Scanning RNA Virus Genomes for Functional Secondary Structures

Dissertation
zur Erlangung des akademischen Grades
Doctor rerum naturalium

Eingereicht an der
Formal- und Naturwissenschaftlichen Fakultät
der Universität Wien

von
Mag. Martin Fekete

Institut für Theoretische Chemie und
Molekulare Strukturbiologie

Wien, im März 2000

Meinen Eltern

Dank an alle, die mitgeholfen haben, besonders an ...

Peter den Jüngeren, Doktorvater und Ziel, was man alles wissen könnte,
Peter den Älteren, Doktorgroßvater, Institution und Inspiration,
Ivo, den unermüdlichen Erklärer der ruhigen Art und personifiziertes Computerwissen,
sowie an alle Kollegen, die mir Herausforderung und Freunde waren.

Zuerst Zeit für ...

Geniale Gedanken, tolle Sprüche ...

Zusammenfassung

Die Vorhersage der nativen dreidimensionalen Struktur von Biopolymeren, wie zum Beispiel von RNA, ist nach dem heutigen Stand der Wissenschaft noch sehr problematisch, mehr noch, in vielen Fällen unmöglich. Im allgemeinen ist die Funktion einer Sequenz nicht bestimmbar. Für die qualitative Beschreibung von RNA Molekülen ist die Sekundärstruktur oft ausreichend, da die Basenpaarungskontakte das Grundgerüst für die 3-dimensionale Struktur bilden.

Für die Berechnung der Sekundärstruktur, gibt es seit einiger Zeit praktikable Faltungsalgorithmen, die von der Sequenz ausgehend, eine Sekundärstruktur zurückliefern, wobei über Energiebeiträge der Basenpaare optimiert wird. Eine bessere Beschreibung der flexiblen Natur der RNA erlaubt die Berechnung der Zustandsumme über alle Strukturen und die Wahrscheinlichkeiten der Paarung einzelner Basen im Ensemble der Strukturen.

Lange RNA-Moleküle findet man in den Genomen von RNA-Viren. Diese viralen RNA Moleküle erfüllen im Virus zweierlei Aufgaben. Zum einen kodiert die Sequenz der RNA die viralen Proteine, zum anderen wird durch die Ausbildung bestimmter Sequenz- Struktur-Motive der Lebenszyklus des Virus reguliert. Eine Reihe spezifischer Strukturelemente, wie z.B. das *TAR* in *HIV* oder *IRES* in *Hepatitis C virus* oder *Picornaviridae* wurden bereits unter diesem Aspekt experimentell untersucht.

Funktionell wichtige Strukturelemente bleiben im Laufe der viralen Evolution konserviert. Schon wenige zufällige Mutationen würden ausreichen, Strukturelemente zu zerstören. Besonders nicht translatierte Bereiche des Virusgenoms sind möglicherweise funktionell bedeutend, da der hohe Selektionsdruck irrelevante Sequenzteile tendenziell eliminiert.

Konservierte Sekundärstrukturelemente können über rein theoretische Methoden identifiziert werden, indem man die vorhergesagten Strukturen verwandter Viren miteinander vergleicht. Eine Kombination von Sequenzvergleich und Sekundärstrukturvorhersage filtert aus einem verwandten Satz von Virusgenomen, z.B. Vertreter eines Genus, konservierte RNA Motive heraus. Dies erlaubt nicht nur eine qualitative Beschreibung von RNA-Viren, sondern könnte auch ein Ansatzpunkt für neue antivirale Strategien sein.

Abstract

The prediction of the native three dimensional structure of biopolymers, such as RNA, is currently problematic and often infeasible. In general the function of a sequence can not be determined. Often secondary structure is sufficient for a qualitative description, since base pairing contacts form the basis of the three dimensional structure.

For several years folding algorithms have been available that compute secondary structures from sequence data alone by energy minimization. A better description of the flexible nature of RNA is obtained by calculating the partition function and the base pairing probability matrix of the ensemble of all structures.

Long RNA molecules are located in virus genomes. These viral RNA molecules are responsible for two functions. On one hand they encode viral proteins, on the other hand they form characteristic RNA motifs regulating the viral life-cycle. Numerous specific RNA motifs, such as the *TAR*-region in *HIV* and the *IRES*-region in *Hepatitis C virus* or *Picornaviridae* have already been experimentally examined.

Functionally important secondary structures are conserved in the course of viral evolution. In the absence of selection, few random mutations are enough to destroy structure motives. Thus, conserved structures must carry some function, that converse a selectional advantage. Especially the non translated regions of virus genomes are probably functionally important, otherwise the high selection pressure would eliminate these regions.

Conserved RNA secondary structure elements can be identified by raw theoretical methods by comparing predicted structures of related virus genomes. A combination of sequence alignment and secondary structure prediction extracts conserved RNA motives from a sample of related sequences, such as members of one virus genus. The result is a qualitative description of RNA virus genomes and furthermore that could lead to establish new anti-viral strategies.

Contents

1 Introduction

RNA molecules are well known to have two functions in nature. The sequence of RNA encodes proteins on the other hand its structure can have functional importance, e.g. ribozymes. All RNA molecules form structures, but the presence of structure does not have any functional significance in itself.

If a structure element is preserved by selection this indicates it must of course have some function. This can be used to search for conserved RNA secondary structure elements in RNA sequences. A purely theoretical approach can be used to detect such elements, based on sequence information only.

This work considers RNA virus genomes, because they show a rather high sequence diversity in a related virus group, and are therefore ideal objects. We can perform this approach even on a small sample of sequences. Thus there are enough sequences in data bases for numerous virus genera.

Our approach is based on a combination of thermodynamic structure prediction and sequence alignment and allows us to detect common structure motifs in a related sample of sequences. The problem computing the secondary structure of long RNA-sequences was solved by porting the folding algorithms to parallel computer architectures. This enables us to fold entire viral genomes and thus to extend our search to sequence lengths up to 13000nt.

Especially long RNA virus genomes yield huge amounts of data and analysis cannot be done without a specialized selection tool. For this purpose a graphical user interface was developed to screen huge sequences for conserved RNA motifs. This makes analysis more efficient and faster, moreover the approach became more user friendly. A collection of software is now available that allows a routine investigation of even the largest viral RNA sequences.

The purpose of this work is to prove that a comprehensive survey of conserved RNA secondary structures in viral genomes is feasible, and that the resulting data provide a valuable basis for further investigations into viral evolution and phylogeny. Members of the virus families *Flaviviridae* and *Bunyaviridae* give an example that this approach is not restricted to already known structure elements, but also detects numerous conserved elements not previously described.

A list of conserved structure motifs, of course can not tell us what the function

of the conserved structure elements might be, nevertheless, knowledge about their location can be used to guide, for instance, deletion studies.

1.1 RNA

The native three dimensional structure of RNA is at present inaccessible to purely theoretical methods. Present day computer algorithms are not in the position to calculate the correct native structure from a given sequence. RNA secondary structure provides a course grained description of RNA structures that is both computationally convenient and biochemically useful. The secondary structure of RNA also enables computer experiments to find out regularities in RNA folding. That is because secondary structure provide the scaffold for tertiary structure formation.

Structure prediction algorithms based on thermodynamic criteria are sufficiently powerful to examine the sequence structure relations. These investigations of relations between sequence- and structure-space have been studied in a series of papers [57, 82, 24, 17], in order to discover regularities in sequence to structure mappings. The results presented in this sections are the base for the development of an approach to find common secondary structures in a set of diverse sequences, which are believed to be functionally important.The following general results for sequence-structure relations of RNA molecules are found.

- **There are more sequences than structures**
  The number of different sequences scales to $N(l) = 4^N$ whereas the number of structures scales to $S(l) \approx 1.48 l^{-3/2} (1.85)^l$. In other words we are dealing with many more sequences than structures. Thus the mapping from sequence space onto structure space is many to one and not invertible.

- **There are many common and few rare shapes**
  For long sequences almost all sequences fold into a vanishingly small fraction of all shapes.

- **Neutral networks are formed by common structures**
  Sets of sequences showing the same structure are connected through mutation in the sequence space [60]. Such connected sets have been

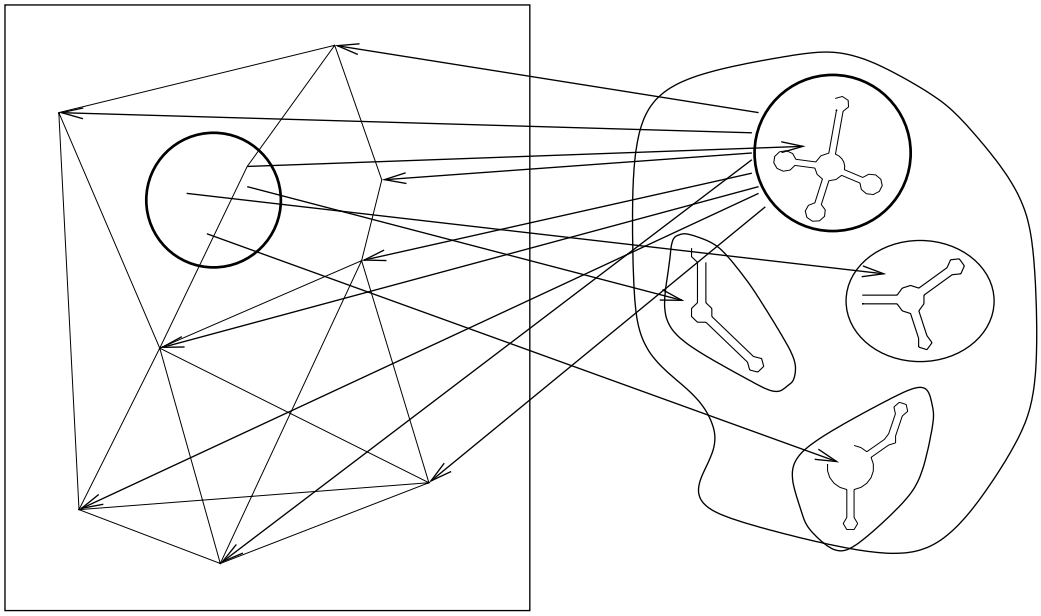


Figure 1: RNA sequence-structure map. Almost all structures can be found almost anywhere in sequence space and a small fraction of mutated positions almost surely changes the structure completely.

termed "neutral networks". Sequences on large neutral nets are characterized by a significant average fraction of nearest neighbors, that is, sequences that differ at a single nucleotide, that also fold into the given structure. A large enough degree of neutrality leads to percolation in the sequence space [57], causing connected neutral networks.

The algorithms used for the prediction of RNA secondary structures are based on thermodynamic rules. The most widely used methods compute a single minimum free energy structure through dynamic programming [53, 73, 82]. Approaches to kinetic folding [46, 18] are also based on the thermodynamic rules. Because of the approximations of the energy model and inaccuracies the measured parameters, the accuracy of these predictions is often insufficient. In cases where the correct structure is known from phylogenetic analysis it has been found that predicted structures contain only 30% to 80% of the correct base pairs [39, 35]. The correct structure can, however,

be found within a relatively small energy interval above the ground state.

There are variants of the folding algorithm for computing a sample of suboptimal folds [80], or even *all* structures within a prescribed energy range [76]. Non-deterministic kinetic folding algorithms [18, 14] can produce ensembles of structures by repeatedly running them with different random numbers. A much more elegant and efficient solution is the computation of the complete matrix of base pairing probabilities [44], which contains suitably weighted information about all possible secondary structures and therefore reduces the impact of inaccuracies in the structure prediction. The disadvantage of these methods is of course that they leave it up to the user to decide which of the proposed structures to believe.

1.2 Conserved RNA in Virus Genomes

Sequences can diverge while their structure remains conserved. On the other hand, a relatively small number of random mutations is sufficient to destroy structural motifs in the absence of selection. Thus, if structures are conserved in spite of sequence variation, the conserved structures must clearly carry an important function.

In this respect RNA viruses are an ideal proving ground. The high mutation rates, estimated to be from 10^{-5} to as much as 10^{-3} errors per nucleotide [30] should lead to unrelated structures. On this account consistently found RNA motifs or also sequences should be functional important.

As a consequence to high mutation rate , the virus populations include large numbers of mutants that allow rapid adaptation to new environmental conditions. This can lead to rapid functional divergence of the RNA viruses, as reflected in the rapid sequence divergence among closely related virus species, and even among progeny of a single virus.

Despite their high mutation rates, cis-acting sequences of RNA viruses, recognized for initiation of transcription or replication, can form conserved RNA motifs. Conserved sequences in related virus genomes can be a hint for functional sequence regions. Note, that computer investigations on RNA sequences have shown, that only 10% sequence diversity is enough to destroy common secondary structures if mutations are placed randomly. The sequence of the minimal promoter of *Alphaviruses* gives an example, that it is

all well conserved as the polymerase protein: the minimal promoter sequences of *Sindbis* and *Semliki Forest viruses* are identical at 83% of the nucleotides positions (range of 71-92% identity among the alpha-viruses). In comparison, their nsP4 genes, that encode the elongation activity of the viral polymerase, have 65% identity at the RNA level, and the corresponding nsP4 proteins are identical at 74% of the amino acid residues, 83% homology if amino acid similarities are included. Although cis-acting sequences are conserved on the sequence level, common secondary structures should be destroyed, if only 10% mutations occur randomly. Well known functional important sequences often show a lot of compensatory mutations, a hint, that folding the sequence to a specific secondary structure is crucial, e.g. *Pestivirus internal ribosome entry site* (IRES).

Conserved, probably cis-acting sequences are found in all families of RNA viruses. Most of them are found at or close to the termini of the genomic RNA, probably recognized for initiation of replication. Cis-acting parts of the genome sequences often play an important role for transcription initiation or termination. This has been already documented for various virus families such as *Picornaviridae, Flaviviridae, Togaviridae*, which include important human and animal pathogens.

Coupled Evolution

The considerations of the last sections can be resumed in a model of coupled evolution. Functional RNA structures are determined by viral or host proteins. Since the native three dimensional structure is crucial for RNA protein interactions and RNA secondary structure is a course grained description, therefore conserved structures can provide a qualitative description of these interactions.

Cis-acting sequences, e.g. the IRES of the 5'end of several RNA viruses are well known to specifically bind translation factors or ribosomal subunits. Suppose that the specificity of recognition is determined by a host protein, this protein evolves at very much slower rate and remains unchanged for relatively long time spans. The virus, however, mutates at high rates, such that the cis-acting sequence, folding in a special shape is rapidly selected to achieve an optimal interaction with the host protein. Once this occurs, most mutations in the cis-acting sequence will be sub-optimal, and be selected

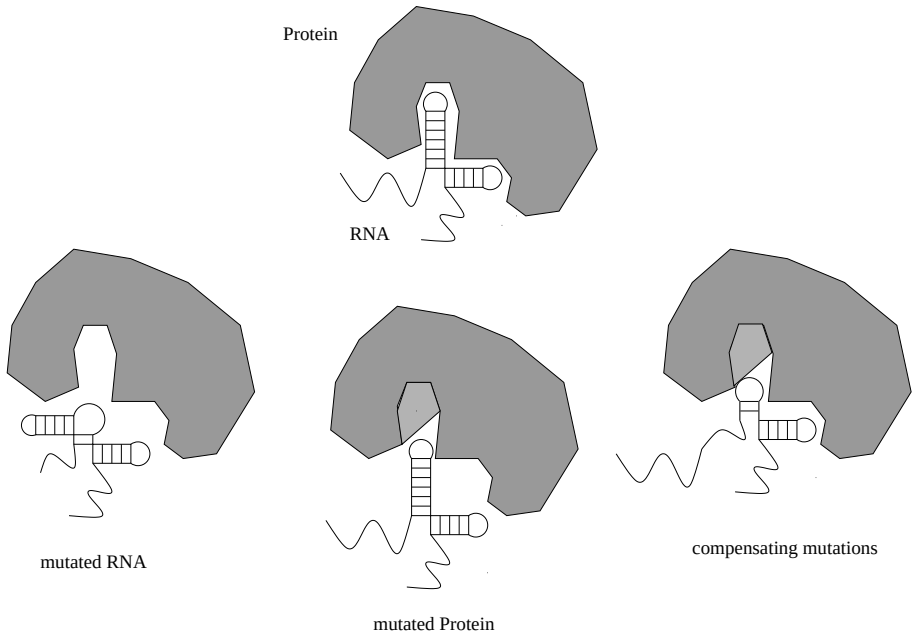


Figure 2: Four examples of protein RNA coevolution. Top figure, the shape of RNA is optimized to protein interaction. Bottom left, RNA is mutated and the shape is different , no interaction possible. Bottom middle, the protein is mutated and alters its surface, interaction to RNA disturbed. Bottom right, protein and RNA change their shapes in a compensatory manner, interaction enabled.

against. Thus, the cis-acting sequence will now evolve only at a rate comparable to the cognate host protein, or mutations inside base paired regions led to consistent or compensatory mutations. If, instead, recognition of the cis-acting sequence is mediated by a viral protein, their interaction should also be rapidly optimized [62]. Once this occurs, they become mutually constrained, neither can change independently without disturbing the optimized interaction. A change is only possible if both mutate coincidentally, and in an exactly compensatory fashion.

Although RNA viruses have high mutation rates, the predominant or *wild-type* genome persists with remarkable stability during passage in culture. This is true even though substantial numbers of mutants are detectable at each passage [7]. To reconcile this apparent paradox, it was proposed that the relevant sequences are quickly optimized when environmental conditions change (e.g., adaptation to culture) resulting in a predominant, *wildtype* se-

quence [62]. The *wildtype* sequence persists because, among the distribution of mutants generated during virus growth, none have a competitive advantage over the *wildtype*, so long as the environmental conditions remain stable [62].

The initial optimization process is likely to be facilitated by the high mutation rates and large population sizes that generate an enormous diversity for selection to operate upon efficiently. If the environmental conditions are altered, some other sequence might be selectively advantageous, and it becomes the dominant species, superior to most of the mutants that arise. This explanation for the persistence of the *wildtype* in culture may be generalized to evolution in nature. As viruses diverge over time, to adapt to disparate niches or environmental conditions, only those features that are the most strongly selected for under a variety of environmental conditions will remain conserved. Whether the cis-acting sequence is recognized by a host or a viral protein, the model predicts that it should evolve quite slowly compared to most of the rest of the genome. If this is true, then the recognition of cis-acting sequences should be functionally conserved, and also the secondary structure responsible for the specific shape for RNA-protein recognition.

2 Methods

Developing a method for searching for conserved RNA secondary structures is quite challenging, because different algorithms have to be linked, such as folding, or multiple alignment algorithms. Additionally a sorting procedure uses aligned sequences and secondary structures to extract common base pairs on a set of diverse sequences. The huge number of extracted base pairs by analyzing long viral genomes cannot be managed without a selection tool, it helps to pick out useful information in other words conserved RNA elements. In this section all necessary steps are explained, to give the reader an overview of our approach, but a parallel implementation of the folding algorithms, the aligned minimum energy folding algorithm and a graphical tool for selecting conserved elements is presented in more detail, because they are developed by the author himself.

Folding of RNA molecules is the most important step in searching for consistent RNA motifs, so an overview of folding algorithms is provided. Several algorithms exist for the prediction of RNA secondary structures based on thermodynamic rules. McCaskill proposed an algorithm to compute the partition function of the thermodynamic ensemble and the matrix of base pairing probabilities P_{hl} of an RNA molecule. The large size of, say, HIV genomes ($n \approx 9200$ nucleotides) implies that there is a huge number of low energy states. For example, the frequency of the minimum energy structure in the ensemble at thermodynamic equilibrium is in general smaller than 10^{-23} for RNAs of the size of a HIV viral genome. Hence one would need a huge number of different structures to adequately describe the ensemble. While such an approach is feasible for RNAs with up to some 100 nucleotides [76], the direct generation and analysis of the necessary amount of structure information for long sequences exceeds by far the capabilities of even the most modern computer systems. Porting these algorithms to parallel computer architecture was therefore desirable. The current implementation adheres to the *Message Passing Interface* (MPI) standard, which allows to use different parallel computer architectures [10]. Short RNA sequences are still folded by the seriell implementations of the folding algorithms. The `Vienna RNA Package`[1] is a software package for predicting and comparing RNA Secondary Structure [24].

[1] `http://www.tbi.univie.ac.at/~ivo/RNA`

Thermodynamic structure prediction of RNA is only the first step towards a search for conserved RNA secondary structures and computer resources are the bottleneck for large virus genomes. Several other algorithms are necessary to obtain a list of RNA motifs. A correct sequence alignment is crucial for the success of the whole procedure, therefore an advanced multiple sequence alignment algorithm was developed by Roman Stocsits [63] called `Ralign`. It is based on the multiple sequence alignment of `Clustal W`. Sorting procedures to extract common base pairs from a sample of aligned sequences are also described in this section.

2.1 RNA Secondary Structures

Most RNA molecules are single stranded *in vivo*, but the molecules can fold back onto itself to form double helical regions stabilized by Watson-Crick G-C and A-U base pairs or the slightly less stable G-U pairs. Base stacking and base pairing are hence the major driving forces of structure formation in RNA. Other, usually weaker, intermolecular forces and the interaction with aqueous solvent shape its spatial structure. As opposed to the protein case, the secondary structure of RNA sequences is well defined, provides the major set of distance constraints that guide the formation of tertiary structure, and covers the dominant energy contribution to the 3D structure. Furthermore, secondary structures are conserved in evolutionary phylogeny [19] and therefore represent a qualitatively important description of the molecules.

The secondary structure can be described as a set of vertices $V = \{1, 2, ..., i, ..., N\}$ and a set of edges $S = \{i \cdot j, 1 \leq i < j \leq N\}$ fulfilling

(1) For $1 \leq i < n$, $i \cdot (i+1) \in S$.

(2) For each i there is at most one $h \neq i-1, i+1$ such that $i \cdot h \in S$.

(3) If $i \cdot j \in S$ and $h \cdot l \in S$ and $i < h < j$, then $i < l < j$.

The first condition simply states that RNA is a linear polymer, the second condition restricts each base to at most a single pairing partner, and the third forbids pseudo-knots and knots. While pseudo-knots are important structural elements in many RNA molecules [75], they are excluded from many studies mostly for a technical reason [74]. In their absence the folding

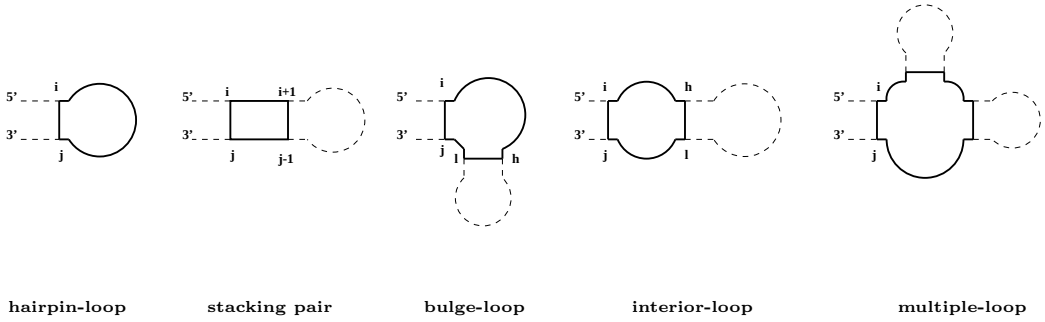


Figure 3: Secondary structures decompose into five distinct loop types, which form the basis of the additive energy model. One distinguishes three loop energy functions: $\mathcal{H}(i,j)$ for hairpin-loops, $\mathcal{I}(i,j,h,l)$ for the three types of loops that are enclosed by base pairs $i \cdot j$ and $h \cdot l$ and the additive model for multi-loops described in the text. Stacked pairs ($h = i+1, l = j-1$) and bulges (either $h = i+1, l \neq j-1$ or $l = j-1, h \neq i+1$) are treated as special cases of interior-loops. The energies depend on the types of closing base pairs indicated by $i \cdot j$ and interior base pairs as well as on the size of the loops.

problem for RNA can be solved efficiently by dynamic programming [81, 74]. In many cases pseudo-knots can be "added" to a predicted secondary structure graph during a post-processing step.

A *base pair* $h \cdot l$ is called *interior* to the base pair $i \cdot j$, if $i < h < l < j$. It is *immediately interior* if there is no base pair $p \cdot q$ such that $i < p < h < l < q < j$. For each base pair $i \cdot j$ the corresponding *loop* is defined as consisting of $i \cdot j$ itself, the base pairs immediately interior to $i \cdot j$ and all unpaired regions connecting these base pairs. In graph theoretical terms, the loops form the unique minimal cycle basis of the secondary structure graph [41].

The standard energy model for RNA contains the following types of parameters: (i) *base pair stacking* energies depend explicitly on the types of the four nucleotides $i \cdot j$ and $(i+1) \cdot (j-1)$ that stack. For the purpose of the recursions in table 1 it is useful to view stacked base pairs as a special type of interior-loop, hence we denote the stacking energies $\mathcal{I}(i,j,i{+}1,j{-}1)$. (ii) *loop energies* depend on the type of the loop, its size, the closing pairs and the unpaired bases adjacent to them, see figure 3. We write $\mathcal{H}(i,j)$ for hairpin-loops and $\mathcal{I}(i,j,h,l)$ for interior loops. Multi-loops energies are assumed to have a linear contribution of the form $\mathcal{M} = \mathcal{M}_C + \mathcal{M}_I \cdot degree + \mathcal{M}_B \cdot unpaired$, in addition the so-called dangling end energies are taken into account which refer to mismatches next to the base pairs that delimit the loop. The im-

plementation of the folding algorithms used in this contribution assumes the energy parameters summarized by [72], except that co-axial stacking of helices is neglected. Co-axial stacking is, strictly speaking not part of the secondary structure graph as defined above. The energy model is thus identical to Zucker's `Mfold 2.3` [78].

Minimum Free Energy versus Base Pairing Probabilities

The additive energy model of RNA secondary structure folding allows a elegant solution for minimum free energy folding and calculation of base pair probabilities, using dynamic programming algorithms. The minimum free energy calculation works by calculating optimal structures for all subsequences of the sequence. The result is an optimal structure and energy over all subsequences. The structure is obtained in a backtracking procedure. The minimum free energy algorithm calculates only one structure the thermodynamic most stable one, and no information about other possible alternate structures is given. This snap shot of the structure space does not tell us a lot about how probably this structure is in the ensemble of structures, or how well determined the ground state is. Calculating only the minimum free energy structure is unsatisfactory for two reasons. An RNA molecule will not always fold in its minimal energy configuration, changes between many structures of similar energy or within a given energy region happens and is often important for functionality. Secondly, if several structures have energies very close to the ground state, choosing one of them becomes arbitrary because of the inaccuracies of the used energy model. One possible solution to this problem is to generate all structures within a prescribed increment of the ground state [79, 76].

A more elegant solution was presented by McCaskill, who noticed, that the partition function Q of all secondary structures can be calculated by dynamic programming as well. The free energy of the ensemble can be obtained as $F = -kTlnQ$. Such an algorithm does not predict a secondary structure, instead one get the probability P_{hl} for the formation of a base pair $h{\cdot}l$. The number of steps necessary for calculating the minimum free energy or partition function scale $\mathrm{O}(n^3)$ with sequence length, the backtracking procedure of the base pair probability too, where backtracking of the ground state structure scales $\mathrm{O}(n)$. The use of integers instead of floating point figures allows a faster computation of the minimum free energy. The memory requirements of both

algorithms scale to O(n^2). Minimum free energy calculation is faster needs less memory. The larger sequences can be computed on serial computers, computing the base pair probabilities makes the use of parallel computers unavoidable. However the base pairing probabilities of a RNA molecules allows us a more detailed view of the structural properties, which is needed for a better understanding of the function of RNA secondary structures.

McCaskill's Algorithm

McCaskill's partition function algorithm naturally decomposes into two parts, namely the computation of the partition function and the subsequent computation of the pairing probabilities. We will refer to the two parts as *folding* and *backtracking*, respectively. The logic of the folding part is essentially the same as for minimum energy folding [81] while the backtracking part is much more elaborate. The recursions of McCaskill's algorithm are summarized in table 1. An efficient implementation for serial machines is part of the `Vienna RNA Package` [24]. In the reminder of this section we briefly review this algorithm.

The partition function of the complete RNA molecule can be derived from the partition functions of all its sub-sequences. For the sub-sequence from i to j we have to distinguish whether $i \cdot j$ forms a base pair or not. We write Q^B_{ij} for the partition function of the substring subject to the constraint that $i \cdot j$ is paired and Q_{ij} for the unconstrained partition function. Consequently, the partition function of the entire molecule is $Q = Q_{1n}$.

If i to j are paired, this pair can close either a hairpin-loop, an interior-loop delimited by $i \cdot j$ and $h \cdot l$, or a multi-component-loop. The three terms in table 1 correspond to these possibilities. Multi-loops can be dealt with efficiently due to a linear ansatz for their energies contributions. This allows for a decomposition into three terms: one for unpaired substructures, one for substructures consisting of single component, and a multi-component reminder. The auxiliary variables Q^M and Q^{M1} are necessary for handling multi-loop contributions. Introducing Q^A and restricting the size of interior-loops to $u \leq u_{\max}$ reduces the CPU requirements from $\mathcal{O}(n^4)$ to $O(n^3)$. Most programs set $u_{\max} = 30$. The restriction on the size of interior loops does not have a serious effect in practice, since long interior loops are energetically unfavorable and therefore very rare. For further details we refer to

Table 1: Recursion for Computing the Partition Function.
The parameter m is the minimum size of a hairpin-loop, usually $m = 3$.

Folding	**Backtracking**
$Q^B_{ij} = e^{-\mathcal{H}(ij)/kT} + \sum_{h=i+1}^{j-m-2} \sum_{\substack{l=h+m+1 \\ u \le u_{\max}}}^{j-1} Q^B_{hl}\, e^{-[\mathcal{I}(i,j,h,l)]/kT} + \sum_{h=i+1}^{j-m-2} Q^M_{i+1,h-1} Q^{M1}_{h,j-1}\, e^{-\mathcal{M}_\mathcal{C}/kT}$	$P^c_{hl} = \frac{Q_{1,h-1} Q^B_{hl} Q_{l+1,n}}{Q_{1n}}$
$Q^{M1}_{ij} = \sum_{l=i+m+1}^{j} Q^B_{il}\, e^{-[\mathcal{M}_\mathcal{I} + \mathcal{M}_\mathcal{B}(j-l)]/kT}$	$P^i_{hl} = \sum_{i=1}^{h-1} \sum_{\substack{j=l+1 \\ u < u_{\max}}}^{n} P_{ij} \frac{Q^B_{hl}}{Q^B_{ij}} e^{-\mathcal{I}(i,j,h,l)/kT}$
$Q^M_{ij} = \sum_{h=i+m+1}^{j-m-1} Q^M_{i,h-1}\, Q^{M1}_{hj} + \sum_{h=i}^{j-m-1} Q^{M1}_{hj}\, e^{-\mathcal{M}_\mathcal{B}(h-i)/kT}$	$P^m_{hl} = Q^B_{hl} e^{-[(\mathcal{M}_\mathcal{C} + \mathcal{M}_\mathcal{I})/kT]} \times \sum_{i=1}^{h-1} (P^{M1}_{il} Q^M_{i+1,h-1} + P^M_{il} Q^M_{i+1,h-1} + P^M_{il} e^{-[(h-i-1)\mathcal{M}_\mathcal{B}/kT]})$
$Q^A_{ij} = \sum_{l=i+m+1}^{j} Q^B_{il}$	$P^M_{il} = \sum_{j=l+2}^{n} \frac{P_{ij}}{Q^B_{ij}} Q^M_{l+1,j-1}$
$Q_{ij} = 1 + Q^A_{ij} + \sum_{h=i+1}^{j-m-1} Q_{i,h-1} Q^A_{hj}$	$P^{M1}_{il} = \sum_{j=l+1}^{n} \frac{P_{ij}}{Q^B_{ij}} e^{-[(j-l-1)\mathcal{M}_\mathcal{B}/kT]}$
	$P_{hl} = P^c_{hl} + P^i_{hl} + P^m_{hl}$

McCaskill's [44] original paper.

