta$ 57
30 Single Node CPU Time versus Sequence Length 58

31 Efficiency Plot Parallelization 60
32 Generalized Mountain Representation of Complete HIV1 61
33 Generalized Mountain Representation of the 5' End of $HIV1_{LAI}$ 62
34 Dot Plot of the RRE Locus of $HIV1_{LAI}$ 64
35 Generalized Mountain Representation of the RRE Locus 65
36 Dot Plot of the RRE Locus of $HIV1_{LAI}$ 66
37 Secondary Structure Distance for $HIV1_{LAI}$ 68
38 Differences in Base Pair Probabilities of $HIV1_{LAI}$ 69

List of Tables

1 Recursion for the Calculation of the Partition Function 31
2 Recursion for the Calculation of Base Pair Probability 37
3 Recursion for the Calculation of the Minimum Free Energy . . . 40
4 Used Quantities 55
5 Test Sequences 56
6 Single Node Times 59

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 1989). While the activity of natural 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 (Noller 1991; Noller, Hoffarth, & Zimniak 1992; Piccirilli *et al.* 1992).

These discoveries have given much support to the idea that an *RNA World* (Gilbert 1986; Joyce 1988; 1989; 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.

RNA secondary structures provide a useful, though coarse grained, description of RNA structure. In many biologically evolved RNA molecules such as viral genomes and tRNA, the secondary structure seems to be more conserved than the sequence. Viruses belonging to the same family show little sequence similarity, yet exhibit strongly conserved secondary structure motifs, e.g. in terminal non-coding regions. The wide variety of tRNA sequences provided by databases fit into almost identical clover-leaf patterns.

While we have at present no satisfactory algorithm for prediction of 3D structures at hand, secondary structures can be computed efficiently by dynamic programming algorithms based on graph enumeration (Waterman & Smith 1978; Zuker & Stiegler 1981). These algorithms usually yield only the ground state structure; there is of course an exponentially large number of other configurations, and even though the ground state is more probable than any other state, its probability within the whole ensemble of structures may be negligible. Moreover, because of the inaccuracies of the energy model, the predicted ground state is not always correct. The correct structure, as known from biochemical analysis, does however appear in the ensemble of structures with high probability.

The approaches are routinely used to overcome this problem: Zuker (Zuker 1989) devised a version of the minimum energy folding algorithm that computes a set of suboptimal structures in a certain energy range, see also (Jacobson & Zuker 1993). A more elegant solution was suggested by McCaskill (McCaskill 1990), who proposed an algorithm to compute the partition function of the thermodynamic ensemble and the matrix of base pairing probabilities of an RNA molecule. The representation of the base pair probabilities is done in a dot plot, where the probability for a base pair is symbolized by the size of a square. Different competing structure elements can be shown, this gives us an idea of the variability of the structure. The Vienna RNA Package (Hofacker *et al.*) provides an efficient serial implementation of both the minimum free energy and the partition function algorithm, the algorithms are only limited by the resources of the present day computers.

The partition function algorithm, however, is quite demanding both in terms of memory and CPU time. The algorithm requires CPU time that scales roughly as the cubic power of the sequence length, and memory that scales quadratically with sequence length. This is not a problem for small RNA molecules such as tRNAs. For large RNA molecules such as viral genomes, memory, rather than computational speed, is usually the fundamental resource bottleneck. The complete genome of bacteriophage $Q\beta$ (4220 bases) was folded on an IBM RISC6000/550 workstation in about 10 h (Hofacker *et al.* 1994a)

using 340 MegaBytes of memory. Folding a large virus like HIV1 with about 9200 bases the folding procedure would last about 105 h, on the same computer, and was performed on a CRAY-M90 (Huynen *et al.* 1996), using 63 h of CPU time and 1.6 GigaBytes of memory. While using a few days of CPU times would often be acceptable, few workstations provide enough memory for this kind of calculations. On the other hand, these resources can easily be provided by modern parallel computers.

The implementation for massively parallel computers of the folding algorithms developed in this diploma thesis was based on the philosophy, that memory is the fundamental resource bottleneck, rather than computational speed. Even though CPU time grows as the cubic power of chain length, sequences such as HIV that are approximately 10000 bases in length still require only on the order of 84 min to fold on 384 nodes of the Intel Delta supercomputer. The use of parallel computers puts us in the position to predict secondary structures of available virus genomes in rather short time.

As a first application and test of the parallel algorithm, the partition function and base pair probabilities for the complete virus genome of HIV1_{LAI} were calculated on the DELTA and compared to the results obtained from the serial program running on the CRAY (Huynen *et al.* 1996), see section 9. The secondary prediction of all available virus genomes now seems to be a feasible computational task, and will help identify important secondary structure motifs and their role in the viral life cycle.

2 RNA Secondary Structures

The presentation of this section follows the dissertation of I. Hofacker (Hofacker 1994). RNA molecules consist of ribonucleotides linked together by covalent chemical bonds. Each ribonucleotides 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 hydrogen bonds with a complementary base: A binds with U, G with C (Watson-Crick base pairs).

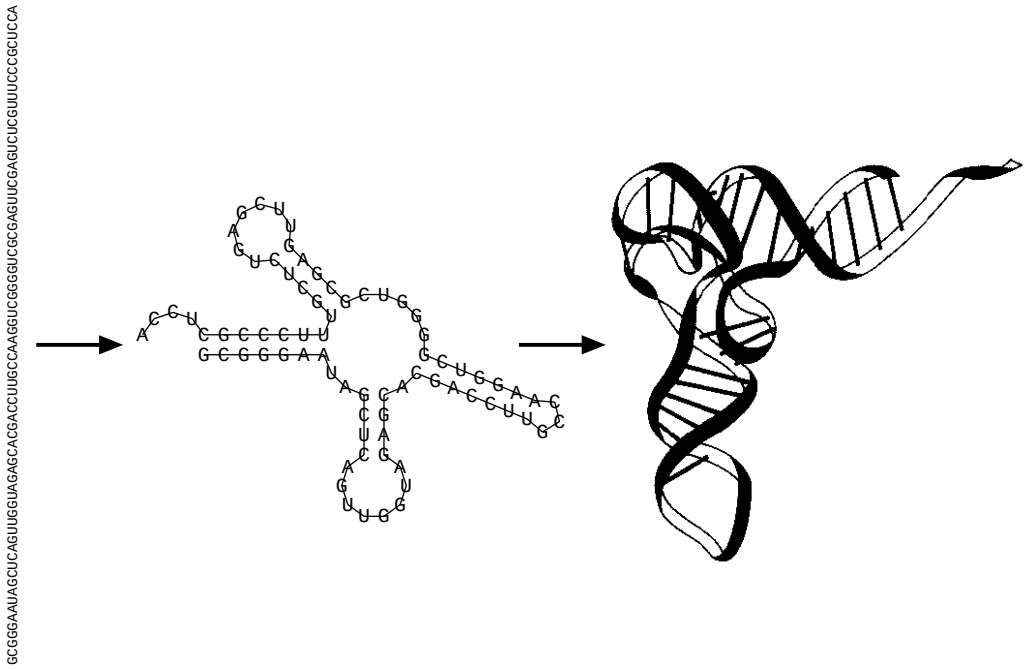


Figure 1: Folding of an RNA sequence into its spatial structure. The process is partitioned into two phases: in the first phase only the Watson-Crick-type base pairs are formed which constitute the major fraction of the free energy, and in the second phase the actual spatial structure is built by folding 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.

Much like DNA, RNA can form stable double helices of complementary 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 4. 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 exceeds that in polypeptides, the full structural prediction problem is hard to solve. However, for RNA it is possible to focus on an intermediate level representation of the folding. This secondary structure representation contains only information on which base pairs are formed and relegates more detailed and additional information to a later and subsequent stage of analysis. The resulting secondary structures are useful in the prediction of the full 3D structures and in the interpretation of the biochemical function of the molecules for several reasons:

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

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

(3) Secondary structures are conserved in evolutionary phylogeny.

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

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

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

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

In the following section we will give a formal definition of secondary structures as graphs: RNA secondary structures can be represented as planar

vertex-labeled graphs or as trees. Note that our definition ranks pseudo-knots as a tertiary interaction. Although pseudo-knots seem to be important for biological function, their inclusion would complicate the mathematical and computational treatment unduly.

3 Secondary Structure

3.1 Definitions

The presentation of this section follows the dissertation of I. Hofacker (Hofacker 1994).

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

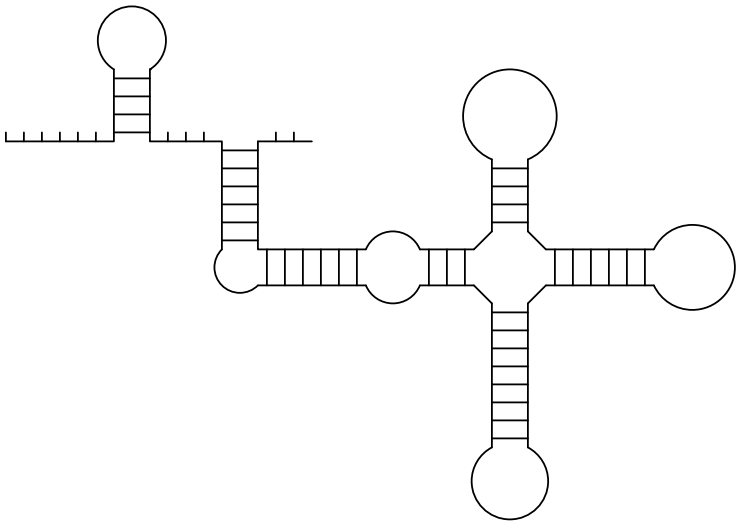


Figure 2: An example for an RNA secondary structure, with free dangling ends, stems and loops.

that the structure contains no pseudo-knots. A vertex i is said to be *interior* to the base pair (k, l) if $k < i < l$. If, in addition, there is no base pair (p, q) such that $p < i < q$, we will say that i is *immediately interior* to the base pair (k, l). A base pair (p, q) is said to be (immediately) interior, if p and q are (immediately) interior to (k, l).

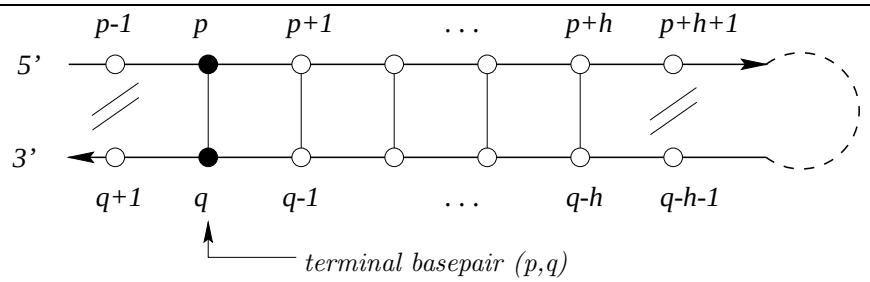
Definition 3.2. A secondary structure consists of the following structure elements

(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.



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 $k = 2$.

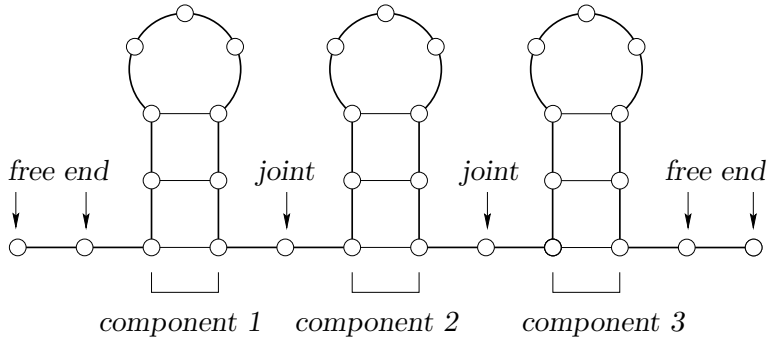


Figure 3: An example for an RNA secondary structure consisting of three components and six external vertices (2 joints and 4 free ends).

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 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 *immediately 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).

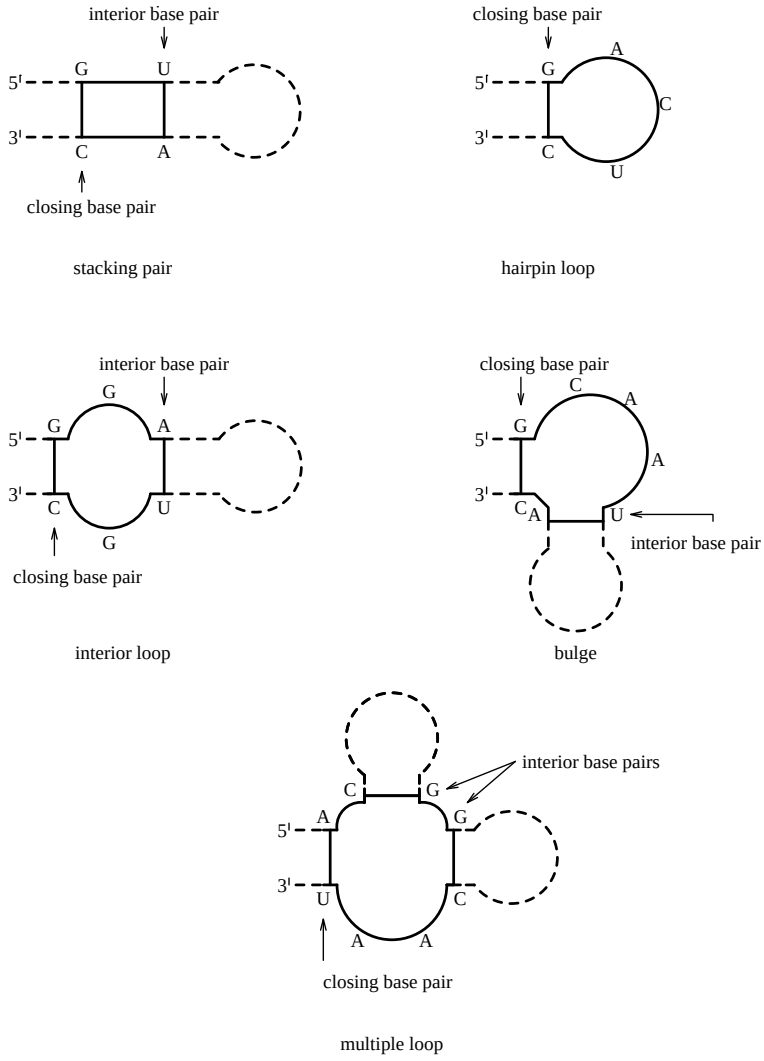


Figure 4: Classification of the loops arising in the decomposition of RNA secondary structure.

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 Structure

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.

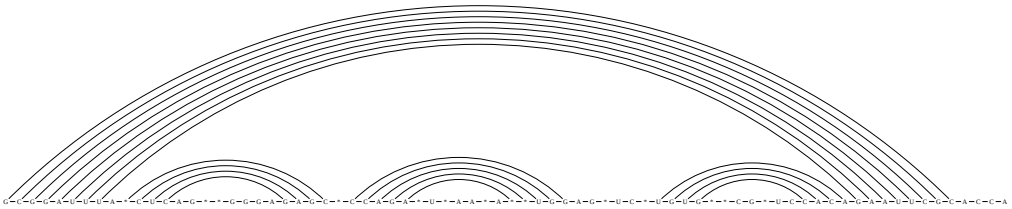


Figure 5: The secondary structure of tRNA^{Phe} in *linked graph representation*.

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 8 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.

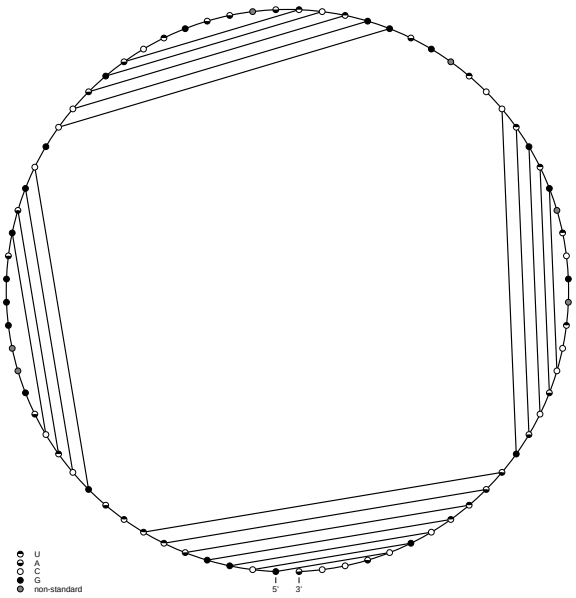


Figure 6: The secondary structure of tRNA^{Phe} in *Circular representation*.

