

Visual Data Mining of RNA Secondary Structure and Folding Pathways as Determined by the Massively Parallel Genetic Algorithm

Bruce A. Shapiro¹ and Wojciech Kasprzak²

¹Laboratory of Experimental and Computational Biology
NCI Center for Cancer Research
Building 469, Room 150
NCI-Frederick
Frederick, MD 21702
bshapiro@ncifcrf.gov

²Basic Research Program
SAIC Frederick, NCI-Frederick National Cancer Institute
Frederick, MD 21702 Building 469, Room 150
Frederick, MD 21702

Abstract

RNA folding pathways are proving to be quite important in the determination of RNA function. Studies indicate that RNA may enter intermediate and multiple conformational states that are key to its functionality. These states may have a significant impact on gene expression and molecular function. It is known that the biologically functional states of RNA molecules may not correspond to their minimum energy state, that kinetic barriers may exist that trap the molecule in a local minimum, that folding often occurs during transcription, and that cases exist in which a molecule will transition between one or more functional conformations. Thus, methods for simulating the folding pathway and dynamic behavior of an RNA molecule are important for the prediction of RNA structure and its associated functions.

We have developed several visual data mining techniques associated with a massively parallel genetic algorithm for RNA structure prediction, as well as with STRUCTURELAB, our RNA/DNA structure analysis workbench. These methodologies are used to determine the significant intermediate and final structures associated with RNA folding. Since the genetic algorithm is essentially stochastic, multiple runs are required. The visualization procedures used give significant feedback concerning the characteristics of the folding runs. This feedback encompasses: interpretation of results from individual genetic algorithm runs that are based on population consensus or best fit structures, this includes the discovery of transition states in the folding process; final results of individual runs; and the interpretation of genetic algorithm results from multiple RNA sequences from the same family to identify common structural elements across the different sequences. In addition, fitness maps as well as results derived from different population sizes are used.

The combination of the visualization techniques as well as other methodologies embedded within the STRUCTURELAB and genetic algorithm environments help to determine the overall picture representing the folding pathway or final structure(s) of a given RNA sequence. This paper will describe several of these techniques and show how they are used to help solve this very highly combinatoric problem.

Introduction

The bioinformatics revolution has led to a tremendous increase in the availability of data on gene location, expression, and function for thousands of species. Because of this vast quantity of data, time and resources are often lacking for in depth experimental analysis of genes and gene products. In the past proteins were the main focus of attention for detailed structural analysis. However, more recently RNA structural studies have become very important to the understanding of complex biological systems.

The number of ways that RNA can interact with its environment is quite extensive. Structure and structural transitions are important in post-transcriptional regulation of gene expression, intermolecular interactions, splice site recognition, and ribosomal frame-shifting to name a few contexts. Ribozymes, for example, constitute a class of RNA molecules whose sequence exists primarily to define their enzymatic properties. The RNA folding problem, i.e. the determination of RNA secondary structure and ultimately three-dimensional structure and function, is a significant area for the use of computational approaches. As with most such applications of high-performance computing, the problem of RNA structure determination is a difficult one. The number of possible secondary structures given a particular sequence varies on the order of 1.8^n for a sequence of n nucleotides. Approaches to the problem are numerous and varied. A wide range of biochemical and biophysical methods may be used to examine RNA secondary and tertiary structure. These methods generally search experimentally for the affect on sequence and structure within a molecule by probing for accessibility to enzymes, calculating optical absorbency, or by measuring the electrophoretic migration rate over a temperature gradient. A given structure generally is verified through phylogenetic analyses, searching among members of a family for compensatory base changes that would maintain base-pairedness in equivalent regions. In addition, the three-dimensional structure of these molecules may be elucidated by X-ray crystallography or nuclear magnetic resonance techniques.

All of this relatively direct data often is supported, or at times even replaced, by theoretical structure calculations. The most familiar variety of these are those that are derived from dynamic programming algorithms (DPA) such as MFOLD [1], which searches for a molecule's thermodynamically optimal structure, as well as a series of suboptimal structures. When the object is secondary structure, that is, a structure that can be defined as a list of base-paired and single-stranded regions (stems and loops), thermodynamic calculations are straightforward. Stems tend to stabilize a structure and most loops tend to destabilize it, and the energies of these stems and loops are additive. Thus, a search for biologically relevant structures is driven by the assumption that a molecule will tend to fold spontaneously into structures that minimize its global Gibbs free energy with respect to the unstructured state. A version of the dynamic programming approach to energy minimization has been able to include H-type pseudoknots and some basic tertiary structure energy contributions at the cost of making the algorithm run in $O(n^6)$ time [2]. By removing pseudoknot predictions and shifting the coaxial stacking energy calculations for multibranch loops to a post-processing reordering phase, this algorithm runs in $O(n^3)$ time [3].