In the backtracking part of the algorithm, the pairing probabilities P_{ij} are obtained by comparing the partition functions Q^B_{ij} and Q_{ij} with and without an enforced pair $i \cdot j$. While the partition function for longer subsequences is computed from shorter ones during the folding part, the backtracking recursion proceeds in the reverse direction. The probability P_{hl} of the pair $h \cdot l$ is the sum of three independent terms: (i) it closes a component with probability P^c_{hl}, (ii) it is an interior base pair of an interior-loop, bulge, or stack with probability P^i_{hl}, or (iii) it is immediately interior to a multi-loop with probability P^m_{hl}. Again, two auxiliary arrays are needed to handle the multi-loop contribution in cubic time. The complete recursion is summarized in Table 1.

For long (sub)sequences the partition functions Q_{ij} become very large since they are the products of a large number of exponential functions. In order

to reduce the numerical problems we rescale the partition function of a subsequence of length ℓ by a factor $\tilde{Q}^{\ell/n}$, where $\tilde{Q}$ is an *a priory* estimate for the partition function. A sufficiently accurate estimate can be obtained from the ground state energy $E_{\min}$:

$$ln\tilde{Q} \approx -1.04 \times E_{\min}/kT \tag{1}$$

We use the message passing implementation of the minimum energy folding algorithm, which is described by [26, 27] to compute $E_{\min}$.

2.2 Representation of Secondary Structure

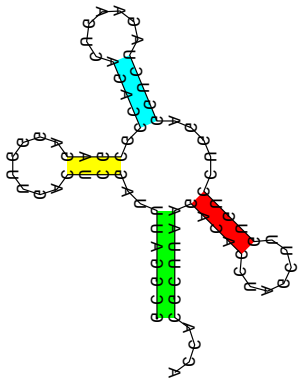
Looking at the raw output of folding algorithms is often less presentive, especially if long RNA sequences are folded. Graphical representation of folding output is therefore more user friendly and enables a better overview at all.

The used folding algorithms parallel or serielle version returns either the minimum free energy structure in bracket notation , its energy, or the free energy of the thermodynamic ensemble and the base pairing probability matrix of the sequence. It also produces `PostScript` files with plots of the resulting secondary structure graph and a *dot-plot* of the base pairing matrix.

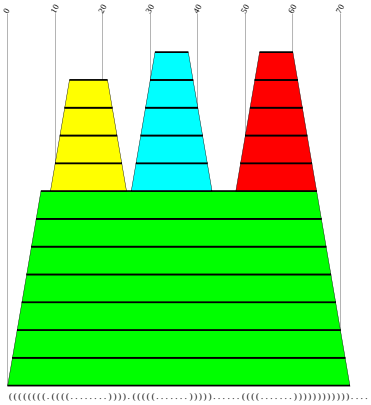
The programs read RNA sequence strings from `stdin` and calculate their minimum free energy structure, partition function and base pairing probability matrix [25, 44]. The output is a minimum free energy structure in bracket notation, its energy, or the free energy of the thermodynamic ensemble and the frequency of the minimum free energy structure in the ensemble. The *dot-plot* shows a matrix of squares with area proportional to the pairing probability in the upper half, and one square for each pair in the minimum free energy structure in the lower half. The results are used as an input for a search for consistently predicted RNA motifs in a set of related sequences [22, 23].

2.3 Parallel Folding of RNA Virus Genomes

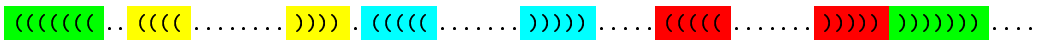
A former parallel computer implementation of the folding algorithms was restricted only to `Intel` hypercube or mesh architecture [9]. A new im-



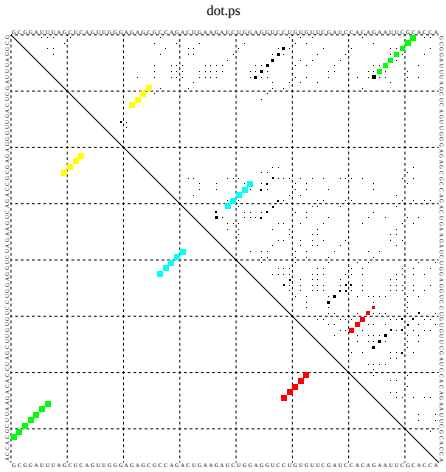
4.1 Structure graph



4.2 Mountain plot



4.3 Bracket notation



4.4 Dot plot

Figure 4: Representation of secondary structures produced by `Vienna RNA Package`, `Alidot`, `Pfrali` or `Alifold`. All drawings contain the same information. **Structure graph.** A two-dimensional drawing of base pairs contacts. **Mountain plot.** Base pairs are draw by horizontal lines, and plateaus symbolize unpaired regions. **Bracket notation.** Dots symbolize unpaired bases in sequence, and matching brackets predicted base pairs. **Dot plot.** The upper right part shows the predicted base pair probabilities computed with the partition function algorithm of the `Vienna RNA Package`. The area of the squares is proportional to the pairing probability. The lower left part gives the minimum free energy structure, see **Structure graph** for comparison. Note that the minimum free energy structure is the ground state structure in the thermodynamic ensemble.

plementation to the common *Message Passing Interface* standard [10] was therefore desirable.

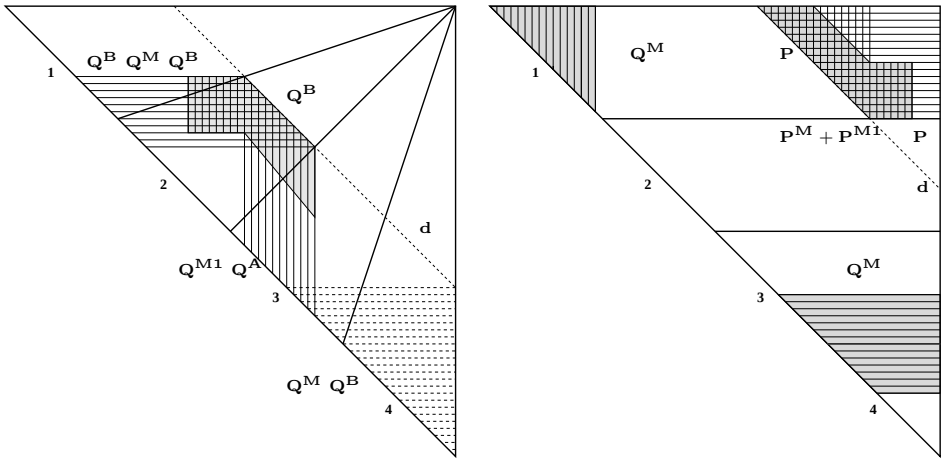
Secondary structure predictions of large RNA molecules with several thousand nucleotides are often performed by folding fairly small subsequences. This has two disadvantages, however, (i) by definition one cannot detect long-range interactions that span more than the size of the sequence window, and (ii) the results depend crucially on the window's exact location. This is because subsequences fold independently of the rest of the sequence only if they form a component by themselves, i.e., if there are no base pairs to the out-side of the sequence window. Often long range base pairs can not be neglected and folding of subsequences results in different predicted base pairs, i.e the panhandle structure of `Hantavirus`. The only way, however, of identifying the component boundaries or long range interactions is to fold the sequence in its entirety.

Folding of large sequences is quite demanding both in terms of memory and CPU time. For a sequence of length n, CPU time scales to $\mathcal{O}(n^3)$ and memory requirements to $\mathcal{O}(n^2)$. While this is not a problem for small RNA molecules, such as tRNAs, the requirements exceed the resources of most computers for large RNA molecules such as viral genomes. In most cases, memory, rather than computational speed, becomes the fundamental resource bottleneck. The use of modern parallel computers thus becomes unavoidable once the memory requirements exceed, say, 1GByte and many viral genome sizes are, unfortunately, well above this limit.

Message Passing

Since the folding and the back-tracking part are independent of each other it seems logical to parallelize them independently. The folding part can be parallelized in a way that is very similar to our earlier message passing implementation of minimum energy folding algorithm [24, 26]. However, some of the intermediate results (partial partition functions Q_{ij} and Q^B_{ij}) are required again during the backtracking stage. Storing these values in such a way that the backtracking recursion can efficiently be distributed among a large number of processors is the main difficulty of our task.

Memory Requirements for the Parallel Partition Function



5.1 Folding 5.2 Backtracking

Figure 5: Logical memory required by a single processor during folding and backtracking, resp., of the entries of sub-diagonal d.

Folding. The work is divided among the processors in sectors by evenly dividing each sub-diagonal d. The matrices Q, Q^M, and Q^B are stored in form of rows, the auxiliary arrays of Q^{M1} and Q^A as columns. Each processor calculates the entries of its part of sub-diagonal d (dashed line). The shaded region representing Q^B does not extend to the diagonal, because we have restricted the maximal size of interior-loops. After the calculation of one sub-diagonal d the rows of the Q^B and Q^M matrices are stored permanently (dashed lines), the memory allocated to the other arrays is recycled.

Backtracking proceeds from the longest subsequences to shorter ones. Each processor computes a horizontal slice of the triangle matrices in order to reduce the number of messages. The computation of P^i_{hl} requires entries of P from the shaded region, while newly calculated values of P are then stored in rows (horizontal stripes). The shaded rows and columns of Q^M (shaded, towards upper left and lower right) are needed for multi-loop contribution P^m. The auxiliary arrays P^M and P^{M1} (vertical stripes) are stored as columns; only those columns intersecting the current sub-diagonal are necessary.

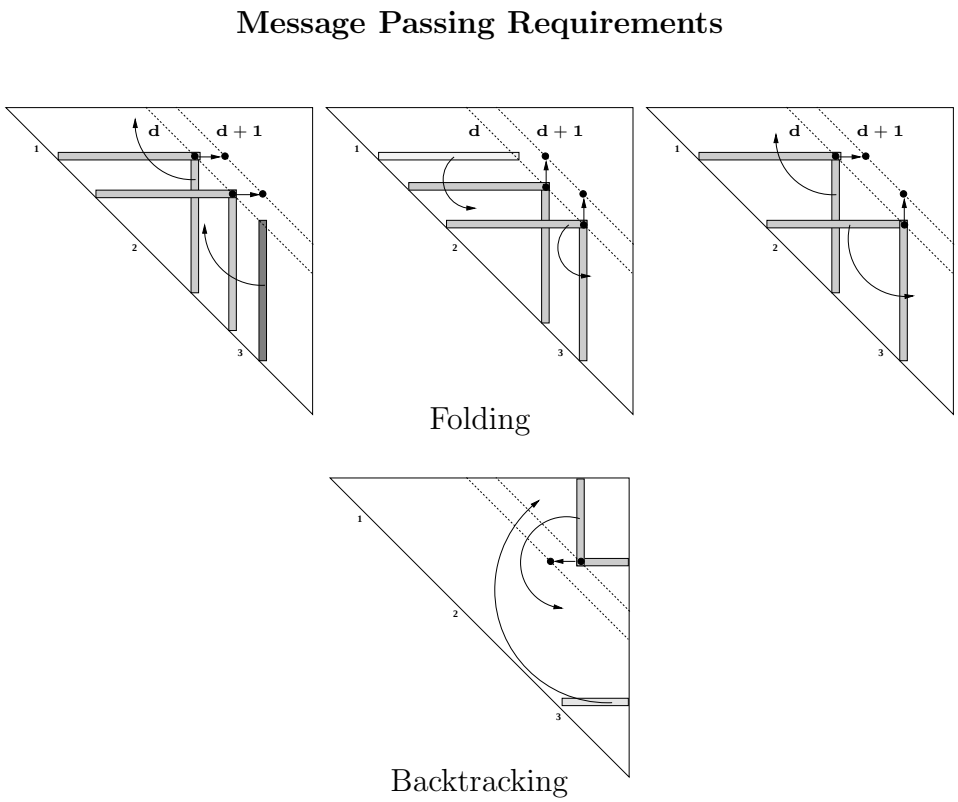


Figure 6: Message passing requirements.
Top: Folding. Each processor has to send an/or receive at most rows or columns of data to its neighbors when the calculation proceeds from diagonal d to $d+1$. We have to distinguish three cases: *left side* The required rows to calculate the sub-diagonal entry for d and $d+1$ are the same, while columns have been shifted. We have to send the left-most column to processor 1 and receive the left-most column of processor 3. *middle* The required columns stay the same for d and $d+1$. The right-most row is not needed anymore and is sent to processor 3, while processor 2 receives the right-most row of processor 1. *right side* In this case the left-most row is the same, we have to send the left-most column to processor 1, while the right-most row is not needed anymore and is sent to processor 3.
Below: backtracking. The required rows to compute a sub-diagonal entry from d to $d+1$ are always the same, while the columns are shifting. We have to send the right-most column of the processor k to processor $k+1$. Additionally we need rows of data, calculated during the folding procedure.

Folding

The crucial observation is that the computation of all those matrix entries that lie on the same sub-diagonal $(i, i+d)$ are independent of each other. Furthermore, they depend only on those entries that are located closer to the main diagonal. The computation therefore proceeds from the diagonal of the matrices Q_{ij}, Q^B_{ij}, etc. towards the corner $(1, n)$. In order to compute the entries $(i, i+d)$ all previously computed data from row i of the arrays Q, Q^B, and Q^M and from column $(i+d)$ of Q^{M1} and Q^A are necessary. Furthermore we need a triangular part of the Q^B array up to depth $u_{\max}$ for the interior-loop contributions. For each processor these triangles add up the trapezoidal area indicated in figure 5.1.

We divide each sub-diagonal as evenly as possible between the available N processors. Set $w = \lfloor (n-d)/N \rfloor$ and $r = n - d \mod N$. Then the first r processors calculate $w+1$ matrix entries, the remaining $N-r$ processors compute only w entries. After completing a sub-diagonal, each processor has to send either the right-most row or the left-most column of its memory to its right or left neighbor, respectively, see figure 6 (upper part). This arrangement, which is the same as for the minimum energy folding [27], is quite efficient since each processor sends and receives only n messages with $\mathcal{O}(n)$ bytes during the entire folding computation.

In contrast to minimum free energy folding, we need to store the entire arrays Q^B and Q^M for the backtracking part, where this information will be needed at different processors. Whenever the last entry of a row in Q^B and Q^M has been calculated, the data are stored for backtracking. A row i will be stored on node $\lfloor i \cdot N/n \rfloor$, so that the same number of rows is kept on each processor. This causes an additional n message passing operations during the folding procedure.

Backtracking

The backtracking part starts in the corner $(1, n)$ and proceeds towards the main diagonal. Again, the entries within each sub-diagonal are independent from each other. To compute a base pair probability P_{ij} we need Q^B and Q^M data that were calculated during the folding part as well as P, P^M and P^{M1} data that were calculated earlier during backtracking. A detailed description

of data required by each processor is given in figure 5.2.

To simplify memory access, we do not divide the sub-diagonals evenly between all processors. Instead, each processor computes a horizontal slice of the triangle matrices as shown in figure 5.2. The first n/N sub-diagonals are therefore computed by a single processor at the beginning of the backtracking part. The poor load balancing during the initial steps, is not crucial, however, since at the beginning all rows and columns are short and the computational effort is small. Towards the end of the backtracking procedure, when all rows and columns are long, the work is distributed ideally among the available nodes. Although the overall load balancing is somewhat worse than in the folding part, this arrangement minimizes the communication overhead, see figure 6 (lower part).

Memory Requirements

Table 2 summarizes the memory requirements of the message passing implementation. In order to ensure a reasonably efficient computation it is necessary to store some of the intermediate data more than once. The trapezoidal arrays are necessary for computing interior-loop contributions. Their height is determined by the constant $u_{\max}$, i.e., the maximum length of interior-loops for which we search rigorously. Their total size is $nu_{\max} + Nu_{max}^2$ and hence negligible in comparison to the triangular arrays. The matrices P^c, P^i, and P^m need not be stored explicitly. In addition, the matrix Q can be reused to store to the newly computed entries of P since in each sub-diagonal we need the Q-values that are located closer to the main diagonal (shorter subsequences) and P-values closer to the upper right corner.

Memory usage is thus dominated by the backtracking part of the algorithm. On each processor we need approximately

$$\mathtt{M} = \frac{1}{N}\left(3n^2 + nu_{max}\right) + 7n \tag{2}$$

real numbers. A number of arrays of length n, such as the last column of the matrix Q, are stored on each processor in order to facilitate memory access. In addition, a few integer fields of length n are used to manage memory and message passing.

For sequences longer that some 3000 nucleotides it is necessary in general

Table 2: Memory requirements.
The folding part requires 5 triangular matrices, while we need 6 such matrices for the backtracking part because the values of Q^M are required in both row and column form.

Matrix	row-wise	column-wise	trapezoidal
Folding			
Q^B	`Qb`		`Qb`
Q^M	`Qm`		
Q^{M1}		`Qmm`	
Q^A		`Qq`	
Q	`Q`		
Backtracking			
	`Qm`	`Qm`	
	`Qb`		
P	`Pr`		`Pr`
P^M		`Prml`	
P^{M1}		`Prmlt`	

to use double precision reals. Hence we need some $2.5G$Bytes to fold a HIV sequence with our implementation.

CPU Requirements

The present implementation is suitable for routinely folding large genomic virus RNAs with a chain length of sometimes more than 10 000 nucleotides, see table 3 for performance data.

The exact number of instructions required for computing the partition function is sequence dependent. We tested the performance of our parallel program on several RNA virus genomes, such as $Q\beta$ bacteriophage, $n = 4220$, polio viruses, $n \approx 7500$, and HIV viruses, $n \approx 10\,000$). 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 CPU time consumed on all processors.

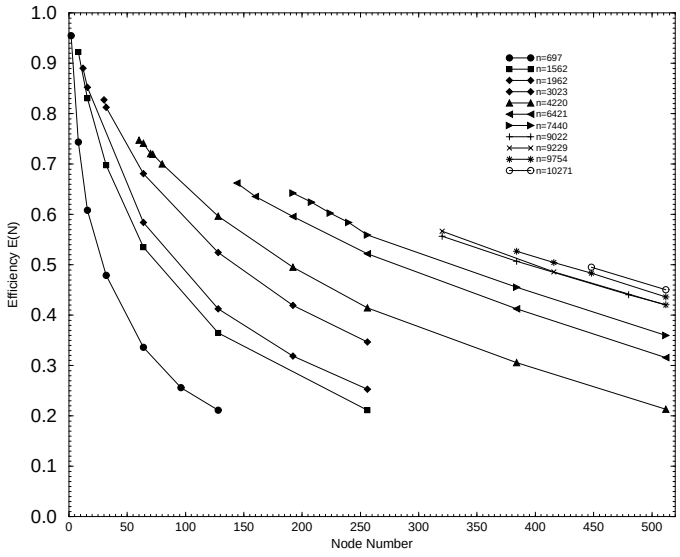


Figure 7: Efficiency of parallelization versus number N of processors on the `Intel Delta`.

The total computational effort is represented quite well by

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

where an^3 comes from the calculation of multi-loops and $bu_{max}^2 n^2$ is determined by the calculation of interior-loops. From several test runs on the $Q\beta$ sequence with different values of u_{max} we obtain $a = 900$ns and $b = 1200$ns. The CPU requirements vary very little with the sequence composition. In order to measure the pure CPU requirements of the folding algorithm (as opposed to I/O and message passing overhead) we have extrapolated folding times for different numbers N of processors to a hypothetical single-node CPU requirement T^*. The *efficiency* of the parallelization is then given by

$$\mathcal{E}(N) := T^*/(Nt) \tag{4}$$

The data in figure 7 show that we achieve efficiencies of more than 50% when the smallest number of nodes satisfying our memory requirements is used. The computation of the minimum energy for estimating $\tilde{Q}$, equ.(1) take less than 20% of the total execution time. Folding and backtracking each need about 40% of the total time.

Table 3: Wall clock times calculating base pairing probability matrix.

Hardware	Sequence	n	N	t (min)
Pentium II 450 Mhz (serial)	Qβ	4220	1	84.0
`Beowulf` Pentium II 450 Mhz	Qβ	4220	16	14.5
	HIV LAI	9229	16	123.6
	Pestivirus	12573	17	315.2
`Intel Delta`	HIV LAI	9229	320	77.0

Recently cost-effective workstation clusters have become widely available. We use a `Beowulf` architecture consisting of 9 two-processor PCs (Pentium II, 450Mhz) with 512MByte each, connected by 100Mbit `Fast Ethernet`, running `Linux` and `LAM 6.3`. This setup is sufficient for the routine computation of base pairing probability matrices from complete RNA virus genomes. Typical execution times are compiled in table 3. For comparison, folding the HIV LAI sequence, $n = 9229$, took about 77min using 320 processors on the `Intel Delta` and 2h on 16 Pentium II 450MHz. The serial code took $42h$ on a `DEC alpha` and $64h$ on `Cray YMP` for the same sequence.

Despite the relatively slow network connection in the `Beowulf` workstation cluster we find efficiencies above 50% on 16 nodes already for the chain length $n = 4000$. The efficiencies increase somewhat for larger n. Executing the parallel code on a single CPU shows that the overhead from the parallelization is about 20% to 25%. This is mainly because some parts of the algorithm can be implemented more efficiently in the serial version, where the memory organization is not constrained by requirements of easy message passing.

2.4 Sequence Alignment

An alignment is the most basic sequence analysis task to realize whether two or more sequences are related and how close this relationship is in terms of sequence similarity. To find the best possible alignment of sequences is of central importance for bioinformatics and data processing after routine laboratory procedures like sequencing nucleic acids. Some alignment algorithms exist which are used to find an optimal alignment, and, of course, a scoring system is necessary to rank alignments. In principle all known algorithms are

based on two criteria, (i) maximum similarity or (ii) minimum (Hamming-) distance [13, 16, 21].

For evaluating the difference between two sequences we have three possibilities of pairs of opposite symbols: (i) identity, (ii) substitution or mismatch and (iii) insertion or deletion. The procedure is usually done by first aligning the sequences and then deciding whether that alignment has occurred because the sequences are related, or just by chance. In any case the scoring system should help to answer this question regarding to identical and similar positions in the alignment. (Similar pairs of residues in amino acid alignments are those which have a positive score in the substitution matrix used to score the alignment, e.g. aspartate-glutamate pairs, D-E, both negatively charged amino acids.)

Careful thought must be given to the scoring system used to evaluate an alignment by looking for evidence when sequences have diverged from a common ancestor by a process of mutation and selection. As mentioned above, the basic mutational processes that are considered are substitutions, which change, and insertions and deletions, which add or remove residues in a sequence and are referred to as 'gaps'. The total score we assign to an alignment is a sum of terms for each aligned pair of residues, plus terms for each gap. Informally, using an additive scoring system we expect identities and conservative substitutions to be more likely in good (biologically relevant) alignments than we expect by chance, and so they should contribute positive score terms. And on the other hand non-conservative changes are expected to be observed less frequently so they contribute negative score terms. This system also corresponds to the assumption that we can consider mutations at different sites in a sequence to have occurred independently (treating a gap of arbitrary length as a single event). All alignment algorithms depend crucially on such a scoring scheme and from a biological point of view the assumption of independence appears to be a reasonable approximation for DNA and protein sequences, although we know that intra-molecular interactions between residues of a protein play a very important role in determining protein structure. Regarding the secondary structures of RNAs, where base pairing introduces very critical long range dependencies, the model of independent mutations is biologically inaccurate [33, 37, 40].

We need score terms for each aligned residue. We derive substitution scores from a probabilistic model that gives a measure of the relative likely-hood

that the sequences are related as opposed to being unrelated. Models assign a probability to the alignment in each of the two cases. Then we consider the ratio of the two probabilities. The *random* model R assumes that a letter in the sequence (for proteins an amino acid or one of the four bases in the case of DNA or RNA) occurs independently with some frequency q, and hence the probability of the two sequences is the product of the probabilities of each amino acid (or base):

$$P(x, y|R) = \prod_i q_{x_i} \prod_j q_{y_j} \tag{5}$$

where x and y is a pair of sequences. In the alternative *match* model M, aligned pairs of residues occur with a joint probability p_{ab}. This value p_{ab} can be thought of as the probability that the residues a and b have each independently been derived from some unknown original residue c in their common ancestor (c might be the same as a and/or b). This gives a probability for the whole alignment:

$$P(x, y|M) = \prod_i p_{x_i y_i} \tag{6}$$

The ratio of these two likelihoods is the odds ratio:

$$\frac{P(x, y|M)}{P(x, y|R)} = \frac{\prod_i p_{x_i y_i}}{\prod_i q_{x_i} \prod_i q_{y_i}} = \prod_i \frac{p_{x_i y_i}}{q_{x_i} q_{y_i}} \tag{7}$$

We want to arrive at an additive scoring system, so we have to take the logarithm of this ratio, known as the log-odds ratio:

$$S = \sum_i s(x_i, y_i) \tag{8}$$

where

$$s(a, b) = \log(\frac{p_{ab}}{q_a q_b}) \tag{9}$$

is the log likelihood ratio of the residue pair(a, b) occurring as an really valid aligned pair, as opposed to an unaligned pair (or by chance joined pair of residues or nucleic acids). We can see that S in this equation is a sum of individual scores $s(a, b)$ for each aligned pair of residues which can be

arranged in a matrix. The highest positive entries in the matrix are given for identical residue pairs, lower, but also positive, values do the conservative substitutions have while non-conservative substitutions give a negative score. So it is possible to derive scores, in fact $s(a, b)$ in the above equation, for every pair of residues in the alignment. Any matrix like this is making a statement about the probability of observing ab pairs in real (biologically relevant) alignments and is called substitution matrix or score matrix or weight matrix.

There are two possibilities for penalizing gaps: the standard cost associated with a gap of length g could be given by a linear score

$$\gamma(g) = -gd \tag{10}$$

where d is called the gap open penalty. It makes a difference whether a gap is newly opened or an existing gap is just extended. A type of score could be used which is known as the affine score

$$\gamma(g) = -d - (g-1)e \tag{11}$$

where e is called the gap extension penalty. This penalty should be smaller than the gap open penalty d, so that extension of existing insertions (or deletions) is penalized less than opening further gaps. Gap penalties also correspond to a probabilistic model of alignment. We assume that the probability of a gap occurring at a particular site in a given sequence is the product of a function $f(g)$ of the length of the gap, and the combined probability of the set of inserted residues,

$$P(\text{gap}) = f(g) \prod_{i \in \text{gap}} q_{x_i}. \tag{12}$$

The form of this equation as a product of $f(g)$ with the q_{x_i} terms corresponds to an assumption that the length of the gap is not correlated to the residues it contains. The natural values for the q_a probabilities here are the same as those used in the random model above, because they both correspond to unmatched independent residues. When we divide by the probability of this region according to the random model to form the odds ratio, the q_{x_i} terms cancel out. This gives a term dependent on length $\gamma(g) = log(f(g))$ where the gap penalties correspond to the log probability of a gap of that length.

After having determined a certain scoring system we need an algorithm for finding an optimal alignment for a pair of sequences. We have

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \simeq \frac{2^{2n}}{\sqrt{2\pi n}} \tag{13}$$

possible global alignments between two sequences of length n. The quantity of possible alignment solutions grows by about 4^n This means for sequences of length 30 there are 10^9 possibilities, and with length 60 we have 10^{18} possible alignments. But in terms of molecular biology sequences of length 30 or even 60 are comparatively short and often it is necessary to find the best alignment between sequences which have a length of a few thousand amino acids or nucleotides (like in the case of virus genomes). It is of course not computationally feasible to enumerate all these, even for moderate values of n.

So we need to find a way which gives us the possibility to gain optimal alignments without testing and valuing every possible solution. The algorithms for finding optimal alignments given an additive alignment score of the type described above is called dynamic programming [45, 51, 52].

Multiple Alignments

Using dynamic programming in order to align just two sequences guarantees a mathematically optimal alignment. But attempts at generalizing dynamic programming to multiple alignments are limited to small numbers of short sequences [42]. For much more than ten or so proteins of average length, the problem is infeasible given current computer power. Therefore, all of the methods capable of handling larger problems in practical time-scales make use of heuristics. Nowadays, the most widely used approach is to exploit the fact that homologous sequences are evolutionary related. Multiple alignment are produced progressively by a series of pairwise alignments, following the branching order in a phylogenetic tree [11]. First all possible pairs of sequences are aligned to derive a distance matrix in order to calculate the initial guide tree which is built up by the distances between the sequences. Then the most closely related sequences get aligned progressively according to the branching order in the guide tree, gradually adding in the more distant ones when we already have some information about the most basic mismatches or gaps.

This approach is fast enough to allow alignments of virtually any size. Further, in most (simple) cases, the quality of the alignments is very good, as judged by the ability to correctly align corresponding domains from sequences of known secondary or tertiary structures [2]. The placement of gaps in alignments between closely related sequences is much more accurate than between distantly related ones. Therefore, the positions of the gaps which were introduced during the early alignments of the closely related sequences are not changed as new sequences are added. One problem is that this approach becomes less reliable if all of the sequences are highly divergent. More specifically, any mistakes like misaligned regions made early in the alignment process cannot be corrected later as new information from other sequences is added. Thus, there is no guarantee that the global optimal solution has been found and the alignment is not captured in a local minimum. This risk increases with the divergence of the initially aligned sequences.

Furthermore, the parameter choice a weight matrix and two gap penalties (one for opening a new gap and one for extension of an existing gap) is very important. When the sequences are closely related identities dominate an alignment, almost any weight matrix will find approximately the correct solution. With very divergent sequences the scores given to non-identical residues will become critically important, because there are more mismatches than identities. The range of gap penalty values which will find the correct or best possible solution can be very broad for highly similar sequences, but the more divergent the sequences are, the more exact values of gap penalties have to be used [71].

Clustal W

A widely used multiple alignment program is `Clustal W` [66]. `Clustal W` addresses the alignment parameter choice problem and dynamically varies the gap penalties in a position- and residue-specific manner. As the alignment proceeds, `Clustal W` chooses different weight matrices depending on the estimated divergence of the sequences to be aligned at each stage. Some matrices are appropriate for aligning very closely related sequences where most weight by far is given to identities, with only the most frequent conservative substitutions receiving high scores. Other matrices work better at higher evolutionary distances where less importance is attached to identities. Besides, sequences are weighted by `Clustal W` to correct for unequal

sampling across all evolutionary distances in the data set [70, 71]. This down-weights sequences which are very similar to other sequences in the data set and up-weights the most divergent ones. The weights are calculated directly from the branch lengths in the initial guide tree [67, 66]. In `Clustal W` the initial guide tree used to guide the multiple alignment, is calculated using the Neighbor-Joining method [58] which is quite robust against the effects of unequal evolutionary rates in different lineages and gives good estimates of individual branch lengths. These branch lengths are used to derive the sequence weights. And finally it is possible for the user to choose between fast approximate alignments [3] or full dynamic programming for the distance calculations used to make the guide tree.

The trees used to guide the final multiple alignment process are calculated from the distance matrix derived in the first step. This produces unrooted trees with branch lengths proportional to the estimated divergence. Then the root of one tree is established at a position where the means of the branch lengths on either side of the root are equal. These trees are then also used to derive a weight for each sequence.

The basic procedure of the progressive alignments is to use a series of pairwise alignments to align larger and larger groups of sequences, following the branching order in the guide tree. First the most similar sequences at the tips of the tree get aligned. Then this alignment gets aligned with the third most similar sequence and so. At each stage a full dynamic programming algorithm [49] is used with a residue weight matrix and penalties for opening and extending gaps. `Clustal W` varies gap penalties used with different weight (substitution) matrices to improve the accuracy of the sequence alignments. Further, the per cent identity of the two (groups of) sequences to be aligned is used to increase the gap opening penalty for closely related sequences and to decrease it for more divergent sequences. Also, if there are already gaps at a position, then the gap opening penalty is reduced in proportion to the number of sequences with a gap at this position and the gap extension penalty is lowered by a half.

The Ralign Algorithm

Alignments of nucleic acid sequences can bear one main problem: the sequence heterogeneity on the level of nucleic acid makes good alignments often

impossible. The resulting alignments contain too many gaps although the sequences should be very similar regarding their high degree of relationship. While protein sequences can still show substantial homology, the corresponding nucleic acid sequences are already essentially randomized. This is caused by the inherent redundancy of the genetic code: most amino acids have more than one codon on the level of nucleic acid. In a protein alignment these amino acids would match each other while the differences on the level of nucleic acids can produce gaps within coding regions in a nucleic acid alignment. Whereas, on the level of protein alignments many of this gaps could have been avoided.

Therefore, in most cases it is possible to obtain better alignments on the level of protein than on the level of nucleic acids. The scores (the per cent homologies) are higher and the number of gaps within the protein sequences is not as high as it would be in the case of nucleic acids. Reducing the gaps within an alignment improves the resulting alignment which may be used as input into other sequence data processing programs like those for secondary structure prediction.