- *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 $i < k$ and $j > k$. The mountain representation allows straightforward comparison of secondary structures and inspired a convenient algorithm for alignment of secondary structures (Konings & Hogeweg 1989).

The presentation of an ensemble of structures obtained by the partition function algorithm can be represented by a matrix of base pair probabilities. Therefore various competitive structures can be displayed in a dot plot, and

give us much more information than only one single minimum free energy structure. A dot plot is a two-dimensional graph in which the size of the dot at position (i, j) within the graph represents the probability P_{ij} of the base pair. We obtained our dot plots here using PSdotplot from the Vienna RNA Package (Hofacker *et al.*). In principle dot plots contain complete base pairing information, in practice we suppress the dots corresponding to base pairs that occur with a probability of less than 10^{-5}. Figure 7 shows the tRNA^{Phe} as an example. The plot is divided into two triangles. The upper right triangle contains the base pairing probability matrix (P_{ij}); the size of the squares is proportional to the pairing probability. The lower-left triangle displays the minimum free energy structure for comparison.

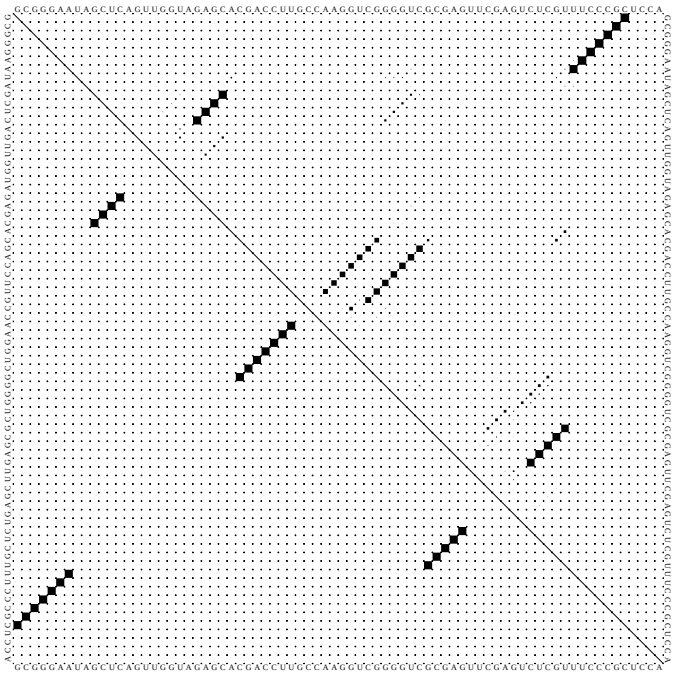


Figure 7: RNA Dot Plot of tRNA^{Phe}. The above triangle shows the base pair probability, and the upper the minimum free energy.

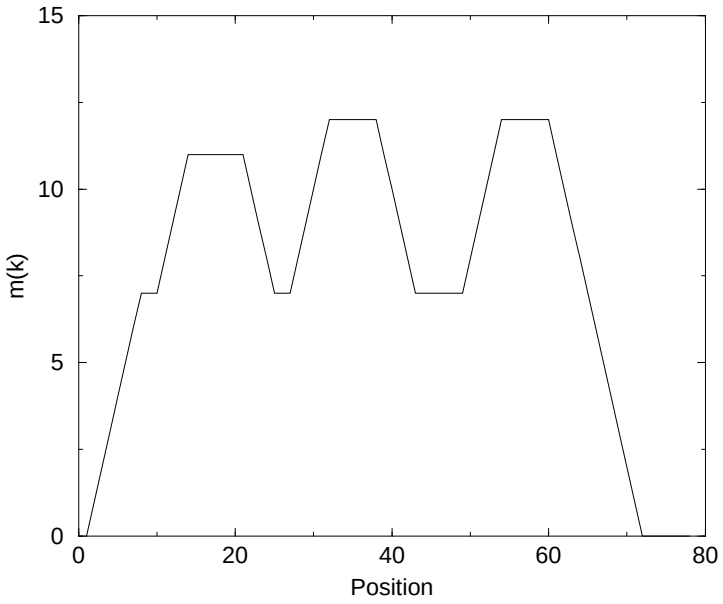


Figure 8: The secondary structure of $tRNA^{Phe}$ from Yeast (see Figure 1) in *original mountain representation*. The same structure in string representation is `(((((((..((((........)))).(((((.......)))))).....(((((.......)))))))))))....`

The generalized mountain representation is displaying the size and distribution of secondary structure elements as a modified version of the *mountain representation*. In the original mountain representation, see Figure 8 only a single secondary structure is represented in a two dimensional graph. In the graph the x-coordinates are the positions in the sequence, whereas the y-coordinates are proportional to the number of base-pairs by which every nucleotide is enclosed. We construct the modified version of the mountain representation as follows: Let us consider the numbers

$$m_k := \sum_{i<k} \sum_{j>k} P_{ij}$$

for all sequence positions k. By definition, m_k counts all base pairs which contain k (in the terminology of Zuker and Sankoff (Zuker & Sankoff 1984) that are all base pairs to which sequence position k is interior, weighted with their respective pairing probabilities.

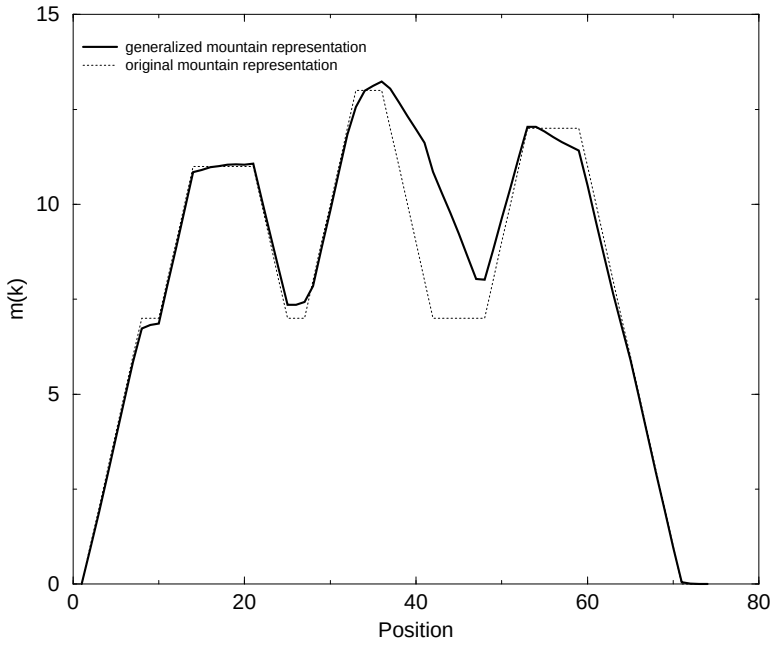


Figure 9: The secondary structure of $tRNA^{Phe}$ from Yeast (see Figure 1) in *mountain representation* and *generalized mountain representation*.

To see that this measure is in fact a close relative of the mountain representation, assume for a moment that P_{ij} is the pairing matrix of a minimum free energy structure. Thus $P_{ij} = 0$ or 1. In this case m_k is the number of base pairs that contain k, i.e., it is constant for all positions in the same loop, increases by one at each paired position at the 5' side of a stack, and decreases by one at each paired position at the 3' side of a stack. The generalized mountain representation gives a weighted average of the ensemble of secondary structures. The y-coordinate of base k corresponds to the number of base pairs that is expected to enclose k on average. In the original mountain representation the steepness of the slope can have any downstream minus the probability of being paired to a base upstream. Figure 9 shows a generalized mountain representation of $tRNA^{Phe}$ and its original mountain representation.

4 The Energy Model

4.1 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 Berlin 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. The presentation of this section follows the diploma thesis of J. Cupal (Cupal 1997).

4.1.1 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 electro-negativity. 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 polarization lead to additivity and cooperativity of the bond forming process: H becomes more electro-positive, 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 10. *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.

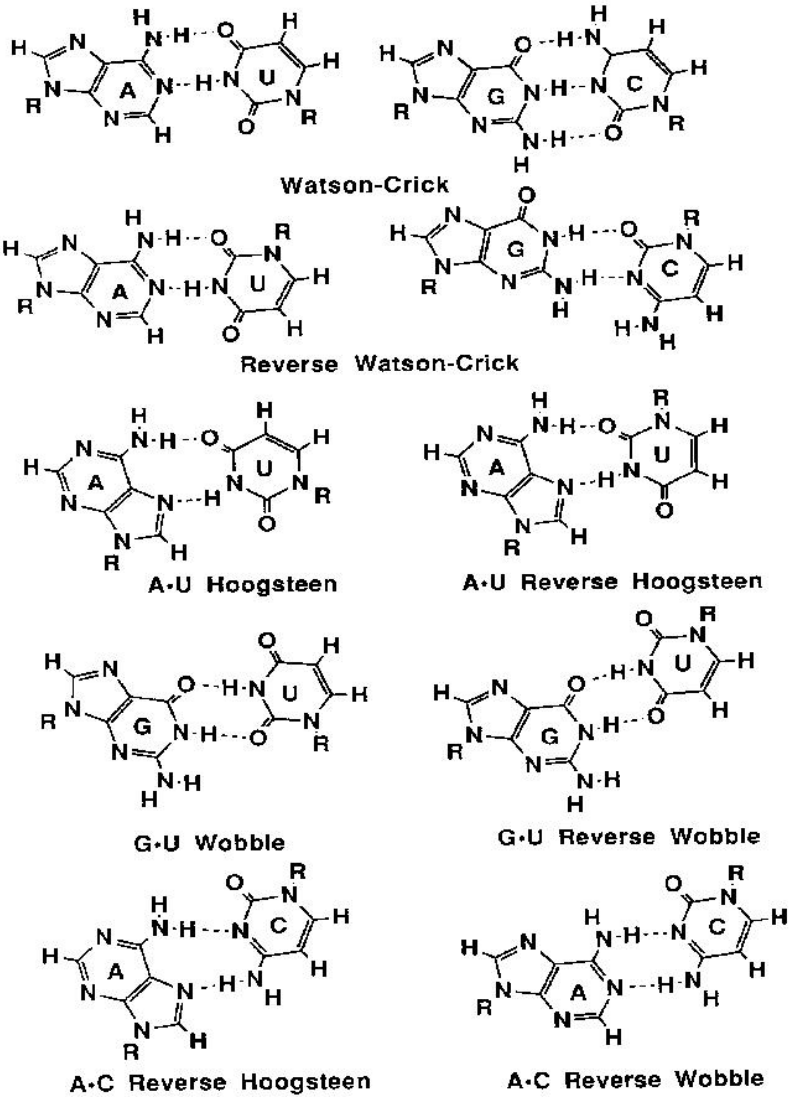


Figure 10: The ten possible purine-pyrimidine base pairs (Saenger 1984; Tinoco CSHL Press 1993).

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 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.

4.1.2 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

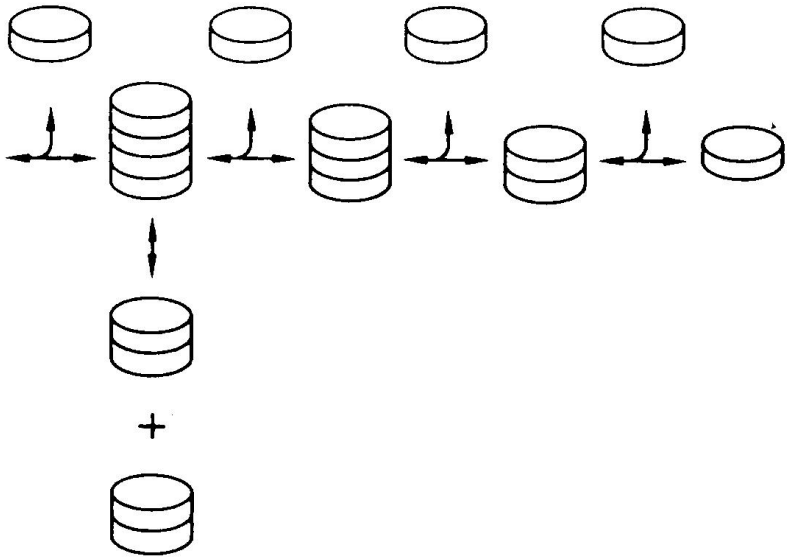


Figure 11: Reaction Scheme of base stacking (Saenger 1984; Poerschke Berlin 1977).

(Poerschke Berlin 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 11. Thermodynamic parameters for the self-association (stacking) of nucleosides 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 $kT = 0.6\,\mathrm{kcal/mol}$. 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 12. 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 non-cooperative (Poerschke Berlin 1977).

Forces mainly contributing to the stabilization 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 $^{5'}\text{C-G}^{3'}_{3'}\text{G-C}^{5'}$. 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 behavior 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

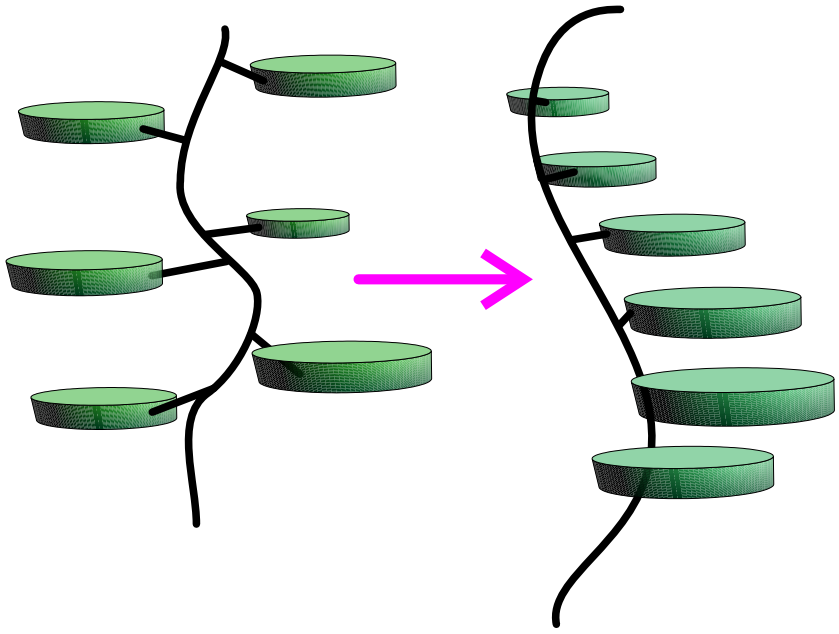


Figure 12: Base stacking to *poly*A single stranded helices (Saenger 1984; Poerschke Berlin 1977).

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.

4.2 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 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.

$$E(\Phi) = \sum_{L \in \Phi} \epsilon(L) + \epsilon(L_{ext}), \tag{1}$$

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 multi-loops 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 ($k > 9$) are extrapolated logarithmically:

$$\mathcal{H}(k) = \mathcal{H}(30) + \text{const.} * \log(k/9) \tag{2}$$

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.

$$\Delta F_{\text{ninio}} = \min\left\{\Delta F_{\max}, |u_1 - u_2| * \Delta F_{\text{ninio}}\left[\min\{4.0, u_1, u_2\}\right]\right\} \tag{3}$$

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 & Ignacio 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

$$\Delta G = \mathcal{M}_{\mathcal{C}} + \mathcal{M}_{\mathcal{I}} \cdot \text{degree} + \mathcal{M}_{\mathcal{B}}, \tag{4}$$

depending on the size of the loop and $\mathcal{M}_{\mathcal{B}}$ the number of base pairs interior to the closing base pair degree, i.e. its degree-1. Good results have been achieved using $\mathcal{M}_{\mathcal{C}} = 4.6$, $\mathcal{M}_{\mathcal{I}} = 0.4$ and $\mathcal{M}_{\mathcal{B}} = 0.1$ 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 parameters compiled in (Freier *et al.* 1986; Turner, Sugimoto, & Freier 1988; He *et al.* 1991), who performed measurements at 37°C in 1 M NaCl. A variety of modifications might be in order in the future. The differences between symmetric and asymmetric loops have newly been reported to be only half the magnitude suggested by Papanicolau (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 G-A 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) *Multi-loops* ($k \geq 2$): Multi-loop energies $\mathcal{M}$ are modeled by the linear ansatz

$$\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot \text{degree} + \mathcal{M}_B \cdot \text{unpaired}, \tag{5}$$

where $\mathcal{M}_C$ denotes the multi-loop 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).

In this description the dangling end energies and mismatch energies for multi-loops are not considered.

