RNA Folding on Parallel Computers

The Minimum Free Energy Structures of Complete HIV Genomes

IVO L. HOFACKER[a], MARTIJN A. HUYNEN[b,c,*],

PETER F. STADLER[a,c], AND PAUL E. STOLORZ[d]

[a]Institut f. Theoretische Chemie, Univ. Wien, Austria

[b]Los Alamos National Lab, CNLS and T-10, Los Alamos, New Mexico, U.S.A.

[c]The Santa Fe Institute, Santa Fe, New Mexico, U.S.A.

[d]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, U.S.A.

[*]Mailing Address: Martijn A. Huynen
CNLS and T-10, Mail Stop K-710, Los Alamos Natl. Lab., Los Alamos, NM 87545, U.S.A.
Phone:(505) 667 9455 Fax: (505) 665-3493
E-Mail: `mah@t10.lanl.gov`

Keywords

RNA secondary structure — HIV — RNA folding — Parallel Computing — Message Passing

Abstract

Secondary structure prediction is a standard tool in the analysis of RNA sequences. The prediction of RNA secondary structures is inherently non-local. This makes the analysis of long sequences (more than 4000 nucleotides) infeasible on present-day workstations. An implementation of the secondary structure prediction algorithm for hypercube-type parallel computers allows to compute efficiently the structure of complete RNA virus genomes such as HIV-1 and other lentiviruses.

1. RNA Secondary Structures

RNA structure can be broken down conceptually into a secondary structure and a tertiary structure. The secondary structure is a pattern of complementary base pairings, see Figure 1. The tertiary structure is the three-dimensional configuration of the molecule. As opposed to the protein case, the secondary structure of RNA sequences is well defined; it provides the major set of distance constraints that guide the formation of tertiary structure, and covers the dominant energy contribution to the 3D structure. Secondary structures are conserved in evolutionary phylogeny, and they represent a qualitatively important description of the molecules, as documented by their extensive use for the interpretation of molecular evolution data. In this paper we will be concerned only with secondary structure.

A secondary structure on a sequence is a list of base pairs i, j with $i < j$ such that for any two base pairs i, j and k, l with $i \leq k$ holds:

$$\begin{aligned} i = k &\iff j = l \\ k < j &\implies i < k < l < j \end{aligned} \tag{1}$$

The first condition implies that each nucleotide can take part in not more that one base pair, the second condition forbids knots and pseudoknots[1]. Knots and pseudoknots are excluded by the great majority of folding algorithms which are based upon dynamic programming concepts.

[1] A pseudoknot is a configuration in which a nucleotide that is inside a loop base pairs with a nucleotide outside that loop.

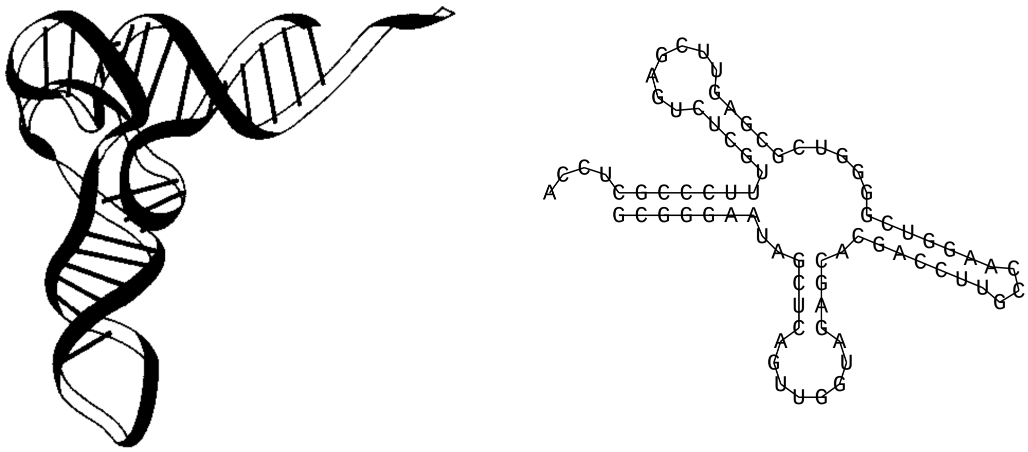


Figure 1: a) The spatial structure of the phenylalanine tRNA form yeast is one of the few known three dimensional RNA structures.
b) The secondary structure extracts the most important information about the structure, namely the pattern of base pairings.

A base pair k, l is *interior* to the base pair i, j, if $i < k < l < j$. It is *immediately interior* if there is no base pair p, q such that $i < p < k < l < q < j$. For each base pair i, j the corresponding *loop* is defined as consisting of i, j itself, the base pairs immediately interior to i, j and all unpaired regions connecting these base pairs. The energy of the secondary structure is assumed to be the sum of the energy contributions of all loops. (Note that two stacked base pairs constitute a loop of size 4; the smallest hairpin loop has three unpaired bases, i.e., size 5 including the base pair.) The types of structural elements are defined in Figure 2.

Experimental energy parameters are available for the contribution of an individual loop as functions of its size and type (stacked pair, interior loop, bulge, multi-stem loop), of the type of its delimiting base pairs, and partly of the sequence of the unpaired strains (Freier *et al.*, 1986; Jaeger *et al.*, 1989). We use a recent version of the parameter set published by Jaeger and co-workers (Jaeger *et al.*, 1989). In the current implementation we do not consider dangling ends. Inaccuracies in the measured energy parameters, the uncertainties in parameter settings that have been inferred from the few known structures, and most importantly, effects that are not even part of the secondary structure model, limit the predictive power of the algorithms. For larger molecules it is furthermore suspected that kinetic

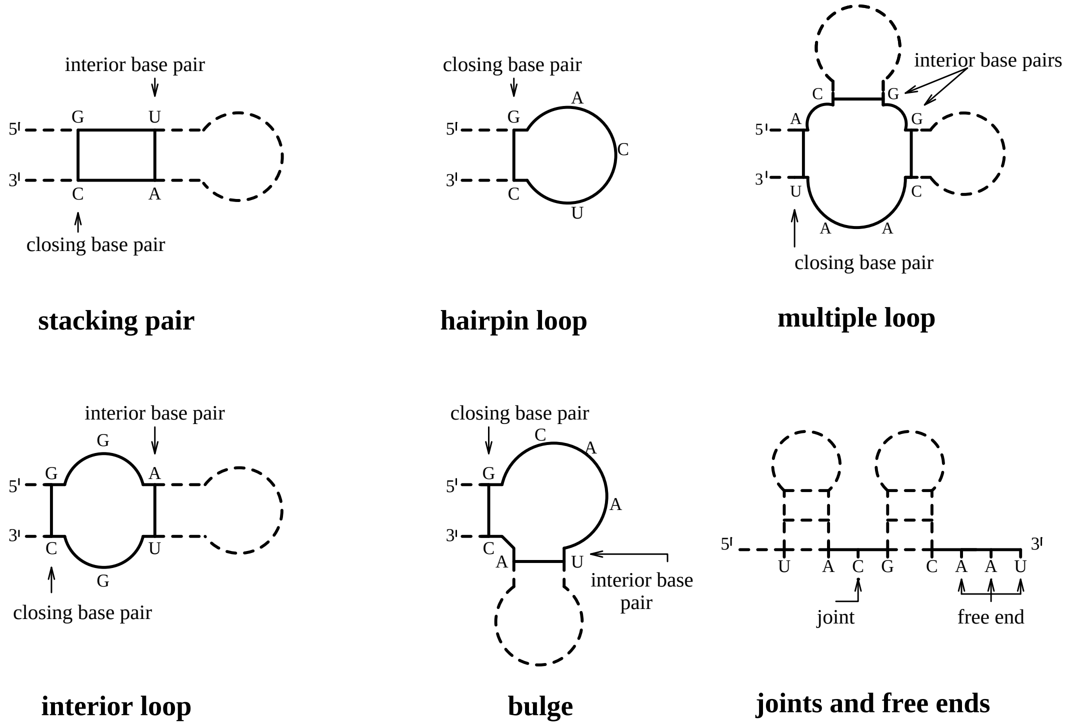


Figure 2: Basic structure elements on nucleic acid secondary structures.
Every structure within the secondary structure model can be decomposed into the basic elements: stems, hairpins, interior loops, bulges, multi-stem loops, joints, and free ends.

effects influence the formation of the secondary structure. Nevertheless, local structures can be computed in quite some detail, and a majority of the base pairs is predicted correctly. Konings and Gutell (1995) recently performed an extensive comparison for ribosomal RNAs (16S RNA molecules contain approximately 1500 bases) between the secondary structure as derived from phylogenetic methods and the secondary structure as derived from minimum free energy folding. They observed that: 1) Short range base pair interactions are generally better predicted than long range interactions and 2) the percentage of correctly predicted base-pairs depends strongly on the taxon from which the sequences are derived. For ribosomal RNAs from Archaea the percentage correctly predicted base pairs was about 70% whereas for ribosomal RNAs from Eukaryotes it dropped to about 30%. This illustrates that elements that are not part of the secondary structure model might play a role in determination of the structure *in vivo*; ribosomal RNAs interact with proteins in the ribosome, apparently this interaction does play a large

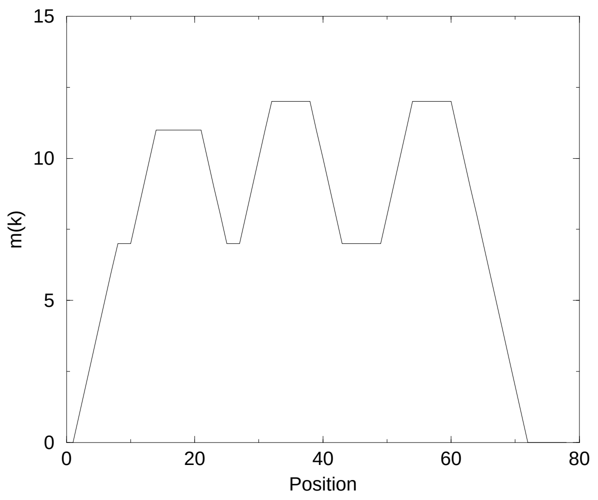


Figure 3: Mountain representation of the tRNA secondary structure shown in Figure 1. The three plateaus correspond to the three hairpin loops of the clover leave structure.

role in determination of secondary structure of the RNA in Eukaryotes, but much less in Archaea.