Virus genomes contain various open reading frames within their nucleic acid sequences as they are available as data sets in various data banks (e.g. `GenBank`). The lengths of small virus genomes can vary from some 3500 bp as in hepatitis B up to about 20000 bp as in the case of Ebola. The typical genome size is about 10000 bp. The genomes can consist of single- or double-stranded DNA or RNA. Retrotranscribing viruses are the retroviruses (e.g. HIV), the hepatitis B viruses as well as caulimoviruses which have a DNA genome but use RNA as an intermediate during their replication. RNA viruses have enormously high mutation rates of up to 10^{-3} per position and replication. The number of the open reading frames depends on the type of virus considered. In addition, the organization of virus genomes is extremely variable. Overlapping open reading frames are possible, hence one part of the nucleic acid sequence codes for more than one protein in different frames. Theoretically, three open reading frames can be covered by the same nucleic acid sequence in all three possible reading frames. This possibility is actually realized in the hepatitis B virus. In addition, various non-coding regions can exist in a certain virus genome.

The idea behind the combined amino acid and nucleic acid based alignments like `Ralign` [64] is that coding regions on the level of protein vary less than

on the level of nucleic acid, because most amino acids are coded by more than one codon (base triplet) and some different nucleic acid sequences can produce the same protein sequence after translation. Thus, this approach was to improve the quality of sequence alignments of RNA viruses by creating and implementing a combined alignment algorithm.

One could argue that the quality of sequence alignments could be raised simply by translating the entire nucleic acid sequence into protein and processing on the level of proteins. But a very important factor is that the viral genomic sequence could consist of more than just one open reading frame (various coding regions in different frames) as well as some non-coding regions. These non-coding regions should, of course, be processed as nucleic acids, and every open reading frame should be processed in the correct frame.

The combined amino acid and nucleic acid based alignment procedure is made available in a program called `Ralign` developed by Roman Stocsits [64] and described in his diploma thesis. The source code of the package is written in the programming language C and will run on computers with a conforming C compiler.

`Ralign` reads `GenBank` nucleic acid sequences from sequence files in Pearson's format and `GenBank` format. Besides, it is possible for the us er to define one or more than one codon tables for each sequence or a group of sequences. Every input file can be processed using its own codon table. The standard codon table is the universal genetic codon table which fits most cases. Entering '`Ralign`' without any options or input files displays a list of the various available codon tables. These user-defined codon tables are then used by the program for translation and, of course, for finding the correct start- and stop-codons in the nucleic acid sequences. Then the program finds all possible open reading frames which have a previously defined minimal length.

`GenBank` files may contain information about the exact positions of start- and stop-codons, the genomic structure of exons and introns or the protein sequence after translation. If some information like this (e.g. regarding exons and introns) is present in the `GenBank` file, it can be obtained and used as preferred information.

The detected coding regions are translated, using the correct codon table, and the resulting proteins are compared to the protein sequences in the `GenBank` file, if available. An output file is created which contains all data about the detected open reading frames, either derived by reading the data in the

GenBank file or as a result of the automatic search done by the program. From this file the user can get information about all open reading frames, about their length, their start and stop, and the lengths of their proteins after translation. Also a second file is created: a PostScript output file which gives a graphical representation of the found open reading frames either in one of the three frames or, beyond these, as derived from the GenBank file input with all introns.

An output file is created which contains all data about the detected open reading frames, either derived by reading the data in the GenBank file or as a result of the automatic search done by the program. From this file the user can get information about all open reading frames, about their length, their start and stop, and the lengths of their proteins after translation. Also a second file is created: a PostScript output file which gives a graphical representation of the found open reading frames either in one of the three frames or, beyond these, as derived from the GenBank file input with all introns.

In many cases we can see significant differences in the genetic structure regarding the number and order of various open reading frames even between very closely related sequences. This makes it difficult to decide which ORFs correspond to each other in the various sequences.

Overlapping open reading frames are quite frequent in virus genomes. If a certain part of the sequence is coding for two or three proteins, a decision has to be made which open reading frame is used for the protein alignment. Ralign constructs a hierarchy which considers the lengths of the open reading frames. The longest coding region has highest priority and gets aligned first as a protein alignment.

The program makes a first decision, which coding regions are maintained through the alignments as protein sequences and what regions get aligned on the level of nucleic acids. The proposed assignment is presented in a file listing the open reading frames chosen for protein alignment. The user now has the possibility to alter this assumption and to tell the program exactly what coding regions are to be used for alignments on protein level. The information in both output files (text and PostScript) turned out to be quite helpful to make meaningful decisions about the choice of the open reading frames.

After the user has either manipulated or accepted the chosen open reading

frames, `Ralign` uses `Clustal W` to align the homologous sequence parts. End gaps are not penalized by the `Clustal W` algorithm. As only a piece of the genomic sequence is aligned end gaps are not desirable. Since `CLUSTAL W` is used as a 'black box' via a system call, a trick is used:

`Ralign` cuts off the end gaps such that the remaining *central alignment block* has no gaps both at the first and the last position. The sequence pieces that have been cut off are joined to the neighboring sequence parts before and after the now aligned protein parts of the sequence. In the case of overlapping coding regions these cut off parts are again handled on the level of the proteins that these regions code for. On the other hand, if the neighboring sequences are non-coding, the cut off sequence pieces are handled directly as nucleic acids.

Then the second protein alignment of the second largest open reading frames (with second priority) is started. In the case of overlapping coding regions, the central block of the first alignment (the alignment of higher priority) is still overlapping the second open reading frame, then the second protein alignment processes only this part of the second open reading frame which is not covered by the prior alignment. In order to be able to smoothly join the first and second central alignment block the generation of end gaps in the second alignment have to be suppressed. This is achieved by adding a tag to each of the sequences to be aligned. In the present implementation this tag consists of 12 copies of the string `THISISATAG`, which is quite unlikely both for a native amino acid and nucleic acid sequence. In almost all cases, therefore, `CLUSTAL W` aligns the artificial tag sequences with each other and hence provides us with well defined edges for the alignment of the real sequence. The ends of the second protein alignment part which lie adjacent to the first central alignment block are therefore forced to lie exactly one above the other.

After having aligned all protein subsequences (all chosen open reading frames), removed all end gap containing regions, and linked them to the neighboring parts of the sequences, the alignments of the non-coding regions start. Again the ends of the aligned sequence parts are forced to lie one above the other, if these ends are adjacent to formerly aligned protein parts. That way all parts can be joined smoothly together.

The protein alignments are then reverse translated. At every position where the protein alignments contain a gap of length n, a gap of length $3n$ is inserted

into the corresponding nucleic acid sequence at the corresponding site.

Finally, all alignments, either on the level of proteins or nucleic acids, get combined and a resulting alignment output file is created which contains the complete nucleic acid sequence alignment.

In some rare cases `CLUSTAL W` will not properly align the tag regions added to suppress end gaps. Gaps inserted into the tags can lead to imperfect removal of the tags and thereby corruption of the sequences. In a last step the final alignment is checked for such errors. Currently, the only recourse is to remove the offending sequence from the alignment.

SplitsTree, Split Decomposition

Evolutionary data is most often presented as a phylogenetic tree, the underlying assumption being that evolution is a branching process. However, real data is never ideal and thus doesn't always support a unique tree, but often supports more than one possible tree. Hence, it makes sense to consider tree reconstruction methods that produce a tree, if the given data heavily favors one tree over all others, but otherwise produces a more general graph that indicates different possible phylogenies. One such method is the `Split Decomposition` introduced by Hans-Juergen Bandelt and Andreas Dress (1992) and its variations.

To show aligned sequences as a tree we used the program `SplitsTree2.4.1`, it is a program for analyzing and visualizing evolutionary data. Input is input a file containing sequences, distances, or a system of splits and produces as output a weakly compatible system of splits and a splits graph representing the given data. It contains a number of transformations to obtain distances from sequences and methods for obtaining compatible or weakly compatible split systems from distances or sequence [34].

2.5 Conserved Structure Detection

The method for detecting conserved secondary structures [23, 29] aims at utilizing the information contained in a multiple alignment of a small set of related sequences to extract conserved features from the pool of plausible structures generated by thermodynamic prediction for each sequence. A

flow chart is shown in figure 9. Our approach is different from efforts to simultaneously compute alignment and secondary structures [59, 65, 6].

One disadvantage of these methods is the much higher computational cost which makes them unsuitable for long sequences such as viral genomes, if no parallel computers are available. Furthermore they assume implicitly that all sequences have a common structure, not just a few conserved structural features. The same is true for the related program `Construct` [43].

The basic two inputs for the algorithm are a multiple sequence alignment and the base pair probabilities from McCaskill's algorithm. We calculate the multiple sequence alignment using `CLUSTAL W` [68]. No attempt is made to improve the alignment based on predicted secondary structures. While this might increase the number of predicted structural elements, it would also compromise the use of the sequence data for verifying these structures. Furthermore we find that most regions that have functional secondary structure tend to align fairly well, at least locally.

While the related `Alidot` method [23] uses only minimum energy structures, i.e., one structure per sequence, `Pfrali` [28] uses base pairing probabilities as obtained from McCaskill's partition function algorithm. Since the base pairing probabilities contain information about a large number of plausible structures, this approach is less likely to miss parts of the correct structures. In both cases, we make explicit use of the sequence variation to select the credible parts of the predicted structures. Thus, we do not assume *a priori* that there is a conserved secondary structure for all (or even most) parts of the sequence.

Base pair probability matrices are conveniently displayed as "dot plots". The `Vienna RNA Package` [24] contains an efficient implementation of McCaskill's algorithm that produces *dot-plots* in `PostScript` format, see figure 4.

The `Pfrali` program reads the pair probabilities from these files as well as a multiple sequence alignment in `CLUSTAL W` format. The gaps in the alignment are inserted into the corresponding probability matrices. We can now superimpose the probability matrices of the individual sequences to produce a *combined dot plot*. To keep the number of base pairs manageable we keep only pairs that occur with a probability of at least $p^* = 10^{-3}$ for at least one sequence. Base pairs with even lower probabilities are very unlikely to be part of an important structure. In the combined *dot-plot* the area of a

dot at position i, j is proportional to the mean probability $\bar{p}_{i.j}$ (averaged over all sequences). In addition we use a color coding to represent the sequence information.

A sequence is *compatible* with base pair $(i.j)$ if the two nucleotides at positions i and j of the multiple alignment can form either a Watson-Crick (**GC**, **CG**, **AU**, or **UA**) pair or a wobble (**GU**, **UG**) pair. When different pairing combinations are found for a particular base pair $(i.j)$ we speak of *consistent* mutations. If we find combinations such as **GC** and **CG** or **GU** and **UA**, where both positions are mutated at once we have *compensatory* mutations. The occurrence of consistent and, in particular, compensatory mutations strongly supports a predicted base pair, at least in the absence of non-consistent mutations.

Phylogenetic methods in general consider only compensatory mutations even though **GU** base pairs are clearly important as evidenced by the fact that **RY**→**YR** conversions are rare [20]. While compensatory mutations of the type **RY**→**RY**, such as **GC**→**AU**, can be obtained by two subsequent consistent point mutations, for instance **GC**→**GU**→**AU**, a double mutation is required for **RY**→**YR** mutations. We argue therefore that all consistent mutations, not only compensatory ones, should be seen as support for a proposed structure.

The sequence variation, the number of non-compatible sequences, and the number $c_{i.j}$ of different pairing combinations is incorporated in the combined *dot-plot* as color information. For the details of the encoding scheme see the caption to the color plate in figure 8.

The base pairs contained in the combined *dot-plot* will in general not be a valid secondary structure, i.e., they will violate one or both of the following two conditions: (i) No nucleotide takes part in more than one base pair. (ii) Base pairs never cross, that is, there may not be two base pairs $(i.j)$ and $(k.l)$ such that $i < k < j < l$. In the reminder of this section we describe how to extract credible secondary structures from the list of base pairs.

In essence, we rank the individual base pairs by their "credibility", using the following criteria:

(1) The more sequences are non-compatible with $(i.j)$, the less credible is the base pair.

(2) If the number of non-compatible sequences is the same, then the pairs are ranked by the product $\bar{p}_{i.j} \times c_{i.j}$ of the mean probability and the number of different pairing combinations.

Then we go through the sorted list and remove all base pairs that conflict with a higher ranked pair by violating conditions (i) or (ii).

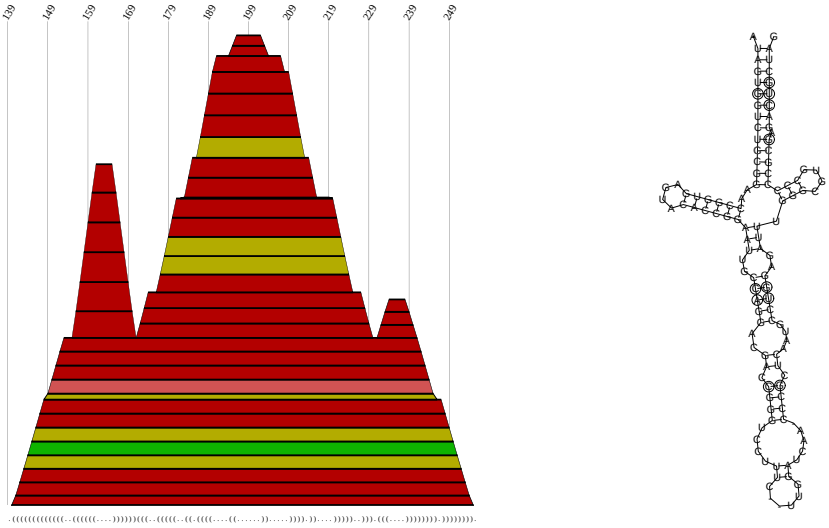


Figure 8: *Hepatitis C virus* IRES, an example of a color *dot-plot*, left picture and the two-dimensional graph of its secondary structure right side. Colors indicate the number of consistent mutations ■ 1, ■ 2, ■ 3 different types of base pairs. Saturated colors, ■, indicate that there are only compatible sequences. Decreasing saturation of the colors indicates an increasing number of non-compatible sequences: ■ 1, ■ 2 sequences that cannot form a base pair (i, j). If there are more than 2 non-compatible sequences the entry is not displayed. In the two-dimensional graph of the secondary structure consistent base pairs are symbolized by a single circle around one base pairing part, compensatory mutations by two circles around both pairing partners.

The list now represents a valid secondary structure, albeit still containing ill-supported base pairs. Since our goal is to produce a list of well-supported secondary structure features that contains as few false positive as possible, we use a series of additional "filtering" steps: First, we remove all pairs with more than two non-compatible sequences, as well as pairs with two

non-compatible sequences adjacent to a pair that also has non-compatible sequences. Helices with so many non-compatible sequences can hardly be called "conserved". (For large samples these rules might have to be modified to tolerate somewhat larger numbers of non-compatible sequences.) Next, we omit all isolated base pairs. The remaining pairs are collected into helices and in the final filtering step only helices are retained that satisfy the following conditions: (i) the highest ranking base pair must not have non-compatible sequences. (ii) for the highest ranking base pair the product $\bar{p}_{i.j} \times c_{i.j}$ must be greater than 0.3. (iii) if the helix has length 2, it must not have more non-compatible sequences than consistent mutations. In general, these filtering steps only remove insignificant structural motifs that one would have disregarded upon visual inspection anyways. The remaining list of base pairs is the conserved structure predicted by the `Pfrali` program.

The final output of the program consists of a color coded *dot-plot* in `PostScript` format, as well as a text output containing the sorted list of all base pairs and the final structure. Additional tools are provided to produce annotated secondary structure plots from these data.

Manual reconstruction of a consensus structure proved to be a time-consuming and error-prone task. In contrast, the structure in figure 8 was produced without human intervention.

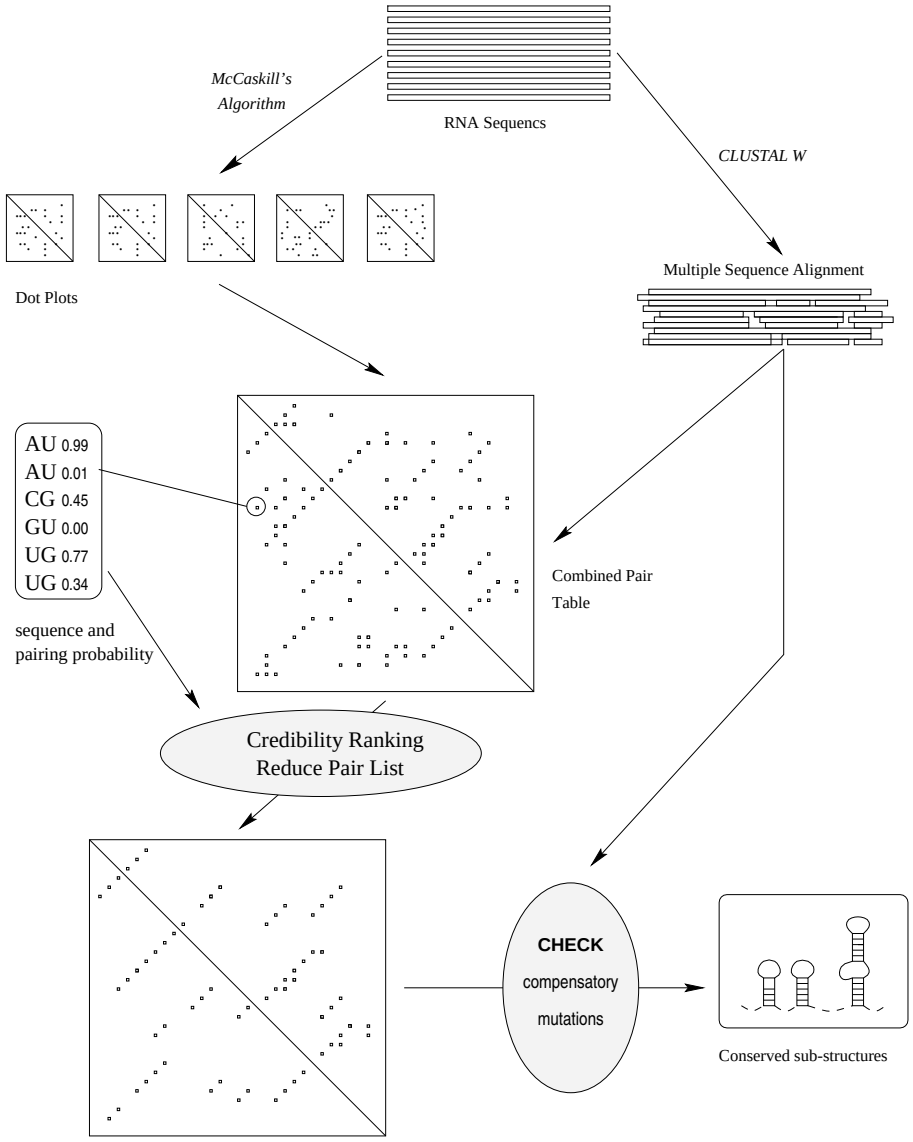


Figure 9: Flow diagram of the algorithm. A multiple sequence alignment is calculated using `CLUSTAL W`. RNA genomes are folded using McCaskill's partition function algorithm as implemented in the `Vienna RNA Package`. The sequence alignment is then used to align the predicted structures. From this structural alignment we extract putative conserved regions. In the final step the sequence information, in particular compensatory mutations, are used for validating or rejecting predicted structure elements.

2.6 Aligned Minimum Energy Folding

Secondary structure prediction is based on folding only a single sequence and common structures of related sequences are detected afterwards. Common RNA structures on a sample of sequences are only presented in a two step process by folding algorithms and the algorithms `Alidot` and `Pfrali`. However it would be nice to compute common structures of a set of sequences in a one step process and the algorithm presented in this section is an attempt in this respect.

The aim is to develop a method that use thermodynamic structure prediction on a sample of aligned sequences, or to combine the algorithm `Alidot` [23] and minimum free energy calculation [24]. The result of this attempt is a new folding algorithm called `Alifold`. Its main idea is to assign to each structural element an mean energy, averaged over all sequences in the alignment. Otherwise it is similar to the usual minimum free energy calculation using the same thermodynamic energy set. Major differences are, that we use a sample of aligned sequences as input, and we get no energy evaluation of the secondary structure. The result of the computation is a common ground state structure over a sample of aligned sequences.

We can use even small data sets, of about 10 sequences, or huge data sets of about 100 sequences to search for consistent RNA secondary structures. For a large number of aligned sequences the algorithm `Alidot` faces the problem of two many sequences not pairing a given base pair $i \cdot j$. The number of unpaired sequences is fixed to three, only three sequences of all may not base pair a given base pair, otherwise the base pair is forbidden and not predicted. The total amount of sequences can influence the result, that is really a problem for a large sample of sequences. In opposite to `AliDot` we use in `Alifold` a user-defined fraction of all sequences to decide whether a base pair can be formed or not. That is a practicable solution to reduce the influence of the total amount of used sequences on predicting base pairs at all.

In our respect conserved RNA elements contain consistent or compensatory mutations, that favors a special structure over the sample of sequences. In our algorithm the same set of energy parameters are used as implemented in the `Vienna RNA Package` and energy contributions of different types of loops are identical, see figure 3 and table 4. A pseudo *minimum free energy*

(`Epseudo`) is calculated over a set of aligned sequences. This pseudo energy consists of the average energy over the aligned sequences plus bonus energies for consistent and compensatory mutations. Adding finally sequences not pairing a given base pair $i \cdot j$ get a penalty energy proportional to the number of sequences not pairing. The bonus and penalty energy contributions are summed up for each $i \cdot j$.

$$E_{Mut}[i,j] = 1.5 * \text{Con}[i,j] + 2 * \text{Comp}[i,j] + 1 * \text{Unp}[i,j] \tag{14}$$

The bonus energy for consistent (Con[i,j]) or compensatory (Comp[i,j]) mutations is practicable to set to -0.05 kcal/mol and the penalty energy (Unp[i,j]) to 0.05 kcal/mol for unpaired sequences, and the fraction of sequences having to pair a base pair $i \cdot j$ is set to 80%.

This settings allows us to find mutations quite well without changing the predicted secondary structure in its entirety. Bonus or penalty energies are only added with hairpin-loops and interior-loops, multi-loops are not yet considered. This may cause that multi-loop could not be predicted as well. One should keep in mind, that good multi-loop energy parameters are not available at present, and all dynamic folding algorithms use a simple estimate to contribute multi-loops, so the prediction of multi-loops is still a problem of thermodynamic folding algorithms. Fortunately most conserved RNA elements can be found in hairpins or interior loops.

Another difference was introduced by the energy contribution of mismatches in stacks. Forming stacking regions mismatches inside a single sequence causes by default a very high penalty energy and would decrease the number of base pairs. That is a crucial problem, because a single mismatch in one sequence can prevent that base pair, although all other sequences can pair. A good solution to this problem is to reduce the penalty energy of single mismatch to 0.1 kcal/mol.

A main problem of the algorithm `Alifold` and predicting conserved structures generally, is a bad sequence alignment. Too many gaps, gaps are contributed to mismatching base pairs, can significantly change the predicted secondary structure, because other base pair contacts can be preferred by the energy model. In the worst case a completely different ground state structure can be predicted, because the aligned sequences are different to the starting sequences and a small amount of mutations (gaps) can change the structure in its entirety.

Table 4: **Pseudocode for the Algorithm `Alifold`.**

```
for(d=1...n)
   for(i=1...d)
      j=i+d
       IsPaired(i,j)
       if(IsPaired(i,j)) else base pair forbidden
         for( s=1...Number_of_Sequences)
             C[i,j] += HairpinEnergy
             C[i,j] += Mutation_energy
             for(p,j... i<p<q<j) {
                  for( s=1...Number_of_Sequences)
                     ali_energy += LoopEnergy
                 ali_energy += Mutation_energy
                 C[i,j] = MIN2(ali_energy,ali_new_c)
         ali_MLenergy = Multiloop_energy
         C[i,j] = MIN( ali_MLenergy,C[i,j])
for(j=5...n)
   f5[j]=MIN2(f5[j-1], C[1,j]+Dangling_energy);
 Epseudo[n]=f5[n]/100;
```

Remark. C[i,j] is the energy given that i and j pair. Function `IsPaired(i,j)` checks whether a given base pair $i \cdot j$ is allowed over all aligned sequences. The fraction of sequences having to pair can be set individually. `Mutation_energy` is either a bonus energy for consistent or compensatory mutation, or a penalty energy proportional to the number of sequences not pairing $i \cdot j$. Multi loop energies are summed up over all sequences, but no bonus energy for consistent and compensatory mutations is given. The array `f5[j]` contribute to subsegment energies. The base pairs are calculated by a backtracking procedure, after the pseudo minimum energy calculation.

Table 5: Folding times of `Alifold` and `Alidot` to predicted conserved RNA structures, performed on Dual Pentium III 450 MHz, 1024MByte. The symbol f denote to the fraction of sequences having to pair, otherwise that base pair is not predicted, default value $f = 80\%$. Testing with identical sequences of sequence length 1000 shows that `Alifold` is about 20% slower than the `Vienna RNAfold 1.3`. Using different sequences e.g. HCV virus genomes, the whole calculation of `Alifold` took about 20% of `Alidot`.

Remark	*Number*	*length*	t (min) `Alifold`	t (min) `Alidot`
identical seq.	10	1000	3.43	2.82
HCV virus	10	9757	285.8	1396.4
HCV virus $f = 70\%$	10	9757	364.9	1396.4
HIV1 virus	51	10678	1514.8	≈ 7038

That happens also if normal minimum free energy folding is used. In this case we have to use a different sequence alignment to predict a correct set of conserved elements. The output is a list of predicted base pairs, additional information on different base pairs for a given $i \cdot j$ is printed to file, the base pair type and the number of sequences not pairing $i \cdot j$.

Performance

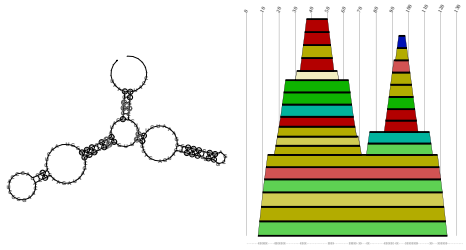
The demand of computational resources folding RNA secondary structures is sequence dependent and for large sequences as complete virus genomes quite demanding both in terms of memory and CPU time, see section 2.3. Although linux parallel clusters are nowadays easier available, the use of single processor computers are still preferred by most scientists. Therefore an algorithm speeding up conserved secondary structure predictions is still desirable.

Secondary structure prediction is the most time and computational resource consuming step in conserved structure prediction. Two different algorithms are used McCaskill's partition function or minimum free energy folding. A more detailed prediction is done by McCaskill's, but the need of computational memory is often to large to be performed on available computers.

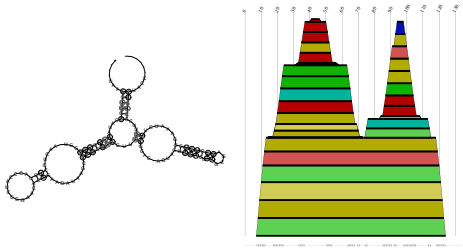
Often conserved RNA secondary structures are well presented in the ensemble, so we can predict them using minimum free energy calculation. Although minimum free energy calculation is fast and less resource consuming than the partition function calculation, large samples of long sequences took a while, a faster alternative is `Alifold`.

The calculation time of `Alifold` depends on the diversity of the used sequences and the fraction f, see table 5. Computation time increases with sequence identity and decreases with the fraction of possible base pairs over all sequences for a given $i \cdot j$.

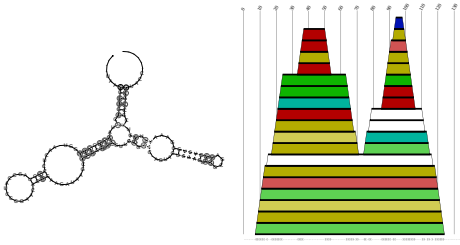
Examples



10.1 AliDot



10.2 Pfrali



10.3 Alifold

Figure 10: Comparison of three differently predicted consensus structures. A set of 21 *Halobacteriales* 5sRNA sequences are taken. In the `Alifold` output 3 additional base pairs could be predicted. Two or more sequences are not consistent with these base pairs, therefore they are not predicted in `Alidot` and `Pfrali`, see figure `Alifold` white colored.

Table 6: List of predicted conserved elements, performed by all three algorithms. The mean pairwise homology of the aligned sequences is 70.7%. Commonly predicted elements are compared. Number of conserved bases (cons.) and the mean pairwise homology (hom.) are listed. Note, length of predicted elements are different, see figures 11,13,14.

Position	Alifold		Alidot		Pfrali	
in Pfrali	cons.	hom.(%)	cons.	hom.(%)	cons.	hom.(%)
141-255	93	94.1	91	94.0	91	94.0
390-429	28	89.8	29	89.8	28	89.8
609-654	52	83.9	52	83.9	28	84.3
680-739	59	90.5	59	90.5	45	89.1
788-818	18	75.6	18	75.6	14	72.4
8046-8095	28	83.8	28	82.5	28	83.8
8715-8749	20	85.2	22	83.5	22	83.5
9143-9216	36	77.5	36	77.5	36	77.5
9330-9373	9	72.2	19	65.8	18	67.9
9377-9424	30	85.8	30	82.4	30	84.5
9433-9467	23	88.7	23	87.3	23	87.3

Found RNA structure motifs are compared to the output of `Pfrali` and `Alidot`. As a first example `Alifold` was tested on a set of 5sRNA of *Halobacteriales*. The output was compared to `Alidot` and `Pfrali` using the `Vienna RNA Package` for secondary structure prediction. Alignment was produced by `Clustal W`.

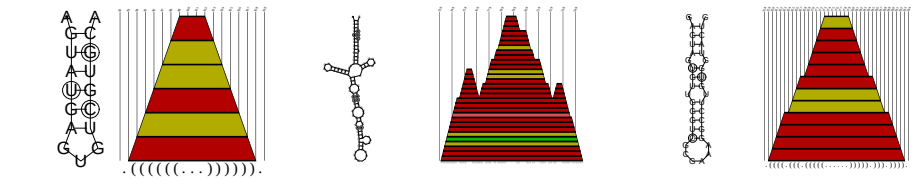
A second example was performed by folding complete *Hepatitis C virus* (HCV) genomes, they are aligned by `Clustal W` and secondary structure motifs are predicted using `Pfrali, Alidot` and `Alifold` for comparison.

The length of the aligned sequences is 9784 and the mean pairwise homology is 70.7%. This sample of *Hepatitis C Virus* sequences is different to the selected virus genomes in section 3.2.1. A more diverse set is used to test the algorithm `Alifold`. The result of the test is that the algorithm `Alifold` predicted all RNA motifs also found by `Pfrali`, four additional motifs can be found not presented by `Pfrali` and `Alidot` predicted one additional structure motif.

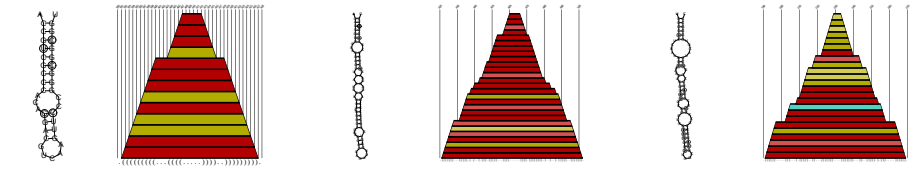
Discussion

The algorithm `Alifold` is a practicable alternative to `Alidot` or `Pfrali` providing us with the same output of conserved secondary structure motifs, see figure 11, 13, 14. The predicted conserved RNA elements are not discussed in detail, because they are only listed to show the quality of this approach. All secondary structure motifs predicted by `Pfrali, Alidot` are also predicted by `Alifold`. Some additional structure motifs are found using `Alifold`, that is a result of using aligned sequences with gaps. Introducing gaps to the sequence can prefer different ground state structures. The selection of conserved RNA structures is still a problem, at present it leave it up to the viewer which elements to believe. An automated and fixed search criteria for the selection is therefore desirable.