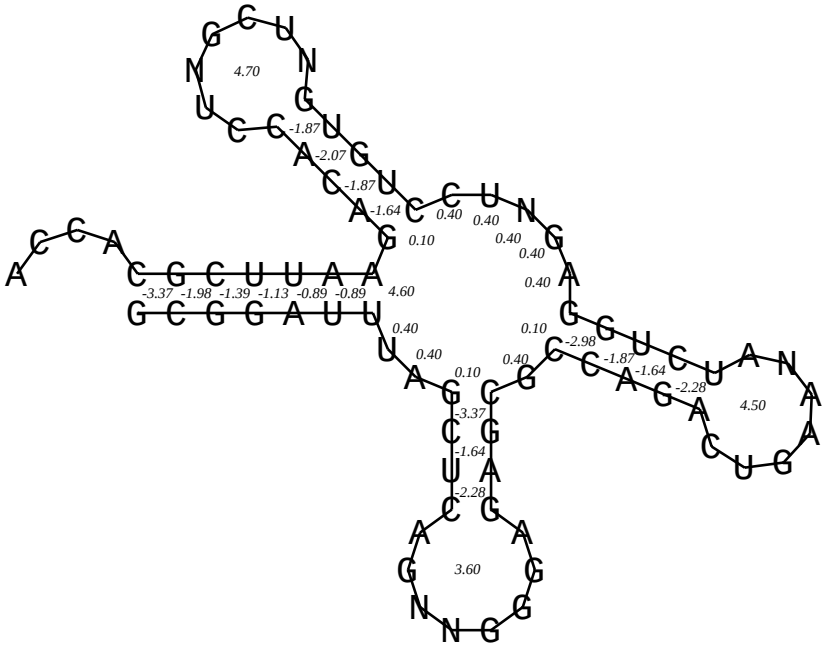


Figure 13: The secondary structure of Yeast tRNA$^{\mathrm{Phe}}$. The sequence ($n = 76$) is taken from the EBI database (Steegborn *et al.* 1995): `GCGGAUUUALCUCAGDDGGGAGA-GCRCCAGABU#AAYAP?UGGAG7UC?UGUGTPCG"UCCACAGAA UUCGCACCA.`
The Free Energy of the structure according to the energy model used in this work without dangling end energies and mismatch energies for multi-loops is -12.26 kcal/mol. Multi-loop energies are $\mathcal{M}_C = 4.60$ kcal/mol, $\mathcal{M}_B = 0.40$ kcal/mol and $\mathcal{M}_I = 0.10$ kcal/mol. See the appendix for the abbreviation and translation of modified bases.

5 Folding Algorithms

5.1 Computing the Partition Function

In this section we explain in detail the algorithm for the computation of the partition function of an RNA molecule first derived by McCaskill (McCaskill 1990). In section 5.2 we explain the recursion that is used for calculating the base pair probability, called backtracking.

The free energy F is related to the *partition function* Q by

$$F = -kT \ln Q, \tag{6}$$

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

$$Q = \sum_{\Phi \in \mathcal{M}} e^{-F(\Phi)/kT}, \tag{7}$$

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

The additivity of free energy contribution of the various loops L of a structure Φ, see equ. (1), implies a multiplicativity in the partition function Q.

$$Q = \sum_{\Phi \in \mathcal{M}} e^{-\left[\sum_{L \in \Phi} F_L\right]/kT} \tag{8}$$

$$= \sum_{\Phi \in \mathcal{M}} \prod_{L \in \Phi} e^{-F_L/kT} \tag{9}$$

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.

$$Q = \sum_{\Phi \in \mathcal{M}} \prod_{\mathcal{S} \in \Phi} \prod_{L \in \mathcal{S}} e^{-F_L/kT} \tag{10}$$

$$= \sum_{\Phi \in \mathcal{M}} \prod_{\mathcal{S} \in \Phi} e^{-F_S/kT} \tag{11}$$

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

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 14. $\mathcal{S}_1$ 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 partition function of structures in this subset is denoted Q^A_{ij}. $\mathcal{S}_3$ 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. 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 15. 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. Using this decomposition we can calculate the whole partition

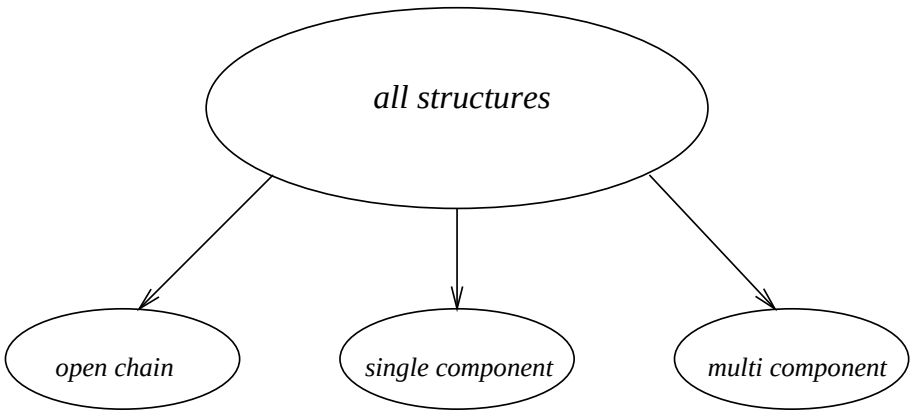


Figure 14: The complete set $\mathcal{S}$ of all secondary structures Φ compatible to string $[\sigma_i ... \sigma_j]$ is split into subsets $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$.

function of the structure out of smaller parts. The complete partition function Q_{ij} on the string $[\sigma_i ... \sigma_j]$ is therefore the sum of the contributions of the three subsets:

$$Q_{ij} = Q_{ij}(\mathcal{S}_1) + Q_{ij}(\mathcal{S}_2) + Q_{ij}(\mathcal{S}_3) \tag{12}$$

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_{ij}. The set of multi-component 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_{kj} 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 15. Therefore, we obtain

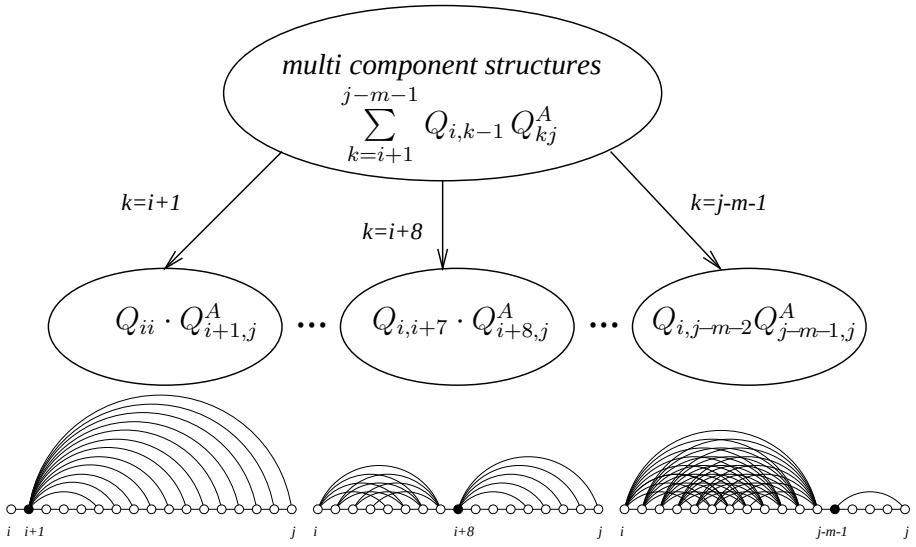


Figure 15: The contribution of each set is derived by the multiplication of the unconstrained partition function on the left substring times the contribution of all single-component structures on the right string. Summing up yields the total contribution of all multi-component structures.

for the complete partition function

$$Q_{ij} = 1 + Q^A_{ij} + \sum_{k=i+1}^{j-m-1} Q_{i,k-1} Q^A_{kj} \tag{13}$$

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

$$Q^A_{ij} = \sum_{l=i+m+1}^{j} Q^B_{il} \tag{14}$$

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

$$Q^B_{ij} = \sum_L e^{-F_L/kT} \prod_{\substack{(h,l)\in L \\ i<h<l<j}} Q^B_{hl} \tag{15}$$

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 multi-loop, 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 types, equ. (15) can be rewritten as

$$\begin{aligned} Q^B_{ij} \;=\; & e^{-\mathcal{H}(ij)/kT} + \sum_{k=i+1}^{j-m-2} \sum_{l=k+m+1}^{j-1} Q^B_{kl}\, e^{-\mathcal{I}(i,j,k,l)]/kT} \\ & + \sum_{k=i+1}^{j-m-2} Q^M_{i+1,k-1} Q^{M1}_{k,j-1} e^{-\mathcal{M}c/kT} \end{aligned} \tag{16}$$

Calligraphic symbols $\mathcal{H}, \mathcal{I}, \mathcal{M}$ refer to the classification of loops described in the previous sections according to the degree of the loops. The number of base pairs immediately interior the loop gives the value of k ($k = 0 \rightarrow$ hairpin loop, $k = 1 \rightarrow$ stack, interior loop, bulge). The computation of interior loops is of order $O(n^4)$ if we do not restrict loop size. Fortunately we can restrict the size of long interior loops to $u < u_{max}$, because larger loops can regarded as

prohibitive. The calculation of interior loops is therefore of order $O(n^2)$ and proportional to u_{max}^2.

$$interiorloop : Q_{ij}^B = \sum_{\substack{k=i+1 \\ u \le u_{max}}}^{j-m-2} \sum_{l=k+m+1}^{j-1} Q_{kl}^B \, e^{-\mathcal{I}(i,j,k,l)]/kT} \tag{17}$$

The third term in equ. (16) represents the multiple loop contribution, see Figure 16. We obtain for the multi-loop contributions

$$Q_{ij}^M = \sum_{k=i+m+1}^{j-m-1} Q_{i,k-1}^M \, Q_{kj}^{M1} + \sum_{k=i}^{j-m-1} Q_{kj}^{M1} \, e^{-\mathcal{M}_{\mathcal{B}}(k-i)/kT} \tag{18}$$

with $Q_{ii}^M = 0$ and $Q_{i+1,i}^M = 0$.

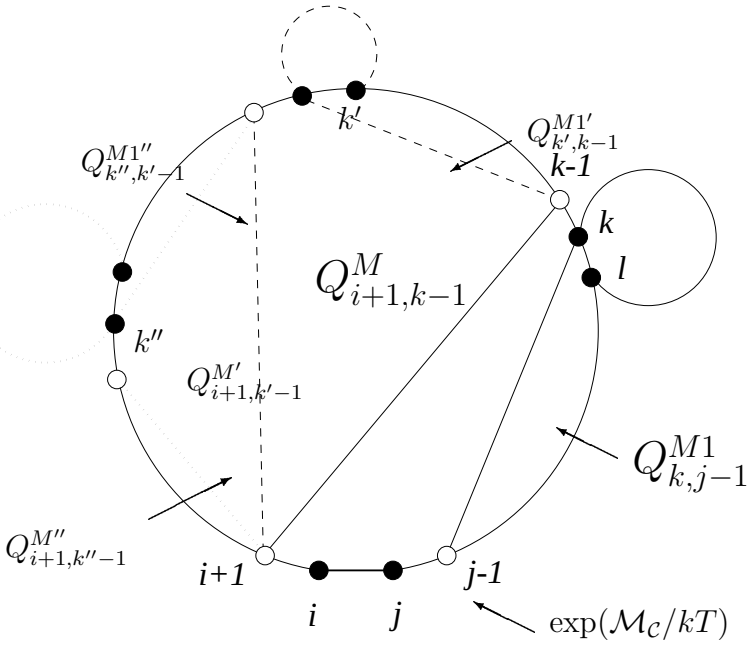


Figure 16: Recursive decomposition of multi-loops and multi-loop energies in the partition function: Multi-loop structures are constructed from the closing base pair (i, j) with *multi-loop closing energy* $\mathcal{M}_{\mathcal{C}}$, a region running from k to $j-1$ containing a single component with a possible tailing end at the right side, and a region running from $i+1$ to $k-1$ containing an arbitrary structure. Multi-loop energy contributions are attributed to individual vertices or base pairs and are additive, see equ. (18).

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

Table 1: **Recursion for the calculation of the partition function:** Calligraphic symbols denote energy parameters for different loop types: hairpin loops $\mathcal{H}(ij)$, 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 $[ij]$ 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 and restricting the size of interior loops to u_{max}, equ. (17) helps reducing the CPU requirements to $O(n^3)$. The unconstrained partition function of the substring $[ij]$ 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.

The contribution of all structures forming a single rightmost stem, Q_{kj}^{M1}, is obtained to:

$$Q_{ij}^{M1} = \sum_{l=i+m+1}^{j} Q_{il}^{B}\, e^{-[\mathcal{M}_{\mathcal{I}}+\mathcal{M}_{\mathcal{B}}(j-l)]/kT} \tag{19}$$

Table 1 summarizes the recursion scheme for the partition function. The next section will extend the recursion scheme to the computation of the base pair probability.

5.2 Calculating the Base Pair Probability: Backtracking

The partition function $Q = Q_{1n}$ can be used to calculate the thermodynamic quantities of the RNA structure. Much more interesting, however, is the wealth of structural information that is contained in the probability of base pairing. It allows us to examine secondary structures much precisely than using only the minimum free energy structure. The base pairing matrix can be obtained by the so called backtracking from the Q_{ij}'s.

The probability of a given structure ϕ with energy E is proportional to the $\exp(-F/kT)$.

$$P(\phi) = \frac{1}{Q} e^{-F(\phi)/kT} \tag{20}$$

We define P_{hl} as the probability that h is bound to l in the equilibrium ensemble of structures

$$P_{hl} = \sum_{\substack{\Phi \\ (h,l)\in\Phi}} P(\Phi) = \frac{Q^{(h,l)}}{Q} \tag{21}$$

where $Q^{(h,l)}$ is the partition function over all structures containing the pair (h,l).

There are three possibilities for a base pair (h,l). It can close up a single component, with no external base pairs, or (h,l) is interior to an interior loop closed by (i,j) or (h,l) is interior a multi-loop.

The probability for a base pair closing a component is calculated as

$$P_{hl}^{component} = \frac{Q_{1,h-1} Q_{hl}^{B} Q_{l+1,n}}{Q_{1n}} \tag{22}$$

As seen in Figure 17 the base pair (h, l) splits the structure into three independent substructures giving rise to the three factors in equ. (22). The calculation uses the quantities Q_{ij} and Q_{ij}^{B} computed in the first part of the program.