The unique decomposition of secondary structures outlined above suggests a simple string representation of structures by identifying a base pair with a pair of matching brackets and denoting an unpaired digit by a dot (downstream is understood in $5'$-$3'$ direction; upstream refers to the opposite direction, RNA sequences are generally displayed in the $5'$-$3'$ direction):

(paired to a downstream base

) paired to an upstream base

. single-stranded base.

This bracket notation is coding for a tree (Fontana *et al.*, 1991). Other tree representations have been proposed for RNA secondary structures as well (Zuker & Sankoff, 1984; Shapiro, 1988; Shapiro & Zhang, 1990; Fontana *et al.*, 1993).

A convenient way of displaying the size and distribution of secondary structure elements is the *mountain representation* introduced by Hogeweg and Hesper (Hogeweg & Hesper, 1984). In this representation a '(' is drawn as a step up, a ')' corresponds to step down, and an unpaired base '.' is shown as horizontal line segment, see Figure 3. The resulting graph looks like a mountain-range where:

- *Peaks* correspond to hairpins. The symmetric slopes represent the stack enclosing the unpaired bases in the hairpin loop, which appear as a plateau.
- *Plateaus* represent unpaired bases. When interrupting sloped regions they indicate bulges or interior loops, depending on whether they occur alone or paired with another plateau on the other side of the mountain at the same height respectively.
- *Valleys* indicate the unpaired regions between the branches of a multi-stem loop or, when their height is zero, they indicate unpaired regions separating the components of secondary structures.

The height of the mountain at sequence position k is simply the number of base pairs that enclose position k; i.e., the number of all base pairs (i, j) for which $i < k$ and $j > k$. The mountain representation allows for straightforward comparison of secondary structures and inspired a convenient algorithm for alignment of secondary structures (Konings & Hogeweg, 1989).

In this contribution we shall be interested in the secondary structure of the RNA genomes of a certain class of single-stranded RNA viruses. Lentiviruses such as HIV-1 and HIV-2 are highly complex retroviruses. Their genomes are dense with information for the coding of proteins and biologically significant RNA secondary structures. The latter play a role in both the entire genomic HIV-1 sequence and in the separate HIV-1 messenger RNAs which are basically fragments of the entire genome. The total length of HIV-1 (about 9200 bases) makes biochemical analysis of secondary structure of the HIV-1 full genome infeasible. For RNAs of this size the prediction of the minimum free energy structure is the only approach that is available at present. By predicting the minimum free energy secondary structure of the full length HIV-1 and other known lentiviruses sequences (HIV-2, SIV, CAEV, visna, BIV and EIAV) and their various splicing products, and by comparison of the predicted structures, a first step can be set into the unraveling of all important secondary structures in lentiviruses. The comparisons of the prediction obtained for different, evolutionarily related RNAs can be used to identify local misfoldings in the same way as a comparative analysis can be used to infer the structure from the phylogeny.

Elucidation of all the significant secondary structures is necessary for the understanding of the molecular biology of the virus. So far a number of significant secondary structures have been determined that play a role during the various stages of the viral life cycle (see section 4). We do expect a high number of undiscovered biologically functional secondary structures to be still present within the various transcripts. A systematic analysis of the 5' end of the HIV genome showed an abundance of functional secondary structures (Baudin *et al.*, 1993). Secondary structures further downstream could be involved in the splicing, regulation of translation of the various mRNAs, or regulation of the stability of the full length sequence and its various splicing products.

The most popular computational approach to the prediction of RNA secondary structure from sequence information is dynamic programming, a topic which is reviewed further in the next section. At this stage the important point to note is the fact that the algorithm, when applied to RNA folding, requires CPU time that scales roughly as the cubic power of the sequence length, and memory that scales quadratically with sequence length.

The basic philosophy driving our implementation on massively parallel platforms is the point of view 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 base pairs in length still require only on the order of 35 minutes to fold on 256 nodes of the Intel `Delta` supercomputer. The same calculation would require of the order 60 hours on a high-end workstation. While this time is hardly ideal, it is certainly an acceptable turn-around time given that there are only a limited number of RNA molecules available to fold. Our implementation is thus suitable as a routine tool allowing for a comparative analysis of the complete set of available sequence data for RNA viruses.

On the other hand, memory requirements are a severe problem for RNA molecules the size of HIV. The simplest RNA folding calculation, which computes just the single minimum free energy structure, requires of the order of 1 Gigabyte of memory for a sequence of the length of HIV-1. More sophisticated algorithms compute averages over a much larger number of structures near the minimum free energy and typically require upwards of 2 Gigabytes. Distributed massively parallel architectures can easily satisfy these memory requirements for viruses such as HIV.

Furthermore, future generations of such machines will provide scalable memory enabling even longer sequences to be studied. These resources simply cannot be matched by conventional architectures. This is the primary reason that scalable architectures are necessary for performing RNA folding computations on large macromolecules.

Recently the implementation of an RNA folding algorithm on a single instruction multiple data (SIMD) machine has been reported (Shapiro *et al.*, 1995). Our approach, which uses a multiple instruction multiple data (MIMD) machine, differs in a number of fundamental ways from this implementation. SIMD machines typically utilize thousands of very simple processors, each of which synchronously executes *the same* program. MIMD machines, like the `Delta` used here, consist of fewer, more powerful processors executing their programs independently. The SIMD implementation on the `MasPar MP-2` uses as many processors as the length of the sequence. It is currently limited by the amount of memory to sequences no longer than 9400 nucleotides. Our implementation on the `DELTA` is able fold sequences at least three times as long, even for the HIV-2 sequence of length 10271 reported here only 96 nodes were needed. We report a comparison of the folding times of our code for the `Delta` and the `MasPar MP-2` implementation in section 3. In terms of wall-clock time, our implementation on the `Delta` is faster by at least a factor of 4, even if we use only the minimal number of processors on the `DELTA` that are required to satisfy the memory requirements. The major advantage of MIMD code and architecture, however, lie in their flexibility. The architecture allows the execution of multiple programs on the same machine in parallel. Hence it is not necessary to use as many processors as possible, which reduces the communication overhead and increases the efficiency of the parallelization.

Our code is written in ANSI C using only a few simple platform specific message passing commands and can therefore be ported easily to other MIMD message passing systems or even workstation clusters. The folding algorithm reported in this paper produces only a single minimum free energy structure, whereas the version used in (Shapiro *et al.*, 1995) has been designed to find also suboptimal structures. Although suboptimal structures can be an important research tool for the investigation of RNA secondary structure, this approach looses most of its power when applied to very long sequences. This is because the number of possible structures increases exponentially with the length of the sequence. For example, the frequency of the minimum energy structure in the ensemble at thermodynamic equilibrium is in general smaller than 10^{-100} for RNAs of length 3000. Hence one needs more than 10^{100} of different structures to adequately describe the ensemble, and the direct generation and analysis of this amount of structure information is way beyond the capabilities of even the most modern computer systems. A better alternative to the issue of suboptimal structures is McCaskill's partition function approach (McCaskill, 1990), which allows for an exact computation of the complete matrix of all base pairing probabilities.

In this contribution we shall demonstrate that message-passing algorithms on distributed-memory parallel architectures represent a feasible approach to RNA-folding. The results are in agreement with known secondary structure features and can be used for analysis of unknown secondary structures. In the following section we outline the folding algorithm and discuss its implementation on a message passing system (more details on the implementation can be found in the appendix). In section 3 we discuss the CPU requirements and the efficiency of our implementation. In section 4 we present a few biophysical results in some detail. Section 5 concludes this paper.

2. Folding Algorithm

As a consequence of the additivity of the energy contributions, the minimum free energy can be calculated recursively by dynamic programming (Waterman, 1978; Waterman & Smith, 1978; Zuker & Stiegler, 1981; Zuker & Sankoff, 1984).

The essential part of the energy minimization algorithm is shown in table 1. The basic logic of this scheme is derived from sequence alignment: In fact, folding of RNA can be regarded as a form of alignment of the sequence to itself (Zuker & Sankoff, 1984). The implementation of sequence alignment algorithms on massively parallel architectures in discussed in detail in (Jones, 1992).