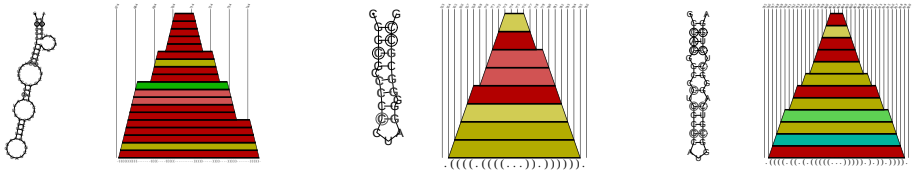
The algorithm `Alifold` allows a lot of parameter setting i.e. to set the fraction of pairing sequences, or the values for mutation bonus or penalty energy and is therefor more flexible than `Alidot`.
The performance of this algorithm is quite good, we observed a speed up of the prediction of about 5 times to normal minimum free energy folding. Folding large sequence numbers is also no problem and quite fast, see table 5. The improved computational speed and the feasible small amount of used memory for detecting conserved structure motifs put this tool into the position to screen easily large numbers of long RNA sequences. The limitation to this algorithm is computational memory than folding times. The prediction of conserved RNA motifs of large RNA viruses up to 16000 nt can be performed on present computers with computational memory up to 1GByte.



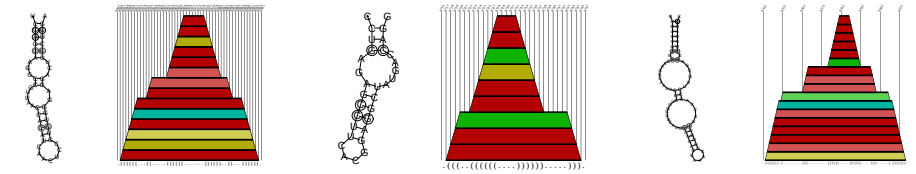
11.1 93-109 11.2 140-256 * 11.3 259-294



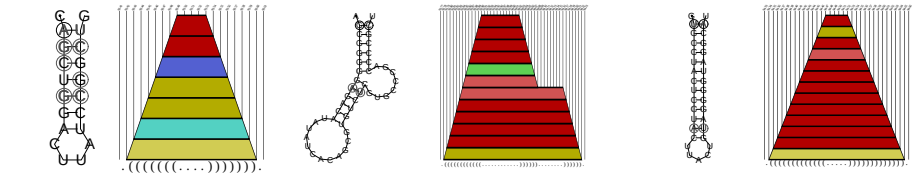
11.4 391-428 * 11.5 430-512 11.6 591-670 *



11.7 675-751 * 11.8 764-786 11.9 786-820 *



11.10 8046-8095 * 11.11 8717-8747 * 11.12 9143-9216 *



11.13 9342-9361 * 11.14 9378-9423 * 11.15 9434-9466 *

Figure 11: Detected conserved secondary structures of HCV. Predicted by `Alifold`, using default parameter settings ($f = 80\%$). Alignment length 9757. Structures labeled by (*) are also predicted by `Pfrali`.

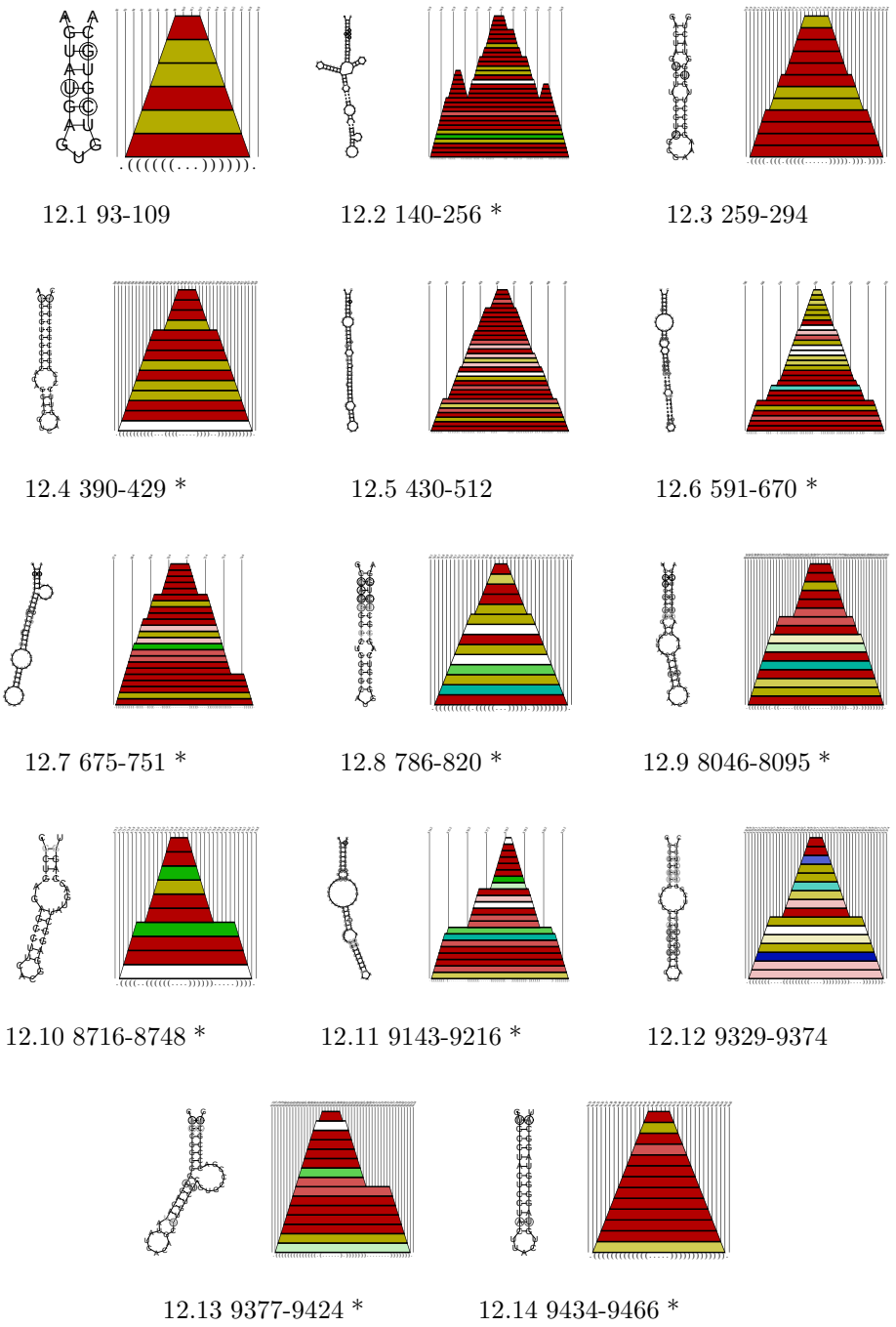


Figure 12: Detected conserved secondary structures of HCV. Predicted by `Alifold`, using a different fraction of sequences having to pair ($f = 70\%$). Alignment length 9757. Structures labeled by (*) are also predicted by `Pfrali`.

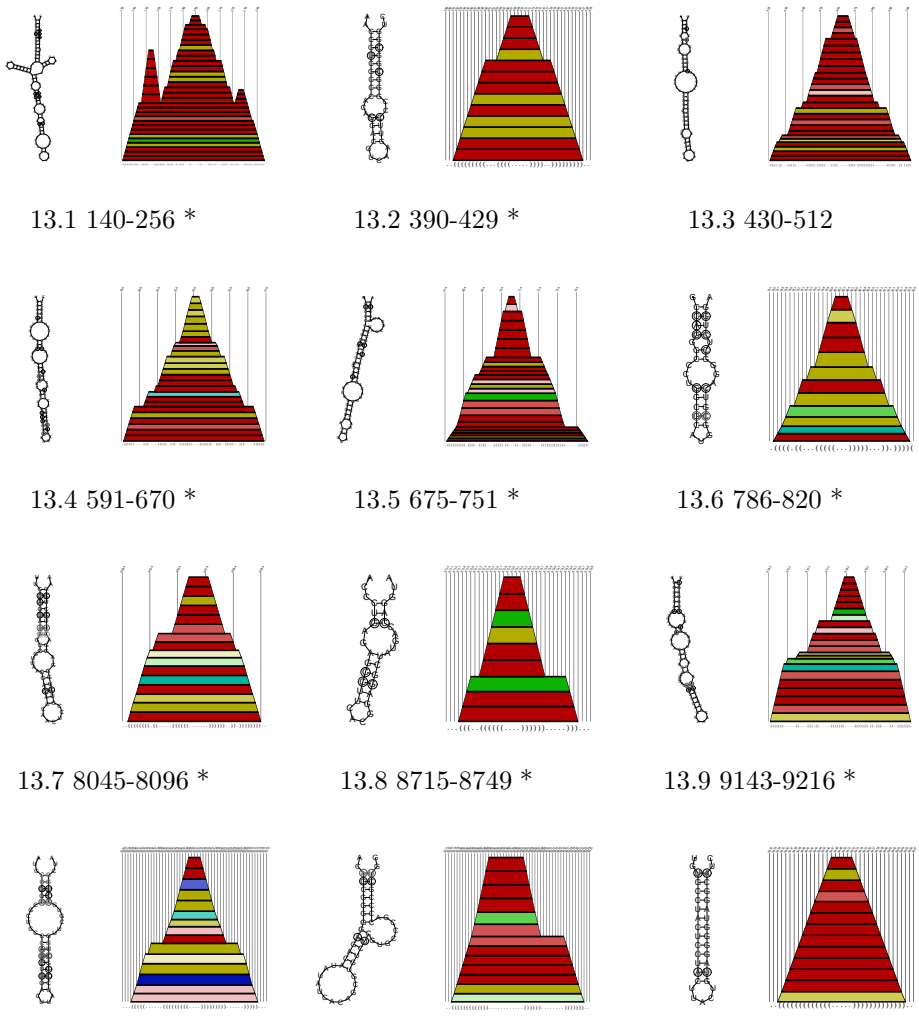


Figure 13: Detected conserved secondary structures of HCV. Predicted by `Alidot`. Alignment length 9757. Structures labeled by (*) are also predicted by `Pfrali`.

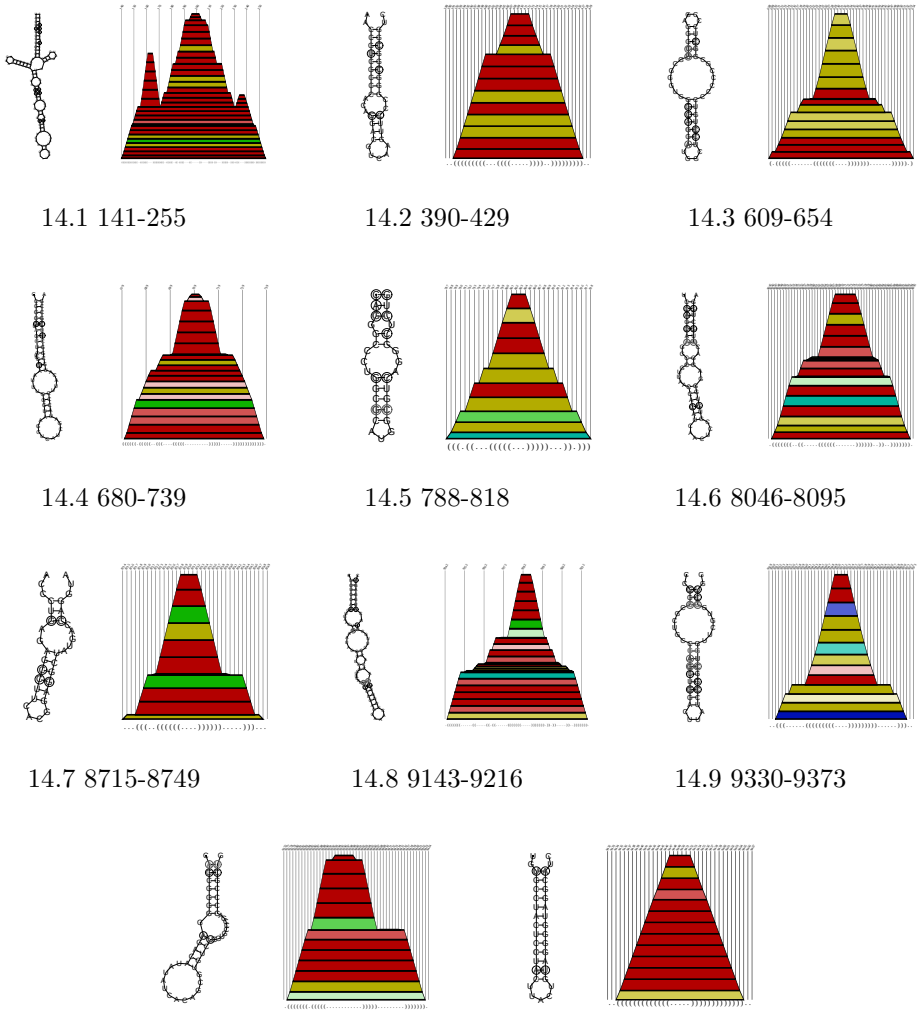


Figure 14: Detected conserved secondary structures of selected HCV. Predicted by `Pfrali`. Alignment length 9757.

2.7 Vienna RNA Viewer

Searching long RNA virus genomes for conserved secondary structure motives by hand is a quite laboriously work. A graphical viewing tool with options for selection of probably conserved regions and a semi automatically generation of detected structure motives is a demand.

The `Vienna RNA Viewer`, a RNA secondary structure viewing tool was developed by Martin Fekete and Ivo Hofacker in `Perl` and `PerlTk` at the *Institute for Theoretical Chemistry and Molecular Structural Biology*. This viewing tool is designed to accept the output formats produced by the `Vienna RNA Package` and the algorithms `Alidot Pfrali`. The program detects automatically the input file type, whether normal RNA dot_plot files, or the special output file format of `Alidot` and `Pfrali`. Although several Viewing tools are known for RNA secondary structures, e.g. `RNAviz`[2], `XRNA`[3] our search for conserved RNA secondary structure patterns made this new viewing tool unavoidable.

RNAfold, Secondary Structure Output

The `Vienna RNA Package` produce a so called dot_plot file format, with the information of the secondary structure of the folded RNA sequence, either only minimum free energy or base pair probability, see figure 15. The main window of the `Vienna RNA Viewer` shows a typical dot_plot file. The lower triangle contains the minimum free energy, and the upper one the base pairing matrix of phenylalanin tRNA sequence. Squares denote to base pairs and its size to the probability in the ensemble of structures. Red colored squares are *minimum free energy (mfe)* base pairs, blue one are base pairs in the ensemble of all structures. Note, the minimum free energy is the ground state structure, but not necessarily the most probable structure in ensemble. Several additional information can be obtained by *left-mouse* click on the colored squares, the base pair position, the pairing nucleotides and the probability is displayed, *minimum free energy* base pairs are labeled "mfe". One can zoom in and out the dot_plot by pressing $(+, -)$. Several functional buttons are available, at top a `GO, Save Screen-button, Redraw, Help` and `Quit`. The current courser position is shown top left side. The `GO`

[2] `http://www-rrna.uia.ac.be/rnaviz`
[3] `ftp://fangio.ucsc.edu/pub/XRNA`

button centers the base pair position inserted right, `Save Screen` prints a `PostScript` screen shot of the main window, `Redraw` deletes all labels and redraws the main window, `Help` provides help on buttons and you can quit the program by clicking the `Quit` button.

The buttons situated at the bottom of the main window provide special functions. Left most button `Basepair List` creates a new window with a list of all drawn base pairs. A *left-mouse* click on a list item centers the selected base pair in the main window and draws a circle around the square. The `Mountain Plot` is disabled for normal dot_plot input files.

The button `Stack List` allows a search for stacking regions inside the dot_plot. You can set the minimal stack size and the minimum probability of stacking base pairs, all stacks matching the search criteria are listed in the stack list window. The sorted list of found stacking regions can be visited by a *left-mouse* click, this centers the stack in the main window and mark the button red for already visited. Labeling with $(+, \sim, -)$ is useful for searching for conserved stacks and the entry field allows to type in a remark for the stack. The selected stack-list can be saved, by clicking button `Save List` in the `Stack List` window, `Draw Stacks` draws only all selected stacks in the main window, and `New Stacklist` allows to create a new list of stacking base pairs.

`Secondary Structure` is a tool to write RNA secondary structure to `PostScript` output files, or `XRNA` compatible structure files. One can select a region to draw by a *left-mouse* click for the start position and a *right-mouse* click for the end position, or by clicking on stacks in the stack list window. A `PostScript` and a structure file of the selected region is drawn to file. For phenylalanin tRNA the `PostScript` output is shown, see figure 16.

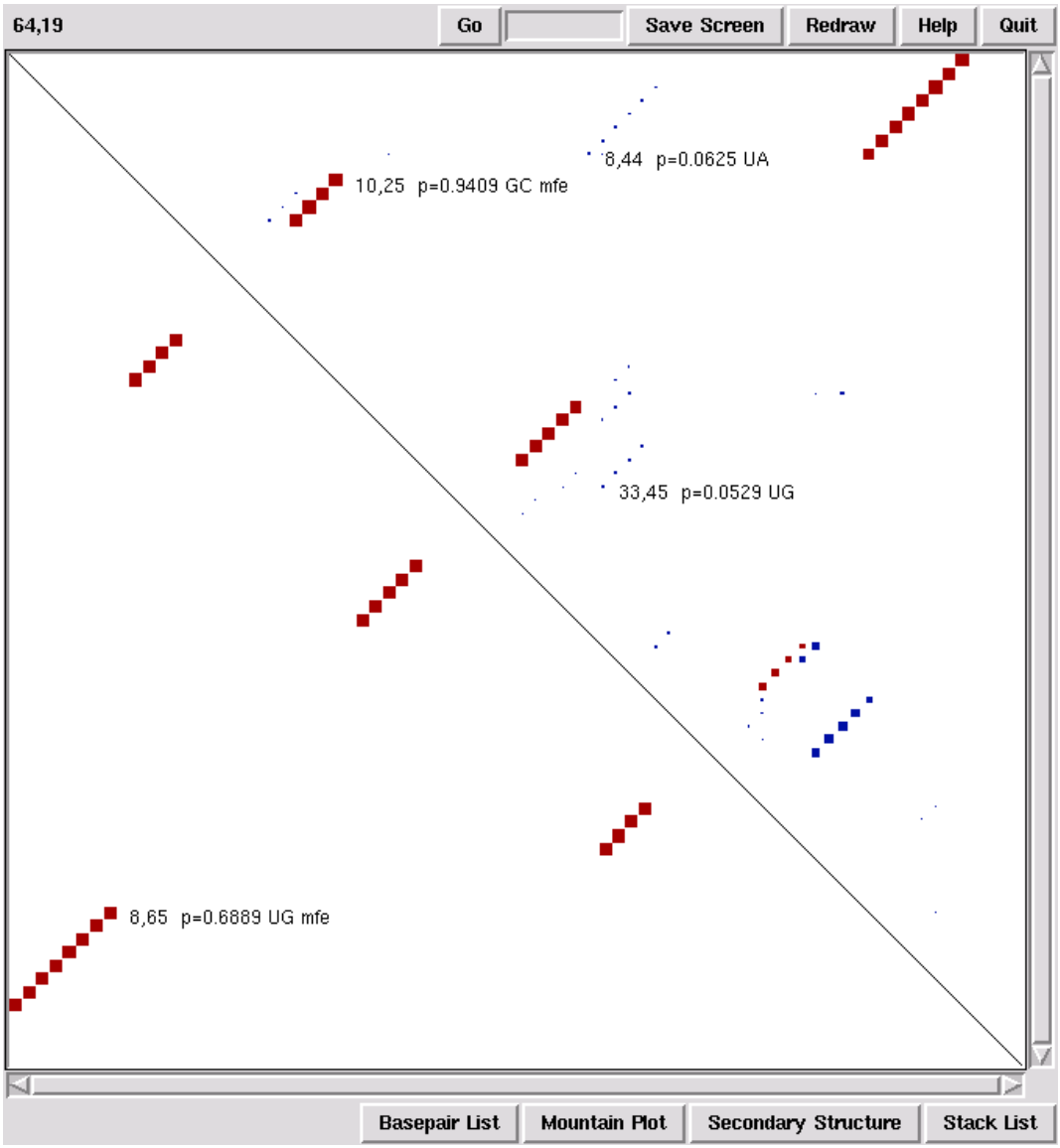
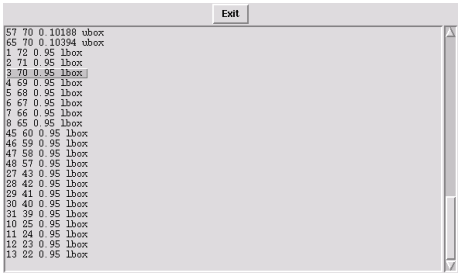
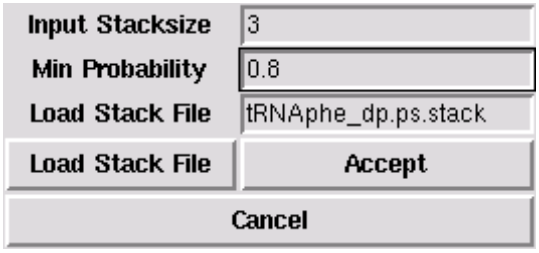


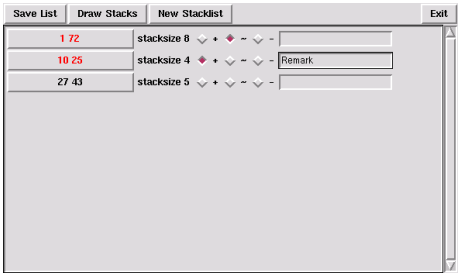
Figure 15: Snap shot of main window of the `Vienna RNA Viewer` displays the secondary structure of phenyalanin tRNA, colored squares denote base pairs. Lower left triangular matrix shows the minimum free energy, red colored, and the upper right triangular matrix show the base probabilities of the ensemble, blue colored. Buttons are explained in text.



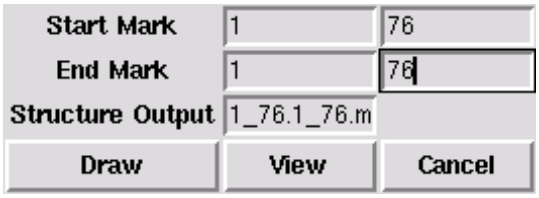
16.1 Base pair list



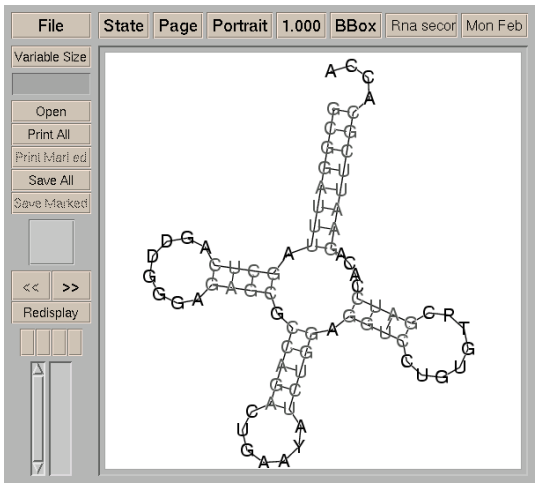
16.2 Select stacking base pairs



16.3 Selected stacks



16.4 Draw secondary structure



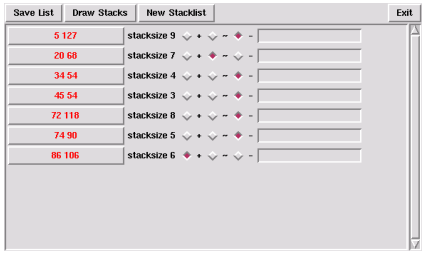
16.5 Secondary structure graph of tRNA

Figure 16: Snap shot of functional windows of the `Vienna RNA Viewer`. The `Base pair list` shows all base pair contacts, minimum free energy and base pair probability, see figure 16.1. A search for stacking regions can be selected the window shown in figure 16.2. Matching stacks are listed in the `Stack list`-window see figure 16.3. The window shown in figure 16.4 allows to select a region to print the secondary structure to file, e.g. the secondary structure graph of phenylalanin tRNA is shown in figure 16.5.

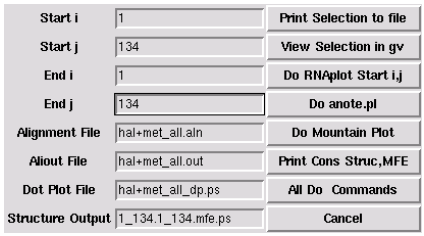
Secondary Structure Output of Pfrali and Alidot

The algorithm `Pfrali Alidot` use a new output format for secondary structures. For searching conserved RNA secondary structures one can use five main functions implemented to the `Vienna RNA Viewer`, see also figure 16.

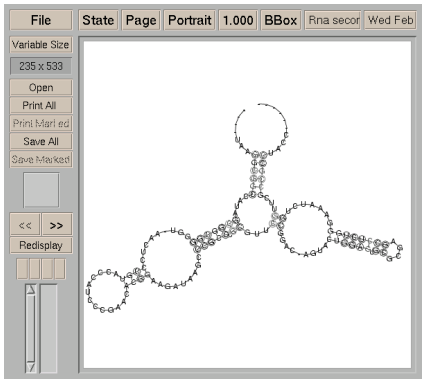
Figure 17: Display of a `Pfrali` output of 21 aligned sequences of 5sRNA of Halobacteriales. The upper left triangle displays the base pair probabilities. Consistent and compensatory base pairs are differently colored. The lower left triangle shows the minimum free energy structure. Additional information on base pairs are obtained by a *left-mouse* click on squares.



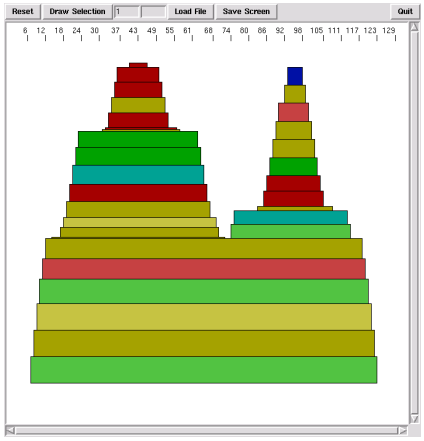
18.1 Selected stacks



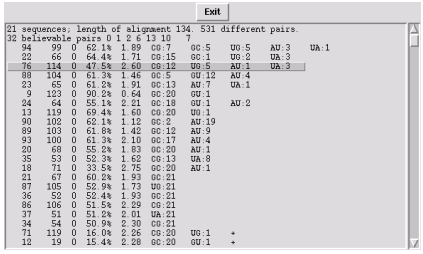
18.2 Draw secondary structure



18.3 Secondary structure



18.4 Mountain Plot



18.5 Base pair list

Figure 18: Snap shot of functional windows. Figure 18.1 displays a sorted list of stacks. Figure 18.2 shows the window to a draw secondary structure to file, the selection of a region follows figure 16, additional the alignment file is needed. The selected secondary structure is shown in figure 18.3. Consistent and compensatory mutations are denoted by circles around the bases. The button `Mountain Plot` draws a colored Hodgewed mountain plot of the `Pfrali` output. One can zoom in the mountain plot by selecting a region by *left-mouse* and *right-mouse* click. Figure 18.4 lists all base pairs by their credibility, a *left-mouse* click centers the base pair in the main window of the `Vienna RNA Viewer`

The button `Mountain Plot` is activated using `Pfrali` and `Alidot` output files and draws a colored Hodgewed mountain plot, see figure 18. One can zoom into the mountain plot by selecting a region by *left-mouse* and *right-mouse* click. `Draw Selection` draws the mountain plot of the selected region. `Reset` draws the entire mountain plot again and `Save Screen` prints out the contents of the window to a `PostScript` file.

Discussion

The `Vienna RNA Viewer` was first designed to view only the dot_plot files produced by either `Pfrali` or `Alidot`. This was the first attempt to visualize the information produced by these algorithms. The analysis of complete virus genomes with several thousand nucleotides, such as *Hepatitis C* virus or *Pestivirus* could be hardly done in reasonable time without an investigation tool, which help to filter useful information.

Conserved RNA secondary structures are always presented in stacking base pairs and a sorted list of stacks is a useful tool to screen through a complete virus genome for possibly conserved motives. A first overview of a large virus genome is provided by the mountain plot, which allows a qualitative analysis, whether conserved motives can be found or not. The result of investigating these files is creating a list of structure files, either mountain plots or structure graphs of possibly conserved RNA secondary structures. This `Vienna RNA Viewer` is a fast and easy tool to screen through even large data files, produced by `Pfrali` or `Alidot`. A first test was done to produce the data files in 3.13.2.

This viewer allows to screen complete virus genomes in rather short time, so a lot of different virus families can be studied. An increase of known conserved RNA secondary structures can also lead to a better understanding of the viral life-cycle of RNA viruses. A first attempt to classify RNA virus species on basis of their conserved structural motives can be tempted and can possibly improve the understanding of virus evolution over time.

3 Results

The procedure was first tested on two different virus families, to give an example that there is no restriction to special RNA viruse genera. The virus family *Bunyaviriade* are anti-sense single strand RNA viruses with a tripartite genome. *Flaviviridae* are sense RNA viruses and beyond it the genera *Hepatitis C virus* and *Pestivirus* have a completely different coding strategy to *Bunyaviridae*.

Numerous sequences were available for *Hepatitis C virus* and *Pestivirus* and *Hantavirus*, but for our search sequences are preselected to improve the quality of the sequence alignment, also the number of used sequences has to be restricted, because to many sequences would have decreased the number of base pairs. Remember in the algorithm `Pfrali` and `Alidot` there is a fixed number of three sequences for not pairing a given base pair otherwise the base pair is forbidden. For the color code of predicted base pairs see figure 8.

The set of selected virus genomes should also represent all available sequences, and the lengths of the genomes were kept in a certain range, to improve the multiple sequence alignment. In spite of this restriction the virus genomes show sequence homologies from approximately 70-90%. Note about 10% sequence diversity is enough to destroy consistent RNA secondary structures, if mutations occur randomly. Detected RNA structures are in this respect conserved.

Extensive testing of our parallel folding algorithms could be performed by folding the entire *Pestivirus* genomes, that scales up to a total sequence length of about 13000 nucleotides. For the first time entire secondary structure data are available for such large virus genomes, that is an improvement to folding only sequence segments, long range interactions are not neglected anymore. Previous investigations focused mainly on the non coding regions. Only sequence of rather short segments of the genome were analyzed. This work extends the search to the entire genome.

3.1 Structure Motifs, Bunyaviridae

Introduction

The virus family *Bunyaviridae* consists of five genera. *Bunyavirus, Phlebovirus, Nairovirus, Hantavirus* and *Tospovirus*. Virions are spherical or pleomorphic, 80-120 nm in diameter, and have a lipid-containing envelope. The genome is tripartite and terminal nucleotides of each viral RNA species are base-paired forming non covalently closed, circular RNAs. Ribonucleocapsids , negative- or ambisense, are single-stranded RNAs, 11-21 kb in overall size. Terminal sequences of gene segments are conserved among different viruses in each genus but are different among genera. The L-segment encodes the viral transcriptase-replicase, the M-segment the envelope glycoproteins, and the S-segment the nucleocapsid protein. Phlebovirus and Tospovirus have an ambisense S-segment; they encode non structural proteins (NSS) in the 5'-half of virion S-segment. The viruses have four structural proteins, two external glycoproteins (G1 and G2), a nucleocapsid protein (N), and a large transcriptase protein (L). Virions contain lipids that are derived from host cell (Golgi) membranes. G1 and G2 proteins contain high mannose glycans. RNA replication involves a primary transcription of mRNA from each segment of the genomic RNA via a virion transcriptase; later using the protein products of this transcription, there is production of full-length complementary RNA for each segment, each of which in turn is used as template for the synthesis of genomic RNA segments. Replication takes place in the cytoplasm, and assembly occurs via budding usually upon Golgi membranes. Closely related viruses can re-assort gene segments during mixed infections. The viruses (except *Hantavirus*) replicate in vertebrates and arthropods. Transovarial and venereal transmission occurs in some vector mosquito species and the viruses are generally cytolytic in their vertebrate hosts, but not in their invertebrate hosts. *Hantavirus* are transmitted by persistently infected rodents via aerosolization of urine, saliva, and feces. Some viruses have narrow host ranges, others have wide host ranges and occur worldwide. Adapted from Fields Virology [12].

All members of the virus family show complementary sequences at the 3' and 5' termini of each segment, which are postulated to form stable panhandle structures [47, 36]. The complementary ends also may play a role in replication, possibly by serving as a transcriptase recognition structure.

Within the family *Bunyaviridae* we have analyzed the genera Bunyavirus and *Hantavirus* in detail. The number of complete genome sequences that are available in `Genbank` of the remaining three genera (*Nairovirus*, *Phlebovirus*, and *Tospovirus*) is too small at present to allow a comparative analysis with our methods.