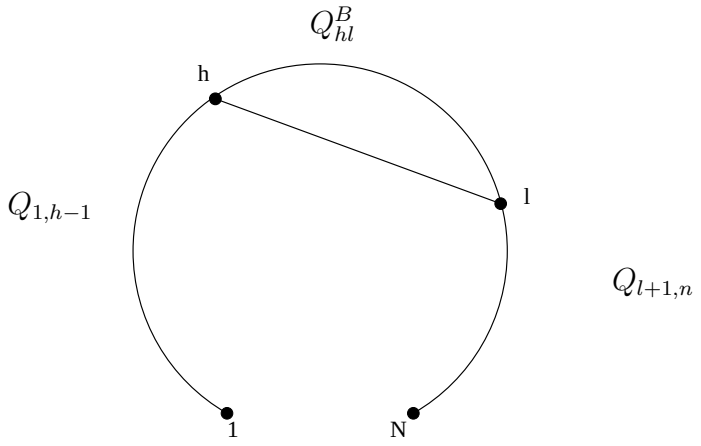


Figure 17: Probability of P_{hl} closing component

The situation where (h, l) forms an interior loop is depicted in Figure 18, and leads to

$$P_{hl}^{interiorloop} = \sum_{\substack{ij \\ i<h<l<j}} P_{ij} \frac{Q_{hl}^{B}}{Q_{ij}^{B}} e^{\mathcal{I}(i,j,h,l)/kT} \tag{23}$$

where $\frac{Q_{hl}^{B}}{Q_{ij}^{B}} \exp(\mathcal{I}(i, j, h, l)/kT)$ is the conditional probability of finding the pair (h, l) given the pair (i, j). Since we have to sum up over all possible base pairs (i, j) fulfilling $(i < h < l < j)$ the calculation, as formulated, would be of order $O(n^4)$. Because we have restricted the size of interior loops in section 5.1, we have to follow the same restriction here, thereby reducing the calculation to

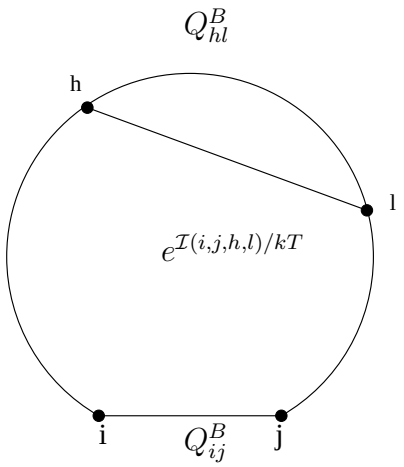


Figure 18: Probability of P_{hl} inside interior loop

order $O(n^2)$.

$$P_{hl}^{interiorloop} = \sum_{\substack{ij \\ i<h<l<j; u<u_{max}}} P_{ij} \frac{Q^B_{hl}}{Q^B_{ij}} e^{\mathcal{I}(i,j,h,l)/kT} \tag{24}$$

For base pairs inside a multi-loop we have to consider the three different cases, depicted in Figure 19. All three cases are summed up to get the base

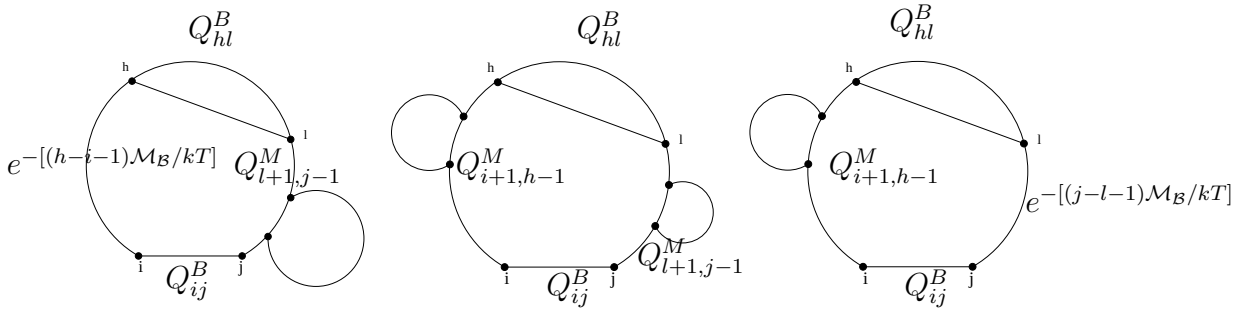


Figure 19: Three different possibilities for a base pair (h, l) interior of a multi-loop. left: the region $[i+1, h-1]$ is unpaired, $[l+1, j-1]$ contains at least one pair; right: $[l+1, j-1]$ unpaired, $[i+1, h-1]$ contains a pair; middle: both regions contain at least one pair.

pair probability for base pairs interior multi-loops.

$$P_{hl}^{multiloop} = \sum_{\substack{ij \\ i<h<l<j}} P_{ij} \frac{Q^B_{hl}}{Q^B_{ij}} e^{-[(\mathcal{M}_\rfloor + M_\mathcal{I})/kT]} \times$$

$$\Big(e^{-[(h-i-1)M_B/kT]}Q^M_{l+1,j-1} + e^{-[(j-l-1)\mathcal{M}_\mathcal{B}/kT]}Q^M_{i+1,h-1} + \\ +Q^M_{i+1,h-1}Q^M_{l+1,j-1}\Big) \tag{25}$$

The computational effort for computing the $P_{hl}^{multiloop}$ can be reduced to $O(n^3)$ at the expense of memory by introducing two additional quantities. We reduce the double sum over (i, j) to a single sum by introducing

$$P^M_{il} = \sum_{i>j} \frac{P_{ij}}{Q^B_{il}} Q^M_{l+1,j-1} \tag{26}$$

and

$$P^{M1}_{il} = \sum_{j>l} \frac{P_{ij}}{Q^B_{ij}} e^{-[(j-l-1)\mathcal{M}_\mathcal{B}/kT]} \tag{27}$$

These sums must be calculated at the appropriate point in the recursion, and have to be stored as two triangle matrices. The expression for P_{hl} can then be written:

$$\begin{aligned} P_{hl}^{multiloop} &= \sum_{i<h} Q^B_{hl} e^{-[(\mathcal{M}_\rfloor + \mathcal{M}_\mathcal{I})/kT]} \times \\ &\quad \Big(P^{M1}_{il} Q^M_{i+1,h-1} + P^M_{il}\big(e^{-[(h-i-1)\mathcal{M}_\mathcal{B}/kT]} + Q^M_{1+1,h-1}\big)\Big) \end{aligned} \tag{28}$$

As discussed previously, the probability of a certain base pair P_{hl} is finally given by the sum of all three possibilities.

$$P_{hl} = P_{hl}^{component} + P_{hl}^{interiorloop} + P_{hl}^{multiloop} \tag{29}$$

5.3 The Problem of Large Numbers

The energies of secondary structures scale roughly linearly with sequence length. The partition function Q, consequently, grows exponentially, and can exceed the range of double precision floating point numbers even for sequences of only

a few hundred bases. The problem can be solved by rescaling. Let $\tilde{Q}$ be an estimate for Q. We can now rescale the partition function by $\tilde{Q}$ to obtain a value near 1. For each subsequence of length l we rescale $Q_{i,i+l+1}$ by $\tilde{Q}^{l/n}$. The same scaling factor is also used for Q^B, Q^M and Q^{M1}. As can be seen by inspection of table 2 the recursions for the rescaled quantities stay essentially unchanged. The estimate $\tilde{Q}$ can be calculated if the minimum free energy E_{min} is already known, we use $\tilde{Q} = exp(1.04E_{min}/kT)$, where T is temperature in Kelvin and k is Boltzmann's constant. The factor 1.04 has been found to yield a good estimate.

$$
\begin{aligned}
P_{hl}^{component} &= \frac{Q_{1,h-1}Q_{hl}^{B}Q_{l+1,n}}{Q_{1n}} \\
P_{hl}^{interiorloop} &= \sum_{\substack{ij \\ i<h<l<j;u<u_{max}}} P_{ij}\frac{Q_{hl}^{B}}{Q_{ij}^{B}}e^{\mathcal{I}(i,j,h,l)/kT} \\
P_{hl}^{multiloop} &= \sum_{i<h} Q_{hl}^{B}e^{-[(\mathcal{M}_{\rfloor}+\mathcal{M}_{\mathcal{I}})/kT]} \times \\
&\quad \Big(P_{il}^{M1}Q_{i+1,h-1}^{M} + P_{il}^{M}\big(e^{-[(h-i-1)\mathcal{M}_{\mathcal{B}}/kT]} + Q_{i+1,h-1}^{M}\big)\Big) \\
P_{il}^{M} &= \sum_{i>j} \frac{P_{ij}}{Q_{ij}^{B}}Q_{l+1,j-1}^{M} \\
P_{il}^{M1} &= \sum_{j>l} \frac{P_{ij}}{Q_{ij}^{B}}e^{-[(j-l-1)\mathcal{M}_{\mathcal{B}}/kT]} \\
P_{hl} &= P_{hl}^{component} + P_{hl}^{interiorloop} + P_{hl}^{multiloop}
\end{aligned}
$$

Table 2: **Recursion for the calculation of base pair probability:** Calligraphic symbols denote energy parameters for different loop types: hairpin loops $\mathcal{H}(ij)$, 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 quantities Q_{ij}^{B}, $Q_{ij}^{M}, Q_{1,j}$ and $Q_{i,n}$ were calculated in section 5.1 and have to be stored for the backtracking. 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 P^{M} and P^{M1} are necessary for handling multi-loops (McCaskill 1990), Restricting the size of interior loops to u_{max}, equ. (25) helps reducing the CPU requirements to $O(n^3)$. The first term, $P_{hl}^{component}$, describes the probability of a base pair closing a component, $P_{hl}^{interiorloop}$ denotes the probability of a base pair closing an interior loop. The calculation of $P_{hl}^{multiloop}$ is reduced to $O(n^3)$ by introducing P_{il}^{M} and P_{il}^{M1}, these quantities have to be stored during computation.

5.4 Computing the Minimum Free Energy

The Problem of large numbers see section 5.3 forces us to calculate a minimum free energy of the sequence. The minimum free energy algorithm of the ground state secondary structure (Zuker & Stiegler 1981; Zuker & Sankoff 1984; Hofacker *et al.* 1994a) relys on the same mechanisms and displays the same CPU requirements as the partition function: (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; earlier computed values for substrings yield values for larger strings, thus reducing CPU requirements to $\mathcal{O}(n^3)$. In the following description of the recursions, dangling ends have been neglected for clarity. Our implementation does, however, include them.

Let F_{ij}^B be the minimum free energy of all structures on $[\sigma_i ... \sigma_j]$, which are enclosed by (i, j), i. e. $(i, j) \in \Phi_{ij}$. Three subsets are contributing to this set of structures, depending on the number of base pairs immediately interior to (i, j), see equ. (16). The minimum energies of these three subsets are again (recursively) obtained from smaller fragments:

$$\begin{aligned} F_{ij}^B \;=\; \min \Big\{ & \mathcal{H}(ij), \min_{\substack{k\in[i+1,j-m-2]\\ l\in[k+m+1,j-1]}} \left\{ F_{kl}^B + \mathcal{I}(i,j,k,l) \right\}, \\ & \min_{k\in[i+1,j-m-2]} \left\{ F_{i+1,k-1}^M + F_{k,j-1}^M + \mathcal{M}_\mathcal{C} \right\} \Big\} \end{aligned} \tag{30}$$

$\mathcal{H}(ij)$ 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_{kl}^B, and the energy of the closing loop, $\mathcal{I}(i, j, k, l)$. Multi-loop structures enclosed by (i, j) are obtained by constructing the multi-loop from two sections, see Figure 16. The minimum free energy is thus the sum of the minimum energy of the two parts, $F_{i+1,k-1}^M$ and $F_{k,j-1}^M$, plus the multi-loop closing energy $\mathcal{M}_\mathcal{C}$. $F_{k,j-1}^M$ denotes the minimum free energy of the rightmost stem plus an arbitrary number of unpaired bases at the right side and is ob-

tained from the sum of the minimum energy of the stem, F^B_{il}, the multi-loop base energy, $\mathcal{M}_\mathcal{B}(j-l)$, which is added for each unpaired base, and the multi-loop internal energy, $\mathcal{M}_\mathcal{I}$.

$$F^M_{ij} = \min_{l\in[i+m+1,j]} \Big\{ F^B_{il} + \mathcal{M}_\mathcal{B}(j-l) + \mathcal{M}_\mathcal{I} \Big\} \tag{31}$$

$F^M_{i+1,k-1}$, equ. (30), denotes the minimum free energy of the remaining section of a multi-loop structure, see Figure 16. This section may contain one or more stems. In analogy with equ. (18), we derive for the minimum free energy

$$F^M_{ij} = \min \Big\{ \min_{k\in[i+m+1,j-m-1]} \big\{F^M_{i,k-1} + F^M_{k,j}\big\}, \tag{32}$$

$$\min_{k\in[i,j-m-1]} \big\{F^M_{kj} + \mathcal{M}_\mathcal{B}(k-i)\,\big\}\Big\}. \tag{33}$$

The first element yields the minimum energy of all multi-loop 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 multi-loop substructures, 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 obtained from the sum of the minimum energy of the stem plus the bases at the right side, F^M_{kj}, see equ. (31), plus the energy of the unpaired bases at the left side of the stem, $\mathcal{M}_\mathcal{B}(k-i)$.

The minimum free energy F_{ij} of *all* structures on string $[\sigma_i...\sigma_j]$ can now be obtained from the F^B as

$$F_{ij} = \min_{k\in[i+1,j-m-1]} \Big\{F_{i.j-1},\ \big[F_{i,k-1} + F^B_{kj}\big]\ \Big\}. \tag{34}$$

The first term, describes the case that j is unpaired, forming a free end. The second term, $F_{i,k-1} + F^B_{kj}$, describes structures where j is paired, and therefore, the pair (k,j) closes a component.

Table 3 summarizes the algorithm for the computation of the minimum free energy.

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

Table 3: **Recursion for the calculation of the minimum free energy:** Calligraphic symbols denote energy parameters for different loop types: hairpin loops $\mathcal{H}(ij)$, 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_{ij}^{B} 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 variable F_{ij}^{M} contains the minimum free energy the substructures on the sequence $[i,j]$ subject to the condition that i and j are part of a multi-loop. The unconstrained minimum free energy of the substring $[i,j]$ is stored in F_{ij}. The first term accounts for the case where j is unpaired and forms a free end. The second term collects all structures that consist of a component closed by the pair (k,j) and an arbitrary structure on the substring $[i,k-1]$. Not all entries of F_{ij} need be computed, it is sufficient to calculate e.g. the first row F_{1j}.

6 Hardware: Parallel Computers

6.1 The Intel iPSC Hypercube Parallel Computer

The parallel algorithm of the the minimum free energy and partition function described in section 5 were developed on an Intel Ipsc/860 distributed memory parallel computer with maximal 16 i860 processors and 8 Mbytes of memory for each processor. A typical iPSC system application has a host program that runs either on a local remote host or on the system resource manager (i386 processor) and a node program that runs on a group of allocated processors called nodes. The processing nodes are interconnected in a hypercube architecture. In a hypercube of dimension n, each node has n neighbors and the total number of nodes in the hypercube is 2^n, for developing our parallel algorithms, we used an Intel Ipsc/860 with 2^4 nodes. The iPSC/860 Operating System is a multi-user, multi-programming system. Each node on the iPSC/860 runs the NX/860 operating system. The host runs the UNIX operating system.

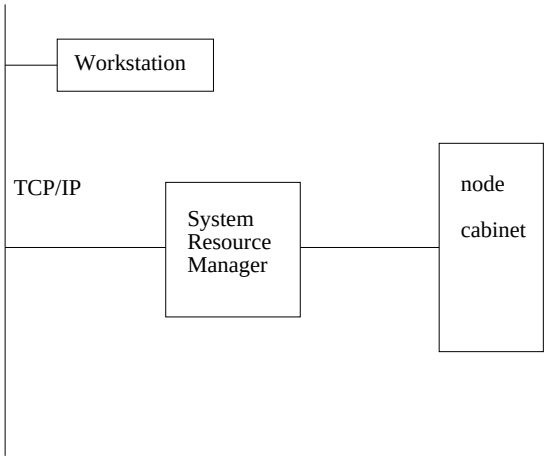


Figure 20: The iPSC distributed memory parallel computer architecture, see (Int 1990). Compilation of the program code is done on the system resource manager, also the host program runs on it. The node program runs on all allocated nodes of the node cabinet. Sending and receiving data from the local workstation is done by TCP/IP.

In our case the host program reads the input file starts the node program on all allocated nodes and sends the input to the first node. At the end of the program the host program receives the output of the node program and deallocates nodes.

6.2 The Intel Delta Parallel Computer

Data production was performed on the Touchstone DELTA System, see sections 8,9. The DELTA is a high-speed concurrent multicomputer, consisting of an ensemble of processors called nodes connected as a two-dimensional mesh. The DELTA System is also a distributed memory parallel computer with maximal 512 nodes and 16 M bytes of memory for each node.

The DELTA System itself contains four types of processing nodes:

- Numeric nodes, doing the calculations of the programs
- Mass storage nodes, providing the numeric nodes with access to the disk and tape drives.
- Gateway node, connecting the System to an Ethernet network.
- Service nodes, supporting a UNIX-like environment for user logged in to the mesh.

To run an application on the DELTA System, one has to put the executable code on the system, and issue on the system a start command that both allocates numeric nodes to the application and runs it on those nodes. The allocated nodes are released after completing the program. The mass storage nodes connected to the disks contain the Concurrent File System (CFS). The CFS is UNIX-like and distributes file blocks on all available disks, using read and write algorithms that allow several users to access the disks simultaneously. Gateway nodes provide the TCP/IP interface used for Ethernet access. Workstations on the network can exchange information with application programs running on the numeric nodes.

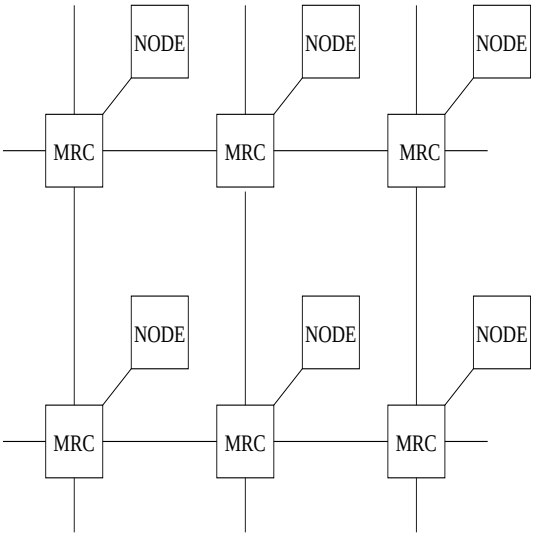


Figure 21: The Touchstone DELTA System mesh arrangement of the MRC node pairs

6.2.1 The Mesh Interconnect

The interconnection network is a two-dimensional mesh, where each node is connected to the mesh through a mesh routing chip (MRC). After the sending node transmits a message to its MRC, the message moves from MRC to MRC until it reaches the receiving node. No intermediate processors are interrupted. Only the sending and receiving nodes participate in the message transfer. Long messages are divided into packets and the MRCs transmit the packets over the network interleaved with messages from other nodes. In the most applications there is no need to minimize the distance of message passing, by allocating a special node arrangement.