Table 1. Pseudo Code of the minimum free energy folding algorithm.

```
for(d=1...n)
   for(i=1..n-d)
      j=i+d
      C[i,j] = MIN(
          Hairpin(i,j),
          MIN( i<p<q<j : Interior(i,j;p,q)+C[p,q] ),
          MIN( i<k<j : FM[i+1,k]+FM[k+1,j-1]+cc )      )
      F[i,j] = MIN( C[i,j], MIN(i<k<j : F[i,k]+F[k+1,j]))
      FM[i,j]= MIN( C[i,j]+ci, FM[i+1,j]+cu, FM[i,j-1]+cu,
                  MIN( i<k<j : FM[i,k]+FM[k+1,j] )      )
free_energy = F[1,n]
```

`F[i,j]` denotes the minimum free energy for the subsequence consisting of bases i through j. `C[i,j]` is the energy given that i and j pair. The array `FM` is introduced for handling multi-stem loops. The energy parameters for all loop types except for multi-stem loops are formally subsumed in the function `Interior(i,j;p,q)` denoting the energy contribution of a loop closed by the two base pairs $i-j$ and $p-q$. We have assumed that multi-stem loops have energy contribution `F=cc+ci*I+cu*U`, where `I` is the number of interior base pairs and `U` is the number of unpaired digits of the loop. The time complexity here is $\mathcal{O}(n^4)$.
It is reduced to $\mathcal{O}(n^3)$ by restricting the size of interior loops to some constant $\mathcal{M}$, i.e., $p-i \leq \mathcal{M}$ and $j-q \leq \mathcal{M}$. In general we use $\mathcal{M}=40$. This can be regarded as a minor correction since loops of that size are extremely rare.
The structure (list of base pairs) leading to the minimum energy is usually retrieved later on by "backtracking" through the energy arrays.

A brief inspection of the algorithm in table 1 shows that is can be parallelized very easily: all entries along a diagonal d are independent of each other and can therefore be computed concurrently. This is the same situation as in the case of sequence alignment (Jones, 1992). The major computational difficulty in the case of folding is that each entry in diagonal d requires the *explicit* knowledge of a large number of the previously computed matrix entries.

On architectures with distributed memory the basic parallel decomposition consists of assigning the matrix entries within each diagonal evenly between all N available processors. Figure 4 shows which entries of the arrays `F`, `C` and `FM` are necessary to compute all entries in the part of diagonal d which is processed by a particular processor $\nu = 1, \ldots, N$.

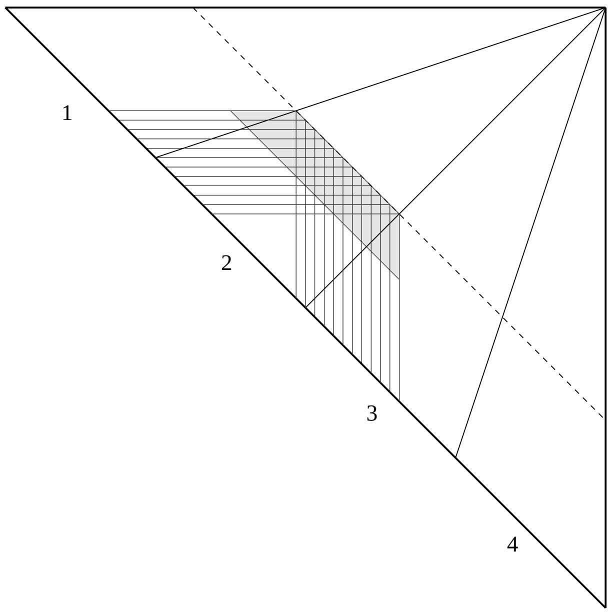


Figure 4: Representation of memory usage by the parallel folding algorithm. The triangle representing the triangular matrices `F`, `C`, and `FM`, respectively, is divided into sectors with an equal number of diagonal elements, one for each processor. The computation proceeds from the main diagonal towards 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 `F` and `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.

In the following we present a parallelized version of the minimum energy folding algorithm for message passing systems (Hofacker *et al.*, 1994). The fragments of the arrays `F` and `FM` required by a particular node are stored both as columns and as rows, while the `C` array is stored in diagonal order. The maximal memory requirement occurs at $d = n/2$, where we need a minimum of

$$M = \frac{n^2}{2N} \times \texttt{sizeof(int)} \quad \text{bytes} \tag{2}$$

each for `F` and `FM`, while the array `C` needs only $(n-d)/N + \mathcal{M}(\mathcal{M}-1)/2$ storage, where $\mathcal{M}$ is the cutoff-parameter for exhaustive search of interior loops, see the caption of table 1 for details.

The length of the required rows and columns increases with d, while the number of required rows and columns decreases with d. One could therefore save memory at the expense of reorganizing the storage after each diagonal, which would result in an additional $\mathcal{O}(n^3)$ computations. If one allocates twice the minimum memory, storage has to be reorganized only once and the total requirement is the same as for the sequential algorithm, namely

$$M = \left[\frac{2n^2}{N} + \mathcal{O}(n)\right] \times \texttt{sizeof(int)} \quad \text{bytes}\,. \tag{3}$$

After completing a diagonal each processor has to either send a row to or receive a column from its right neighbor, and it has to either receive a row from or send a column to its left neighbor. Details of the message passing requirements are given in the appendix.

Since we do not store the entire matrices, we cannot do the usual (Zuker & Sankoff, 1984) backtracking to retrieve the structure corresponding to the minimum energy. Instead, the necessary information is written to a file: For each index pair (i, j) we store 2 integers which identify the term that actually produced the minimum. The file size is hence $\mathcal{O}(n^2)$. The backtracking can then be done with $\mathcal{O}(n)$ readouts.

All in all we need $\mathcal{O}(n)$ communication and I/O steps each transferring $\mathcal{O}(n)$ integers, while the computational effort is asymptotically $\mathcal{O}(n^3)$. The communication overhead — for a constant number N of nodes — therefore becomes negligible for sufficiently long sequences.

3. Performance

The exact number of instructions required for computing a minimum free energy structure is sequence dependent. Furthermore, the calculation of the energy contributions themselves is quite involved. We are therefore not in a position to measure the performance of program in terms of Flops. In this section we will discuss the CPU requirements for folding complete genomes of RNA viruses, such as the Qβ phage ($n = 4220$), polio viruses ($n \approx 7500$), and HIV viruses ($n \approx 10000$). The set of sequences used for determining the performance of the algorithm is given in table 2.

Table 2. Sequences used for the performance analysis on the `Delta`.

Length	Name	Description
697	`mit16sce`	16S RNA
1562	`eub16stm`	16S RNA
1962	`mit16szm`	16S RNA
3023	`eub23stm`	23S RNA
4220	`QBETA`	Qβ 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

The HIV database entries do not necessarily represent the exact sequence of the viral genome, they often include the terminal repeats, which are present in the proviral genome. For the biological analysis described in section 4 we consider of course the foldings of the exact viral RNAs. The performance analysis described in the present section was carried out with the "raw" database entries.

The computations reported in this section have been performed on the `Delta` at the California Institute of Technology. The `Delta` system is a message-passing multicomputer, consisting of an ensemble of individual and autonomous nodes that communicate across a two-dimensional mesh interconnection network. It has 513 computational `i860` nodes, each with 16 Mbytes of memory and each node has a peak speed of 60 double-precision Mflops, 80 single-precision Mflops at 40 MHz. The operating system is Intel's Node Executive for the mesh (NX/M).

In the following we will use t to denote the time required to perform the folding in real time on the `Delta`, while $T = tN$ refers to the total time used for the computation.

In order to measure the pure CPU requirements of the folding algorithm (as opposed to I/O and message passing overhead) we have folded our test sequences on a single node, or in the cases where this was not possible because of the memory requirements we have extrapolated a hypothetical single-node CPU requirement T^* from T versus N curves. For short sequences we have $T^* = T(1)$. Fig. 4 shows a plot of T^* as a function of the chain length n. A linear regression yields

$$\log(T^*) \approx (2.34 \pm 0.04) \log n \; - \; (3.86 \pm 0.15) \tag{4}$$

for the logarithm of the single-node execution times.

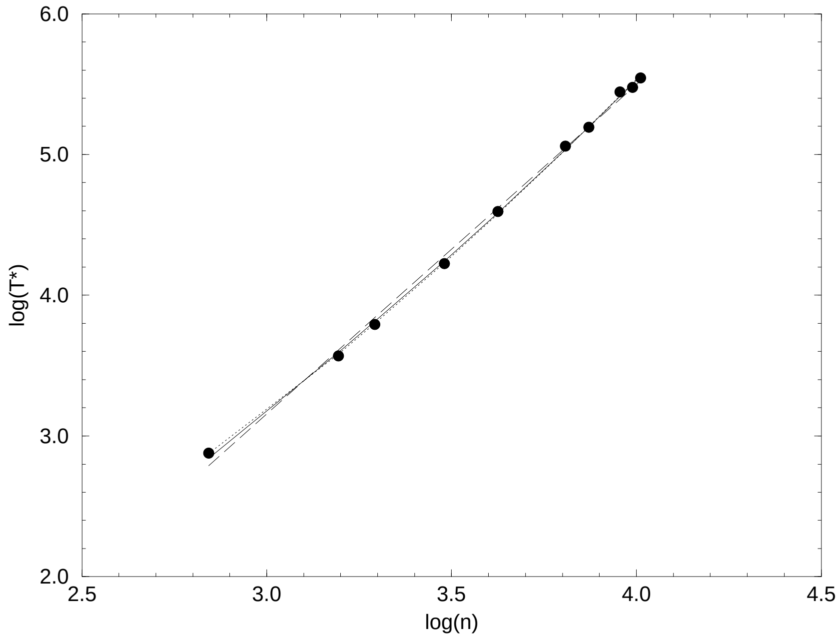


Figure 5: Plot of T^* versus chain length n. The fitted lines are a linear regression (dashed), equ.(5), and $T = an^3 + b\mathcal{M}^2 n^2 + c$. The latter fit yields a=226ns, b=729ns, and the residual c=108s. The direct fit to equ.(5) yields a=209ns and b=803ns, which is in reasonable agreement with the data in table 2, which have been obtained on $N \gg 1$ nodes.