3.1.1 Genus Bunyavirus

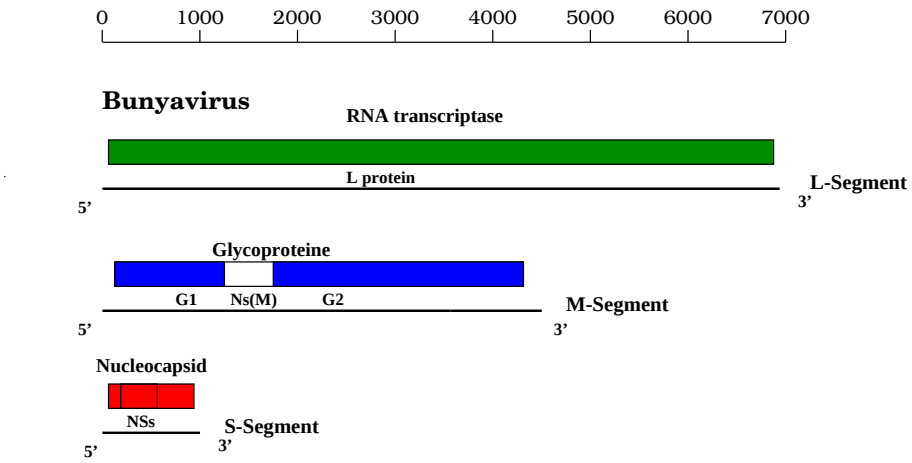


Figure 19: *Bunyavirus* genome map. Translation and processing products of the tripartite anti-sense genome. The L-segment encodes the specific viral transcriptase, M-segment codes for glycoproteins and nucleocapsid proteins are encoded in the S-segment. Replication takes place in cytoplasm via full length complementary segment RNA

Introduction

Virions contain three segments of circular negative-sense and ambi-sense single stranded RNA, which encode for RNA transcriptase, glycoproteins and nucleocapsid proteins.

Total genome length is 12300-12450nt. The largest segment is 7000 nts and labeled L-segment; the second largest 4450-4540nt (M-segment); the third

850-990nt (S-segment). Genome sequences have terminal repeated sequences, at both ends. Terminal repeats at the 5'-end about 11 nucleotides long are well known, also the 3'-terminal sequences are complementary to similar regions on the 5' end, thus forming a panhandle structure.

For our analysis we searched sequence databases for all available complete *Bunyavirus* sequences. For the L-segment too few complete genome sequences are available to use our methods, the tripartite genome is analyzed separately, anti-sense and sense RNA.

Bunyavirus M-segment

Our analysis of *Bunyavirus* M-segment is based on 8 complete M-segment sequences which where found in sequence data banks, see table 15. No selection of genomes was done to improve the alignment.

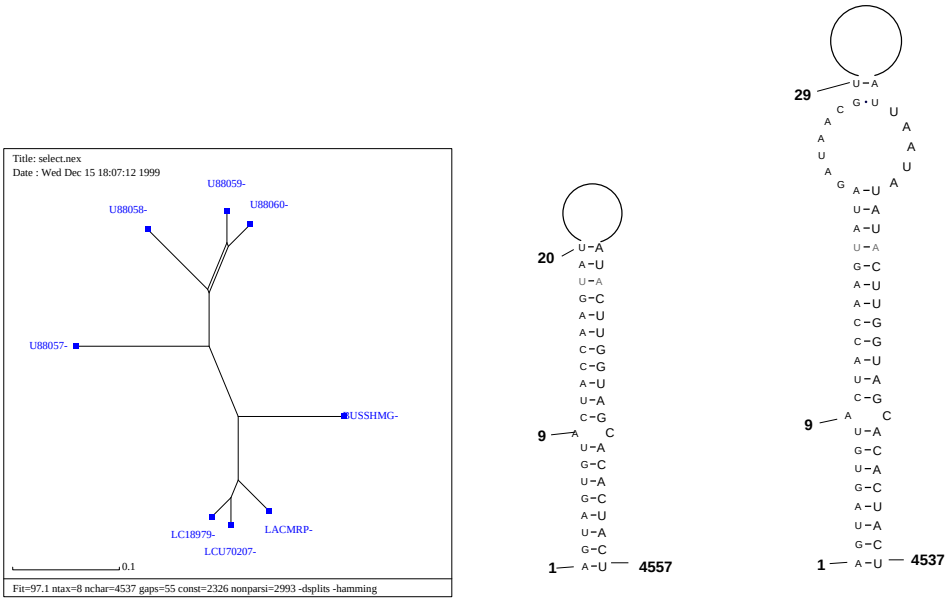


Figure 20: Left most figure the `SplitsTree` plot of the aligned sequences of *Bunyavirus* M-segment, negative sense RNA. The panhandle structure in the middle is from sense RNA and the right most panhandle structure predicted from anti-sense RNA.

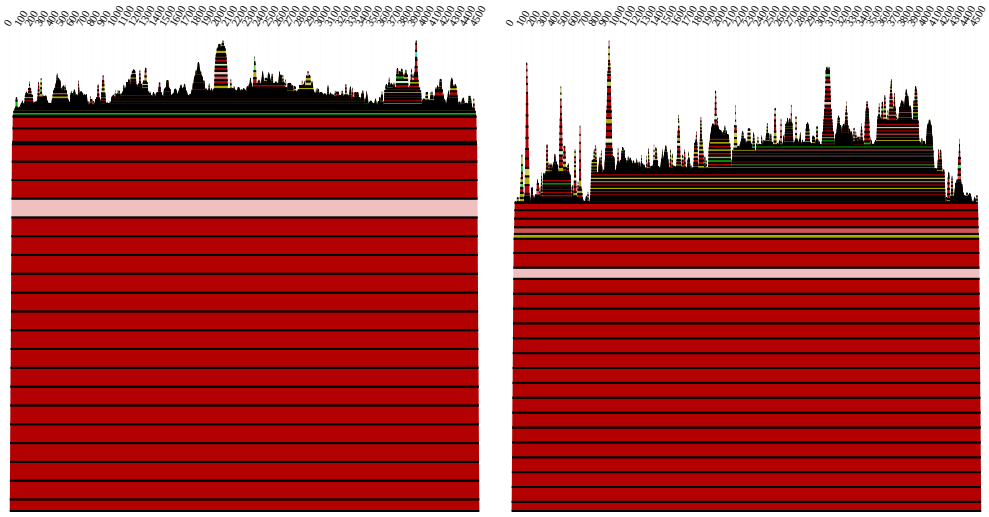


Figure 21: Mountain plots of *Bunyavirus* M-segment, left figure shows the mountain plot of anti-sense RNA and right side of sense RNA. The panhandle structure is the best and only predicted conserved structure motif, see figure 20. The red colored base pairs are conserved, no consistent or compensatory mutations are detected inside the panhandle structure.

The length of alignment using `Clustal W` is 4537 bases long and the mean pairwise homology is 74.8% for the anti-sense RNA. The same sequence files are used to get the sense RNA sequences. Their alignment length was a little bit different in length 4557 and the mean pairwise homology was 74.7%.

Bunyavirus S-segment

The *Bunyavirus* S-segment analysis is based on 9 complete S-segment sequences which where found in sequence data banks, see table 15, these segments are selected to improve the alignment and to represent all 49 sequences available. Sequences with rather different length has be removed. The length of alignment using `Clustal W` is for anti-sense RNA 1010 bases and the mean pairwise homology 76.9%, using sense RNA alignment length is 1043 and mean pairwise homology 76.9%.

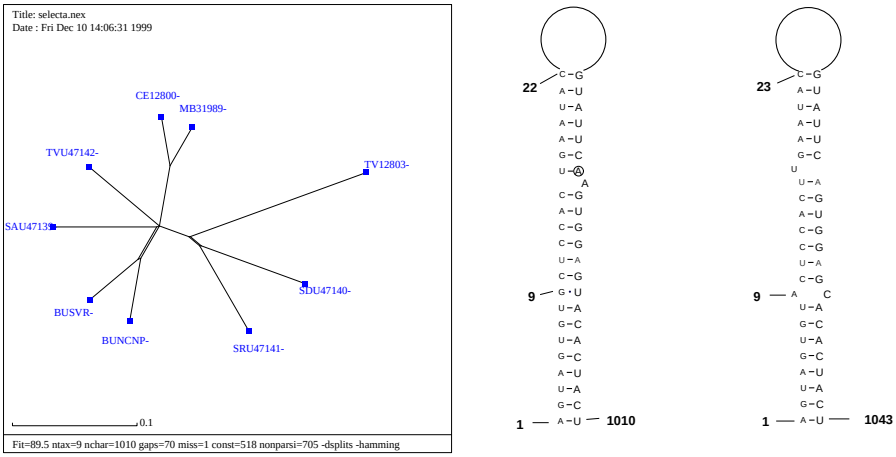
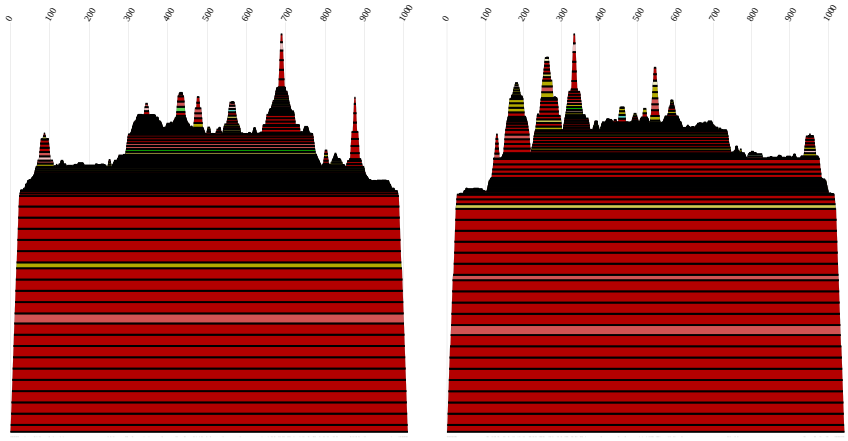


Figure 22: Left most figure the `SplitsTree` plot of the aligned sequences of *Bunyavirus* S-segment, negative sense RNA.The panhandle structure in the middle is from anti-sense RNA and the right most panhandle structure predicted from sense RNA.



23.1 Mountain plot anti-sense RNA

23.2 Mountain plot sense RNA

Figure 23: Mountain plots of *Bunyavirus* S-segment. The left figure shows the mountain plot of anti-sense RNA and left side of sense RNA. The panhandle structure is the best and only predicted conserved structure motif in the S-segment.

Discussion

It is well known that the complementary sequences of the 5'- and 3'-ends of *Bunyavirus* can base pair to each other and form a so called panhandle structure. This structural feature is presented in all 3 virus segments. The panhandle structure, a stacking region of base pairs of different length form a multi-loop over the entire segment RNA. Inside the multi-loop no other conserved RNA structure motif is detected.

The sequences of *Bunyavirus* are rather diverse on the sequence level, approximately 25% of the nucleotides are different inside the genus. Remarkably the 5' and 3' ends of the viral segment RNAs are highly conserved. Formation of the discussed panhandle structure could be essential in the viral life-cycle, maybe the conserved sequence at the 5' and 3' ends play an important role.

Panhandle motifs are also described in *Influenza virus* and it has been experimentally proven that they are functional important for replication, translation and packaging into the virion. *Bunyavirus* may also use such a strategy to regulate replication, translation and packaging. The fact, *Bunyavirus* show only the panhandle structure and no further structural important motifs, is a hint that the panhandle structure possible has that functional importance.

3.1.2 Genus Hantavirus

Introduction

Hantavirus contain a single stranded RNA genome of negative polarity that is divided into three segments. Total genome length is 11800-13800nt, the largest segment 6500-8500nt (L-segment), the second largest 3600nt (M-segment) and the third 1700nt (S-segment).

Hantavirus genome sequence has terminal repeated sequences. Terminal repeats are at the 5'-end 8 nucleotides long and at the 3'-terminus, 11 nucleotides, complementary to similar regions on the 5' end, thus forming a panhandle structure. The tripartite *Hantavirus* genome was found in one particle only.

Genomic segments from different viruses can re-assort when cell cultures are coinfected with two viruses within a group or serocomplex. The L-segments

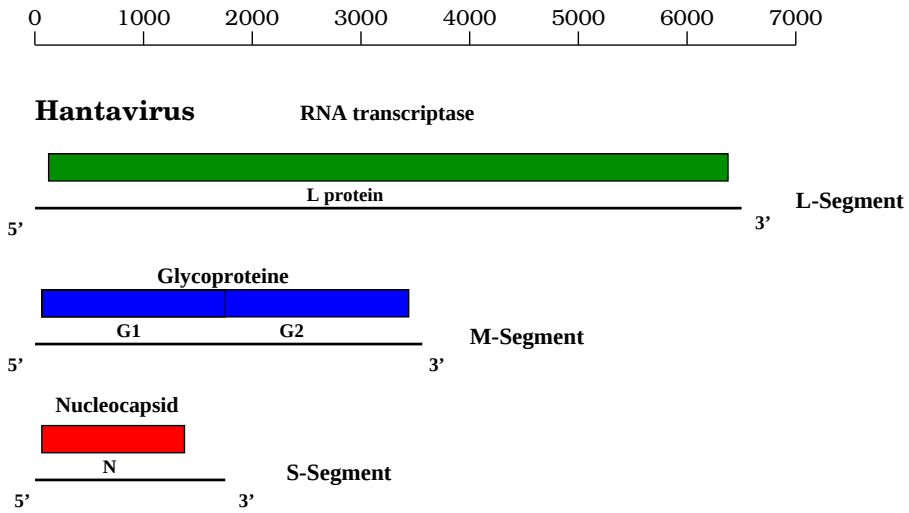


Figure 24: *Hantavirus* genome map. Translation and processing products of the *Hantavirus* tripartite genome. The L-segment encodes the specific viral transcriptase, M-segment codes for glycoproteins and nucleocapsid proteins are encoded in the S-segment. Replication takes place in cytoplasm via full length complementary segment RNA.

codes for a large L protein or polymerase, the M-segment codes for viral glycoproteins (G1, G2), and the S-segment for a nucleocapsid protein (N). Each viral particle contains three internal nucleocapsids composed of genome associated with many copies of the N protein and a few copies of the L protein. The negative single stranded genome follows an anti-sense coding strategy.

Hantavirus L-segment

For our analysis we searched sequence databases for all available complete *Hantavirus* L-segment sequences, and found 8 complete L-segment sequences, see table 12. The length of alignment using `Clustal W` is 6582 bases and the mean pairwise homology 72.4%. Aligning the sense RNA sequences alignment length is 6584 with a pairwise homology of 73.0%.

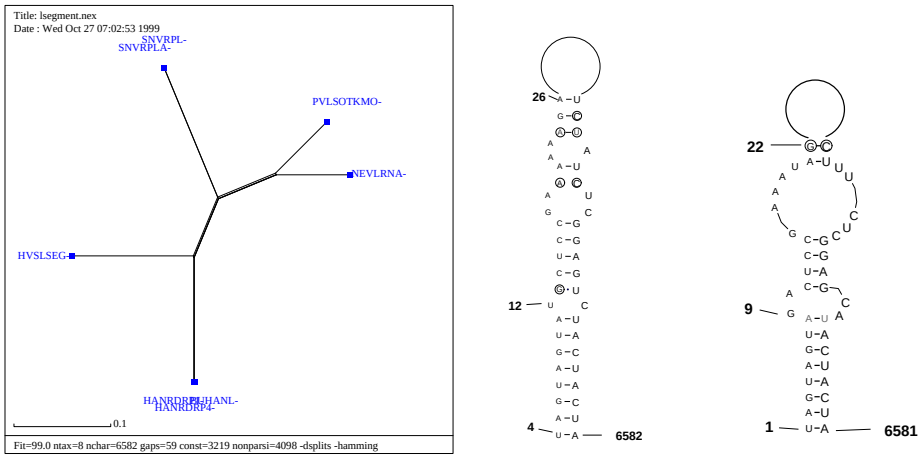
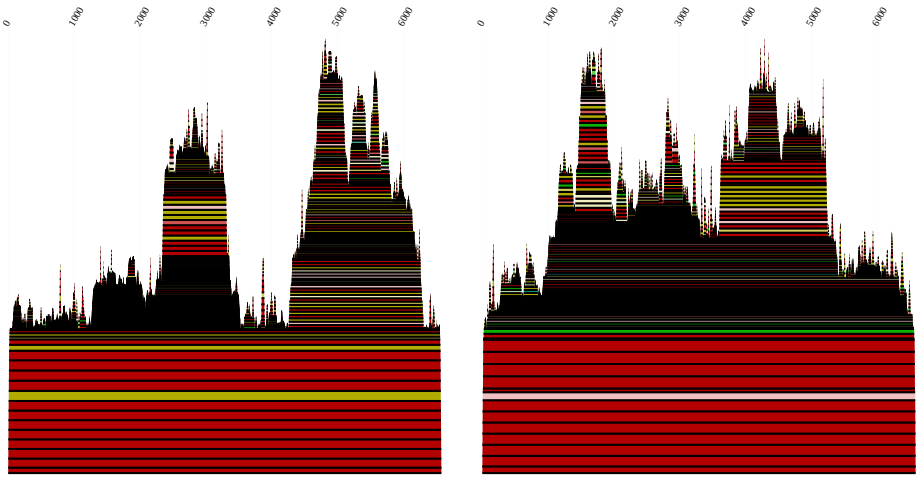


Figure 25: Left most figure the `SplitsTree` plot of the aligned sequences of *Hantavirus* L-segment, negative sense RNA. The panhandle structure in the middle is from anti-sense RNA and the right most panhandle structure predicted from sense RNA. A mismatch at position 9 is characteristic for the panhandle structure of the virus family *Bunyaviridae*, the sense RNA panhandle shows an additional unpaired position at 10.

Table 7: Detected conserved structures of *Hantavirus* L-segment, anti-sense and sense RNA. Position denotes the outmost base pair in aligned genomes. An additional structure motif is found in the sense sequences. Only the panhandle motif is found in anti-sense and sense RNA segment

anti-sense RNA		sense RNA	
Position	Seq. homology (%)	Position	Seq. homology (%)
779-808	77.4	153-182	87.3
3014-3040	87.6	3743-3764	93.2
		5166-5186	95.2



26.1 Mountain plot, anti-sense RNA

26.2 Mountain plot, sense RNA

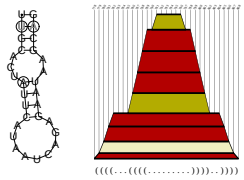
Figure 26: Mountain plot of *Hantavirus* L-segment, left side shows the anti-sense mountain plot and right side the sense mountain plot. The panhandle structure is the best predicted conserved structure motif, see figure 25. Other conserved RNA secondary structures could be detected inside a multiloop formed by the panhandle structure, see figure 27.

The mountain plots of sense and anti-sense RNA show the panhandle structure as the best conserved RNA motif, but there are other possible conserved structures inside the coding region of the L-segment of *Hantavirus*, see figures 25,27.

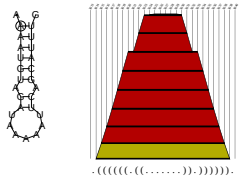
Hantavirus M-segment

For this analysis 24 different sequences are selected from all available M-segment sequences. This selection represents the total amount of *Hantavirus* M-segments, see table 12. The aligned sequences are rather diverse on sequence level, the mean homology after multiple alignment was for anti-sense RNA 65.3% and the alignment length 3754, and for sense RNA 3756 and 65.4%. For such a diverse group of sequences errors in the alignment probably destroys any conserved RNA secondary structure.

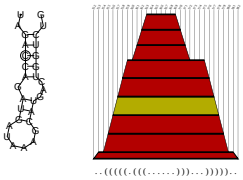
Four different groups are selected and aligned separately to improve the pre-



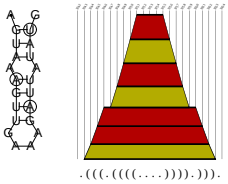
27.1 anti-sense RNA 779-808



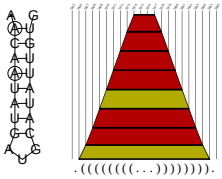
27.2 anti-sense RNA 3014-3040



27.3 sense RNA 153-182



27.4 sense RNA 3743-3764



27.5 sense RNA 5166-5186

Figure 27: Detected conserved structures motifs of *Hantavirus* L-segment. The sense RNA shows an additional conserved motif to anti-sense RNA. All motifs are quite well presented in the ensemble of structures and few consistent mutations occur.

diction. All of them are analyzed separately first the anti-sense RNA than the sense RNA, see table 14.

Four groups are formed with 6 sequences each. Group 1 contains mostly *Hantaan viruses ,Thailand virus* and *Sin Nombre virus*. Group 2 mostly *Puumala viruses, Prospect Hill virus* and *Tula virus*. Group 3 viruses are located mostly in Argentina. Group 4 located mostly in the USA.

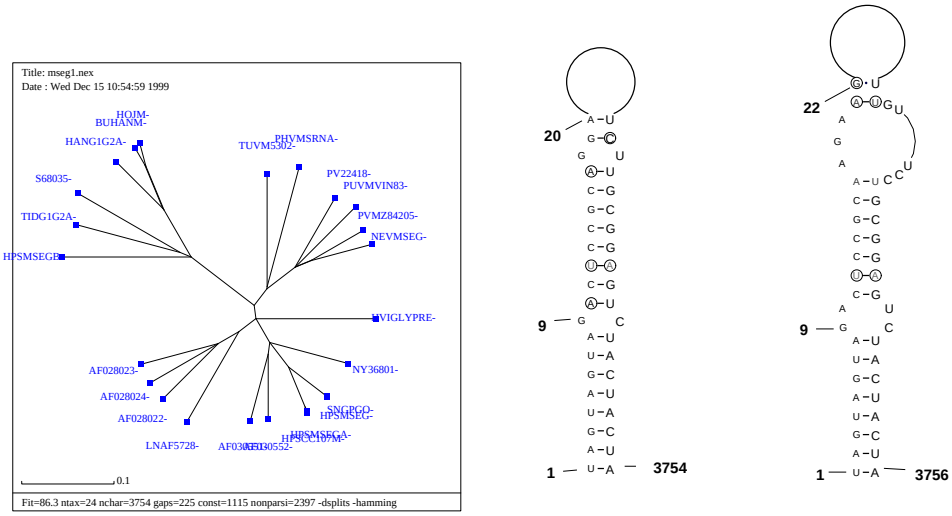


Figure 28: Left most figure the `SplitsTree` plot of the aligned sequences of *Hantavirus* M-segment, negative sense RNA. The panhandle structure in the middle is from anti-sense RNA and the right most panhandle structure predicted from sense RNA. A mismatch at position 9 is characteristic for the panhandle structure of the virus family *Bunyaviridae*, the sense RNA panhandle shows an additional mismatch at position 10.

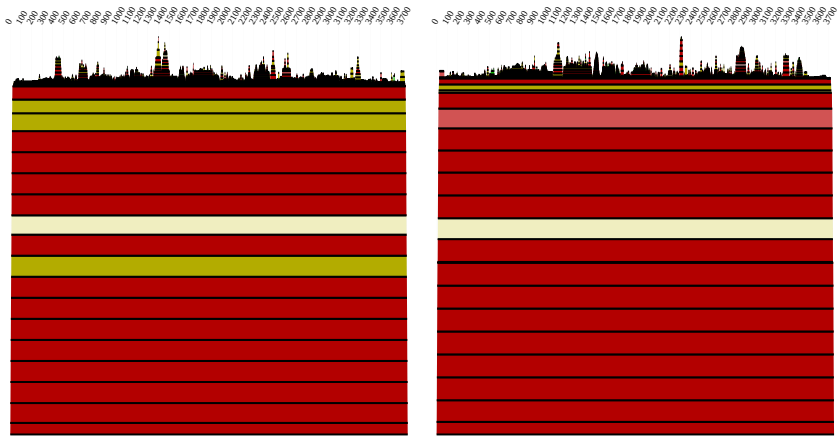
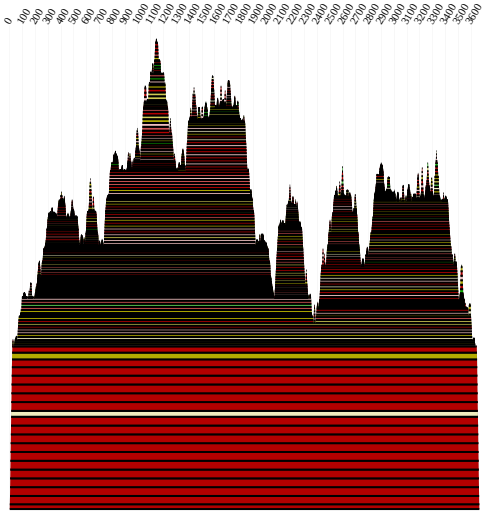
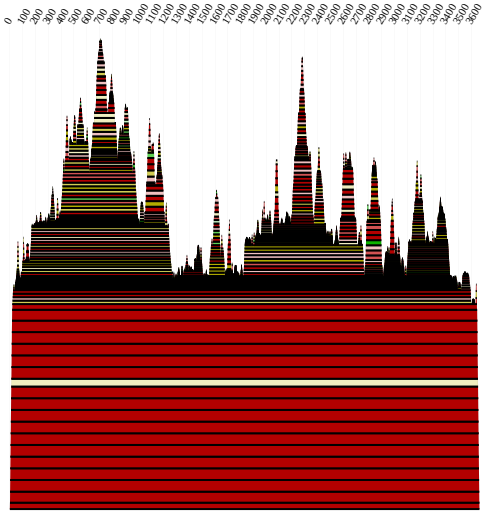


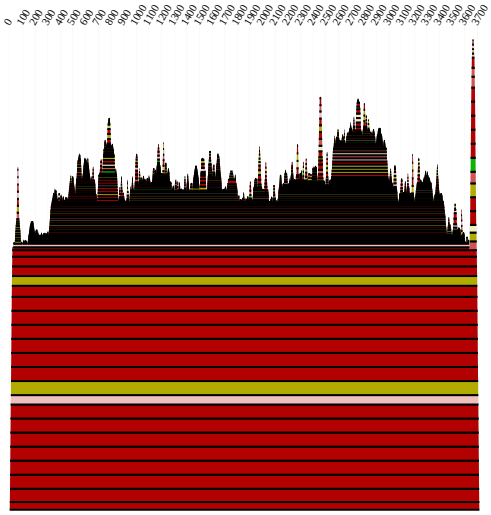
Figure 29: *Hantavirus* M-segment mountain plots of all 24 selected sequences. The left figure is the mountain plot of anti-sense RNA and left side sense RNA. A panhandle structure is the only conserved RNA secondary structure, see figure 28. The green colored base pairs symbolize mutations in base pairs, there are more in the anti-sense RNA.



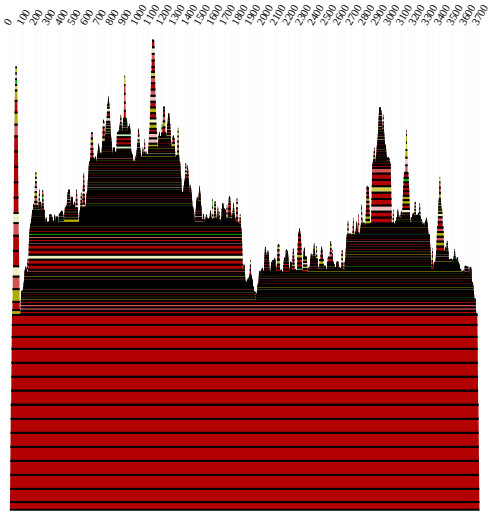
30.1 Group 1, anti-sense RNA



30.2 Group 1, sense RNA

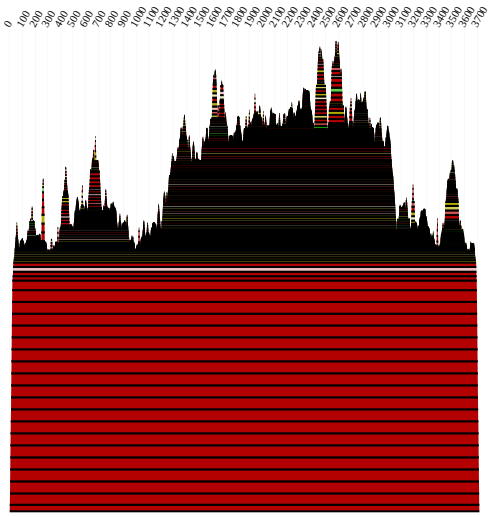


30.3 Group 2, anti-sense RNA

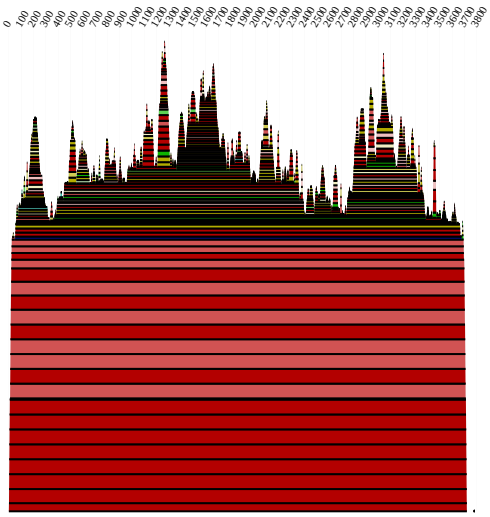


30.4 Group 2, sense RNA

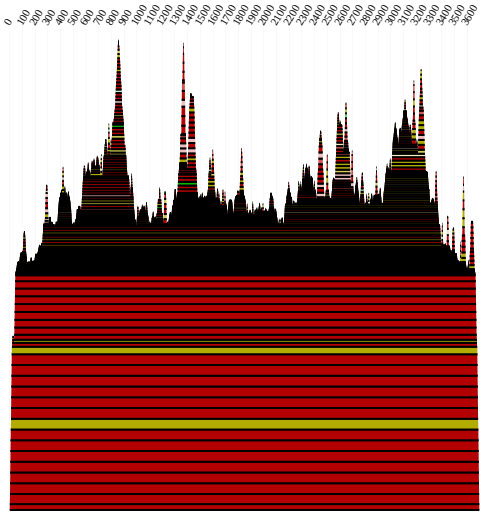
Figure 30: Anti-sense and sense RNA mountain plots of *Hantavirus* M-segment. Mountain plot of group 1 and group 2 are compared. In all groups the panhandle structure is well predicted. Group 1 shows only a panhandle structure. A list of other secondary structures is presented in figure 32.



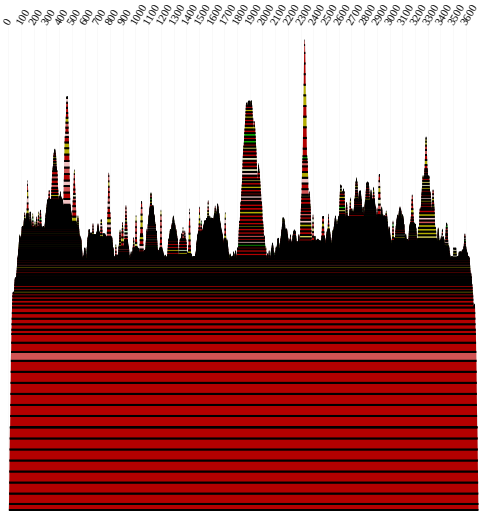
31.1 Group 3, anti-sense RNA



31.2 Group 3, sense RNA

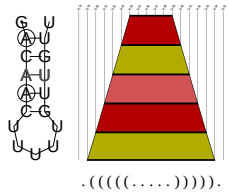


31.3 Group 4, anti-sense RNA

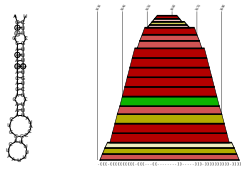


31.4 Group 4, sense RNA

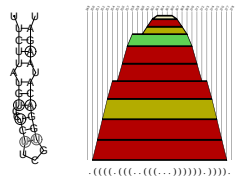
Figure 31: Anti-sense and sense RNA mountain plots of *Hantavirus* M-segment. Mountain plot of group 3 and group 4 are compared. In all groups the panhandle structure is well predicted. A list of secondary structures is presented in figure 32.