6.2.2 System Description

The complete DELTA System contains 576 nodes, which are arranged as a 16 by 32 mesh in nine cabinets. Each cabinet contains four card cages, each with 16 nodes and arranged 4 by 4 mesh. In addition to the 16 node boards, each card cage also contains a Unit Service Module (USM) that performs system initialization and runs diagnostics under the control of the system console.

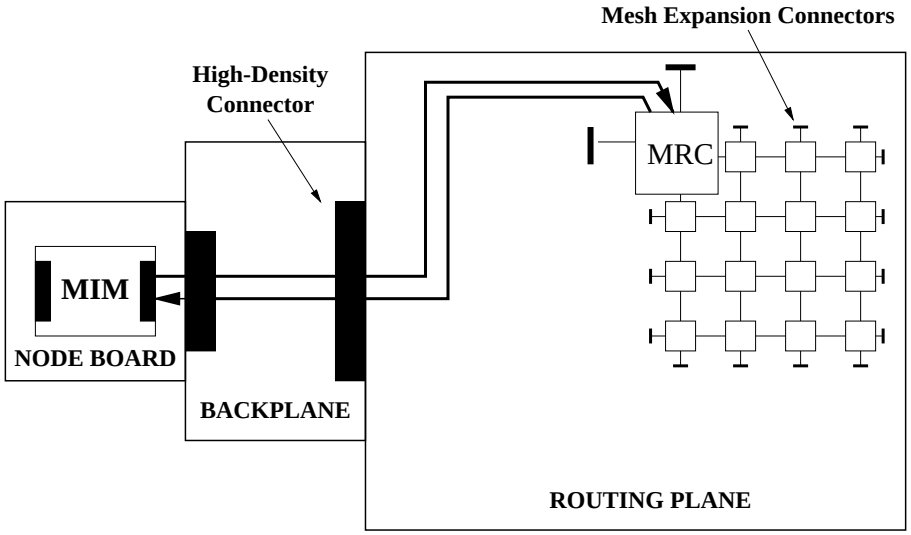


Figure 22: Mesh interconnection Hardware. Each node board has a daughter card called Mesh Interface Module (MIM) that is connected through the backplane to an MRC on a mesh routing plane. Each card cage is connected to its own routing plane. Each routing plane contains a 4 by 4 array of 16 MRCs, one MRC for each node in the card cage, see (Int 1991).

6.2.3 Types of Nodes

The numeric nodes are i860 processing boards. They are dedicated to executing numerically intensive tasks, like those found in scientific applications. The i860 operates at 40 MHz and is rated at 33 MIPs, 80 (peak) single-precision MFLOPS and 60 (peak) double-precision MFLOPS. Mass storage nodes provide I/O services to the numeric nodes. They are i386 processing boards with a SCSI interface. Either the disk drives containing the Concurrent File System or tape drives can be connected to the SCSI interface of the mass storage nodes. Gateway nodes, also i386 processing boards, connect the system to an Ethernet network. For the DELTA System it is a combination of a gateway node, a Bus Interface Adapter (BIA), and a Ethernet controller. Each gateway node has an Intel's Parallel Bus Interface (PBX), this is connected to a BIA that resides in a slot adjacent to the gateway node. Service nodes are i386 processing boards that allow login access to the system.

6.2.4 Message Passing

When a packet arrives at an MRC, the MRC examines the routing information in the packet header, there are the X-direction and the Y-direction determined. Routing is always first X then Y. If the MRC finds the X displacement to be none zero it routes the packet to the next MRC and decrements the X displacement. If the X displacements is zero routing is done in the Y direction. At the end of the routing the packet is transferred into the node. An error occurs if the Cycling Redundancy Check (CRC) word calculated by the receiver does not match the CRC at the end of the packet, or if the sender sends a packet to a non existing XY address, in which case the message is thrown away. If a message is too long it is divided into packets. Each packet is treated as an unique message and is sent through the MRC to the receiving node,there all packets are collected to reconstruct the whole message.

6.2.5 The NX/M Operating System

The node memory is limited to 16M-byte, but the NX/M operating system takes up 500K bytes and message passing buffering 3M-bytes, leaving 12.5M-bytes per node for user applications. The NX/M operating system provides message passing, memory management and process management capabilities. The message passing calls allow synchronous and asynchronous message exchange as well as interrupt-driven message handling. Synchronous calls block processing until the exchange is complete, while asynchronous calls permit processing to continue as the message is being passed. The handling of the message passing commands is the same as on the iPSC hypercube, but on the iPSC we use a host program necessary for allocating the nodes and for input output of the program. On the Delta mesh architecture we used a single node program loaded on the nodes. To log in to the mesh from a remote workstation you issue the rlogin command and specify the name of one of the gateway nodes.

6.2.6 The concurrent File System

The Concurrent File System treats all of the disks as a single logical disk with a single file system. Each file is distributed across the disk volumes in 4K-byte logical blocks. The volumes are numbered so that consecutive volumes are not on the same I/O node. The way CFS partition files makes it possible to transfer file portions in parallel when more than one compute node requests portion of a file residing on disks on different mass storage nodes.

7 Implementation of the Parallel Partition Function Algorithm

7.1 Parallel MFE fold

For calculating the partition function of an RNA sequence we have to calculate first the minimum free energy (MFE) of the sequence. The MFE is used to avoid overflows in floating point figures, see section 5.3. Although the memory requirements for calculating the MFE are less (we use integers instead of floats or doubles) than calculating the partition function we ported it to a parallel computer. A brief inspection to the algorithm in section 5.4 shows, that it can be parallelized quite easily. A message passing version of the MFE algorithm was already available (Hofacker *et al.* 1996). It required some modifications in order to accustom the improved energy model. At present we have omitted the backtracking, since we need the MFE only to rescale our partition function.

The principle of a parallel program is distributing the computational work on several processors, to speed up the calculation. In our case we have to compute all entries of the triangle matrices `C` and `FM` (corresponding to F^B and F^M in section 5.4). Since an entry for a substructure of length d depends only on smaller substructures $d' < d$, all entries on the diagonal d are independent of one another and can therefore be computed concurrently. The major computational difficulty is, that each entry on requires the explicit knowledge of a large number of previously calculated entries. This leads to a complicated message passing for each diagonal d on parallel computers with distributed memory. All entries of the diagonal d are distributed among the N processors, such that each processor has to calculate the same amount of entries. This guarantees good load balancing and consequently good parallel performance, see section 8.

In the following we present a parallelized version of the serial folding algorithm for distributed message passing systems. The distribution of data to the

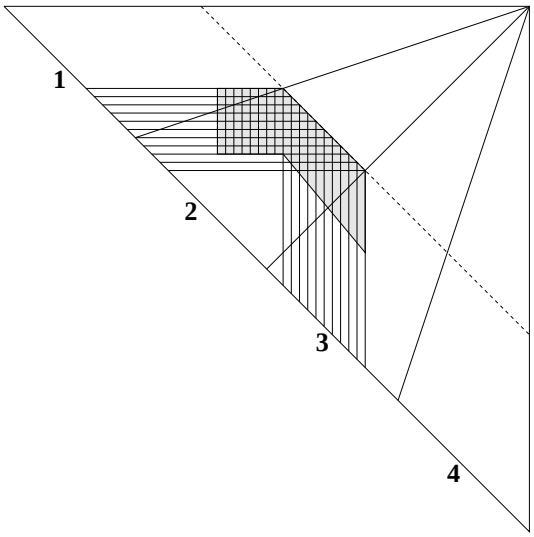


Figure 23: Memory requirement for calculating the diagonal d. The triangle representing the triangular matrices `C` and `FM` is divided into sectors with an equal number of diagonal elements, one for each processor. The computation proceeds from the main diagonal towards the upper right corner.The information needed by processor 2 in order to calculate the elements of the dashed diagonal are highlighted. To compute its part of the dashed diagonal processor 2 needs the horizontally and vertically striped parts of the arrays `FM` and the shaded part of the array `C`. The shaded part does not extend to the diagonal, because we have restricted the maximal size of interior loops to u_{max}. For efficiency reasons the `FM` array is stored both as rows and columns. The `C` is also stored for later use in the backtracking, to simplify communication it is stored both as rows and columns.

nodes is shown in Figure 23. To avoid reorganizing the data along the computation, all arrays are initially allocated to the maximum size. Apart from the 2 large triangle matrices the algorithm also uses several arrays of length n, as well as the trapezoid part of `C` shown in Figure 23. Neglecting linear arrays, the memory requirement `M` per processor is therefore:

$$\texttt{M} = \left(\frac{4d^2 + 2d(u_{max}+2)}{2N}\right) \cdot \text{ sizeof(int) bytes.} \tag{35}$$

The first term in equ. (35) counts the large `C` and `FM` arrays, the second term the additional trapezoid array `C`. Additional memory could be conserved by storing `C` as columns only. However, since the partition function uses more

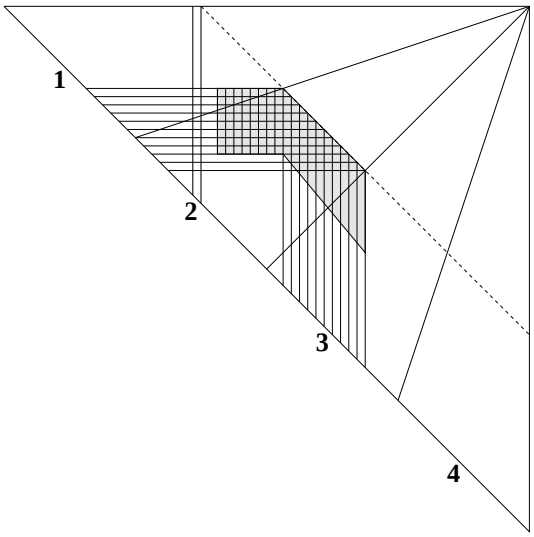


Figure 24: Calculation of the `F5` array is done by node 1. Two columns of the `C` array contribute to `F5`, see table 3. Because the expense of calculating the `F5` arrays is small, parallelizing this part would actually slow down the algorithm by introducing more communication delays.

memory anyways, conserving memory within the MFE part is of minor importance.

In addition to the large `C` and `FM` arrays we have to calculate the linear array `F5`, containing the minimum energy for subsequences on the

interval $[1, j]$ ($F_{1,j}$ in section 5.4). This is a relatively small amount of work, which is not worth parallelizing. Rather, we the entries are always calculated by node 1, which also holds the necessary columns of `C`, so that no additional message passing is necessary.

After completing a diagonal each processor has to either send a row or receive a column from his right neighbor or send a row or receive a column from his left neighbor.

An ideal parallel algorithm should share the amount of computational work equally to all available processors. In our case we have to divide up all diagonal entries of the diagonal d equally to the processors. Therefore, the amount of work is simply d/N, because each node can only calculate a direct number of

entries, we have to divide up the rest amount of entries equally to all processors. $N - (dmodN)$ calculate $(d/N) + 1$ diagonal entries per node and the other processor have to compute only d/N diagonal entries.

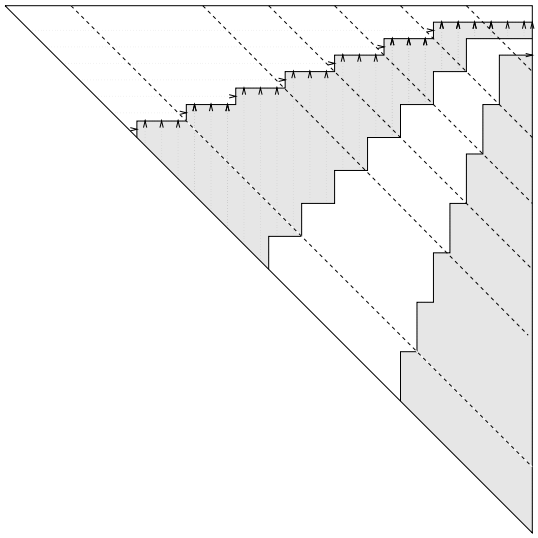


Figure 25: Distribution of work between processors and communication events. Horizontal arrows mark occasions where rows of data are sent from processor 1 to processor 2. Vertical arrows symbolize columns of data being sent from processor 2 to processor 1. The dashed lines show where communication switches from sending rows to columns and vice versa. The shaded areas are computed by nodes 2 and 4, respectively, light areas by nodes 1 and 3.

7.2 Parallel Partition Function

We have divided the parallel partition function algorithm (PPFA) into two parts. The first calculates the partition function of the sequence and all data needed for backtracking. All data are kept in memory during the entire computation. This leads to a quite fast algorithm, on the other hand the space requirement is enormous. Parallelization of the PPFA proceeds similarly to the PMFEA described in the previous section. Each diagonal is divided equally among all processors, this is done in the same way as discussed in section 7.1.

We need a lot of message passing, thus each node has to know the part of the diagonal each processor is computing. To manage all the message passing easily, that information is kept in an additional array. In the backtracking of the program, we decided to use a different strategy for parallelizing the program to reduce the amount of message passing, see section 7.3.

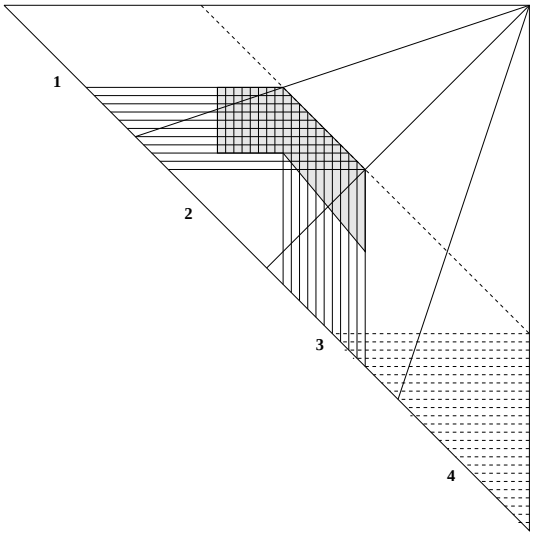


Figure 26: Representation of memory usage for the PPFA. The triangle represents the triangular matrices of the required arrays. For calculating the partition function `Q` we need at least 5 triangular matrices. Some of them have to be stored to do the backtracking calculation. The arrays `Qm, Q` and `Qb` are stored in form of rows, `Qmm` and `Qq` as columns. The diagonal d is divided to an equal number of entries for each processor. Each processor calculates the dashed line entries of it's part of d. To compute its part processor 2 needs the rows of `Qm, Q, Qb` and the columns of `Qmm, Qq` additionally the shaded part of the `Qb` array. The shaded part does not extend to the diagonal, because we have restricted the maximal size of interior loops. After the calculation of one diagonal d the rows of the `Qb` and `Qm` arrays are stored permanently (dashed lines), otherwise rows and columns we do not need any more for the ongoing calculation are removed from the memory.

The message passing for calculating the whole matrix of data is done in the same way as in the parallel MFE, but additional messages are necessary to distribute the data that are needed for the backtracking. All data for the backtracking, i.e. the `Qb` and `Qm` are stored in form of rows. The last element of a row is always calculated by node N, which then sends it to the node that

will store it permanently, as shown in Figure 26.

7.3 Calculating Base Pair Probabilities: Backtracking

The backtracking is parallelized in a different way, to simplify communication and reduce the number of messages. Each processor has to compute a horizontal slice of the triangle matrices as shown in Figure 27. Although the load balancing is somewhat worse in this method, it minimizes the communication overhead. Backtracking proceeds from the longest subsequences to shorter ones, i.e. in reverse order than calculation of the partition function.