This result is surprising at first sight, since the analysis of the algorithm reveals an eventually cubic dependence of the execution time on the chain length as discussed in the previous section. However, the exhaustive search for interior loops of sizes up to a cut-off $\mathcal{M}$ requires $\mathcal{O}(n^2\mathcal{M}^2)$ instructions. This term dominates the CPU requirements for "small" n.

In order to estimate this term we have performed a few computations with $\mathcal{M} = 15$ instead of the usual value of $\mathcal{M} = 40$. Assuming that the CPU requirements are approximately of the form

$$T \approx a\,n^3 + b\,\mathcal{M}^2\,n^2 \tag{5}$$

we may obtain the coefficients from

$$an^3 = \frac{T_2\mathcal{M}_1^2 - T_1\mathcal{M}_2^2}{\mathcal{M}_1^2 - \mathcal{M}_2^2} \qquad \text{and} \qquad bn^2 = \frac{T_1 - T_2}{\mathcal{M}_1^2 - \mathcal{M}_2^2} \tag{6}$$

This approach yields much more accurate estimates for the parameters a and b than obtaining these coefficients directly from a T^* versus n plot by least square fitting. The results are compiled in table 3. The cubic term dominates for chain lengths $n > (b/a)\mathcal{M}^2 \approx 7000$.

Table 3. Estimated coefficients for equation (5)

n	N	$a\,[ns]$	$b\,[ns]$
3023	8	203	837
3023	16	211	872
4220	16	234	984
6421	64	243	973
7440	64	225	890
9022	64	223	915
		223 $\pm$14	912 $\pm$53
T^*		210	800

Values obtained from equ.(6) for different values of n and from a direct fit of equ.(5) with $\mathcal{M} = 40$ are shown.

The *efficiency* of the parallelization is measured by

$$\mathcal{E}(N) := \frac{T^*}{Nt} \tag{7}$$

where T^* is the (hypothetical) single node execution time, N is as usual the number of nodes used for the calculation and t is real time used for the folding (including the backtracking step). The data in figure 6 show that we achieve efficiencies of more than 90% when the smallest possible number of nodes is used for the computation.

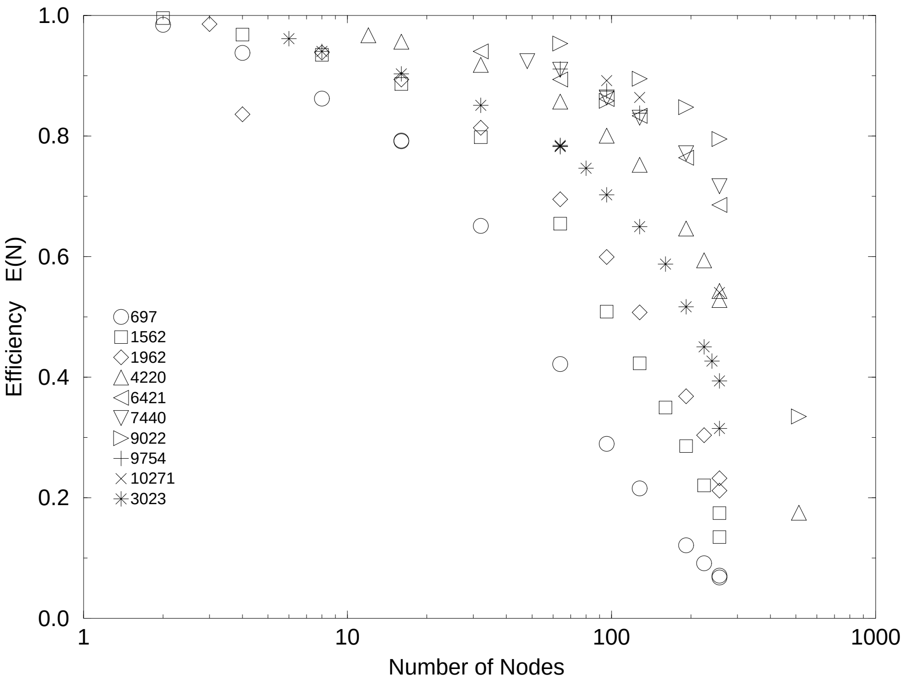


Figure 6: Efficiency of the parallelization versus the number N of nodes.

Since the number of message passing events is $\mathcal{O}(nN)$, with the size of the messages not depending on the number of nodes N, we expect that to first order the execution time will depend on N as

$$T = T^* + \alpha N + \dots \tag{8}$$

with a coefficient α which depends in turn on the chain length n. Consequently, the efficiency should decrease linearly for moderately small N. This is observed

Table 4. Comparison of performance of our implementation with the performance of the MP-2 implementation described in (Shapiro *et al.*, 1995). All timing results are wall clock times.

Sequence	GenBank Name	Length	Platform	Nodes	Time
rhinovirus	PIHRV14	7208	MP-2	8192	3 : 59 : 05
Polio Virus	POL2LAN	7440	Delta	48	0 : 58 : 35
				256	0 : 14 : 10
HIV-1	HIVNY5CG	9022	Delta	64	1 : 13 : 44
				256	0 : 22 : 05
HIV-1	HIVRF	9128	MP-2	16384	6 : 26 : 41
HIV-2	HIV2UC1GNM	10271	Delta	96	1 : 08 : 04
				256	0 : 42 : 09

in fact. As expected, the efficiency loss due to the parallelization decreases with increasing chain length.

In table 4 we display a comparison between the performance of the `MasPar MP-2` implementation (Shapiro *et al.*, 1995) and our code for the `Delta`. Note that the algorithm used in (Shapiro *et al.*, 1995) does i nclude the prediction of suboptimal structures. It involves the computation of the full square matrices where our algorithm needs only triangular matrices and hence it requires about twice the number of operations. Taking this into account, our implementation still runs 2 to 3 times faster on the minimum number of nodes that satisfy the memory requirements. Performance analyses like the one presented above are of course always as much comparisons between the hardware as they are comparisons of the software implementations. The important issue is the fact that our implementation is flexible in the number of nodes is uses, and hence the communication overhead between the processors can be reduced. Indeed Shapiro and coworkers use the maximum number of nodes and do report a communication overhead of 50%. Our code needs only a small fraction of the `Delta` supercomputer for about an hour per folding. This fact greatly facilitates the use of secondary structure prediction as a routine method, and put us into the position to report a comparative study of lentivirus genomes in the following section.

4. Applications: Secondary Structure of Lentiviruses

Retroviruses are viruses that in their life cycle alternate between a single stranded RNA stage and a double stranded DNA stage. The lentiviruses are a taxon with the retroviruses, they are characterized by long incubation times and a similar genomic organization (see Figure 7). The genome of a lentivirus consist of a single RNA molecule with about 7000 to 10000 nucleotides. Almost all of this genome is used for coding for various viral proteins and RNA secondary structures. Below we sketch the life cycle of lentiviruses in which we highlight the role of some of the known functional secondary structures.

After the viral RNA has entered the host cell, it needs to be *reverse transcribed* into DNA. The initiation of the reverse transcription requires the interaction of the so called Primer Binding Site (PBS) in the viral RNA with a tRNA, the primer (Rhim *et al.*, 1991). The specific secondary structure conformation in HIV-2 of the PBS and its surrounding RNA exposes part of the PBS for its interaction with the primer (Berkhout & Schoneveld, 1993). After the virus has been reverse transcribed into DNA and integrated into the host DNA it is called a provirus. The provirus DNA has to be *transcribed* into messenger RNAs to express its proteins. The transcription process is activated by an RNA secondary structure located at the 5' end of the viral genome, the so called *trans*-Activating Responsive element (TAR) (Jeang *et al.*, 1991).