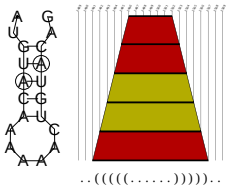
32.1 Group 2, anti-sense RNA 3179-3195



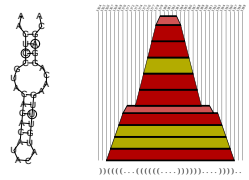
32.2 Group 2, anti-sense RNA 3637-3694*



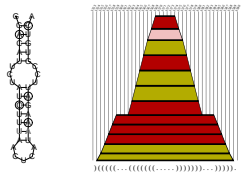
32.3 Group 3, anti-sense RNA 250-278



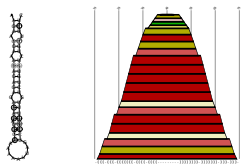
32.4 Group 4, anti-sense RNA 2490-2509



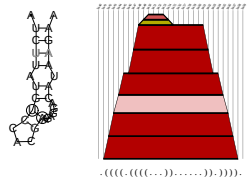
32.5 Group 4, anti-sense RNA 3165-3199



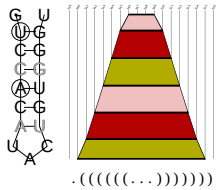
32.6 Group 4, anti-sense RNA 3554-3590



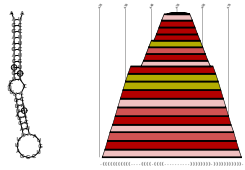
32.7 Group 2, sense RNA 20-79*



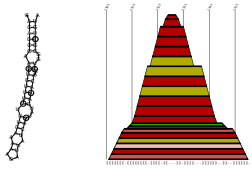
32.8 Group 3, sense RNA 3441-3469



32.9 Group 4, sense RNA 140-156



32.10 Group 4, sense RNA 429-484



32.11 Group 4, sense RNA 2304-2359

Figure 32: Secondary structure graphs and mountain plots of possible conserved *Hantavirus* M-segment structures, sorted by groups anti-sense and sense RNA. The panhandle structure is a common structure motif and is presented in figure 28, only group 2 shows another common RNA motifs (labeled *), predicted in anti-sense and sense RNA.

Table 8: List of detected conserved structures of four different groups of *Hantavirus* M-segment. The start position and sequences diversity of the structural motifs are listed. Position denotes to the start position of conserved elements. Remember the panhandle motif is the best predicted structure motif, see figure 28. Except of group 1 all other groups show at least one additional structure motif. In group 2 a stem-loop structure (labeled *) is predicted in anti-sense and sense RNA, see figure 32.

Group	RNA anti-sense		RNA sense	
	Position	Seq. homology (%)	Position	Seq. homology (%)
1	1-3663	75.6	1-3662	75.6
2	1-3714	75.3	1-3714	75.3
	3179-3195*	77.1	20-79*	76.7
	3637-3694	77.1		
3	1-3718	74.7	1-3709	74.5
	250-278	70.0	3441-3469	74.5
4	1-3696	82.5	1-3696	82.5
	1338-1393	91.0	140-156	86.7
	3165- 3199	87.0	429-484	85.7
	3554-3594	89.0	2304-2359	91.0

Hanta virus S-segment

For the investigation of the shortest *Hantavirus* segment a total amount of 20 sequences is selected, representing all available sequences, see table 13. The `Clustal W` multiple alignment shows 3 different groups on the sequence level. The length of alignment of all 20 S-segments is 2049 bases and the mean pairwise homology is 63.3% for the anti-sense RNA, for the sense RNA the alignment length is 2045 and the mean pairwise homology 63.3%.

To improve the secondary structure prediction three groups are formed, see table 14. The analysis is done separately and the results were compared.

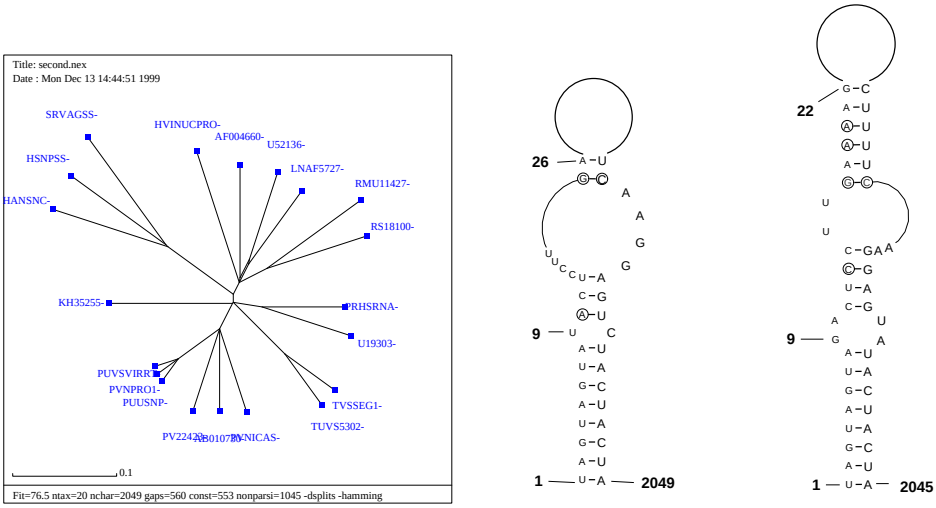
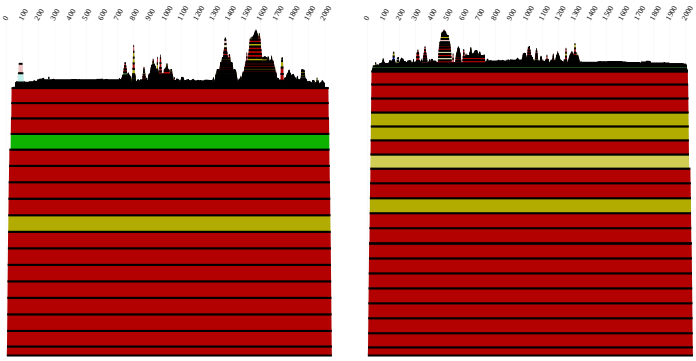


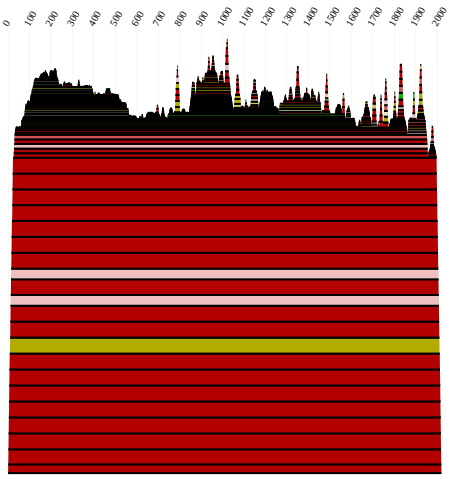
Figure 33: Left most figure shows the `SplitsTree` plot of the aligned sequences of *Hantavirus* S-segment, negative sense RNA. The panhandle structure in the middle is from anti-sense RNA and the right most panhandle structure predicted from sense RNA. A mismatch at position 9 is characteristic for the panhandle structure of the virus family *Bunyaviridae*, the sense RNA panhandle shows an additional mismatch at position 10.



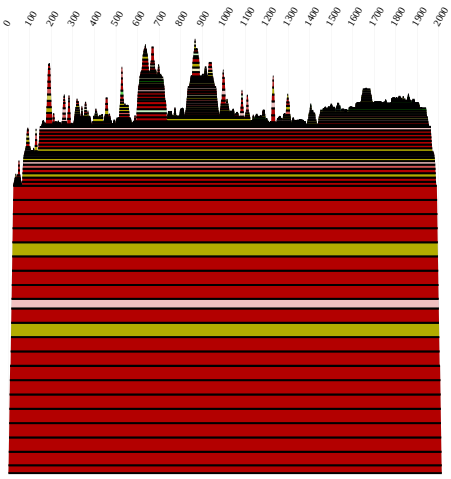
34.1 Mountain Plot anti-sense RNA

34.2 Mountain plot sense RNA

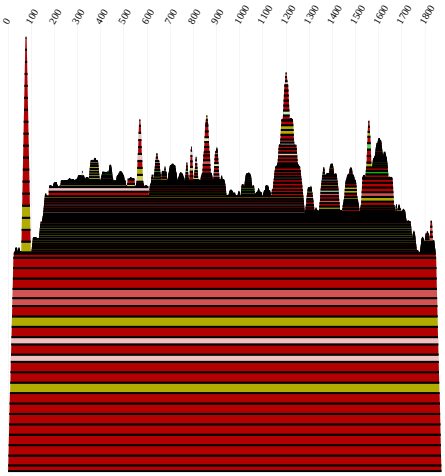
Figure 34: Mountain plots of *Hantavirus* S-segment. Left figure shows the mountain plot of anti-sense RNA and right side of sense RNA. Only the panhandle structure can be predicted.



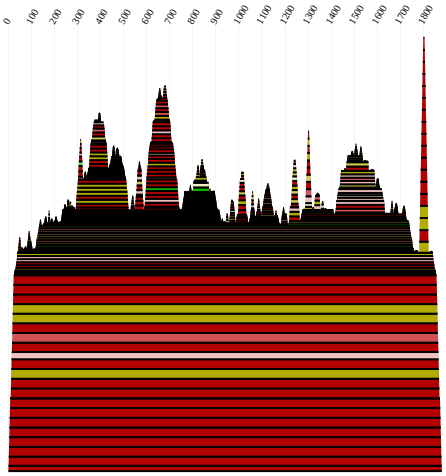
35.1 Group 1, anti-sense RNA



35.2 Group 1, sense RNA

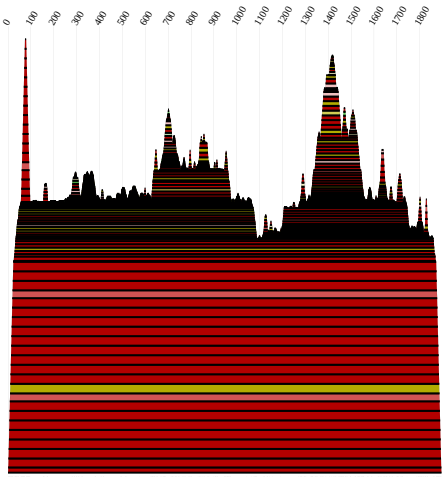


35.3 Group 2, anti-sense RNA

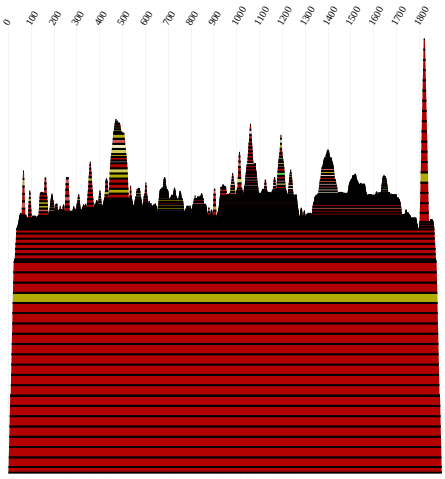


35.4 Group 2, sense RNA

Figure 35: Anti-sense and sense RNA mountain plots of *Hantavirus* S-segment. Mountain plot of group 1 and group 2 are compared. In all groups the panhandle structure is well predicted. Group 1 sequences show only a panhandle structure.



36.1 Group 3, anti-sense RNA

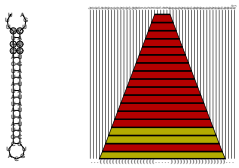


36.2 Group 3, sense RNA

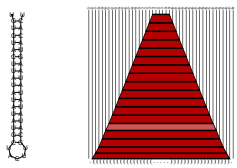
Figure 36: Anti-sense and sense RNA mountain plots of *Hantavirus* S-segment. Mountain plot of group 3 and group 4 are compared. In all groups the panhandle structure is well predicted. For other structure motifs, see figure 37

Table 9: List of the start position and the sequence homology of detected conserved RNA structures of selected groups. The consensus panhandle structure of all sequences is shown in figure 33. In group 2 and 3 additional RNA secondary structures are found. Conserved RNA motifs labeled (*) are commonly predicted in group 2, 3 and also in sense and anti-sense RNA, see figure 37.

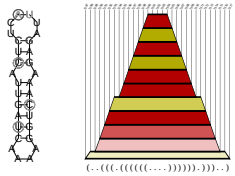
Group	RNA anti-sense		RNA sense	
	Position	Seq. homology (%)	Position	Seq. homology (%)
1	1-2020	67.7	1-2014	67.5
2	1-1871	69.9	1-1880	70.6
	55-101 *	88.7	1288-1317 *	88.0
			1782-1824	90.7
3	1-1895	81.6	1-1898	81.7
	56-98 *	98.4	58-75	93.3
			1801-1843 *	98.4



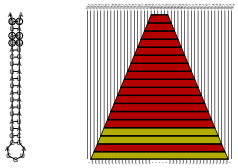
37.1 Group 2, anti-sense RNA 55-101*



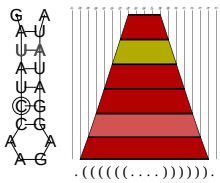
37.2 Group 3, anti-sense RNA 56-98*



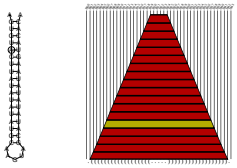
37.3 Group 2, sense RNA 1288-1317



37.4 Group 2, sense RNA 1782-1824*



37.5 Group 3, sense RNA 58-75



37.6 Group 3, sense RNA 1801-1843*

Figure 37: List of *Hantavirus* S-segment conserved secondary structures. Anti-sense and sense RNA structures of selected groups are shown. Remember in all groups a panhandle structure is detected, see figure 33. RNA structures labeled (*) are predicted in different groups and in the sense and anti-sense RNA.

Discussion

Hantavirus genomic sequences are rather diverse on the level of genus, that could cause problems with the used procedure for detecting conserved RNA secondary structure motifs. Aligning all available sequences either L-,M- or S-segments allows only the prediction of the highly conserved panhandle structure. This result is not very surprising, that the 5' and 3' terminal sequences are highly conserved among genus *Hantavirus* and complementary. Both ends can base pair to each other, forming a panhandle structure.

The formation of groups decreases sequence diversity, and increases on one hand the sequence alignment and on the other hand the number of possibly conserved structures. Analysis of selected groups gives use a small number of additional RNA motifs, see figures 32,37, but the panhandle is still the best one.

These RNA motifs are mostly presented in their groups, only one stem-loop structure, at the 5'-end (anti-sense) or 3'-end (sense), can be commonly predicted in M-segment. Inside the S-segment also such stem-loop structure can be detected, even among different groups. This large stem loop structure is identical among the selected groups and well predicted. It is located at the border of the panhandle structure, inside the antisende RNA at the 5'end and for sense RNA at the 3'end. Aligned sequences are rather diverse. Previous only the panhandle structure is discussed in literature. Group 3 consists mainly of *Puumala virus* genomes, and group 2 of *Tula virus, Prospect Hill virus, Prairie vole hantavirus* and *Khabarovsky hantavirus*. This conserved elements can lead mutation experiments to find out its function. Beside this RNA motif each group forms its individual set of secondary structure motifs, different to others.

Little is known about functional RNA secondary structures in *Hantavirus* genomes at all. The conserved 5' terminal nucleotide extension are already examined, and the possible panhandle structure of *Bunyavirus* was already discussed by Paradigon 1992 [50], but the functional importance was not discussed either.

Viruses faces the problem of genome shortening by replication, the panhandle structure maybe can play an important role to overcome this problem. All of the viral RNA polymerase described to date initiate their chains with triphosphates do so with either ATP or GTP. The overhang arrangements of genomes ends is maintained because the 3' A is presumably added in a non templated manner by the viral replicase, in the act of terminating RNA synthesis. The propensity of RNA polymerase to slip back on the template during initiation while retaining the nascent chain, may cause repetitions at the 5' end of the nascent RNA, may also be a more general property of these enzymes [15].

The 5'end of *Hantaan virus* genome is exact the complement of its 3' end. A prime-and-realign (or slip-back or jump-back) mechanisms which initiate viral genome synthesis require terminal sequence repetitions, and all *Bunyaviridae* genera contain such di- or trinucleotide repeats at their ends. There is one other feature of this mechanism that require comment, its ability to repair damaged genome ends by restoring small terminal deletions and mutations. The regeneration of damaged ends by using pseudotemplated synthesis and terminal sequence repetitions would, of course also apply to RNA

viruses and may be important in maintaining virus infective when these ends undergo limited damage. Genomes which lack a few nucleotides at the 3' end can be repaired by simply extending these ends on an intact complementary 5' end require a different mechanism for repair, as conventional RNA synthesis takes place only 5'-to 3' direction. The prime-and realign mechanism allows growing of RNA 3'-to 5' direction [15].

Analysis of other negative-strand RNA viruses has shown that 5' and 3' terminal nucleotides sequences, as well as putative panhandle like structures formed by 5' and 3' termini of RNA molecules, are involved in the process of initiation and regulation of viral transcription, replication, and encapsidation [5].

Panhandle structures at least 17 bp are formed by highly conserved complementary regions of the 5' and 3' termini of each segment. Complementarity is incomplete in all cases, with a mismatch at position 9. Our investigation also shows a bulge at position 9 of the panhandle structure. Position 10 is only unpaired in the coding RNA segments, different to anti-sense RNA. The observation of incomplete complementarity of RNA termini is similar to the situation seen in other negative-strand viruses. For instance, *Influenza virus* has been shown to possess a mismatch bulge in the panhandle structure formed by genome segment termini. This mismatch region has been determined to be the virus polymerase binding site [69].

Conversion of the termini to exact complementarity destroys polymerase binding. In *Vesicular Stomatitis Virus* RNA termini has been shown to influence the balance between transcription and replication. By analogy, one can speculate that this unpaired base pair is a binding site for polymerase. Analysis of the role of various 3' terminal regions of the *Vesicular Stomatitis Virus* genome RNA in the encapsidation and replication of defective interferrring particles demonstrate that bases 1-12 were involved in the encapsidation process, whereas bases 13-18 were not. In addition, bases 19-24 were involved in replication and virus assembly.

By analogy, the highly conserved bases 1-14 found at the 3' termini of *Hantavirus* sense and anti-sense RNA templates may be involved in initiation of encapsidation and/or binding virus RNA polymerase, whereas the nucleotide differences in positions 20-28 between different RNA segments could determine the differential rate of RNA segment transcription or replication.

The conservation and experimental analysis point out the importance of the

panhandle structure, but do not imply, that other functional important RNA secondary structure can not exist. Although all *Hantavirus* segments present the panhandle as the best conserved motif, there are also few other RNA motifs. The function of these RNA elements can not be determined by theoretical methods alone. We can only present them. Our selected elements could guide further experiments to determine, whether they are functionally important or not. At least the fact any other conserved RNA element can be detected is a new discovery for *Hantavirus* genomes.

3.2 Structure Motifs, Flaviviridae

Introduction

The virus family *Flaviviridae* contains the genera *Flavivirus*, *Pestivirus* and *Hepatitis C virus*. In this section the genera *Hepatitis C virus* and *Pestivirus* are examined.

The *Hepatitis C virus* is responsible for chronic liver infections in man and was first identified in 1975 as a non *Hepatitis A* and *Hepatitis B* virus. The viral infection is a leading cause of cirrhosis and liver cancer, and is now the main reason for liver transplantation in the United States. Recovery from infection is uncommon, and between 70 and 85 percent of infected persons become chronic carriers of the virus. There is no cure or vaccine for *Hepatitis C virus* which is spread primarily by direct contact with blood.

The *Pestivirus* contain three different species *Bovine diarrhea virus* (BVDV) infecting cattle, *Hog cholera virus* or *Classical swine fever virus* infecting swine, the third one *Border disease virus* is infecting sheep. The different species are closely related, both antigenically and structurally. The virus is not restricted to a single host, for example BVDV can also infect sheep and swine.

Virions contain one molecule of linear positive-sense single stranded RNA. Total genome length is 9500-12500nt. Translation of the virus polyprotein occurs cap independent. An *internal ribosome entry side* (IRES) inside the 5' non coding region is responsible for ribosome binding and translation start. Both non coding region 5' and 3' are supposed to have regulatory effects for polyprotein translation.

The polyprotein encode for three to four structural virion proteins. Virion structural proteins are usually glycosylated, or not glycosylated (in some viruses). The non-structural proteins including protease, helicase and polymerase, are encode at the 3' end of the coding region [12, 48].

3.2.1 Genus Hepatitis C virus

Introduction

Hepatitis C virus (HCV) was first recognized as *non-A, non-B Hepatitis* in 1975. Disease was transmitted to chimpanzees in 1978. The genome of non-A, non-B HCV was cloned and sequenced in 1989 and renamed the *Hepatitis C virus*.

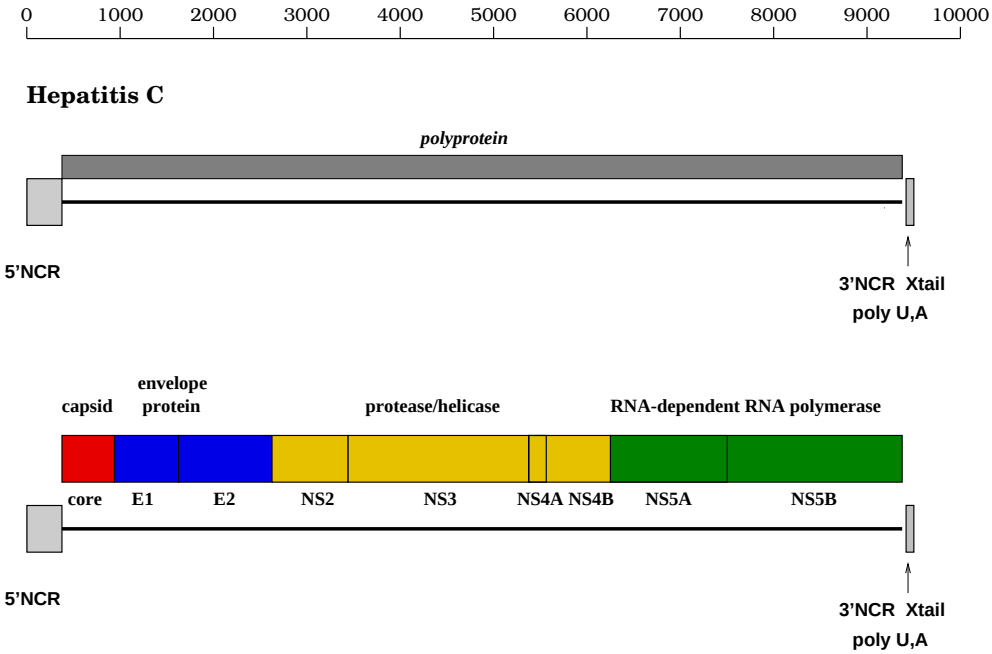


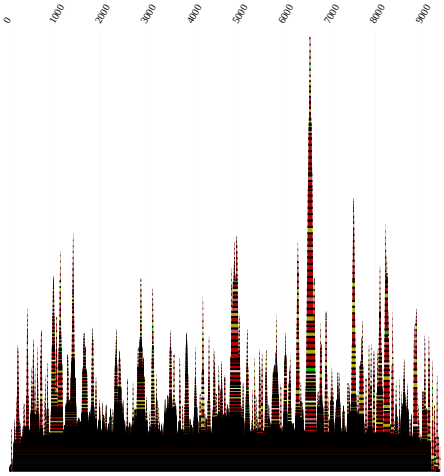
Figure 38: HCV genome map. Translation and processing of the HCV polyprotein. At the top is the viral genome with structural and non structural protein coding regions. Boxes below indicate mature proteins generated by the proteolytic processing cascade [12, 48].

HCV is a spherical, enveloped, single stranded, linear RNA virus which is arranged in a positive sense configuration. The genome contains 9.5 Kb with a 9Kb open reading frame which codes for a single 3K amino acid polyprotein. The open reading frame is flanked by 5' and 3' non-coding regions of approximately 340 and 100 nucleotides, respectively. The virus is bound to

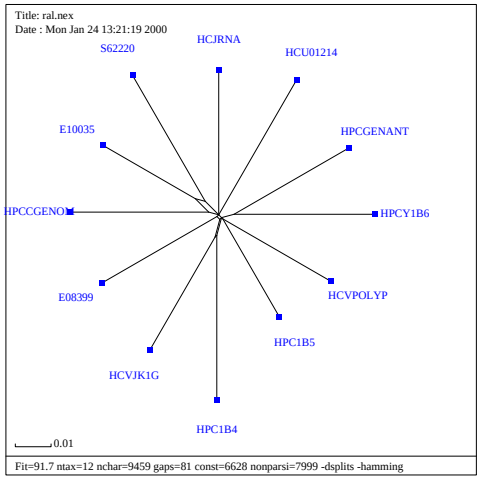
low density lipoproteins in vitro [12, 48].

RNA Structure Motifs

For the analysis 12 sequences were selected, see table 10. The Multiple alignment is done by `Ralign`, its length is 9459 bases and the mean pairwise homology is 90.9%.



39.1 Mountain plot of HCV genome



39.2 Aligned HCV sequences

Figure 39: Mountain plot of genome and `SplitsTree` plot of aligned sequences of HCV. Left side shows the colored mountain plot of the entire HCV genome. Right side shows all aligned sequences and their alignment distance.

With the help of the `Vienna RNA Viewer` possible conserved secondary structures were selected form the data set. This resulted in a rather huge list of RNA motifs, which could be functional important and could play a role in the viral life cycle. Previously known motifs are discussed and examined later. The RNA motifs which have been selected show a relative high number of compensatory mutations and are well presented in the ensemble of structure.

The non coding regions of a virus play an important role, see IRES function and discussed hairpin loops at 3' terminus of the HCV genome. A rather huge number of structures are also found inside the coding region, which is a hint, that possibly important regulatory regions can be situated also inside the coding region of the virus genome.

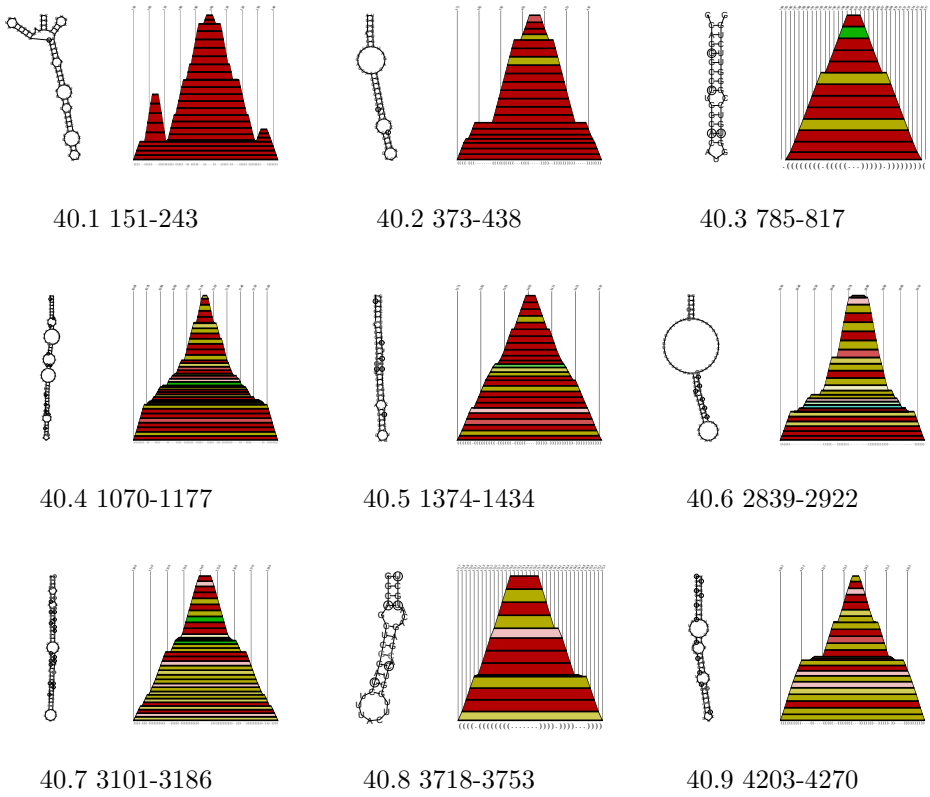
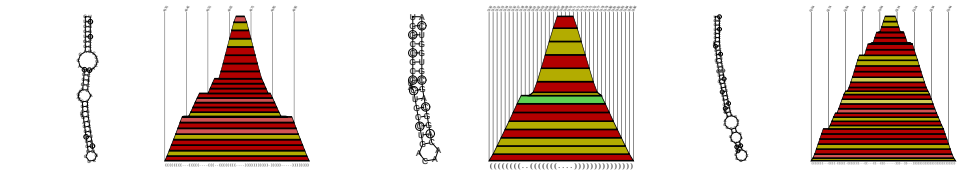
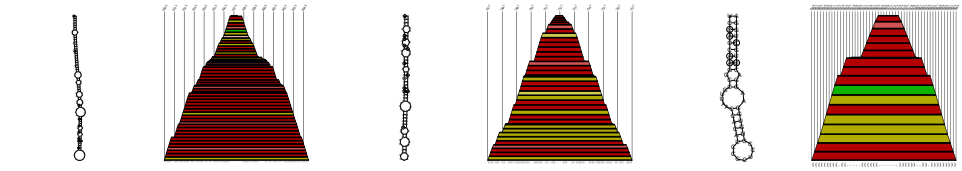


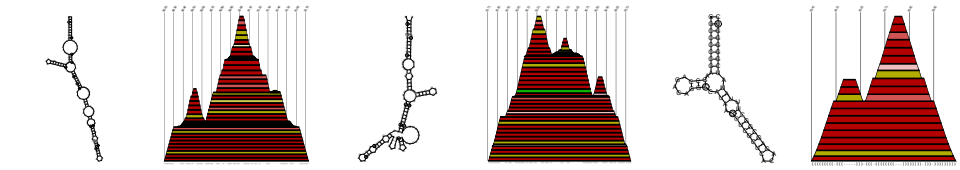
Figure 40: A list of probably conserved secondary structures of HCV. The IRES, see figure 40.1 was also proposed by Brown [4]. Numbers indicate starting position of base pairs in the aligned sequences.



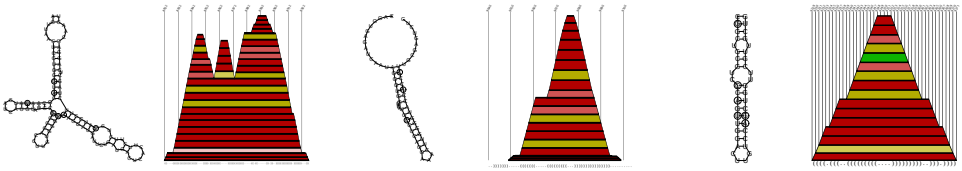
41.1 4336-4402 41.2 5351-5386 41.3 6265-6348



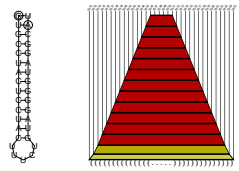
41.4 6504-6648 41.5 7478-7577 41.6 7945-7994



41.7 8021-8173 41.8 8176-8321 41.9 8346-8404



41.10 8823-8927 41.11 9047-9110 41.12 9230-9271



41.13 9333-9365

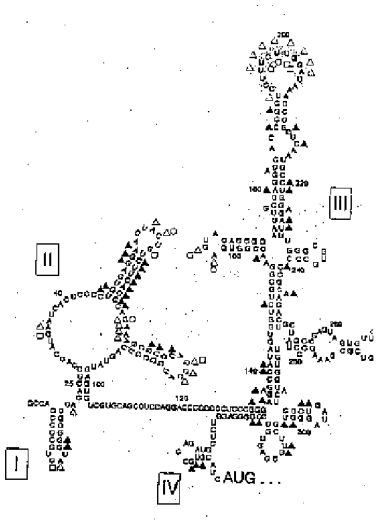
Figure 41: A list of probably conserved secondary structures of HCV. Numbers indicate starting position of base pairs in the aligned sequences.