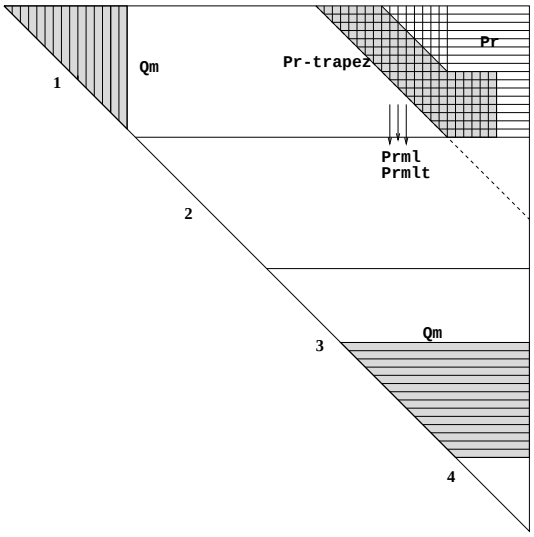


Figure 27: Data needed by processor 1 for calculation of its part of the diagonal d: The shaded trapezoid regions of `Pr` is needed for contributions from interior loops, and newly calculated values of `Pr` are then stored in rows (horizontal stripes). The shaded rows and columns of `Qm` (upper left and lower right) are needed for multi-loop contributions. In this decomposition, the same columns of `Qm` are needed for every diagonal d, reducing the amount of message passing. The auxiliary arrays `Prmlt` and `Prml` are stored as columns (vertical stripes); only those columns intersecting the current diagonal are needed. The calculation proceeds from the upper right corner towards the main diagonal.

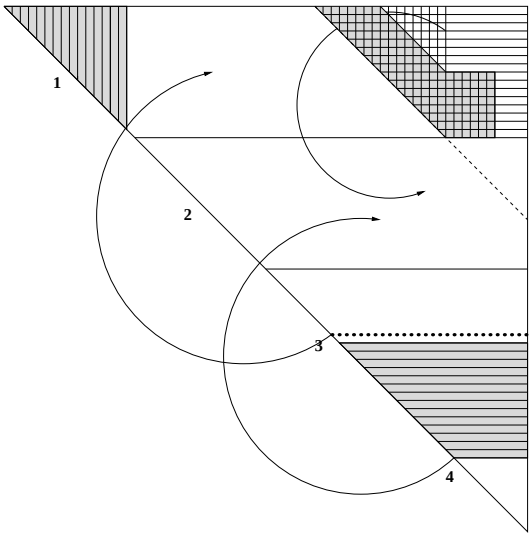


Figure 28: Message Passing for the Backtracking. All solid rows and columns are in the processors 1's memory and necessary for computing the current diagonal d, for the next d a lot of message passing has to be done. First processor 1 needs the next row of `Qm` (dotted row), currently stored by processor 3. At the same time, the lowest row of `Qm` is no longer needed on processor 1 and is sent to processor 2. Columns of Pr-arrays are also sent from processor 1 to processor 2.

Not shown in Figure 27 is the formation of the `Qm` columns. Each processor needs a certain amount of these columns, not shifting during calculation. For each diagonal d we have to create `Qm` colum $j = 1 + d$. To create it each processor sends its part of the column to the processor requiring it. One can easily see, that the algorithm is serial at the very beginning and becomes more parallel as it progresses. Towards the end the work is distributed ideally on the nodes. The advantage of the algorithm is a simpler communication structure, the disadvantage is a poorer load balancing. The loss of efficiency is not too great, because at the beginning all rows and columns are short and the computational effort is small. At the end of backtracking, when the computational requirements are greatest there, we have ideal parallelism.

In total, the backtracking algorithm uses 6 triangular arrays: `Qm` stored both as rows and columns, `Pr` and `Qb` as rows, and `Prml` and `Prmlt` as columns. In addition a triangular array with values from `Pr` is used on each processor. The

total memory requirement `M` per processor, neglecting small linear arrays, is therefore:

$$\texttt{M} = \left(\frac{6n^2 + 2n(u_{max} + 2)}{2N}\right) \cdot \text{sizeof(floats, doubles) bytes.} \tag{36}$$

The memory requirement for the whole algorithm is therefore dominated by the backtracking.

Quantity	row-wise	column-wise	trapezoid
1) Parallel minimum free energy			
F^B	`C`	`C`	`C`
F^M	`FM`	`FM`	
F		`F5`	
2) Parallel partition function			
Q^B	`Qb`		`Qb`
Q^M	`Qm`		
Q^{M1}		`Qmm`	
Q^A		`Qq`	
Q	`Q`		
3) Backtracking: Base pair probability			
	`Qm`	`Qm`	
	`Qb`		
P	`Pr`		`Pr`
$P^M + P^{M1}$		`Prmlt,Prml`	

Table 4: Used quantities in the parallel algorithms. 1) To calculate the minimum free energy we need 5 triangular matrices of integers and an additional trapezoidal array with values from `C`. The `F5` array holds the first row of the array F and is stored only on the first processor. 2) To compute the partition function we need 5 triangular matrices of floats or doubles and an additional trapezoidal array of `Qb`. `Qb` and `Qm` have to be stored permanently for the backtracking. 3) For the backtracking 6 triangular matrices and one additional trapezoidal array are needed.

8 Performance of the Parallel Algorithm

In this section we discuss the performance of the parallel partition function algorithm (PPFA), consisting of the minimum free energy calculation, calculation of the partition function and backtracking. The calculation of the MFE is only done to obtain an estimate for the partition function that can be used for rescaling. Here we discuss the performance of both programs together, called the PPFA. The exact number of instructions needed in the whole PPFA is sequence dependent. Therefore, we are not in the position to measure the performance of the program in terms of Flops (floating point operations). We tested the performance of our parallel programs on several RNA virus genomes, such as $Q\beta$ bacteriophage ($n = 4220$), polio viruses ($n \approx 7500$), and HIV viruses ($n \approx 10000$), see table 5.

Length	Name	Description
697	mit16sce	16S RNA
1562	eub16stm	16S RNA
1962	mit16szm	16S RNA
3023	eub23stm	23S RNA
4228	QBETA	$Q\beta$ viral genome
6421	CGMMV	Cucumber green mottle mosaic virus
7440	POL2LAN	Poliovirus type 2 (Lansing strain)
9022	HIVNY5CG	HIV 1 viral genome
9754	HIVANT70	HIV 1 viral genome
10271	HIV2UC1GNM	HIV 2 viral genome

Table 5: Test sequences used for the performance analysis on DELTA.

The implementation of the PPFA was done for distributed memory parallel computers as described in section 6. In the following we will use t to denote the time required to perform the computation of the PPFA in real time ("wall clock time"), while $T = tN$ refers to the total CPU time consumed on all processors. In order to measure the parallel performance of our algorithm, we

need to compare the performance on multiple processors to the performance on a single node. In our case this was impossible because of the memory requirements, instead we derive hypothetical single node CPU times T^* below.

An ideal parallel algorithm would compute its computational work N times faster than one processor would. In practice there is always a loss of efficiency caused by communication overhead and poor load balancing. For constant sequence length n the number of messages is proportional to N, we therefore expect a roughly linear dependence of T as a function of N. Figure 29 shows that this is indeed the case. The single node execution times T^* can therefore be estimated from a linear regression.

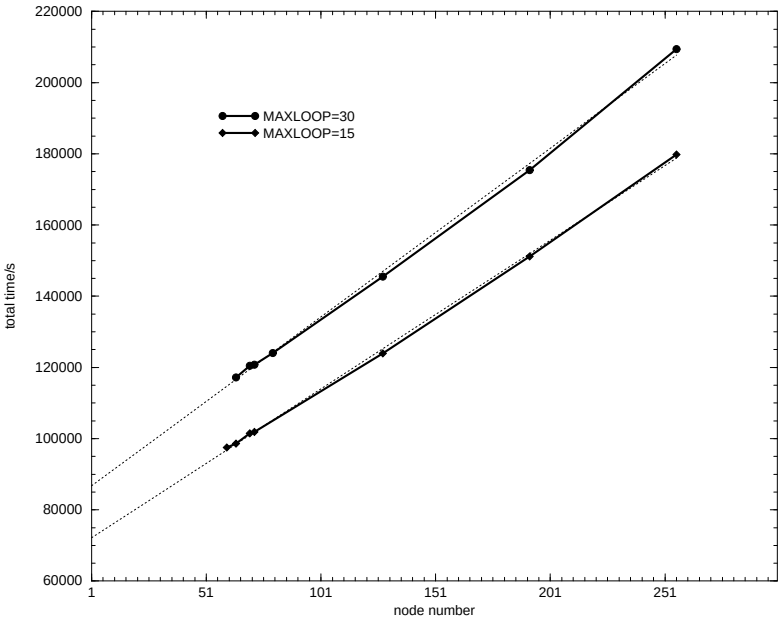


Figure 29: Plot of $T = tN$ versus number of nodes. Data are for the $Q\beta$ sequence ($n = 4220$) and two different values of u_{max}, $u_{1max} = 30$ and $u_{2max} = 15$. The linear regression (dotted lines) gives a good estimate of the execution time on a single node.

Since one has to calculate the sequence many times using different numbers of nodes, this method of obtaining single node execution times is somewhat tedious. If one examines the partition function algorithm, see section 5.1, one

notices that the total computational work follows quite well:

$$T^* \approx an^3 + bu_{max}^2 n^2 \tag{37}$$

Where the term an^3 comes from the calculation of multi-loops and the term $bu_{max}^2 n^2$ from the calculation of interior loops. By fitting the parameters a, b in equ. (37), one can obtain easily single node times for any sequence length n without computation. Two different methods were used to estimate the parameters a and b. A good fit can be obtained from the values of T^* for different u_{max}, as done in Figure 29, where u_{max} is the maximal size of interior loops.

$$an^3 = \frac{T_2 u_{1max}^2 - T_1 u_{2max}^2}{u_{1max}^2 - u_{2max}^2} \text{ and } bn^2 = \frac{T_1 - T_2}{u_{1max}^2 - u_{2max}^2} \tag{38}$$

Alternatively, a and b can be obtained from a nonlinear fit of T^* versus n, as done in Figure 30. Single node execution times for various sequence lengths and the resulting values for a and b are summarized in table 6.

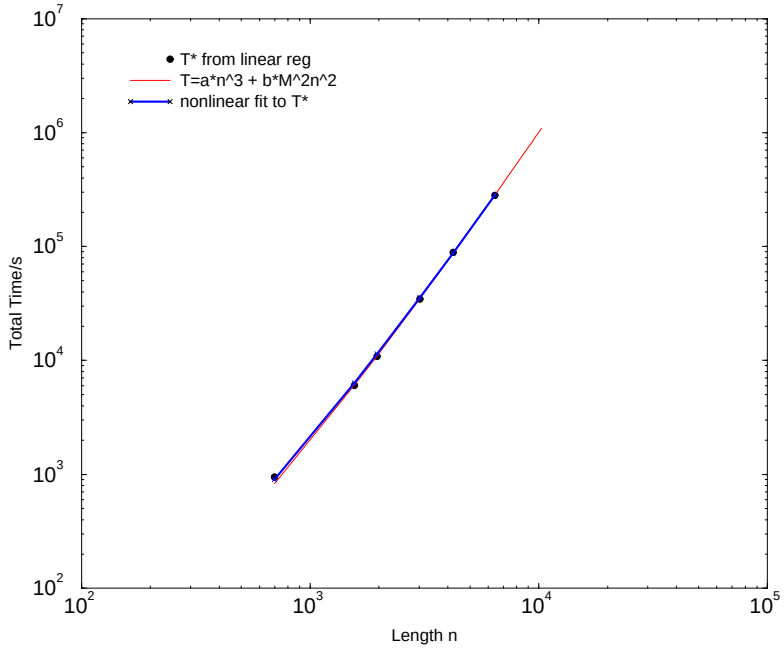


Figure 30: Plot of T^* versus sequence length n. To calculate the hypothetical single node CPU time we used the values of the a and b coefficients listed in table 6. On the other hand we got single node times from a linear regression tN versus N. One can see the nonlinear regression fit quite well to the curve obtained using a and b from table 6.

Length	`Min Nodes`	T^* [s]
697	2	829
1562	8	6065
1962	12	10955
3023	30	34733
4220	60	86869
6421	144	282787
7440	192	430430
9022	320	748831
9229	320	799456
9754	384	937952
10271	448	$1.089 \cdot 10^6$
1)	a=900 ± 20 ns	b=1200 ± 150 ns
2)	a=870 ns	b=1400 ns

Table 6: Single Node Times. `Min Nodes` denotes the minimal number of nodes necessary for folding, and T^* is the hypothetical single node execution time for the computation. 1) are the parameters obtained from equ. (37) using T^* from Figure 29 for different u_{max}. 2) shows the values for a and b obtained from Figure 30 doing a nonlinear fit.

Now that we know T^*, we are in the position to calculate the efficiency of the PPFA. The efficiency of a parallel algorithm is defined as,

$$E(N) := \frac{T^*}{(Nt)} \tag{39}$$

where T^* is the single node execution time, N is the number of processors, and t is the real time used for the computation. An ideal parallel algorithm would have efficiency of 1. In practice any efficiency above 50% is satisfactory for highly parallel algorithms. As shown in Figure 31 our algorithm reaches good efficiencies when the minimal necessary number of nodes is used.

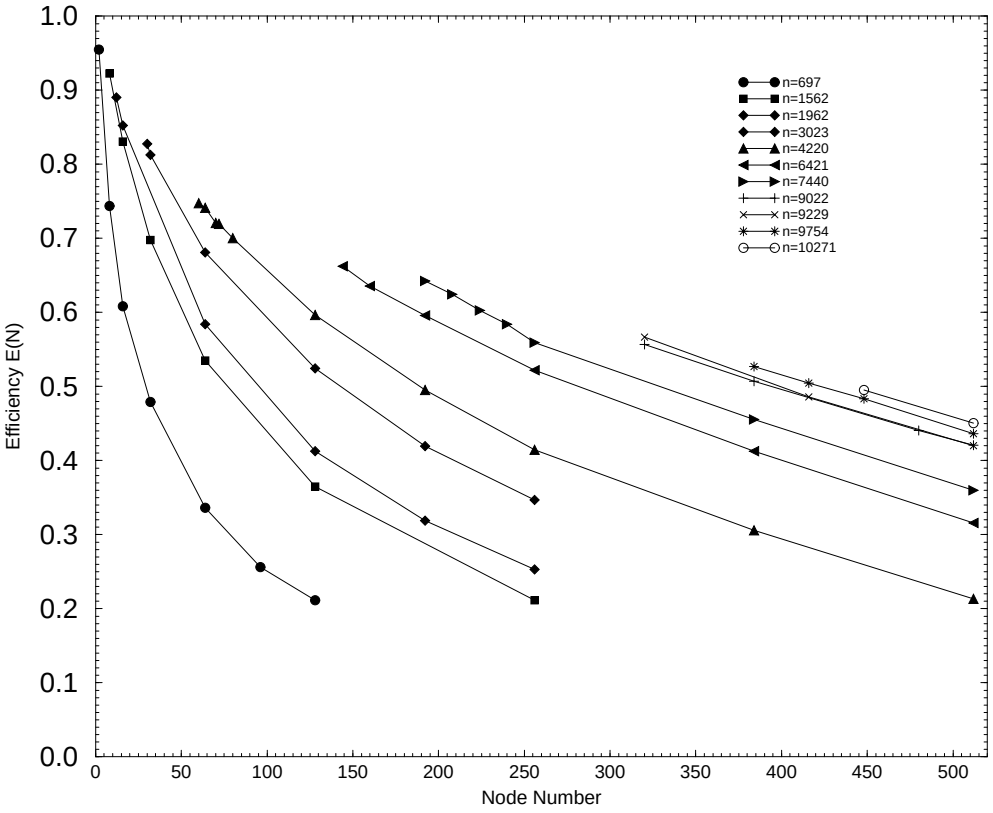


Figure 31: Plot of efficiency versus number of processors. Longer sequences require more memory, therefore the minimum number of nodes increases. By using more processors the amount of computation of a node decreases, but the amount of message passing is constant, resulting in a loss of efficiency. Conversely, efficiencies improve with sequence length for constant number of processors.

9 Base Pair Probabilities in $HIV1_{LAI}$

As a first application we have calculated the base pair probability and partition function of a full length $HIV1_{LAI}$ genome using the parallel partition function algorithm (PPFA).$HIV1$ is a highly complex retrovirus. Its genome is densely packed with information for the coding of proteins and biologically significant RNA higher order structure.