Proteins in retroviruses can be encoded in different, overlapping reading frames. In retroviruses this is among others the case for the *gag* and *pol* genes. In order to get the different proteins expressed, translation has to shift from one reading frame into another. This process, called ribosomal frame-shifting, is facilitated by the presence of hairpins and pseudoknots downstream of the position of the shift (for a review see (Hatfield & Oroszlan, 1990)). A hairpin structure downstream of the translation shift site has been observed for a wide variety of retroviruses (Le *et al.*, 1989). In order to get the genes downstream from *gag* and *pol* expressed, a number of which are not contiguous, the transcription product has to be spliced in a number of alternative ways. The major splice donor (SD) lies in a conserved secondary structure (Harrison & Lever, 1992). Since the transcription product of

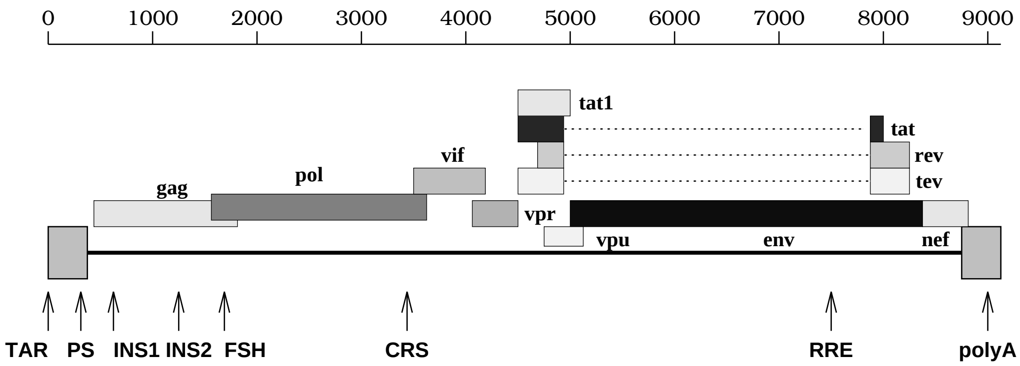


Figure 7: Organization of the HIV-1 genome. Proteins are shown on top, known features of the RNA are indicated by arrows below.
The major genes are *gag*, *pol*, *env*, *tat* and *rev*. These genes are present at similar locations in all lentiviruses. The *gag* gene codes for structural proteins for the viral core. The *pol* gene codes among others for the reverse transcriptase and the protein that integrates the viral DNA (after reverse transcription) into the host DNA. The *env* gene codes for the envelope proteins. The *tat* and *rev* genes code for regulatory proteins Tat and Rev that can bind to TAR and the RRE respectively. INS1, INS2 and CRS are RNA sequences that destabilize the transcript in the absence of the Rev protein. FSH refers to the hairpin that is involved in the ribosomal frameshift from *gag* to *pol* during translation. PolyA refers to the polyadenylation signal.

retroviruses can both serve as messenger RNA for the production of proteins and as genetic material for its next generation, the virus faces a regulatory issue: when and how to switch from using the transcription product for protein production to using it as genetic material. An important aspect of the switch is to stop the splicing of full length transcripts. The Rev response element (RRE), a secondary structure located in the *env* gene in lentiviruses, plays a crucial role in this switch. The interaction of the Rev protein with the Rev response element (RRE) reduces splicing and increases the export of unspliced viral RNA from the nucleus of the cell, see (Kimura & Ohyama, 1994) and references therein. As the newly produced viral RNAs need to be packed into virion particles, they need to be discriminated from other viral and cellular RNAs. The packaging signal, which has a conserved secondary structure, serves as a recognition site for the packing (Hayashi *et al.*, 1993). A schematic drawing of an HIV-1 genome, which includes the positions of the above discussed RNA structures and the major genes, is shown in Figure 7.

The minimum free energy structure was predicted for the 22 available full-length

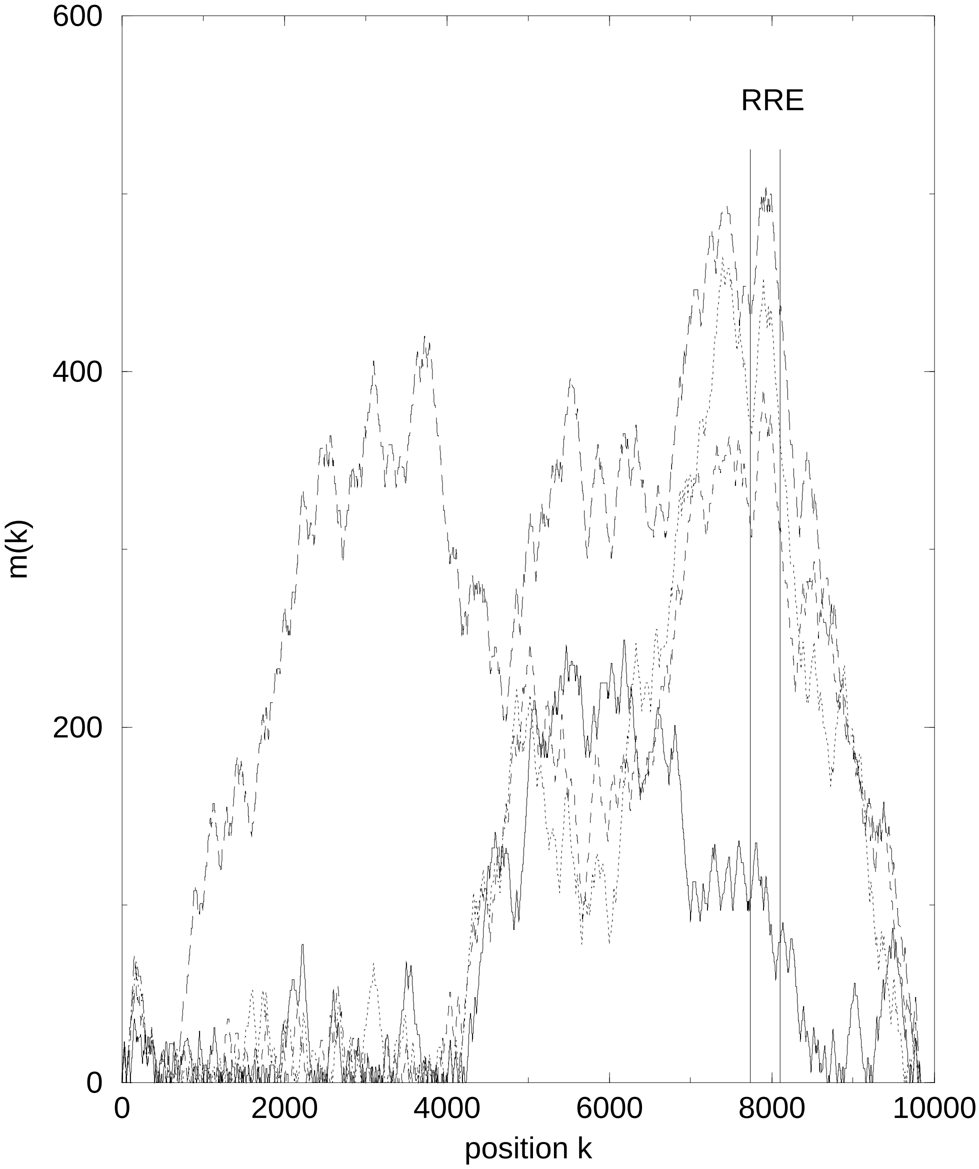


Figure 8: Mountain representation of the secondary structure of the full length genomes of HIVELI (long-dashed line), HIVANT70 (solid line), HIVLAI (dotted line) and HIVMAL (dashed line). The sequences represent three subtypes of HIV-1 of which full length genomes are available, and a one recombinant of two subtypes (HIVMAL). Variation of secondary structure within one subtype was considerably less than between subtypes. The vertical lines designate the position of the Rev response element as defined in (Mann *et al.*, 1994), including a correction for the alignment at the level of the sequences.

HIV-1 sequences[2] and for 9 sequences of related lentiviruses[3]. Figure 8 shows the mountain representation of 7 out of the 22 HIV-1 sequences. These sequences represent three HIV-1 types (Korber *et al.*, 1994) for which full-length sequences are known and one recombinant of two types. The majority of the secondary structures exhibit two distinct domains: whereas the 5' half consists of a large number of fairly small components, the 3' part is a single component (except for the 3' repeat of the about a hundred nucleotides which form the TAR and the polyadenylation signal). This was also observed in the most of the HIV-1 sequences which are not displayed in Figure 8. The boundary between the two structural domains coincides roughly with the end of the *pol* gene.

At the 5' end of the viral HIV-1 RNA molecule resides the *trans*-Activating Responsive (TAR) element (Berkhout, 1992), which interacts with the regulatory Tat protein. The binding of the Tat protein to TAR is responsible for the activation and/or elongation of transcription of the provirus (Feng & Holland, 1988; Klaver & Berkhout, 1994). On the basis of biochemical analysis (Baudin *et al.*, 1993) and computer prediction of the 5' end of the genome it is known that the TAR region in HIV-1 forms a single, isolated stem loop structure of about 60 nucleotides with about 20 base pairs interrupted by two bulges. This structure is indeed predicted in the minimum free energy structures of six of the seven sequences in Figure 9. Besides in HIV-1, a functional TAR structure has also been observed in HIV-2 and various SIV types (reviewed in Berkhout, 1992), while all other known lentiviruses have a *tat* gene, see (Saltarelli *et al.*, 1993) and the references therein. Although the secondary structure of TAR is strongly conserved within HIV-1, it varies considerably between the various human (HIV-1 and HIV-2) and simian (SIV) lentiviruses, as is also reflected in the minimum free energy foldings. Our analysis (Figure 10) shows that CAEV, visna, EIAV and BIV all have a short hairpin structure at their 5' end. These might have a similar function as TAR in HIV and SIV, although their small size makes specific recognition by a protein unlikely.

[2] HIVANT70, HIVBCSG3C, HIVCAM1, HIVD31, HIVELI, HIVHAN, HIVHXB2R, HIVJRCS-F, HIVLAI, HIVMAL, HIVMN, HIVMVP5180, HIVNDK, HIVNL43, HIVNY5CG, HIVOYI, HIVRF, HIVSF2, HIVU455, HIVYU10, HIVYU2, HIVZ2Z6

[3] EIAV (equine infectious anemia virus), CAEV (caprine arthritis encephalitis virus), BIV106 (bovine immunodeficiency virus), VLVCG (visna virus), three monkey immunodeficiency viruses SIVMM239, SIVMM251 (from macaque) and SIVSYK (from Syke's monkey), and two HIV-2 sequences HIV2BEN and HIV2ST.