5' Non Coding Region (5'NCR) of Hepatitis C Virus

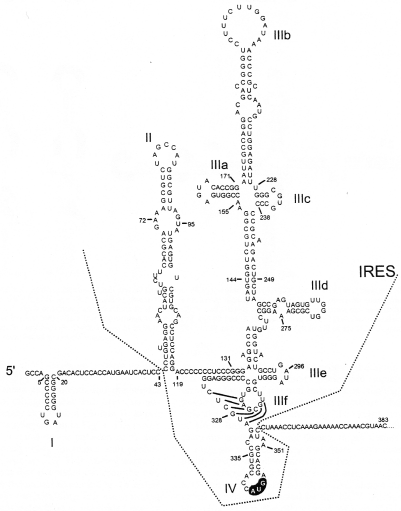
The RNA genomes of human *Hepatitis C virus* (HCV) have relatively lengthy 5' non-translated regions (5'NCR) sharing short segments of conserved primary nucleotide sequences. This 5'NCR region of HCV is responsible for cap independent translation of the HCV genome. With the help of comparative sequence analysis and thermodynamic modeling Brown proposed a secondary structure model [4]. In this model the detected *internal ribosomal entry site* (IRES) mainly consists of 4 different domains, see figure 42. There are conflicting views, which region of the viral sequence is responsible for IRES activity. A lot of discrepancy could have resulted from the inclusion of less than full-length 5'NCR in constructs studied for translation initiation. Brown analyzed only the 5'NCR segment of the HCV genome, a full length HCV genome was studied by Honda et al. [32].

In our study we folded the virus genome in its entirety, long range interaction as the panhandle structure of *Bunyaviridae* play an important role in viral life-cycle and can not be neglected. Our proposed secondary structure model only shows part of the structure model of Brown, see figure 42. The stem loop structure labeled I is not predicted in the aligned consensus structure. The sequence alignment introduces several gaps at the very 5' terminal end of the virus genome. Investigations of secondary structures of the used virus genomes show most of them have this stem-loop structure at the 5' end of the virus, this implies sequence alignment destroys domain I in the consensus structure. Domain II is a multi-loop structure with two stacking regions. One of the stems with the sequence (ACUACUGU) in the hairpin matches the stem in our prediction from position 49-70 (IIa), Honda published two alternatives for the domain II structure [31], one of them is presented in figure 42. Honda labeled this stem-loop region IIa, which we detected in our consensus structure.

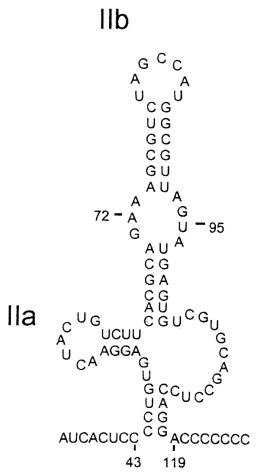
Domain structure III shows a highly conserved secondary structure motif. The hairpin structure labeled IIIb is representing a complementary sequence to ribosomal RNA, this complementary sequences (CCUUUCUUGGA) is highly conserved among HCV virus strains and is complementary to bases 461-471 of human 18S RNA. In our prediction domains IIIa-IIIc match the predicted structures of Brown and Honda, the rest of domain IIId-IIIf can not be found in the consensus structure.



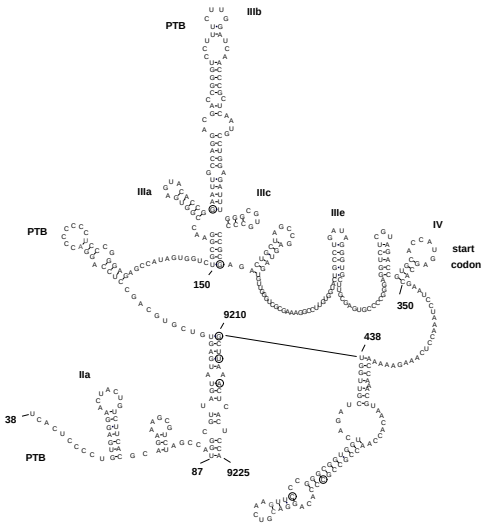
42.1 Brown 1992



42.2 Honda 1998



42.3 Honda 1998



42.4 5' NCR, consensus structure

Figure 42: Proposed secondary structure models of the HCV 5'NCR element and stem-loop structure IIa. Figure 42.1 secondary structure model proposed by Brown, four different domains are labeled [4]. Figure 42.2 structure model proposed by Honda [32]. Figure 42.3 stem-loop structure IIa proposed by Honda [31]. Figure 42.4 consensus structure of our prediction of the 5' and 3' NCR regions. A multi-loop connects both ends, it is formed between nucleotides 87 and 9225 and its length is 16. Consensus structure model predicts well domains IIa, IIIa-IIIc, IIIe and IV. Polypyrimidine binding protein regions are labeled PTB, and circles around base pairs denote to consistent mutations.

In the 5'NCR region a pseudo-knot seems to be established, and our folding algorithm do not contribute to such RNA interactions at all. In domain IV, there are different structure models proposed by Brown and Honda, our prediction favours a stem-loop structure with the starting codon of the open reading frame in the unpaired hairpin region, see stem-loop IV in figure 42.

The occurrence of domains II and IV should play an important role in IRES function. Deletion of nucleotides 28-69 of the 5' NCR (stem-loop IIa) sharply reduced capsid translation both in vitro and vivo, and deletion mutants directly upstream the initiator AUG also resulted in a nearly complete inhibition of translation [32].

Honda reported that domains II and III of the 5' NCR are both essential to activity of the IRES while conservation of sequence downstream of the initiator AUG is required for optimal IRES-directed translation. The 5' terminal region may bind a polypyrimidine tract-binding protein (PTB). PTB binding could be important for determining the higher-order structure of the 5' NCR and might interact with other factors involved in RNA replication. Three distinct PTB binding site has been detected within the 5' NCR of HCV [1], see figure 42. PTB is believed to be a homo-dimer which, in theory, might initiate or stabilize interactions between the HCV 5'NCR and 3' NCR region. Such interactions could be important for modulating translation versus replication of HCV genome RNA.

Little is known about the molecular interactions required for HCV viron assembly. The highly basic core protein is rich in arginine and lysine residues and can form specific interactions with the 5' NCR of HCV. This could be important for virus encapsidation. Interactions with other virus proteins could not be detected [8].

Our predictions show a different structural feature of HCV 5'NCR region, where we found a multi-loop structure combining the 5' with the 3' terminus of the virus genome.

3' Non Coding Region (3'NCR) of Hepatitis C Virus

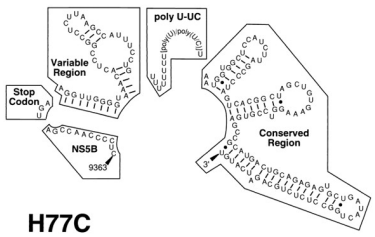
Following the long ORF, most reports suggests that HCV genome RNA contains a short 3' NCR followed by a poly(U) homo-polymer tract. In contrast, the genome RNA of HCV-1 (genotype 1a) has been reported to contain a 3'-terminal poly(A) tract.

The 5'- and 3'- terminal sequences and structures of positive stranded RNA viruses often function as cis acting elements important for RNA replication and/or packaging, such elements are typically highly conserved. Correct terminal sequences can therefor be of critical importance for recovery of infectious RNA transcripts. The function of the HCV 3' NCR, including the highly conserved 3'-terminal element, remains to be determined. For other RNA viruses, conserved terminal sequences or structures play critical roles in initiation of minus and plus strand RNA synthesis and in packaging of viral RNAs. Such processes are mediated via interactions with trans acting proteins encoded by the virus or host and, in some cases, other cis RNA elements elsewhere in the genome. For instance, conserved tRNA-like structures at the 3' termini of *Bromovirus* RNAs are required for initiation of minus-strand synthesis.

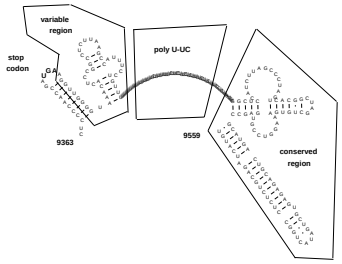
For negative-strand viruses, such as *Influenza virus* and *Vesicular stomatitis virus* conserved sequences at the 5' and 3' termini can base pair and constitute the cis regulatory elements for transcription, replication, and packaging [54, 55, 38]. Terminal cis RNA elements important for translation and RNA replication have also been identified for positive-strand animal viruses, including *Alpha viruses, Flaviviruses* and *Picornaviruses*. The 3' terminal region may also bind a polypyrimidine tract-binding protein (PTB). PTB binding could be important for determining the higher-order structure of the 3' NCR [1]. PTB interaction with the variable region of HCV can cause translation enhancement. Alternatively, other translation factors or primary sequence or secondary structure RNA may also be involved in translation enhancement.

Our consensus structure of 12 selected HCV genomes lacks a 3' terminal consensus sequence, see figure 43. The aligned sequences are about 100nt shorter than the HCV H77C strain, see table 10, where the length of the conserved element is 98 nt. This so called X-tail is not present in the published "complete" genomes with rare exceptions.

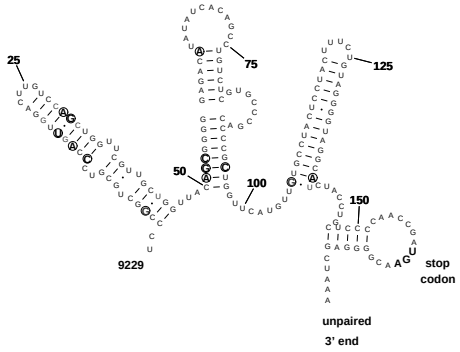
A comparison of the proposed model to the thermodynamically predicted consensus structure of HCV strains H77C show alternatives structures, in both regions conserved and variable, see figure 43.1. The used set of HCV genomes present a three stem loop motif downstream the stop codon.



43.1 3' NCR of H77C



43.2 consensus struct. H77C



43.3 3'NCR consensus structure

Figure 43: Figure 43.1 shows the 3' NCR of HCV strain H77C, its length 225nt, consisting of the open reading frame (ORF) stop codon, a short sequence of 40nt (variable region), a poly(u-UC) region of 81nt, and a 3' terminal sequence of 101nt (conserved region). Sequences in the 3' end of the NS5B protein coding frame and in the variable region of the 3' UTR could potentially form two stem-loop structures, and sequences of the conserved region of the 3' UTR could potentially form three stem-loop structures [77].
Figure 43.2 consensus structure of the 3'NCR of HCV strain H77C. The conserved region is different to H77C, and the variable region shows the same secondary structure. The stop codon can be found in the hairpin loop.
Figure 43.3 Predicted consensus structure of 12 different HCV genomes, see table 10. These genomes lack the X-tail sequence. Circles around base pairs denote to mutations, either consistent or compensatory. A three stem loop motif can be found at the 3' end of the ORF.

These stem-loops are well presented in the ensemble of structures and several mutated base pairs are detected. The function of this region is not known, maybe it has also regulatory function for virus replication, as proposed for the X-tail by Yanagi 1999 [77].

Conserved RNA Element inside the Open Reading Frame

The non coding regions are known to contain several important structural domains. Our investigations show a lot of interesting secondary structures inside the open reading frame, not yet described. One of the possibly functional structures is examined in detail.

A huge stem loop structure was found at position 6466 to 6714 inside the coding HCV genome, its size is 249nt, nearly all nucleotides form base pairs and the calculated minimum free energy is -90.2 kcal/mol (`Vienna RNAfold 1.3`). The mean pairwise homology of the structure is 92.3% and 179 bases are conserved and 18 base pairs has either consistent or compensatory mutations. A good example for a RNA element well conserved in our respect.

The function in HCV life-cycle is unknown, its location inside the ORF at the beginning of the polymerase coding region (position 6364-9354). Maybe this element is important for translation attenuation. The amount of translated polymerase can be regulated by this RNA motif. This function is highly speculative, and has to be proven by experimental data.

Discussion

The HCV genome gives a good example where RNA secondary structures play an important role in the viral life-cycle, regulating virus replication and protein translation. Non coding regions are best known for this kind of functional RNA secondary structures. Cap independent translation is mainly controlled by the 5' non translated region, where an *internal ribosome entry site* (IRES) allows docking of human ribosome subdomains. A complementary sequence in domain IIIb to 18S human ribosomal RNA gives a good example.

Part of four well known functional domains can be detected by our methods. Without knowledge of the HCV functional RNA secondary structures, part

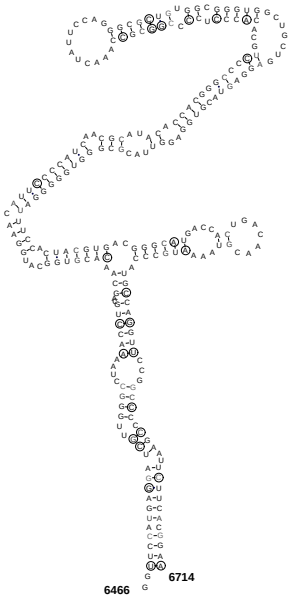


Figure 44: Stem-loop structure inside ORF of HCV genome. Its size is 250nt, nearly all nucleotides are base paired. Circles around base pairs denote to mutations.

of IRES domain III is predicted as conserved element. The stem-loops IIIa to IIIc show highly conserved structures, such important feature as complementary ribosomal sequence, or binding site for a PTB protein for translation control are known. A pseudo-knot also plays an important role for IRES function, our secondary structure prediction programs do not contribute to such RNA contacts. Stem-loop IIa, where the functional region of IRES starts, is also well predicted, the same with stem-loop IV, this hairpin contains the start codon of the open reading frame.

The predicted consensus structure of twelve selected HCV genomes also gives an example of different RNA secondary structures as already described. HCV 3' non coding region is well known for its importance of translation control. A 98nt highly conserved sequence at the very end of the genome shows a three stem-loop motif, which binds PTB. Mutational analysis shows translation enhancement of this so called X-tail. The same HCV sequences as used

for this investigation fold into a different secondary structure as proposed. Unfortunately most "complete" genomes lack this important region. There are reports, that this structure is important for PTB binding and translation control, but there are also alternative binding possibilities. Maybe our detected three stem loop motif inside the coding region can also bind PTB, and can be responsible for translation control. Several compensatory mutations inside the stem-loops are a hint for the functional importance for the virus. Two of these stem-loops are also detected by our algorithm as possibly important.

A structural completely new motif in our proposed 5'NCR region is represented by a multi-loop structure. This multi-loop consists of 13 base pairs and three consistent mutations, it starts right in front of the ORF and ends before the predicted 3' stem-loop motif begins. This functional feature of HCV genome has not been detected before, and no function is known. Maybe the multi-loop is responsible for interaction of the 3' end with the 5' end of the genome. It is proved, that PTB binds both termini of the genome and is functional important for translation control. This translation factor is homodimeric in its structure. The multi-loop can be important to get the virus ends close together, to establish PTB's translational control function. For an open chain it would be rather difficult binding both ends at once.

The results of our algorithm shows a lot of possibly important RNA secondary structure inside the coding region of HCV. The best motif is selected to give an example for possible important regions inside the coding HCV genome. The selected stem-loop motif is well predicted in the ensemble of structures and a lot of compensatory mutations are detected. Its location inside the start region of the polymerase gene is a hint, that its function could be translation attenuation of the polymerase protein. The amount of polymerase for virus replication is rather low to structural proteins, so it is useful to use a regulatory element to decrease it's translation. Large hairpin structures are known being responsible for stopping protein translation. This RNA stem-loop motif can also be a cis regulatory element for polymerase translation, which have to be proved.

At present only little is known about functional elements in the coding region of HCV. The HCV genome is rather complicatedly regulated, and several reports underline the importance of functional RNA secondary structures. Our analysis is the first attempt so search also inside the coding region for

such elements, and several interesting RNA motifs have been detected. They can guide mutation experiments to find functional RNA secondary structures.

3.2.2 Genus Pestivirus

Introduction

The genome RNA of prototype strains of *Bovine viral diarrhea virus* (BVDV), *Classical swine fever virus* (CSFV) and *Border disease virus* are single stranded RNAs 12.3 to 12.6 kb in length. Larger genome RNAs contain duplications and rearrangements, have been found for some BVDV. *Pestivirus* genome RNAs do not contain a 3' poly (A) but appear to terminate with a short poly (C) tract.

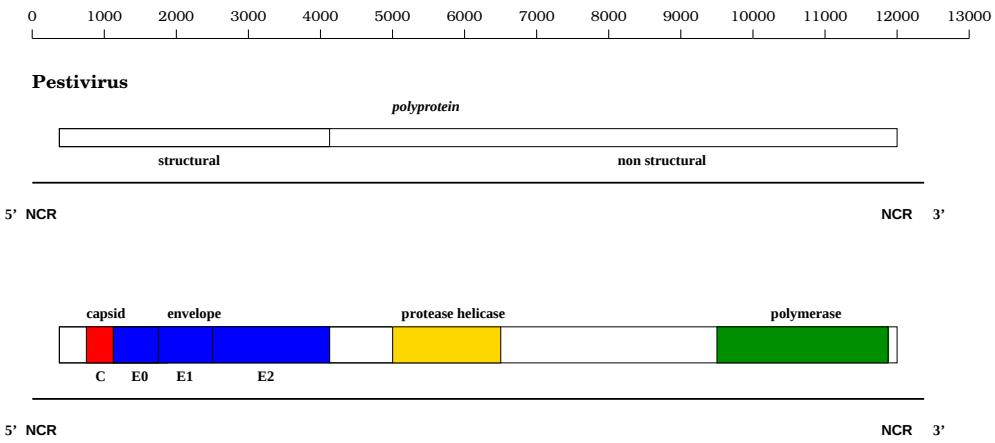


Figure 45: *Pestivirus* genome map. Translation and processing of the *Pestivirus* polyprotein. At the top is the viral genome with structural and non structural protein coding regions. Boxes below indicate mature proteins generated by the proteolytic processing cascade [12, 48]. Red and blue colored boxes denote to structural proteins, yellow boxes protease and helicase proteins and green colored virus specific polymerase.

The 5' terminus has not been analyzed directly, but it has been suggested that the genome RNAs lack a 5' cap structure. As for the *Flaviviruses*, no *Pestivirus* sub genomic RNAs have been detected. The long 5' non coding

region (NCR) contains several short ORFs of unknown function and has been predicted to form a highly structured RNA element that may serve as an *internal ribosome entry site* (IRES) to initiate cap-independent translation of the long ORF [12, 48].

RNA Structure Motifs

Our analysis of *Pestivirus* RNA is based on 10 complete virus genomes, representing all available *Pestivirus* genomes in data banks, see table 11. Multiple sequences alignment was performed by `Ralign`, the length of alignment is 12709 bases and the mean pairwise homology is 71.2%.

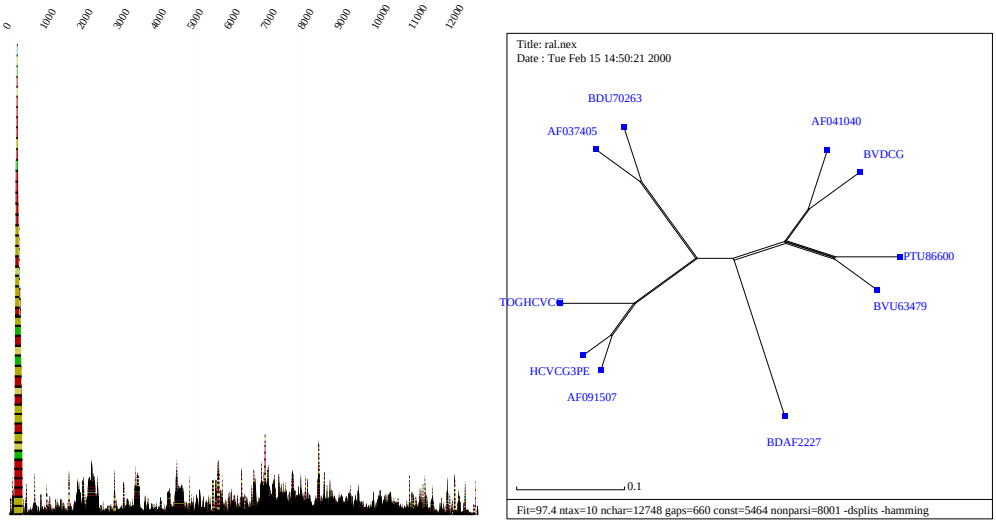


Figure 46: Mountain and sequence distance plot of aligned *Pestivirus* genomes. The Left side shows the mountain plot of all selected *Pestivirus* genomes. The IRES motif can be detected as an peak at the utmost left side, no other conserved motif is predicted. Right side, `SplitsTree` representation of the sequence distances after the multiple alignment.

The entire *Pestivirus* genome show only a single conserved RNA motif. This RNA motif is part of the already known IRES structure, situated at the 5' NCR of the virus genome.

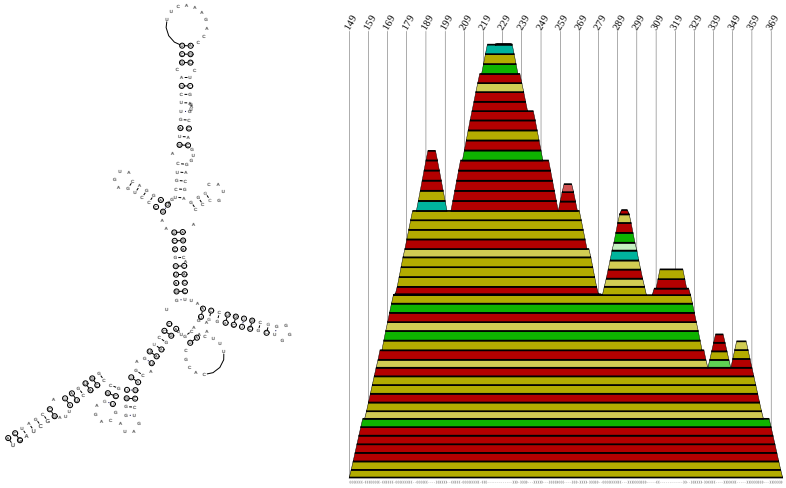


Figure 47: Detected conserved secondary structures of all *Pestivirus* sequences. Only one conserved RNA structure could be detected, the IRES motif situated in the 5' NCR region of the virus genome [4].

5' Non Coding Region (5'NCR) of Pestivirus

The 5' terminal sequence of *Classical swine fever virus* (CSFV) *Bovine viral diarrhea virus* (BVDV) and *Border disease virus* (BDV) is about 374nt long. Translation of the genome is cap independent, an IRES, located inside the 5' non coding region is responsible for RNA translation.

The pestiviral 5' NCR is highly conserved structurally, despite substantial differences in the primary nucleotide sequence. A structure model of this region was proposed by Brown [4]. Brown examined the phylogenetically-related 5'NCR sequences of BVDV for the presence of covariant nucleotide substitutions predictive of conserved, base paired helical RNA structures, the model is based on thermodynamic and phylogenetic considerations.

Brown reported four structural motifs I-IV inside the 5'NCR region. He detected a large complex structure consisting of a long irregular helix with multiple branching stem-loops labeled domain III (nucleotides 142 -358), see figure 48.

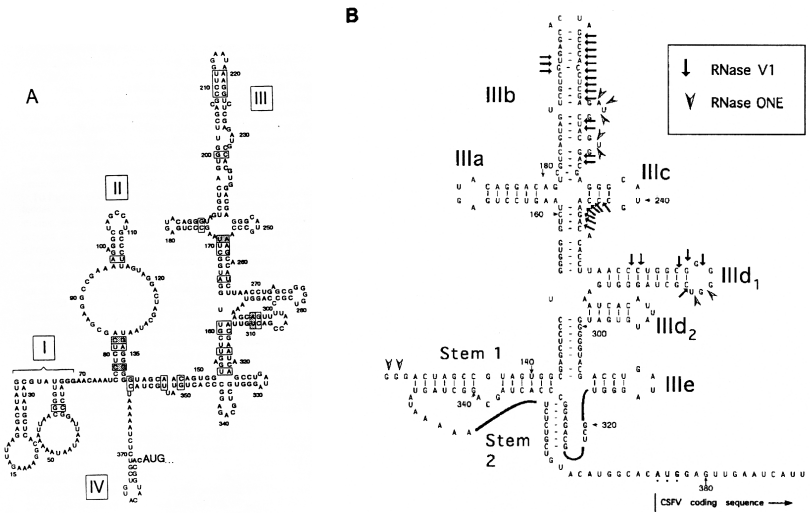
The IRES plays an important function in assembly of the ribosome. A specific binding of a translation initiation factor to the 5'NCR of *classical swine fever*

virus (CSVF) was reported by Sizova [61]. The translation factor *eIF3* binds strongly and specifically to the apical region of domain III of the CSFV IRES. The binding site consists of a large clover-leaf-like structure composed of the central helix of domain III and hairpins IIIa-IIIc. These observations led to propose a model for IRES function in which these large RNA contains a specific binding site for incoming 40S subunits and factors associated with them in 43S preinitiation complexes and structural elements that orient these binding sites in such a way that their interaction with components of the 43S complex correctly places the initiation codon of the mRNA at or in the immediate vicinity of the ribosomal P site.

Our predicted consensus structure of 10 selected sequences contains only part of these already described modules. Stem-loops at the very start of the virus genome labeled I can not be detected at all. The sequence alignment introduced a lot of gaps, so no base pairs are predicted at the very 5' terminus of the sequences. Part of module II is found in the consensus structure.

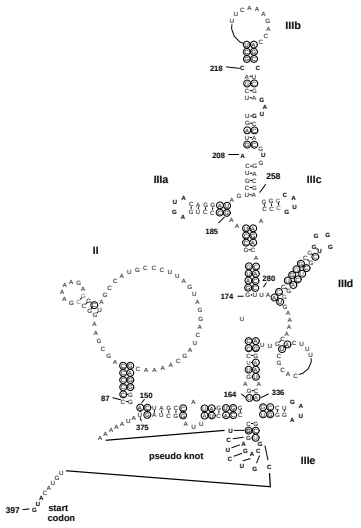
The most structural conserved module is labeled III, a huge stem loop structure. The predicted consensus structure nearly match in its entirety. The importance of the given secondary structure for *Pestivirus* is underlined by numerous compensatory mutations inside module III. The mean sequence homology of the 5'NCR region is 77.7% and a total amount of 165 conserved bases is found.

The 5'NCR of *Pestivirus* gives a good example where very different sequences fold nearly into the same secondary structure, which is a hint of functional importance of this region. The hairpin loops of IIIa, IIIc, IIId and IIIe are conserved compared to the proposed models of Brown and Sizova, but hairpin IIIb is different to the others, see figure 48. Specific nucleotides in this hairpin loops are probably of importance and specific nucleotides in hairpin IIIb are of less importance. A pseudo knot structure is shown in the model of Sizova. Our thermodynamic folding algorithm do not contribute to such base pair interaction, to complete our proposed model this interaction is also shown. The secondary structure of stem-loop IIIe is stabilized by a consistent and compensatory mutation, which implies that this base pairs should be established although the pseudo knot interaction. The thermodynamic stability of the secondary structure model is (-60.99 kcal/mol).



48.1 Brown 1992

48.2 Sizova 1998



48.3 5'NCR Consensus Struct

Figure 48: Proposed secondary structure models of *Pestivirus* IRES. The upper left model is designed by Brown 1992 [4]. Upper right is a slightly different model proposed by Sizova 1998 [61]. Arrows in the figure denote to cleavage sites for RNase V1 (ds specific) and RNase ONE (ss specific). The model at the bottom is the consensus structure of our thermodynamic prediction. The shown pseudo-knot was not predicted by our algorithms. Circles around base pairs denote consistent or compensatory mutations. The domain III is best conserved and also predicted in the other models.

Comparison of BVDV, CSFV and Border Disease Virus IRES

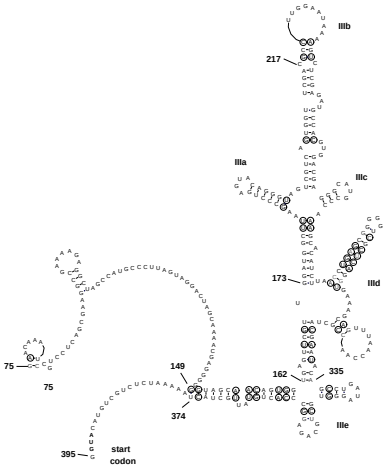
The genus *Pestivirus* includes 3 different subspecies infecting bovine (*Bovine diarrhea virus*), sheep (*Border disease virus*) and swine (*Classical swine fever virus*). All of them translate their genome cap independently using an IRES structure. Figure 49 compares the IRES region of all three subspecies.

The selected conserved secondary structure is module III. This module is commonly predicted. Although pairing nucleotides are rather diverse unpaired loop regions are highly conserved, see stem-loops IIIa,IIIc,IIId and IIIe. The unpaired region of stem-loop IIIb is different in all different viruses, also differences in part of stem-loop IIId occur. All three viruses show nearly the same secondary structure, although sequences are not homolog at all.

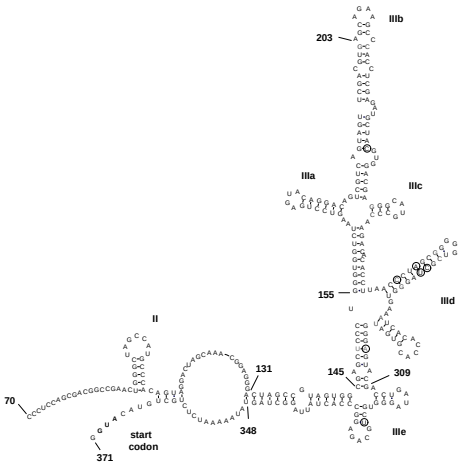
Bovine diarrhea virus, 8 different sequences are selected, see table 11. The mean pairwise homology of the selected region is -75.6% and 201nt are conserved. The calculated minimum free energy of the consensus structure is -77.16 kcal/mol.

For hog cholera virus 13 sequences, see table 11, are aligned the mean pairwise homology is 94.9%. The number of consistent and compensatory mutations is reduced to the other and the minimum free energy is -96.06 kcal/mol. *Border disease virus*, three genomes are aligned , see table 11, and the mean pairwise homology of the region is 87.3%, and the minimum free energy is -78.09 kcal/mol, only three different sequences show a lot of consistent and compensatory mutations.

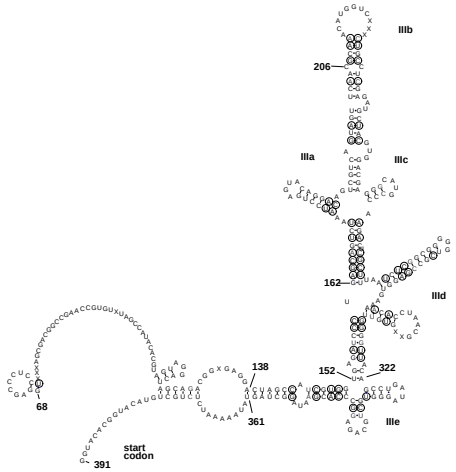
The main structural differences among *Pestivirus* are located inside stem-loop IIId, where *bovine diarrhea virus* forms a shorter stem-loop to *Hog cholera* and *border disease virus*, additionally *Bovine diarrhea virus* does not contain a base paired region upstream to module III, which is well predicted in both other viruses.



49.1 Bovine diarraeha IRES, cattle



49.2 Hog Cholera IRES, swine



49.3 Border disease IRES, sheep

Figure 49: Consensus structure models for all three *Pestivirus* species. Domain III is well predicted in all three species. *Pestivirus* in cattle show a slightly different structure to sheep and swine at the the start position of the domain III. Circles around bases denote consistent and compensatory mutations.

Discussion

The results of our investigation of the *Pestivirus* genomes show one highly conserved RNA motif at the 5' end of the non coding region. This element well known as an *internal ribosomal entry site* (IRES) show a common secondary structure over all three *Pestivirus* species. Sequences of the analyzed genomes are rather diverse on the sequence level, the overall mean pairwise homology was about 71% and about 78% for the consensus structure of the selected IRES region, see figure 48. Note, that about 10% difference in the nucleic acid sequence leads almost surely to unrelated structures if the mutated sequence positions are chosen randomly.

Over the rest of the *Pestivirus* genome no further conserved RNA motifs can be detected, that could be caused by the rather diverse sequences, note a bad alignment with a lot of gaps reduce the amount of RNA motifs significantly.

Investigations by Pestova and Sizova show the function of ribosome assembly initiated by the IRES and specific interaction of a translation factor to the IRES region [61, 56]. The Stem-loops IIIa,c,d,e, are important in this respect. It is remarkable, that the unpaired hairpin loops of these stem-loops contain the same nucleotides, see figure 48, although the stacking sequences are rather diverse. Maybe the conservation of the bases is important for ribosome assembly.