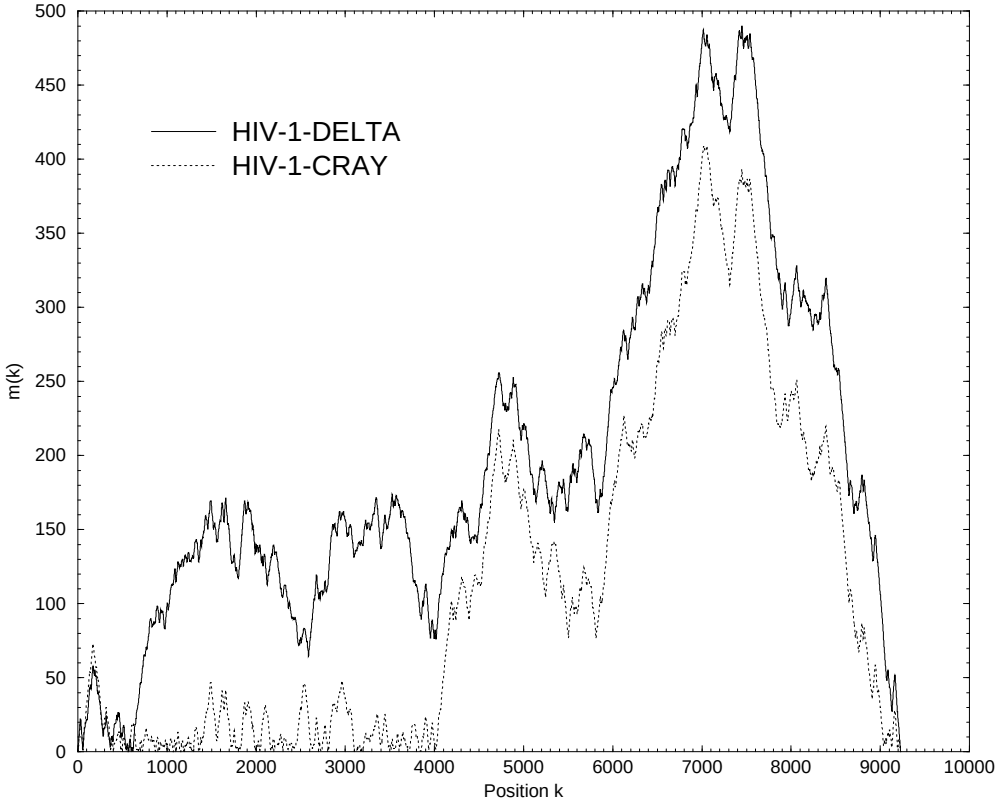


Figure 32: Generalized mountain representation of $HIV1_{LAI}$ obtained from the base pair probability matrix. This representation gives a good impression of the average structure of the RNA molecule, see section 3.2. The older serial folding on the CRAY shows multi-loops were somewhat less stable than in the new version, because dangling end energies were neglected, In the new folding a number of components from 500 to 4000 are replaced by a huge multi-loop. Although the curves do well on the first part of $HIV1_{LAI}$ the slopes and the peaks of the generalized mountain representation are very similiar.

We compare our results with an older computation on a CRAY-M90 (a large memory configuration of the CRAY YMP) (Huynen *et al.* 1996).The serial algorithm used in CRAY study did not account for dangling end energies and uses a slightly different set of energy parameters, this causes a slightly different base pair probability output and generalized mountain plot see Figure 32. The impact of differences will be discussed here. The data for two important regions of the $HIV1$ genome (5' end and the Rev response element RRE) are compared in detail.

At the 5' end of $HIV1$ resides the *trans*-activating responsive (TAR) element, which interacts with the regulatory **Tat** protein, see Figure 33. The binding of the **Tat** protein to TAR increases transcription rates (Feng & Holland 1988; Jeang *et al.* 1991). The TAR hairpin presents itself in the folding data as a beautiful stem-loop structure (5-54). The entire TAR motif seems to be quite well separated from the rest of the structure.

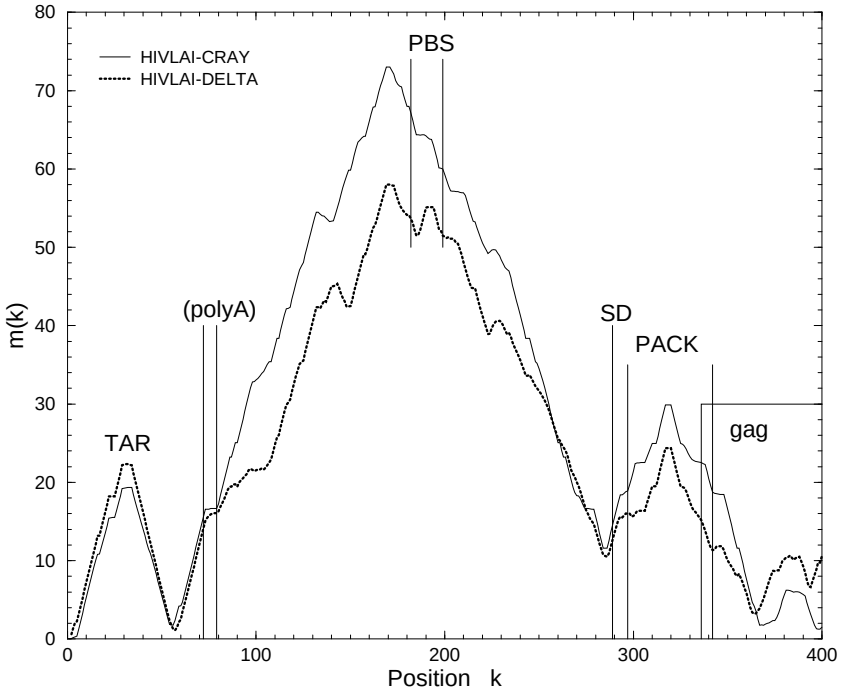


Figure 33: Generalized mountain representation of the 5' end of $HIV1_{LAI}$ The full line is obtained from the base pairing probability matrix of the complete $HIV1_{LAI}$ sequence computed on the CRAY.

Reverse transcription of $HIV1$ RNA into DNA is primed by a $tRNA_{lys}$ that is bound to a region of 18 nucleotides in the 5' LTR, positions 182-199, see Figure 33. The nucleotide sequence in this region, the so called Primer Binding Site (PBS), is complementary to the nucleotides at the 3' end of the tRNA. Figure 33 shows that the PBS is located in a partly unpaired region belonging to an interior loop (the mountain representation is partly horizontal in the PBS region).

The Packing Signal Region forms a quite well defined linear stem-loop structure immediately following the PBS region. The major-splice donor (SD) is located at the 5' side of this region, position 289. It resides in a strongly conserved (Harrison & Lever 1992) and stable (Baudin *et al.* 1993) secondary structure a few positions up-stream from the packaging signal. Figure 33 shows that it is located close to a minimum in the generalized mountain representation, enclosed only by a few base pairs, which form a multi-loop carrying both the packaging signal region and the PBS region.

The packaging signal (PACK) forms a very well defined linear stem-loop structure from position 297 through position 342. Both Figure 33 and Figure 34 show this structural motif very clearly. It consists of three or four stems separated by interior loops. One of the stems, which is not present in the minimum free energy structure, is quite unstable. The packaging signal itself extends a few nucleotides into the coding region of the **gag** gene. A well defined component boundary separates this region from the rest of the genome. These results are in agreement with (Hayashi, Ueno, & Okamoto 1993).

The Rev response element (RRE, also called CAR), is an RNA structure that is located within the **env** gene. The binding of the **Rev** protein to RRE promotes the transport of unspliced HIV transcripts to the cytoplasm (Malim *et al.* 1989; Malim & Cullen 1989; Mann *et al.* 1994; Kimura & Ohyama 1994). The RRE region forms a well defined structure on the outside of a large bulk of secondary structure, enclosed by more than 350 base pairs. The stem-root structure (I) contains a total of 32 base pairs in the MFE structure, which do not show any significant alternative structures. It separates the binding region well from the rest of the RNA.

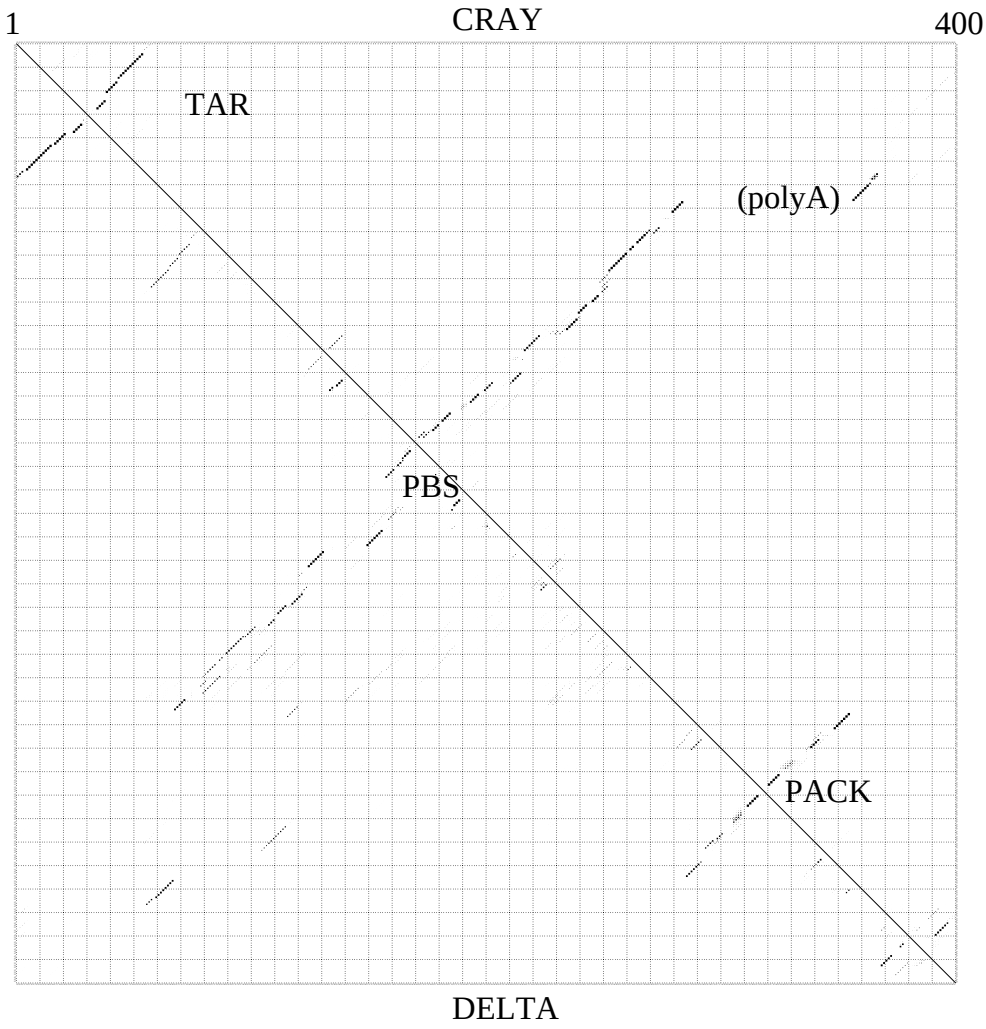


Figure 34: Dot plot of the TAR region at the 5' end of $HIV1_{LAI}$. The upper right triangle contains the base pairing probability matrix (P_{ij}) obtained on the CRAY; the size of the squares is proportional to the pairing probability. The lower-left triangle displays (P_{ij}) obtained on the DELTA for comparison. Hairpin loops appear as diagonal patterns close to the separating line between the two triangle, with the distance from this line indicating the loop size.

The long stem-loop structure furthermore indicates that the structure is easily accessible. There is very little interaction from the outside into the RRE region. The consensus secondary structure for the RRE in HIV1 consists of 5 hairpins in a multiple branched conformation closed by a single stem structure (Konings 1992). An alternative structure of only 4 hairpins, in which

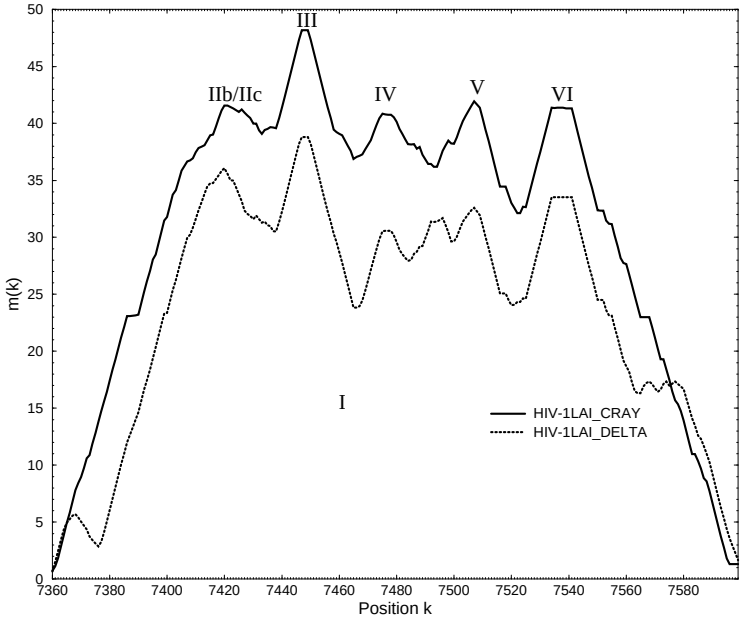


Figure 35: Generalized mountain representation of the RRE locus of $HIV1_{LAI}$. The baseline of both plots has been shifted to zero for easy comparison. The five-fingered structural motif of the CRAY folding is different to the DELTA folding. The DELTA folding contains an additional small hairpin between loops number IV and V. Loop III, VI are almost identical. The region IIb/IIc is not well defined in both foldings.

the hairpins III and IV of the consensus model merge to form one hairpin, has however been proposed (Mann *et al.* 1994). Note that this alternative structure matches the predicted minimum free energy structure, see Figure 36. Extensive computer analysis has shown that the alignment of the RRE at the level of the sequence does not coincide with the alignment at the level of the secondary structure (Konings 1992). This has two important implications: 1) methods that predict secondary structure of RNA on the basis of co-variation of positions within the sequence (Gutell 1993) can not be used here, and 2) the RRE has structural versatility.

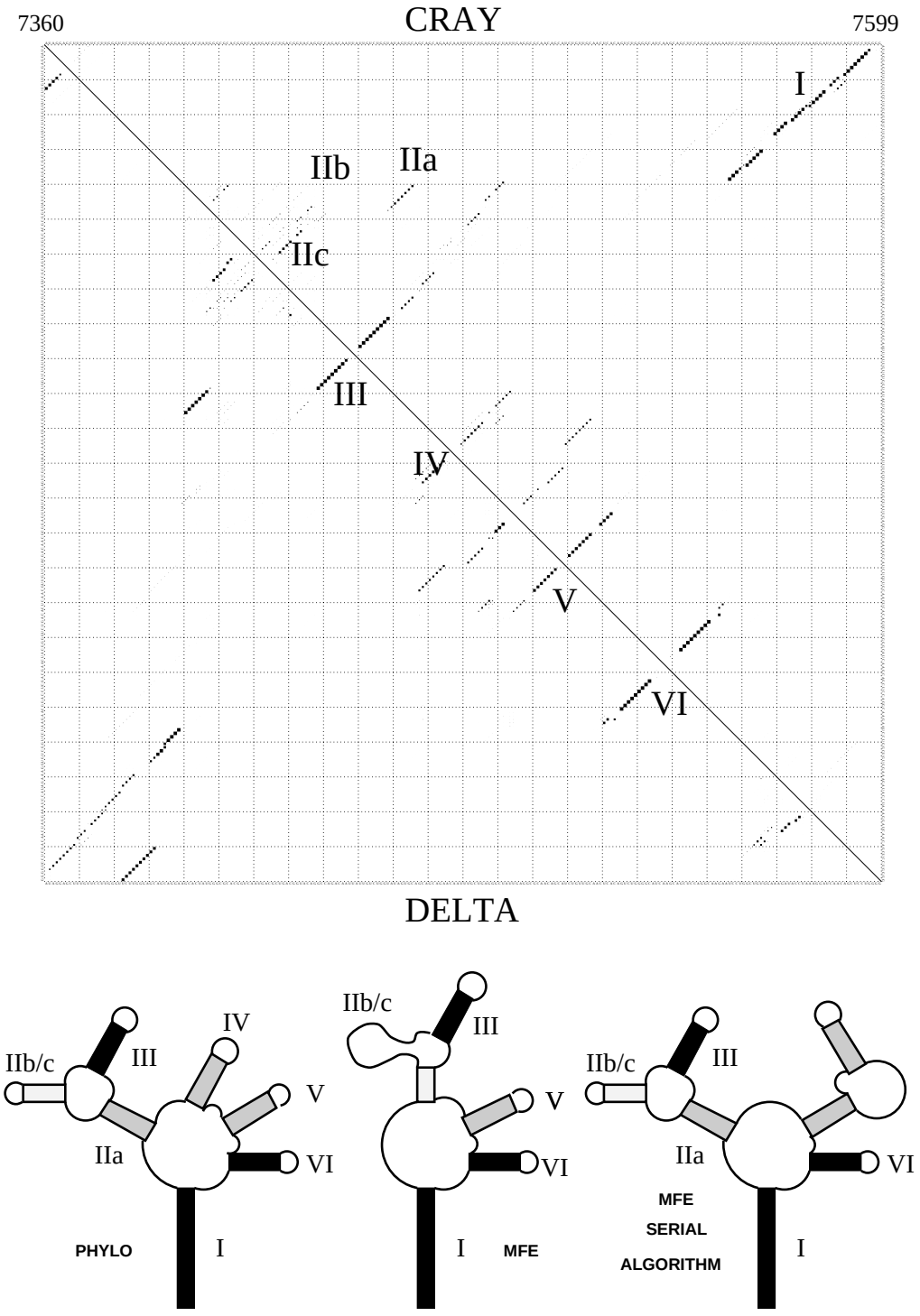


Figure 36: The picture on top of the page shows the dot plot of the RRE locus of $HIV1_{LAI}$. The base pair probabilities dot plots are labeled CRAY and DELTA. The picture below shows three possible structures of the RRE region. The minimum free energy structure obtained with the new energy parameters is shown to the right. In middle we see the MFE structure from the CRAY folding (Huynen *et al.* 1996). The left structure, labeled PHYLO, has been inferred from a comparison of several $HIV1$ RREs (Konings 1992). Stacks occurring in all structures are shown in black.