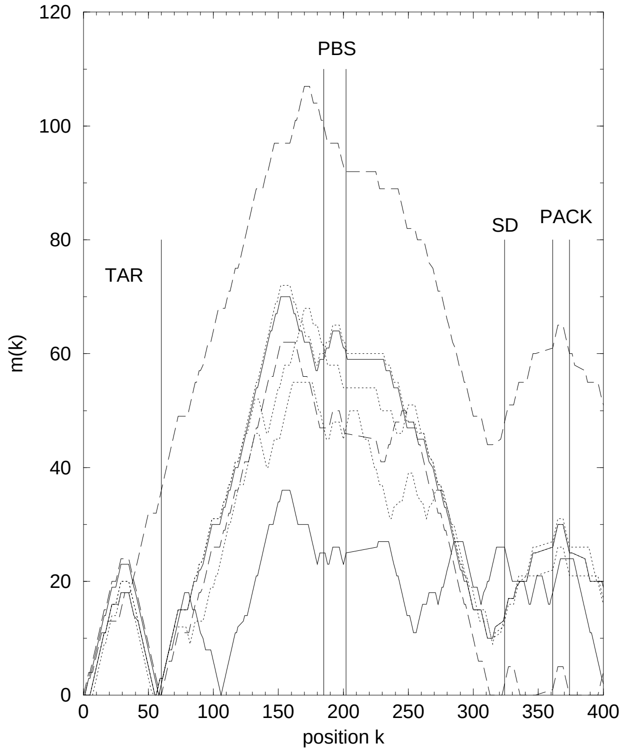


Figure 9: Mountain representation of the secondary structure of the 5' end of seven HIV-1 sequences (HIVLAI, HIVOYI, HIVBCSG3C: dotted line, HIVELI,HIVDNK: dashed line, HIVANT70: solid line, HIVMAL: long-dashed line) The secondary structures were aligned at the sequence level. Although the structures do show considerable variation, some features are conserved.
1) The TAR hairpin structure is present in six out of seven sequences.
2) The center of the Primer Binding Site (PBS) is always single stranded (sometimes as a hairpin loop, sometimes as an internal loop), thus exposing this part of the sequence for base pairing with the tRNA primer.
3) The center of the packaging signal (PACK) is always present as a hairpin.

The packaging signal is essential for the packaging of full length genomes into new virion particles. All analyses of its secondary structures are consistent with a short (5 base pairs) hairpin structure that carries a **GGAG** loop (Harrison & Lever, 1992; Sakaguchi *et al.*, 1993; Hayashi *et al.*, 1993; Baudin *et al.*, 1993; Clever *et al.*, 1995). Indeed, this feature is shared by all the sequences in Figure 10. However, the predictions in the literature for the more global secondary structure of this region of the RNA (beyond the 6 base pair hairpin) vary considerably. A

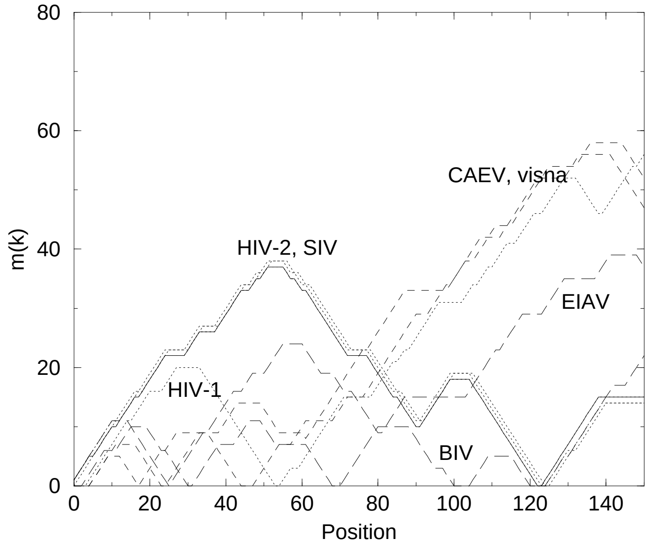


Figure 10: Mountain representation of the secondary structure of the first 150 nucleotides for different lentiviruses, SIVMM239, SIVMM251, HIV-2, BIV, EIAV, CAEV and Visna. The secondary structure of HIV-1LAI is shown for comparison. For HIV-1 (57 bases), HIV-2 and SIV (120 bases), the TAR structure is the first structure from the left. Which structures serve as TAR in EIAV, CAEV, Visna and BIV is not known. The HIV-2 and SIV sequences show a different secondary structure (with two hairpins) than the consensus structure (Berkhout & Schoneveld, 1993), which has three hairpins. This consensus secondary structure has only a slightly higher free energy and hence an only a slightly lower chance of occuring than the two hairpin structure reported here (data not shown). The other sequences, EIAV, CAEV, Visna and BIV, all show a short hairpin structure at the 5' end. CAEV and Visna are similar in their structure.

large variation in the predicted secondary structures is also present in the minimum free energy structures of the various HIV-1 sequences.

The Primer Binding Site (PBS) at the 5' of the viral genome (Berkhout & Schoneveld, 1993; Baudin *et al.*, 1993) is necessary for the initiation of reverse transcription of the HIV genomic RNA into DNA. It is a sequence of 18 nucleotides that is complementary to the nucleotides at the 3' end of the tRNA with which it base pairs. The tRNA serves as a primer to initiate the reverse transcription of the viral RNA. In absence of the primer, part of the Primer Binding Site is paired to bases outside the PBS (Figure 9). The binding of the primer could therefore lead to a rearrangement of the secondary structure of the 5' end of the molecule.

Indeed, such rearrangements were observed up to 69 nucleotides upstream and 72 nucleotides downstream of the PBS after the binding of the primer (Isel *et al.*, 1995). Computer prediction of the secondary structure of RNA can play a role in guiding these types of experiments and explaining their results.

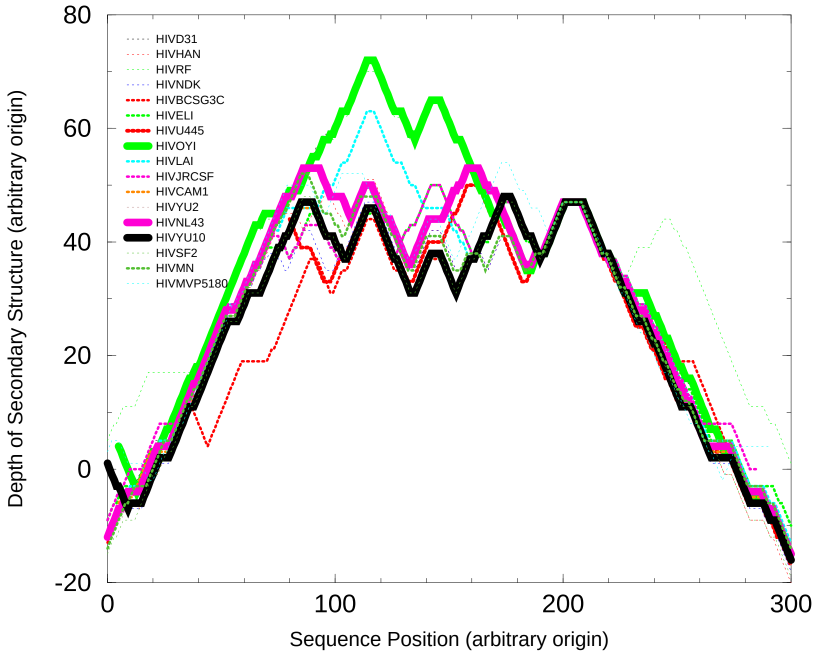


Figure 11: Alignment of the RRE regions of 17 sequences based solely on the minimum free energy secondary structure. The mountain representation reveals the five-fingered motif, the Roman numerals correspond to the numbering of the hairpins in (Dayton *et al.*, 1989). 5 out of 22 sequences showed a different pattern here. We find three different folding patterns each highlighted by one example. The first one (thick black line) corresponds to the consensus five-fingered motif that is presented in (Konings, 1992). The second one (light gray) is present among other in HIVLAI. It shown in (Huynen *et al.*, 1995) that this structure has high structural versatility; one of its alternatives with comparable thermodynamic stability is the five-fingered consensus structure. The third (dark gray) corresponds to the structure proposed (Mann *et al.*, 1994).
The alignment in this figure is based solely on the secondary structure and does not contain gaps. It is centered around the hairpin VI which appears in all the foldings.

Within the *env* gene of lentiviruses resides the Rev response element (RRE). The consensus secondary structure of the RRE in HIV-1 is a multi-stem loop structure consisting of five hairpins supported by a large stem structure (Konings, 1992). The interaction of RRE with the Rev protein reduces splicing and increases the transport of unspliced transcripts to the cytoplasm, which is necessary for the formation of new virion particles (Malim *et al.*, 1989; Malim & Cullen, 1989; Mann *et al.*, 1994; Kimura & Ohyama, 1994). Figure 11 shows an alignment of the RRE region of 17 out of the 22 HIV-1 sequences based entirely on the predicted secondary structures and without gaps. Most of the secondary structures show the five-fingered hairpin motif. An alternative structure is present in which hairpin III is relatively large and a few of the other hairpins have disappeared from the minimum free energy structure. A complete analysis of the base pair probability distribution of HIV_{LAI} showed that hairpin II, IV, and V, as well as the basis of hairpin III are meta-stable in the sense that they allow for different structures with nearly equal probabilities (Huynen *et al.*, 1995). This structural versatility within a single sequence is here reflected in the variation in the minimum free energy secondary structure of closely related sequences. Although there is structural versatility in the hairpins, the stem structure on top of which the hairpins are located is generally present in the minimum free energy folding.