Pestivirus can either infect cattle, swine or sheep and cause severe diseases. The comparison of the IRES region of single groups may cause different RNA structures. The consensus structures are shown in figure 49 and at the first glance they are nearly identical. *Pestivirus* in swine and sheep show a difference at the beginning of the overall conserved structure where they form an interior loop and a small stem-loop , which was not found in cattle.

The 5'NCR of the genus *Pestivirus* show identical IRES structure. Conserved RNA secondary structures could be important to be compatible to other hosts. Cross-infection can occur among *Pestivirus*, it was reported for Bovine diarraeha this virus can also infect sheep and swine. The structural similarities of IRES implies that viruses from sheep and swine could also use other hosts.

4 Discussion

4.1 Conclusions

In this work, techniques for detecting conserved RNA secondary structures have been improved and extended. Investigation of entire virus genomes of the largest sequence lengths are now possible and only limited by available computer resources. The analyzed virus genera show conserved RNA secondary structures, which are detected by our theoretical approach using sequence data alone. Moreover it is fast and conserved RNA elements provide a qualitative description of virus genomes.

The combination of alignment, structure prediction and comparative sequence alignment uses a parallel version of McCaskill's partition function algorithm. It allows us to search to viral genome lengths of some 13000nt. We preferred the base pairing probability matrix, because it provides a better description of RNA molecules, thus this approach can be preferred to pure minimum free energy structure investigations.

In view of restricted computer resources an alternative algorithm to `Alidot` was developed. Minimum free energy folding needs less computer resources than McCaskill's algorithm and structure prediction is faster. The new algorithm `Alifold` computes common RNA secondary structures on a sample of aligned sequences based on the minimum free energy algorithm. It is even faster and more user friendly and common RNA structures are provided by a one step process.

Analysis of rather short RNA molecules can not be done without a practicable viewing tool. Especially large virus genomes of several thousand nucleotides overwhelm the investigator with data. Therefore we decided to develop a graphical viewing tool called `Vienna RNA Viewer`, that presents RNA secondary structure more user friendly. Even unskillful users can easily handle our viewer, and are able to select conserved RNA elements by hand, or use a lot of filter functions. A semi-automatically analysis of RNA sequences can be done, and with the help of the viewer a list of conserved RNA elements can be selected quickly.

First we applied our approach to analyzing the virus family *Bunyaviridae*. For the genera *Bunyavirus* and *Hantavirus* we found enough sequences in databanks. *Bunyaviridae* are well known to form a panhandle structure, be-

cause terminal sequences of the tripartite genomes are complementary. This structural element was the best detected in the virus family. *Hantavirus* genomes show some additional RNA elements but they are predicted with less quality than the panhandle. Moreover the panhandle seems to be the most important structural feature in the family *Bunyaviridae*. The second virus family we analyzed was *Flaviviridae*. Two genera are analyzed *Hepatitis C virus* and *Pestivirus*. *Hepatitis C virus* shows a large number of conserved RNA elements. Our results are compared to phylogenetically and experimentally known RNA secondary structures. Especially the non coding regions are compared to literature, and the well known *internal ribosome entry site* (IRES) could be verified by other works. Differences could be detected to proposed structure models of the non coding region, and the most promising new element was discussed in detail. We detected this element inside the coding region of the virus genome, besides many other interesting RNA elements.

Pestivirus show only one conserved RNA element inside the non coding region. This RNA element is an IRES and it is similar to one found in *Hepatitis C virus*. Numerous consistent and compensatory mutations are presented. *Pestivirus* genomes are isolated form cattle, swine and sheep and aligned sequences are rather diverse, maybe this is a reason that no further elements could be detected. A comparison of the IRES of cattle, swine and sheep show no crucial structural differences. *Pestivirus* isolated from cattle can also infect other hosts, a common conserved IRES structure could therefore be critical for cross-infection to occur.

4.2 Outlook

Conserved RNA secondary structures establish a new method for describing functional important regions of viral genomes. As long the three-dimensional structure is not available by theoretical approaches alone, they will be an essential description. Improving this approach is therefore desirable.

A too diverse sample of sequences is still a major problem of our method. Although sequence alignments have been improved, alignment errors often significantly decrease the quality of our results. At present this problem is solved by selecting sequences by hand. This method leaves it up to the investigator which sequences to select. Often numerous virus genomes can

be arranged in multiple groups, that makes the analysis more laborious, if all groups are analyzed.

We used the sorting procedures called `Alidot` or `Pfrali`, which have been optimized for a rather small sample of sequences. Therefor analysis of numerous sequences is problematic. On the other hand it is desirable to use many sequences, since the quality of predicted conserved elements increases. We are confident of solving the problem if more different virus groups have been analyzed.

Selecting conserved RNA elements by hand is laborious and subjective. At present different researchers set individually the threshold, so different conserved RNA elements are selected. However the best RNA elements are selected commonly, but the quality of the other elements can differ substantially. On this account results from different investigators could be hardly compared, because they contain individually selected RNA elements. More objective results are crucial for comparison and to extended our approach to all available virus genera. Common search criteria are also essential to automate our method, though automation makes our method faster and more user friendly. We plan two steps towards automation. First the development of selection criteria for conserved RNA secondary structures, where sequence selection is still done by hand. The second one is a fully automated approach, so only a sample of related sequences is provided. Conserved RNA elements are predicted fully automatic.

Automation is in many respects desirable, but analysis by hand will be important in some cases. Although a practical viewing tool has been developed in this work, some additional filter functions are still desirable to improve the search for conserved RNA elements.

The immediate objective of this approach is to analyze all available RNA virus groups. At all a global overview of conserved RNA secondary structures is necessary to improve our approach, beyond a qualitative description of viral genomes is still a demand, because of improving our knowledge of viruses generally. The analysis of all RNA virus genera should lead to a data bank based on conserved RNA elements, which should be public. Such a data bank will be a useful tool to guide deletion studies for research.

At present taxonomy for sequences of viral genomes is an unsolved problem. There is no consensus on how to group different virus genomes and often viruses with completely different coding strategies are grouped within one

virus family. Family *Flaviviridae* include single stranded viral genomes with either cap dependent translation (*Flavivirus*) and cap independent translation *Hepatitis C virus* or *Pestivirus*. Conserved RNA elements can help to group different virus genomes, because common RNA secondary structures can guide taxonomy for phylogenetically related virus genomes. Investigations of the viral phylogeny can be based on conserved RNA elements, and unknown viruses can be assigned to virus families by comparing their secondary structures to already analyzed conserved RNA elements.

Our theoretical approach show that conserved elements are crucial for the viral life-cycle by definition. Even the high mutation rate of virus genomes can not destroy these elements, which makes them to ideal targets for new anti-viral strategies.

Sequences

Table 10: *Hepatitis C Virus* Sequences

Selected Sequences				
REM	ID	Accession No	length (nt)	organism
1	E08399	E08399	9413	Hepatitis C virus
2	E10035	E10035	9416	Hepatitis C virus
3	HCJRNA	D14484	9427	Hepatitis C virus
4	HCU01214	U01214	9446	Hepatitis C virus
5	HCVJK1G	X61596	9408	Hepatitis C virus
6	HCVPOLYP	AJ000009	9379	Hepatitis C virus
7	HPC1B4	D50484	9410	Hepatitis C virus
8	HPC1B5	D50485	9410	Hepatitis C virus
9	HPCCGENOM	L02836	9400	Hepatitis C virus
10	HPCGENANT	M84754	9425	Hepatitis C virus
11	HPCY1B6	D50480	9410	Hepatitis C virus
12	S62220	S62220	9440	Hepatitis C virus
Example Sequences for Alifold				
REM	ID	Accession No	length (nt)	organism
1	AF009606	AF009606	9646	Hepatitis C virus
2	AF054247	AF054247	9595	Hepatitis C virus
3	D84262	D84262	9449	Hepatitis C virus
4	D84263	D84263	9426	Hepatitis C virus
5	HC45476	U45476	9431	Hepatitis C virus
6	HCJK046E2	D63822	9461	Hepatitis C virus
7	HCV4APOLY	Y11604	9355	Hepatitis C virus type 4a
8	HCVJK1G	X61596	9408	Hepatitis C virus
9	HPCHK6	D28917	9454	Hepatitis C virus
10	HPCPOLP	D00944	9589	Hepatitis C virus
Hepatitis C virus strain H77C				
REM	ID	Accession No	length (nt)	organism
1	AF011751	AF011751	9599	Hepatitis C virus strain H77
2	AF011752	AF011752	9599	Hepatitis C virus strain H77
3	AF011753	AF011753	9599	Hepatitis C virus strain H77

Table 11: *Pestivirus* Sequences

Selected *Pestivirus* Sequenes				
REM	ID	Accession No	length (nt)	organism
1	AF037405	AF037405	12333	
2	AF041040	AF041040	12260	pestivirus type 1
3	AF091507	AF091507	12310	Hog cholera virus
4	BDAF2227	AF002227	12255	
5	BDU70263	U70263	12268	pestivirus type 3
6	BVDCG	M31182	12573	pestivirus type 1
7	BVU63479	U63479	12247	pestivirus type 1
8	HCVCG3PE	M31768	12283	Hog cholera virus
9	PTU86600	U86600	12267	pestivirus type 1
10	TOGHCVCG	J04358	12284	Hog cholera virus
Selected *Pestivirus* Sequenes for Comparison Cattle, Swine and Sheep				
REM	ID	Accession No	length (nt)	organism
cattle				
1	AF041040	AF041040	12260	pestivirus type 1
2	AF091605	AF091605	12310	bovine viral diarrhea virus
3	BV18059	U18059	12513	pestivirus type 1
4	BVDCG	M31182	12573	pestivirus type 1
5	BVDPOLYPR	M96751	12308	pestivirus type 1
6	BVDPP	M96687	12480	pestivirus type 1
7	BVU63479	U63479	12247	pestivirus type 1
8	E01149	E01149	12492	pestivirus type 1
swine				
1	A16790	A16790	12284	Hog cholera virus
2	AF091507	AF091507	12310	Hog cholera virus
3	AF091661	AF091661	12297	Hog cholera virus
4	HC45477	U45477	12298	Hog cholera virus
5	HC45478	U45478	12278	Hog cholera virus
6	HCSEQB	L49347	12144	Hog cholera virus
7	HCVCG3PE	M31768	12283	Hog cholera virus
8	HCVCOMGEN	X87939	12298	Hog cholera virus
9	HCVCOMSEQ	X96550	12297	Hog cholera virus
10	HCVPOLYP1	D49532	12298	Hog cholera virus
11	HCVPOLYP2	D49533	12298	Hog cholera virus
12	HCVPOLYPR	Z46258	12311	Hog cholera virus
13	TOGHCVCG	J04358	12284	Hog cholera virus
sheep				
1	AF037405	AF037405	12333	border disease virus
2	BDAF2227	AF002227	12255	border disease virus
3	BDU70263	U70263	12268	pestivirus type 3

Table 12: *Hantavirus* L,M-segment Sequences

REM	ID	Accession No	length (nt)	organism
L-segment				
1	BUHANL	X55901	6533	Hantaan virus
2	HANRDRP1	D25528	6533	Hantaan virus
3	HANRDRP4	D25531	6533	Hantaan virus
4	HVSLSEG	X56492	6530	Hantavirus
5	NEVLRNA	M63194	6550	Puumala virus
6	PVLSOTKMO	Z66548	6550	Puumala virus
7	SNVRPL	L37901	6562	Sin Nombre hantavirus
8	SNVRPLA	L37902	6562	Sin Nombre hantavirus
M-segment				
1	AF028022	AF028022	3653	Lechiguanas virus
2	AF028023	AF028023	3654	Hu39694 virus
3	AF028024	AF028024	3646	Oran virus
4	AF030551	AF030551	3664	Blue River virus
5	AF030552	AF030552	3662	Blue River virus
6	BUHANM	Y00386	3616	Hantaan virus
7	HANG1G2A	L08753	3616	Hantaan virus
8	HOJM	D00376	3613	HoJo virus
9	HPSCC107M	L33474	3696	Pulmonary syndrome
10	HPSMSEG	L25783	3696	Sin Nombre 0
11	HPSMSEGA	L33684	3696	Pulmonary syndrome
12	HPSMSEGB	L33685	3644	Hantavirus
13	HVIGLYPRE	L36930	3677	Bayou hantavirus
14	LNAF5728	AF005728	3698	Laguna Negra virus
15	NEVMSEG	M29979	3682	Puumala virus
16	NY36801	U36801	3668	New York hantavirus
17	PHVMSRNA	X55129	3707	Prospect Hill virus
18	PUVMVIN83	Z49214	3682	Puumala virus
19	PV22418	U22418	3681	Puumala virus
20	PVMZ84205	Z84205	3682	Puumala virus
21	S68035	S68035	3655	Hantavirus
22	SNGPGO	L37903	3696	Sin Nombre hantavirus
23	TIDG1G2A	L08756	3613	Thailand virus
24	TUVM5302	Z69993	3694	Tula virus

Table 13: *Hantavirus* S-segment Sequences

REM	ID	Accession No	length (nt)	organism
1	AB010730	AB010730	1833	Puumala virus
2	AF004660	AF004660	1876	Andes virus
3	HANSNC	M14626	1696	Hantaan virus
4	HSNPSS	L41916	1670	Hantavirus sp.
5	HVINUCPRO	L36929	1958	Bayou hantavirus
6	KH35255	U35255	1845	Khabarovsk hantavirus
7	LNAF5727	AF005727	1904	Laguna Negra virus
8	PRHSRNA	M34011	1675	Prospect Hill virus
9	PUUSNP	X61035	1830	Puumala virus
10	PUVSVIRRT	Z69985	1837	Puumala virus
11	PV22423	U22423	1847	Puumala virus
12	PVNICAS	U14137	1828	Puumala virus
13	PVNPRO1	Z30702	1832	Puumala virus
14	RMU11427	U11427	1896	El Moro Canyon hantavirus
15	RS18100	U18100	1749	Mexicanus hantavirus
16	SRVAGSS	M34881	1769	Sapporo rat virus
17	TUVS5302	Z69991	1831	Tula virus
18	TVSSEG1	Z30941	1847	Tula virus
19	U19303	U19303	1722	Prairie vole hantavirus
20	U52136	U52136	1975	Rio Mamore hantavirus

Table 14: *Hantavirus* M,S-segments of Groups

REM	ID	Accession No	organism
M-segment			
group 1	BUHANM	Y00386	Hantaan virus
group 1	HANG1G2	M14627	Hantaan virus
group 1	HOJM	D00376	HoJo virus
group 1	HPSMSEG	L25783	Sin Nombre hantavirus
group 1	S68035	S68035	Hantavirus
group 1	TIDG1G2A	L08756	Thailand virus
group 2	NEVMSEG	M29979	Puumala virus
group 2	PHVMSRNA	X55129	Prospect Hill virus
group 2	PUVMVIN83	Z49214	Puumala virus
group 2	PV22418	U22418	Puumala virus
group 2	PVMZ84205	Z84205	Puumala virus
group 2	TUVM5302	Z69993	Tula virus
group 3	AF028022	AF028022	Lechiguanas virus
group 3	AF028023	AF028023	Hu39694 virus
group 3	AF028024	AF028024	Oran virus
group 3	HVIGLYPRE	L36930	Bayou hantavirus
group 3	LNAF5728	AF005728	Laguna Negra virus
group 4	AF030551	AF030551	Blue River virus
group 4	AF030552	AF030552	Blue River virus
group 4	HPSCC107M	L33474	Pulmonary syndrome
group 4	HPSMSEG	L25783	Sin Nombre hantavirus
group 4	HPSMSEGA	L33684	Pulmonary syndrome
group 4	NY36801	U36801	New York hantavirus
S-segment			
group 1	AF004660	AF004660	Andes virus
group 1	HVINUCPRO	L36929	Bayou hantavirus
group 1	LNAF5727	AF005727	Laguna Negra virus
group 1	RMU11427	U11427	El Moro Canyon hantavirus
group 1	RS18100	U18100	Reithrodontomys mexicanus hantavirus
group 1	U52136	U52136	Rio Mamore hantavirus
group 2	KH35255	U35255	Khabarovsk hantavirus
group 2	PRHSRNA	M34011	Prospect Hill virus
group 2	TUVS5302	Z69991	Tula virus
group 2	TVSSEG1	Z30941	Tula virus
group 2	U19303	U19303	Prairie vole hantavirus
group 3	AB010730	AB010730	Puumala virus
group 3	PUUSNP	X61035	Puumala virus
group 3	PUVSVIRRT	Z69985	Puumala virus
group 3	PV22423	U22423	Puumala virus
group 3	PVNICAS	U14137	Puumala virus
group 3	PVNPRO1	Z30702	Puumala virus

Table 15: *Bunyavirus* Sequences

REM	ID	Accession No	length (nt)	organism
M-segment				
1	BUSSHMG	K02539	4527	Snowshoe hare virus
2	LACMRP	D10370	4526	La Crosse virus
3	LC18979	U18979	4526	La Crosse virus
4	LCU70207	U70207	4526	La Crosse virus
5	U88057	U88057	4501	Melao virus
6	U88058	U88058	4510	Jamestown Canyon virus
7	U88059	U88059	4506	Inkoo virus
8	U88060	U88060	4506	Inkoo virus
S-segment				
1	BUNCNP	K00108	981	La Crosse virus
2	BUSVR	J02390	982	Snowshoe hare virus
3	CE12800	U12800	978	California encephalitis virus
4	MB31989	U31989	976	Morro Bay virus
5	SAU47139	U47139	976	San Angelo virus
6	SDU47140	U47140	967	Serra do Navio virus
7	SRU47141	U47141	984	South River virus
8	TV12803	U12803	973	Trivittatus virus
9	TVU47142	U47142	976	Tahyna virus

List of Figures

List of Tables

References

[1] N. Ali and A. Siddiqui. Interaction of polypyrimidine tract-binding protein with the 5' noncoding region of hepatitis c virus RNA genome and its functional requirement in internal initiation of translation. *J. Virology*, 1995.

[2] G. J. Barton and M. J. E. Sternberg. A strategy for the rapid multiple alignment of protein sequences. confidence levels from tertiary structure comparisons. *J. Mol. Biol.*, 198:327–337, 1987.

[3] D. Bashford, C. Chothia, and A. M. Lesk. Determinants of a protein fold. unique features of the globin amino acid sequences. *J. Mol. Biol.*, 196:199–216, 1987.

[4] E. A. Brown, H. Zhang, L.-H. Ping, and S. M. Lemon. Secondary structure of the 5' nontranslated regions of hepatitis C virus and pestivirus genomic RNAs. *Nucl. Acids Res.*, 20:5041–5045, 1992.

[5] V. E. Chizhkov, C. F. Spiropoulou, S. P. Morzunov, C. J. Peters M. C. Monroe, and S. T. Nichol. Complete geneitic characterization and analysis of isolation of sin nombre virus. *Journal of Virology*, 1995.

[6] J. Corodkin, L. J. Heyer, and G. D. Stormo. Finding common sequences and structure motifs in a set of RNA molecules. In T. Gaasterland, P. Karp, K. Karplus, Ch. Ouzounis, Ch. Sander, and A. Valencia, editors, *Proceedings of the ISMB-97*, pages 120–123, Menlo Park, CA, 1997. AAAI Press.

[7] E. Domingo, D. Sabo, T. Taniguchi, and C. Weissmann. Nucleotide sequence heterogeneity of an RNA phage population. *Cell*, 1978.

[8] Z. Fan, Q. R. Yang, J. Twu, and A. H. Sherker. Specific in vitro association between the hepatitis c viral genome and core protein. *Journal of Medical Virology*, 1999.

[9] M. Fekete. RNA secondary structure prediction using parallel computers. Master's thesis, Faculty of Sciences, University of Vienna, Austria, 1997.

[10] M. Fekete, I. L. Hofacker, and P. F. Stadler. Prediction of RNA base pairing probabilities using massively parallel computers. *J. Comp. Biol.*, 2000.

[11] D. F. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, 25:351–360, 1987.

[12] B. N. Fields, D. M. Knipe, P. M. Howley, R. M. Chanock, J. L. Melnick, T. P. Monath, B. Roizmann, and S. E. Straus, editors. *Fields Virology.* Lippincott, 3rd edition, 1996.

[13] W. M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155:279–284, 1967.

[14] C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster. RNA folding at elementary step resolution. *RNA*, 2000.

[15] D. Garcin, M. Lezzi, M. Dobbs, R. M. Elliot, C. Schmaljohn, C. Yong Kang, and D. Kolakovsky. The 5' ends of hantaan virus (bunyaviridae) RNAs suggest a prime-and realign mecchanism for the initiation of RNA synthesis. *Journal of Virology*, 1995.

[16] W. B. Goad and M. I. Kanehisa. Pattern recognition in nucleic acid sequences. A general method for finding local homologies and symmetries. *Nucl. Acids Res.*, 10:247–263, 1982.

[17] W. Grüner, R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster. Analysis of RNA sequence structure maps by exhaustive enumeration. II. Structures of neutral networks and shape space covering. *Monath. Chem.*, 127:375–389, 1996.

[18] A. P. Gultyaev, F. H. D. vanBatenburg, and C. W. A. Pleij. The computer simulation of RNA folding pathways using a genetic algorithm. *J. Mol. Biol.*, 250:37–51, 1995.

[19] R. R. Gutell. Evolutionary characteristics of RNA: Inferring higher-order structure from patterns of sequence variation. *Curr. Opin. Struct. Biol*, 3:313–322, 1993.

[20] P.G. Higgs. The influence of RNA secondary structure on the rates of substitution in RNA-encoding genes. Preprint, Univ. Manchester, 1998.

[21] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Comm. Assoc. Comp. Mach.*, 18:341–343, 1975.

[22] I. L. Hofacker, M. Fekete, C. Flamm, M. A. Huynen, S. Rauscher, P. E. Stolorz, and P. F. Stadler. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucl. Acids Res*, 26:3825–3863, 1998.

[23] I. L. Hofacker, M. Fekete, C. Flamm, M. A. Huynen, S. Rauscher, P. E. Stolorz, and P. F. Stadler. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. *Nucl. Acids Res.*, 26:3825–3836, 1998.

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

[25] I. L. Hofacker, W. Fontana, P. F. Stadler, and P. Schuster. `Vienna RNA Package`.
`ftp://ftp.itc.univie.ac.at/pub/RNA/`
`http://www.tbi.univie.ac.at/~ivo/RNA/`,
1994. (Free Software).

[26] I. L. Hofacker, M. A. Huynen, P. F. Stadler, and P. E. Stolorz. Knowledge discovery in RNA sequence families of HIV using scalable computers. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR*, pages 20–25, Portland, OR, 1996. AAAI Press.

[27] I. L. Hofacker, M. A. Huynen, P. F. Stadler, and P. E. Stolorz. RNA folding and parallel computers: The minimum free energy structures of complete HIV genomes. Technical report, SFI, Santa Fe, New Mexico, 1996. # 95-10-089.

[28] I. L. Hofacker and P. F. Stadler. Automatic detection of conserved base pairing patterns in RNA virus genomes. *Comp. & Chem.*, 23:401–414, 1999.

[29] Ivo L. Hofacker and Peter F. Stadler. Automatic detection of conserved base pairing patterns in rna virus genome. *Comp & Chem.*, 1999.

[30] J. J. Holland, J.C. De La Torre, and D.A. Steinhauer. RNA virus populations as quasispecies. *Curr Top Microbiol Immunol*, 1992.

[31] M. Honda, Li-Hua Ping M. R. Beard, and S. M. Lemon. A phylogenetically conserved stem-loop structure at the 5' border of the internal ribosome entry site of hepatitis c virus is required for cap-independent viral translation. *Journal of Virology*, 1999.

[32] M. Honda, Li-Hua Ping, R. C. A. Rijnbrand, E. Amphlett, B. Clarke, D. Rowlands, and S. M. Lemon. Structural requirements for initiation of translation by internal ribosome entry whithin genome-length hepatitis c virus RNA. *Virology*, 1996.

[33] J. W. Hunt and M. D. McIlroy. An algorithm for differential file comparison. Technical Report Comp. Sci. 41, Bell Laboratories, 1976.

[34] D.H. Huson. Splitstree: a program for analyzing and visualizing evolutionary data. *Bioinformatics*, 14(1):68–73, 1998.

[35] M. A. Huynen, R. Gutell, and D. A. M. Konings. Assessing the reliability of RNA folding using statistical mechanics. *J. Mol. Biol.*, 267:1104–1112, 1997.

[36] Patterson JL, Kolakofsky D, Holloway BP, and Obijeski JF. Isolation of the ends of LaCrosse virus small RNA as a double-stranded structure. *J Virol*, 1983.

[37] M. I. Kanehisa and W. B. Goad. Pattern recognition in nucleic acid sequences. An efficient method for finding locally stable secondary structures. *Nucl. Acids Res.*, 10:265–277, 1982.

[38] A. Kolykhalov, S. Feinstone, and C. M. Rice. Identification of a highly conserved sequence element at the 3'terminus of hepatitis C virus genome RNA. *J. Virology*, 70:3363–3371, 1996.

[39] D. Konings and R. Gutell. A comparison of thermodynamic foldings with comparatively derived structures of 16S and 16S-like rRNAs. *RNA*, 1:559–574, 1995.

[40] M. Kunze and G. Thierrin. Maximal common subsequences of pairs of strings. *Congr. Num.*, 34:299–311, 1982.

[41] J. Leydold and P. F. Stadler. Minimal cycle basis, outerplanar graphs. *Elec. J. Comb.*, 5:R16, 1998. See `http://www.combinatorics.org`.

[42] D. J. Lipman, S. F. Altschul, and J. D. Kececioglu. A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA*, 86:4412–4415, 1989.

[43] R. Lück, G. Steger, and D. Riesner. Thermodynamic prediction of conserved secondary structure: Application to the RRE element of HIV, the tRNA-like element of CMV, and the mRNA of prion protein. *J. Mol. Biol.*, 258:813–826, 1996.

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

[45] D. L. Mills, editor. *A new algorithm to determine the Levenshtein distance between two strings*, Conference on Sequence Comparison. University of Montreal, 1978.

[46] A. A. Mironov, L. P. Dyakonova, and A. E. Kister. A kinetic approach to the prediction of RNA secondary structures. *Journal of Biomolecular Structure and Dynamics*, 2:953, 1985.

[47] Collet Ms, Purchio AF, Keegan K, and et al. Complete nucleotide sequence of the M RNA segment of Rift Valley fever virus. *Virology*, 1985.

[48] F. A. Murphy, C. M. Fauquet, D. H. L. Bishop, S. A. Ghabrial, A. W. Jarvis, G. P. Martelli, M. A. Mayo, and M. D. Summers, editors. *Virus Taxonomy*. Springer-Verlag, 6th edition, 1995.

[49] E. W. Myers and W. Miller. Optimal alignments in linear space. *CABIOS*, 4:11–17, 1988.

[50] P. Vialat N. Paradigon, M. Girard, and M. Bouloy. Panhandles and hairpin structures at the termini of germiston virus RNAs (bunyavirus). *Virology*, 122:191–197, 1982.

[51] S. B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.*, 48:443–453, 1970.

[52] J. M. Norman. *Elementary dynamic programming.* Crane, Russak and Co., New York, 1975.

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

[54] R. E. O'Neil and P. Palese. Cis-acting signals and trans-acting factors involved in influenza virus RNA synthesis. *Infect. Agents Dis.*, 1994.

[55] A. K. Pattnaik, L. A. Ball, A. W. LeGrone, and G. W. Wertz. The termini of VSV DI particle RNAs are sufficient to signal RNA encapsidation, replication, and budding to generate infectious particles. *Virology*, 1995.

[56] T. V. Pestova, I. N. Shatsky, S. P. Fletcher, R. J. Jackson, and C. U.T. Hellen. A prokaryotic-like mode of cytoplasmic eukaryotic ribosome binding to the initiation codon during internal translation initiation of shepatitis c and classical swine fever virus RNAs. *Genes & Development*, 1998.

[57] C. Reidys, P. F. Stadler, and P. Schuster. Generic properties of combinatory maps: Neural networks of RNA secondary structures. *Bull. Math. Biol.*, 59:339–397, 1997.

[58] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.

[59] D. Sankoff. Simultaneous solution of the RNA folding, alignment, and proto-sequence problems. *SIAM J. Appl. Math.*, 45:810–825, 1985.

[60] P. Schuster, W. Fontana, P. F. Stadler, and I. L. Hofacker. From sequences to shapes and back: A case study in RNA secondary structures. *Proc. Royal Society London B*, 255:279–284, 1994.

[61] D. Sizova, V. G. K., T. Pestova, Ivan N. Shatsky, and C. U. T. Hellen. Specific interaction of eukaryotic translation initiation factor 3 with the 5' nontranslated regions of hepatitis c virus and classical swine fever virus RNAs. *J. of Virology*, 1998.

[62] D. A. Steinhauer and J.J. Holland. Rapid evolution of RNA viruses. *Annu. Rev. Microbiol*, 19987.

[63] R. Stocsits. Improved alignments based on a combination of amino acid and nucleic acid information. Master's thesis, Faculty of Sciences, University of Vienna, Austria, 1999.

[64] R. Stocsits, I. L. Hofacker, and P. F. Stadler. Conserved secondary structures in hepatitis B virus RNA. In *Computer Science in Biology*, pages 73–79, Bielefeld, D, 1999. Univ. Bielefeld. Proceedings of the GCB'99, Hannover, D.

[65] J. E. Tabaska and G. D. Stormo. Automated alignment of RNA sequences to pseudoknotted structures. In T. Gaasterland, P. Karp, K. Karplus, Ch. Ouzounis, Ch. Sander, and A. Valencia, editors, *Proceedings of the ISMB-97*, pages 311–318, Menlo Park, CA, 1997. AAAI Press.

[66] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignments through sequence weighting, position specific gap penalties and weight matrix choice. *Nucl. Acids Res.*, 22:4673–4680, 1994.

[67] J. D. Thompson, D. G. Higgins, and T. J. Gibson. Improved sensitivity of profile searches through the use of sequence weights and gap excision. *CABIOS*, 10:19–29, 1994.

[68] J. D. Thompson, D. G. Higgs, and T. J. Gibson. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties, and weight matrix choice. *Nucl. Acids Res.*, 22:4673–4680, 1994.

[69] L. S. Tiley, M. Hagen, J. T. Matthews, and M. Krystal. Sequence-specific binding of the influenza virus RNA polymerase to sequences located at the 5' ends of viral RNAs. *J. Virology*, 1994.

[70] M. Vingron and P. R. Sibbald. Weighting in sequence space: A comparison of methods in terms of generalized sequences. *Proc. Natl. Acad. Sci. USA*, 90:8777–8781, 1993.

[71] M. Vingron and M. S. Waterman. Sequence alignment and penalty choice. Review of concepts, case studies and implications. *J. Mol. Biol.*, 235:1–12, 1994.

[72] A. E. Walter, D. H. Turner, J. Kim, M. H. Lyttle, P. Müller, D. H. Mathews, and M. Zuker. Co-axial stacking of helixes enhances binding of oligoribonucleotides and improves predicions of RNA folding. *Proc. Natl. Acad. Sci. USA*, 91:9218–9222, 1994.

[73] M. S. Waterman. Secondary structure of single - stranded nucleic acids. *Adv. math. suppl. studies*, 1:167–212, 1978.

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

[75] E. Westhof and L. Jaeger. RNA pseudoknots. *Current Opinion Struct. Biol.*, 2:327–333, 1992.

[76] S. Wuchty, I. L. Hofacker W. Fontana, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 1998. in press, Santa Fe Institute Preprint 98-05-040.

[77] M. Yanagi, M. St. Claire, S. U. Emerson, R. H. Purcell, and J. Bukh. In vivo analysis of the 3' untranslated region of the hepatitis c virus after in vitro mutagenesis of an infectious cDNA clone. *PNAS*, 1999.

[78] M. Zuker. `mfold-2.3`. `ftp://snark.wustl.edu/`. (Free Software).

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

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

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

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

Curriculum vitae

Martin Fekete

*6. 8. 1970
Waidhofen/Thaya, NÖ.

Schulbildung

1976 – 1980	Volksschule in Heidenreichstein
1980 – 1981	Hauptschule in Heidenreichstein
1981 – 1989	Neusprachliches Gymnasium in Waidhofen/Thaya
Mai 1989	Reifeprüfung in Waidhofen/Thaya

Studium

1989 – 1997	Studium Chemie, Studienzweig Biochemie, an der Universität Wien
1998 – 2000	Doktorat Chemie, Universität Wien

Forschungsaufenthalt

Sept. 1999	Santa Fe Institute, New Mexico, USA