The structural versatility could also play a role in a single HIV clone; i.e. as long as the structural conformation is close to the consensus conformation the RRE is functional. This is exactly the motivation for analyzing and presenting the secondary structure as an ensemble of base-pair probabilities instead of a single or a few (alternative) structures. It is puzzling thus, that the high affinity binding site for the **Rev** protein, which lies in structure IIb, has a relatively ill defined secondary structure. Apparently the local secondary structure of the sequence is not that relevant for the binding of **Rev**. An alternative hypothesis is that the structural versatility of this region actually has a function. The binding of **Rev** proteins is cooperative process; the binding of the first **Rev** at the high affinity site facilitates the binding of other **Rev** proteins along stem IIa and stem I. This process has been attributed to protein-protein interactions (Mann *et al.* 1994). An alternative possibility is that the binding of **Rev** to the high affinity site stabilizes a specific conformation of the secondary structure, therewith giving the other binding sites, in particular the one stem IIa, the right secondary structure for the binding of other Rev proteins.

To present the differences between the pairing probabilities obtained from the two implementations of Mc Caskills algorithm (McCaskill 1990), we used a modified version of the RNApdist program from The Vienna RNA Package (Hofacker *et al.* 1994b) to get the distance between the thermodynamic ensemble of RNA secondary structure. For each base i we calculated a distance as follows: First we calculated the probabilities $P_i^($, $P_i^)$, P_i° of i being paired upstream, downstream or unpaired, respectively; $T[1] = P_i^(= \sum_{j>i} P_{ij}$, $T[2] = P_i^) = \sum_{j<i} P_{ij}$, $T[0] = P_i^\circ = 1 - P_i^(- P_i^)$. The distance $\texttt{dist}(T_1, T_2)$ was then calculated as:

$$\texttt{dist}(T_1, T_2) = 2 - \sum_{k=0,1,2} 2\sqrt{T_1(k)T_2(k)} \tag{40}$$

The output of the program is the distance at each position i, see Figure 37. As another way of comparing the two secondary structures, we calculated the differences in probabilities for all possible pairs $|P_{ij}^{DELTA} - P_{ij}^{CRAY}|$). In

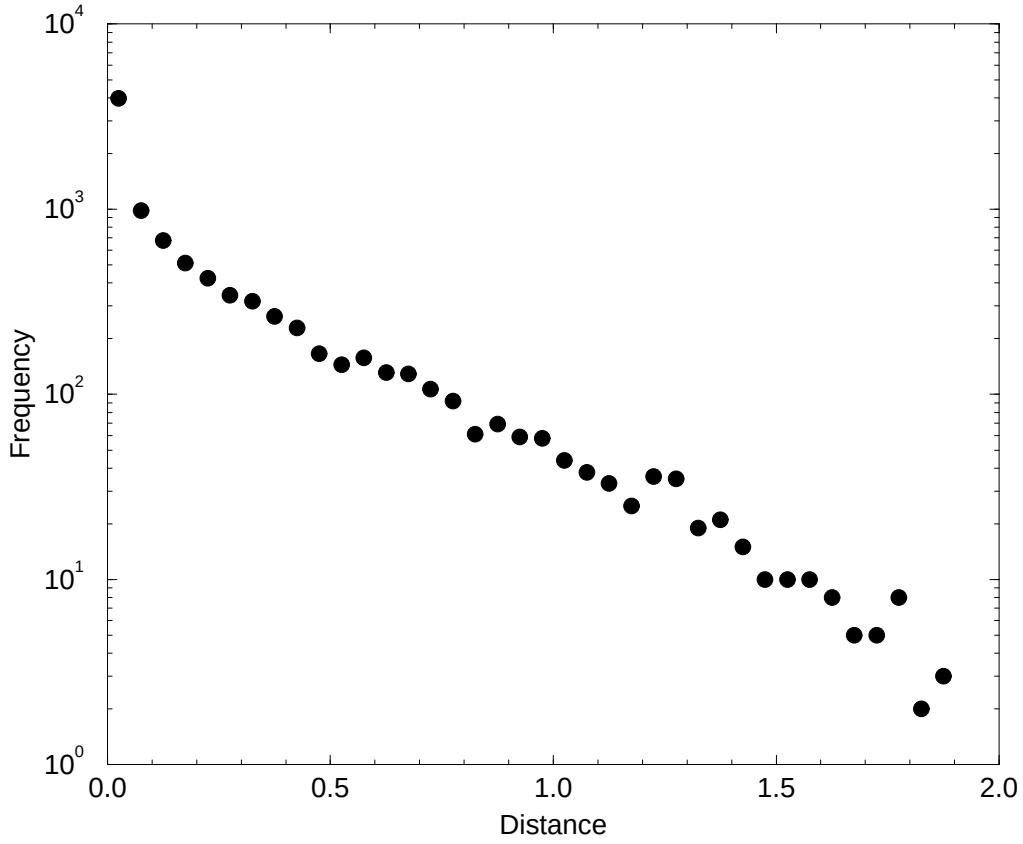


Figure 37: Frequency distribution of distances. In most positions the distance between the two structures is very small, almost no positions display distances close to the maximum of 2. The mean structure distance over all positions is 0.21, for unrelated structures we would expect a distance of about 0.6.

Figure 38 we plot the differences versus the distance of the base pair $(j - i)$. Comparing the results in Figure 38 to 32 we detect a high diversity for base pairs with $(j - i)$ about 8500. This is a consequence of the large multi-loop predicted by the PPFA at this base distance which is not present in the CRAY prediction.

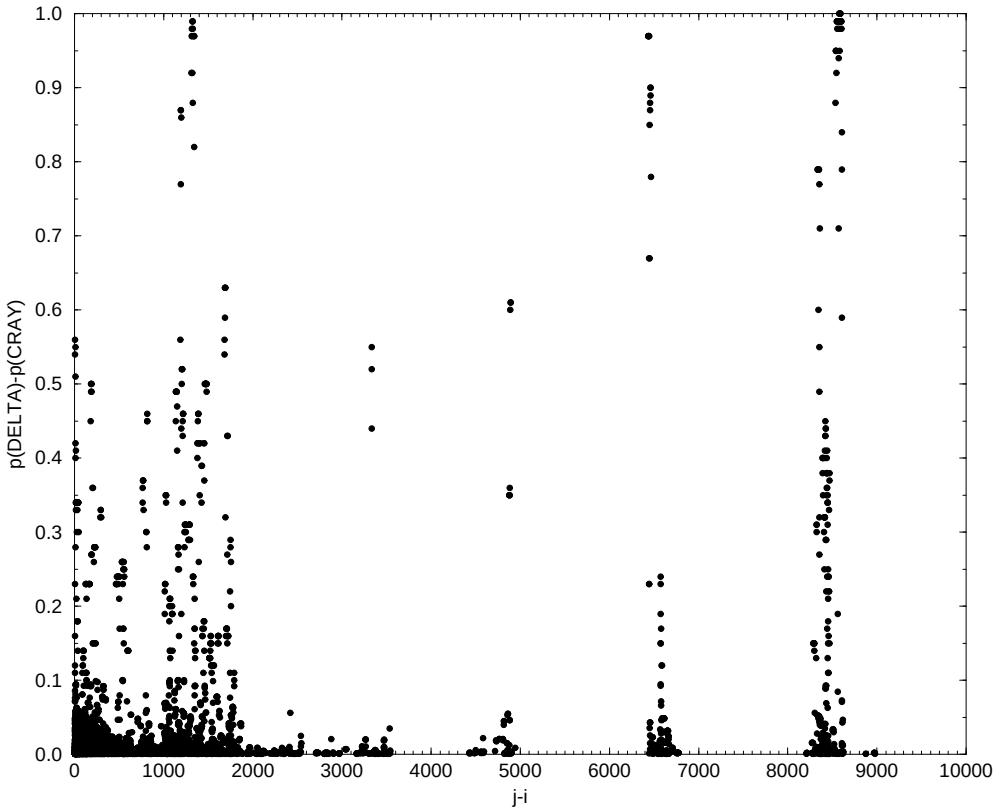


Figure 38: Differences in base pair probabilities versus distance $j - i$ of the pair. Clearly visible are several bands, e.g. at $j - i \approx 8500$ and 6500, corresponding to long range pairs predicted by the DELTA folding, but not by the CRAY.

10 Conclusion and Outlook

RNA structures play a significant role in a wide range of problems today. The secondary structures provide a convenient way of coarse graining, and their study yields important information about RNA useful in the prediction of the full 3D structures and in the interpretation of the biochemical function either. Secondary structures are discrete and therefore well suited for computational methods.

Earlier algorithms for structure prediction provided information only about the thermodynamically optimal structure, the minimum free energy structure. To understand the biological role of an RNA molecule, it is not sufficient, to know only one single structure. RNA molecules are not fixed in a single structure, they can vary over an ensemble of structures. The partition function algorithm of (McCaskill 1990), evaluates the complete thermodynamic ensemble of structures and provides us with a wealth of information on optimal and suboptimal structures in form of the base pair probabilities matrix. This puts us in the position to learn more about the stability and structural flexibility of these molecules.

For small RNA molecules like tRNAs RNA structure prediction has long been a useful tool providing us with information concerning the relationship between sequence and structure. For long RNA molecules it becomes a computationally demanding task, requiring computation time that scales as $O(n^3)$ and memory proportional to $O(n^2)$. In the past these requirements have made investigations of large RNA molecules, of great biochemical and medical importance, such as genomes of RNA viruses all but impossible. A single HIV1 Virus structure prediction for sequence length $n = 9229$ requires supercomputers such as the CRAY and is far away from a routine method, because of the time consumption.

On the other hand such requirements are easily met by modern massively parallel computers such the DELTA. We have therefore developed an implementation of the folding algorithms to message passing machines. With the

help of our program running on powerful parallel computers secondary structure prediction and analysis of the complete set of presently available RNA virus genomes has become a feasible task. Calculating the base pair probabilities for all virus genomes may cause a better understanding of functional tasks of the virus structure in respect of exploring evolutionary questions. As a first example we tested our parallel partition function algorithm (PPFA) on HIV1, to give an example that structure prediction of large virus genomes is now a feasible computational task.

References

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

Baudin, F.; Marquet, R.; Isel, C.; Darlix, J.; Ehresmann, B.; and Ehresmann, C. 1993, Functional sites in the 5' region of human immunodeficiency virus type 1 RNA form defined structural domains. *J. Mol. Biol.* 229:382–397.

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

Cupal, J. 1997. The density of states of RNA secondary structures. Master's thesis, University of Vienna.

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

Feng, S., and Holland, E. 1988, HIV-1 tat trans-activation requires the loop sequence within tar. *Nature* 334:165–167.

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 predictions 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.

Gutell, R. 1993, Comparative studies of RNA: Inferring higher order structure from patterns of sequence variation. *Current Opinion in Structural Biology* 3:313.

Harrison, G., and Lever, A. 1992, The human immunodeficiency virus type 1 packaging signal and major splice donor region have a conserved stable secondary structure. *J. Virology* 66:4144–4153.

Hayashi, T.; Ueno, Y.; and Okamoto, T. 1993, Elucidation of a conserved RNA stem loop structure in the packaging signal of human immunodeficiency virus type 1. *FEBS* 327:213–218.

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

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

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

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

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.

Huynen, M. A.; Perelson, A. S.; Vieira, W. A.; and Stadler, P. F. 1996, Base pairing probabilities in a complete HIV-1 RNA. *J. Comp. Biol.* 3:253–274. SFI preprint 95-07-057, LAUR-95-1613.

Intel, Corporation. 1990. *iPSC/2 and iPSC/860 User's Guide.*

Intel, Corporation. 1991. *Touchstone Delta System User's Guide.*

Jacobson, A. B., and Zuker, M. 1993, Structural analysis by energy dot plot of large mRNA. *J. Mol. Biol.* 233:261–269.

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.

Jeang, K.-T.; Chang, Y.; Berkhout, B.; Hammarskjöld, M.-L.; and Rekosh, D. 1991, Regulation of HIV expression: mechanisms of action of tat and rev. *AIDS* 5 (suppl 2):3–14.

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. 1989, 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.

Kimura, T., and Ohyama, A. 1994, Interaction with the rev response element along an extended stem I duplex structure is required to complete human immunodeficiency virus type 1 rev-mediated trans-activation in vivo. *J. Biochemistry* 115:945–952.

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.*

Konings, D. 1992, Coexistence of multiple codes in messenger RNA molecules. *Comp. & Chem.* 16:153–163.

Malim, M., and Cullen, B. 1989, HIV-1 structural gene expression requires the binding of multiple Rev monomers to the viral RRE: implications for HIV-1 latency. *Cell* 65:241–248.

Malim, M.; Hauber, J.; Le, S.; Maizel, J.; and Cullen, B. 1989, The HIV-1 Rev trans-activator acts through a structured target sequence to activate nuclear export of unspliced viral mRNA. *Nature* 338:254–257.

Mann, D.; Mikaelian, I.; Zemmel, R.; Green, S.; Lowe, A.; Kimura, T.; Singh, M.; Butler, P.; Gait, M.; and Karn, J. 1994, A molecular rheostat. Co-operative rev binding to stem I of the rev-response element modulates human immunodeficiency virus type-1 late gene expression. *J. Mol. Biol.* 241:193–207.

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

Morse, S., 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.

Noller, H. F.; Hoffarth, V.; and Zimniak, L. 1992, Unusual resistence of peptidyl transferase to protein extraction procedures. *Science* 256:1416–1419.

Noller, H. F. 1991, Ribosomal RNA and translation. *Ann.Rev.Biochem.* 60:191–227.

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.

Piccirilli, J. A.; McConnell, T. S.; Zaug, A. J.; Noller, H. F.; and Cech, T. R. 1992, Aminoacyl-esterase activity of the *tetrahymena* ribozyme. *Science* 256:1420–1424.

Poerschke, D. Berlin 1977, Elementary steps of base recognition and helix-coil transitions in nucleic acids. *Molecular Biology, Biochemistry and Biophysics* volume 24:191–218.

Saenger, W. *Principles of Nucleic Acid Structure*. Springer 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 *In Encyclopedia of Physical Science and Technology*, volume 6. Academic Press 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* 33:14289–965.

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 of tRNA genes. *Nucl. Acids Res.* 24(1).

Tinoco, J. CSHL Press 1993., Structures of base pairs involving at least two hydrogen bonds. *The RNA World.* 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 Zucker, M. 1994, Coaxial stacking of helices enhances binding of oligoribunucleotides 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 pairs. *Biochemistry* 33:9218–22.

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.

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. *Bulletin of Mathematical Biology* 46(4):591–621.

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

Zuker, M. The use of dynamic programming algorithms in RNA secondary structure prediction. In Waterman, M. S., ed., *Mathematical Methods for DNA Sequences*. CRC Press 1989. 159–184.

Curriculum Vitae

Martin Fekete

* 6.8.1970, Waidhofen/Thaya

1976 – 1980	Volksschule in Heidenreichstein
1980 – 1981	Hauptschule in Heidenreichstein
1981 – 1989	Neusprachliches Gymnasium in Waidhofen/Thaya
Mai 1989	Reifeprüfung am Gymnasium in Waidhofen/Thaya
1989 – 1997	Studium der Chemie, Hauptfach Biochemie, an der Universität Wien
Jänner 1993	1. Diplomprüfung Biochemie
1995 – 1997	Diplomarbeit am Institut für Theoretische Chemie, an der Universität Wien
1994 – 1995	Musikstudium am Praynerkonservatorium, Hauptfach Klarinette
1995 – 1996	Präsenzdienst im Österreichischen Bundesheer