5. Discussion

Our implementation of dynamic programming RNA folding algorithms on up to 512 nodes of the `Delta` supercomputer demonstrates that massively parallel distributed memory computer architectures are well-suited to the problem of folding the largest RNA sequences available. With sequences comprising several thousand nucleotides, efficiencies above 80% are obtained on partitions of the machine containing about 100 nodes. As the partition size increases beyond 100 nodes the efficiency deteriorates to only 20-40%, even for the larger sequences studied. Not surprisingly, the optimal partition sizes are those for which the total available memory on each node is utilized. These results are extremely encouraging. Apart from the insight that can be shed on the HIV virus itself, they indicate that even longer virus genomes containing up to 30000 nucleotides can be folded on the existing `Delta` architecture, with future scalable machines promising to extend this range even further. One long molecule of special interest is the Ebola virus, which contains roughly 20000 nucleotides.

The minimum free energy structure of a set of HIV-1, HIV-2, and related lentivirus has been determined. The results show the presence of known secondary structures such as TAR, RRE, and the packaging signal that have been predicted on the basis of biochemical analysis, phylogenetic analysis, and the folding of small fragments of the sequence. In HIV-1 we observe a striking difference between the secondary structures of the first half and the second half of the molecule. Whereas the first 4000 nucleotides form a large number of independent components, the second 5000 nucleotides form a single huge component, on top of which the RRE is located. In general, although the overall pattern with short range interactions in the 5' end and long range interactions at the 3' end, and although some relatively local patterns appear conserved, there is extensive variation in the secondary structure between the various HIV-1 sequences.

The folding algorithm discussed in this paper predicts only the thermodynamically most stable secondary structure. Under physiological conditions, i.e., at or above room temperature, however, RNA molecules do not take on only the most stable structure, they seem to rapidly change their conformation between structures with similar free energies. A realistic investigation of RNA structures has to account for this fact which is of utmost biological importance.

The simplest way to do this is to compute not only the optimal structure but all structures within a certain range of free energies (Waterman & Byers, 1985). A more recent algorithm (McCaskill, 1990) is capable of computing physically-relevant averages over all possible structures, by calculating an object known as the partition function. From it, the full matrix $P = \{p_{ij}\}$ of base pairing probabilities, which carries the biologically most relevant information about the RNA structure, can be obtained. The additivity of the free energy contributions in the secondary structure model implies a factorization of the partition function which allows a dynamic programming scheme analogous to the Zuker-Stiegler algorithm described in this contribution. CPU requirements will be comparable, the main difference being that double-precision floating point variables are required for the main arrays instead of integers. This doubles the memory requirements. In fact, a sequential implementation (Hofacker *et al.*, 1993) has been ported recently to a `CRAY-Y-MP` and has been successfully applied to analyzing the base pair probabilities of a complete HIV-1 genome (Huynen *et al.*, 1995). A comparative analysis of base-pair probabilities of RNA viruses requires an implementation of the partition function algorithm on massively parallel computers. Work in this direction is in progress.

Acknowledgments

This research was performed in part using the CACR parallel computer system operated by Caltech on behalf of the Center for Advanced Computing Research. Access to this facility was provided by the California Institute of Technology. Partial financial support by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung*, Proj. No. P 9942 PHY, is gratefully acknowledged. PES acknowledges support from the Aspen Center for Physics. This work was supported by the Los Alamos LDRD program and by the Santa Fe Institute Theoretical Immunology Program through a grant from the Joseph P. and Jeanne M. Sullivan Foundation.

Appendix

In this appendix we describe the details of the message passing steps. Let d index the diagonal. Note again that all elements within one diagonal can in principle be calculated in parallel.

In order to describe how exactly the diagonals are divided between the processors some notation needs to be introduced: Let $i^{\nu}_{(l)}(d)$ and $i^{\nu}_{(r)}(d)$ be the left-most and the right-most rows that are processed by node $\nu = 0, \ldots, N-1$ in diagonal d. Each node ν processes the entries $[i, i+d]$ for all i between $i^{\nu}_{(l)}(d)$ and $i^{\nu}_{(r)}(d)$. Furthermore, let

$$p = \lfloor \frac{n-d}{N} \rfloor \qquad \text{and} \qquad q = n - d \,(\mathrm{mod}\, N), \tag{A.1}$$

i.e., $pN + q = n - d$, the total number of entries in diagonal d.

We define the boundaries between the nodes by

$$\begin{aligned} i^{\nu}_{(l)}(d) &= \begin{cases} \nu p + 1 & \Longleftrightarrow \quad \nu < N - q \\ \nu p + [\nu - (N-q)] + 1 & \Longleftrightarrow \quad \nu \geq N - q \end{cases} \\ i^{\nu}_{(r)}(d) &= \begin{cases} (\nu+1)p & \Longleftrightarrow \quad \nu < N - q \\ (\nu+1)p + [\nu - (N-q)] + 1 & \Longleftrightarrow \quad \nu \geq N - q \end{cases} \end{aligned} \tag{A.2}$$

It is easy to check that in fact $i^{\nu+1}_{(l)}(d) = i^{\nu}_{(r)}(d) + 1$, and that the number of entries processed by each node is

$$i^{\nu}_{(r)}(d) - i^{\nu}_{(l)}(d) + 1 = \begin{cases} p & \Longleftrightarrow \quad \nu < N - q \\ p+1 & \Longleftrightarrow \quad \nu \geq N - q. \end{cases} \tag{A.3}$$

Now consider the diagonal $d' = d + 1$. We have two cases: (i) $q = 0$, i.e. all processors in diagonal d have exactly p entries. Then $p' = p - 1$ and $q' = N - 1$; (ii) otherwise we have $p' = p$ and $q' = q - 1$.

If $q = 0$ then the length of the segment changes for node $\nu = 0$ only. All other nodes have to deal with $p' + 1 = p$ nodes again, i.e., their left and right margins are shifted by -1.

If $q \neq 0$ it follows immediately that, with $X = N - q$.

$$
\begin{aligned}
i^{\nu}_{(l)}(d+1) &= \begin{cases} i^{\nu}_{(l)}(d) & \text{if} \quad \nu < X \\ i^{\nu}_{(l)}(d) & \text{if} \quad \nu = X \\ i^{\nu}_{(l)}(d) - 1 & \text{if} \quad \nu > X \end{cases} \\
i^{\nu}_{(r)}(d+1) &= \begin{cases} i^{\nu}_{(r)}(d) & \text{if} \quad \nu < X \\ i^{\nu}_{(r)}(d) - 1 & \text{if} \quad \nu = X \\ i^{\nu}_{(r)}(d) - 1 & \text{if} \quad \nu > X \end{cases}
\end{aligned} \tag{A.4}
$$

That is, the length of the segment stays the same for all nodes except for the one with index $\nu = X$.

Both cases can be subsumed in the above formula if X is redefined to be

$$
X = N - q \,(\mathrm{mod}\, N) = \begin{cases} N - q & \iff \quad q > 0 \\ 0 & \iff \quad q = 0 \end{cases} \tag{A.5}
$$

It is trivial to check that q satisfies the recursion

$$
q' = \begin{cases} q - 1 & \iff \quad q > 0 \\ N - 1 & \iff \quad q = 0 \end{cases} = q - 1 \,(\mathrm{mod}\, N) \tag{A.6}
$$

From this we obtain immediately the following recursion for X:

$$
X' = X - 1 \ (\mathrm{mod}\, N). \tag{A.7}
$$

Hence we can use (A.4) and (A.7), together with the initial conditions defined by (A.1), (A.2), and (A.5), for describing the updating of the boundaries between the processors. These relations are important for understanding the message passing requirements.

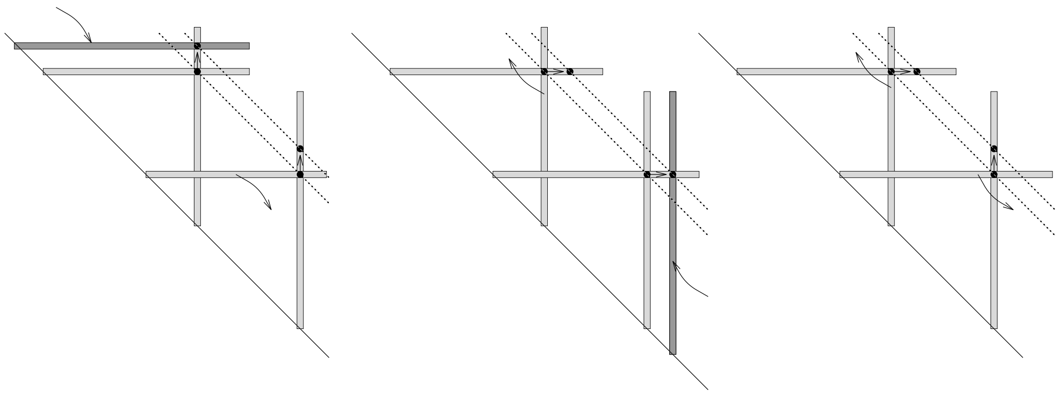
At a given diagonal d the information stored on each processor is the following

(1) The trapezoidal piece of memory needed for the array C, described in the main text.

(2) The rows $i^{\nu}_{(l)}(d) \leq i \leq i^{\nu}_{(r)}(d)$ for both the F and the FM array.

(3) The columns $j^{\nu}_{(l)}(d) = i^{\nu}_{(l)}(d) + d \leq j \leq\leq i^{\nu}_{(r)}(d) + d = j^{\nu}_{(r)}(d)$.

The trapezoidal piece of the C array is also delimited by these row and column numbers. The only difference is that only a piece of length $\mathcal{M}$ has to be stored



(1) (2) (3)

Figure 12: Each processor has to send an/or receive at most one row or column of data to its neighbor when the computation proceeds from one diagonal to the next. For the details consult the text of the appendix.

for each row or column. Hence message passing becomes necessary after each diagonal, see Figure 12.

We have to distinguish only three cases:

(1) $i^{\nu}_{(l)}(d+1) = i^{\nu}_{(l)}(d) - 1$ and $i^{\nu}_{(r)}(d+1) = i^{\nu}_{(r)}(d) - 1$.
In this case the required columns are the same for both step d and step $d+1$. The last row, $i^{\nu}_{(r)}(d)$ is no longer needed at processor ν. We have now $i^{\nu+1}_{(l)}(d+1) = i^{\nu}_{(r)}(d)$, hence this row has to be sent to node $(\nu+1)$. Correspondingly, $i^{\nu}_{(l)}(d+1) = i^{\nu-1}_{(r)}(d)$, thus this row has to be received from node $(\nu-1)$.

(2) $i^{\nu}_{(l)}(d+1) = i^{\nu}_{(l)}(d)$ and $i^{\nu}_{(r)}(d+1) = i^{\nu}_{(r)}(d)$.
In this case the required rows are the same for both step d and $d+1$, while the columns have been shifted. By the same reasoning as above we have to send the column $j^{\nu}_{(l)}(d) = j^{\nu-1}_{(r)}(d+1)$ to node $(\nu-1)$ and we have to receive the column $j^{\nu}_{(r)}(d+1) = j^{\nu+1}_{(l)}(d)$ from node $(\nu+1)$.

(3) $i^{\nu}_{(l)}(d+1) = i^{\nu}_{(l)}(d)$ and $i^{\nu}_{(r)}(d+1) = i^{\nu}_{(r)}(d) - 1$
In this case we have to send the column $j^{\nu}_{(l)}(d) = j^{\nu-1}_{(r)}(d+1)$ to node $(\nu-1)$, but we need no additional information to the right.

Of course, node $\nu = 0$ never sends columns, and node $(N-1)$ never receives rows. Inspecting the conditions for the three cases above allows us to link them to the recursions for i_r and i_l. We have the following scheme

$$\begin{array}{lcccc} & \text{send} & & & \text{receive} \\ \nu < X & \text{col}[j^{\nu}_{(l)}(d)] & \rightarrow & (\nu-1) & \text{col}[j^{\nu+1}_{(l)}(d)] \\ \nu = X & \text{col}[j^{\nu}_{(l)}(d)] & \rightarrow & (\nu-1) & - \\ \nu > X & \text{row}[i^{\nu}_{(r)}(d)] & \rightarrow & (\nu+1) & \text{row}[i^{\nu-1}_{(r)}(d)] \end{array}$$

where again node 0 never sends and node $(N-1)$ never receives. Consequently we have $\mathcal{O}(Nn)$ message passing events, and the size of each messages is $\mathcal{O}(n)$.

References

Baudin, F., Marquet, R., Isel, C., Darlix, J., Ehresmann, B., & 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.

Berkhout, B. (1992). Structural features in TAR RNA of human and simian immunodeficiency viruses: a phylogenetic analysis. *Nucl. Acids Res.* **20**, 27–31.

Berkhout, B. & Schoneveld, I. (1993). Secondary structure of the HIV-2 leader RNA comprising the tRNA-primer binding site. *Nucl. Acids Res.* **21**, 1171–1178.

Clever, J., Sassetti, C., & Parslow, T. (1995). RNA secondary structure and binding sites for gag gene products in the 5' packaging signal of human immunodeficiency virus type 1. *J. Virol.* **69**, 2101–2109.

Dayton, E., Powell, D. M., & Dayton, A. I. (1989). Functional analysis of CAR, the target sequence for t he Rev protein of HIV-1. *Science,* **246**, 1625–1629.

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

Fontana, W., Griesmacher, T., Schnabl, W., Stadler, P., & Schuster, P. (1991). Statistics of landscapes based on free energies, replication and degradation rate constants of RNA secondary structures. *Monatshefte der Chemie,* **122**, 795–819.

Fontana, W., Konings, D. A. M., Stadler, P. F., & Schuster, P. (1993). Statistics of RNA secondary structures. *Biopolymers,* **33**, 1389–1404.

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

Harrison, G. & 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.

Hatfield, D. & Oroszlan, S. (1990). The where, what and how of ribosomal frameshifting in retroviral protein synthesis. *Trends Biochem. Sci.* **15**, 186–190.

Hayashi, T., Ueno, Y., & 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.

Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, L., Tacker, M., & Schuster, P. (1993). `Vienna RNA Package`. `ftp://ftp.itc.univie.ac.at/pub/RNA/ViennaRNA-1.03`. (Public Domain Software).

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, S., Tacker, M., & Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie,* **125** (2), 167–188.

Hogeweg, P. & Hesper, B. (1984). Energy directed folding of RNA sequences. *Nucleic acids research,* **12**, 67–74.

Huynen, M., Perelson, A., Viera, W., & Stadler, P. (1995). Base pairing probabilities in a complete HIV-1 RNA. Los Alamos Preprint LA-UR 95-1613.

Isel, C., Ehresmann, C., Keith, G., Ehresmann, B., & Marquet, R. (1995). Initiation of reverse transcription of HIV-1: secondary structure of the HIV-1 RNA/tRNA(3Lys) (template/primer). *J. Mol. Biol.* **247**, 236–250.

Jaeger, J. A., Turner, D. H., & 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., Hammarskjold, M.-L., & Rekosh, D. (1991). Regulation of HIV expression: mechanisms of action of Tat and Rev. *AIDS,* **5 (supple 2)**, 3–14.

Jones, R. (1992). Protein sequence and structure aligments on massively parallel computers. *Int.J.Supercomp.Appl.* **6**, 138–146.

Kimura, T. & 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.

Klaver, B. & Berkhout, B. (1994). Evolution of a disrupted TAR RNA hairpin structure in the HIV-1 virus. *Embo J.* **13**, 2650–2659.

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

Konings, D. & Gutell, R. (1995). A comparison of thermodynamic foldings with comparatively derived structures of 16s and 16s-like rrnas. *RNA,* .

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

Korber, B., MacInnes, K., Smith, R., & Myers, G. (1994). Mutational trends in V3 loop protein sequences observed in different genetic lineages of human immunodeficiency virus type 1. *J. Virol.* **68**, 6730–6744.

Le, S.-Y., Chen, J.-H., & Maizel, J. (1989). Thermodynamic stability and statistical significance of potential stem-loop structures situated at the frameshift sites of retroviruses. *Nucl. Acids Res.* **17**, 6143–6152.

Malim, M. & 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., & 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., & 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.

Rhim, H., Park, J., & Marrow, C. D. (1991). Deletions in the tRNA(Lys) primer-binding site of human immunodeficiency virus type 1 identify essential regions for reverse transcription. *J. Virol.* **65**, 4555–4564.

Sakaguchi, K., Zambrano, N., Baldwin, E., Shapiro, B., Erickson, J., Omichinski, J., Clore, G., Gronenborn, A., & Appella, E. (1993). Identification of a binding site for the human immunodeficiency virus type 1 nucleocapsid protein. *Proc. Natl. Acad. Sci., USA,* **90**, 5219–5223.

Saltarelli, M., Schoborg, R., Gdovin, S., & Clements, J. (1993). The caev tat gene trans-activates the viral ltr and is necessary for efficient viral replication. *Virology,* **197**, 35–44.

Shapiro, B., Chen, J., Busse, T., Navetta, J., Kasprzak, W., , & Maizel, J. (1995). Optimization and performance analysis of a massively parallel dynamic programming algorithm for RNA secondary structure prediction. *Int.J.Supercomp.Appl.* **9**, 29–39.

Shapiro, B. A. (1988). An algorithm for comparing multiple RNA secondary stuctures. *CABIOS,* **4** (3), 387–393.

Shapiro, B. A. & Zhang, K. (1990). Comparing multiple RNA secondary structures using tree comparisons. *CABIOS,* **6**, 309–318.

Waterman, M. & Byers, T. (1985). A dynamic programming algorithm to find all solutions in a neighborhood of the optimum. *Math. Biosci.* **77**, 179–188.

Waterman, M. & Smith, T. F. (1978). RNA secondary structure: A complete mathematical analysis. *Math. Biosci.* **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.

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

Zuker, M. & Stiegler, P. (1981). Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. *Nucl. Acids Res.* **9**, 133–148.