Searching experimentally and theoretically for these equilibrium structures, either optimal or suboptimal, however, is often insufficient. The

biologically functional state of a given molecule may not be the optimal state. The issue then is how does one determine the relevant suboptimal structures? A structured RNA molecule, moreover, is not a static object. A molecule may pass through several active and inactive states over its lifetime, due to the kinetics of folding, to the simultaneity of folding with transcription, i.e. sequence elongation, or to interactions with its environment. A molecule may become trapped in a local energy minimum with a high-energy barrier to surmount before reaching a more stable state.

We have developed methods using a massively parallel Genetic Algorithm (GA) optimization approach that have proven highly amenable to the exploration of RNA secondary structure folding pathways [4-10]. This algorithm was designed using the same basic considerations as the dynamic programming algorithm; that is, with thermodynamic calculations to optimize the global free energy of an RNA molecule. It is reasonably successful at finding optimal or near-optimal equilibrium structures, including pseudoknots, given a particular sequence. The properties of this massively parallel, iterated, stochastic algorithm, however, have also been shown to be well suited to the problem of predicting the dynamic folding process of a given molecule. In addition, the algorithm allows for the incorporation of some types of experimental data, allowing it both to verify and to predict the outcome of experiments under known conditions. The Genetic Algorithm operates on a population of thousands of possible solution structures, evolving them toward thermodynamic fitness. It may be run multiple times and in multiple phases. STRUCTURELAB, an interactive RNA structure analysis workbench, has proven indispensable in analyzing the large quantities of data generated by the GA [14-15]. STRUCTURELAB used in combination with a set of GA visualizers is used to datamine the GA's output.

What is an RNA Secondary Structure?

An RNA sequence can be viewed as a string derived from an alphabet consisting of the letters {A, G, C, U}. These letters represent respectively the nucleic acid bases, Adenine, Guanine, Cytosine and Uracil. The bases tend to form Watson-Crick base pairs, i.e. G-C and A-U and a wobble base pair G-U that are stabilized by hydrogen bonds and base stacking interactions. Other base pairs are possible, but for the sake of brevity will not be brought into the current discussion.

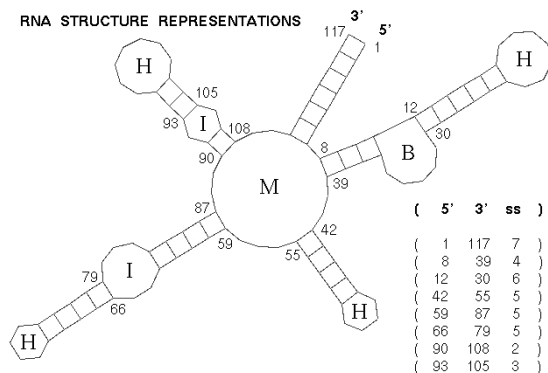


Figure 1. A typical RNA secondary structure indicating the various morphologies that are commonly present. B – bulge loop, H – hairpin loop, I – internal loop and M – multibranch loop. Also illustrated is the stem table that defines the topology for the given structure.

The interaction of these bases will form different types of motifs, which are depicted in Figure 1. Each crosshatched line represents a base pair, while other positions represent bases that are not normally paired and thus form loop regions. Loop regions are labeled according to their motif type. M, I, B, and H stand for multibranch loop, internal loop, bulge loop and hairpin loop respectively. The region table, which represents the topology of the structure, is also shown. The region table represents individual helical stems sorted based on their 5' positions. Thus, the first stem, represented by the triple (1, 117, 7) is a stem whose 5' position is 1, its 3' position is 117 and its size is 7.

Each loop motif and hydrogen bonded stem contributes to the free energy of the secondary structure (the more negative the free energy the more stable the structure). For the most part, loop regions tend to destabilize the structure (more positive free energy), while base paired regions tend to stabilize the structure (more negative free energy). Energy rules, with different degrees of context sensitivity, are used to compute the free energy of a structure. Given a sequence of size n , it is estimated that there are 1.8^n possible secondary structures. Thus, the number of possible structures is quite significant for even relatively small sequences, e.g. 600-700 bases.

The Massively Parallel Genetic Algorithm

A massively parallel genetic algorithm was developed to predict RNA secondary structures from a given a sequence. The genetic algorithm borrows from the biological concepts of evolution and the survival of the fittest [11-13]. In its current incarnation, the algorithm is capable of running on several different computer architectures and operating systems. It was originally designed to run on a massively parallel 16384 processor SIMD (single instruction multiple data) MasPar [4-7]. It was then adapted to a MIMD (multiple instruction multiple data) architecture, i.e. an SGI ORIGIN and Cray T3E, using the SHMEM libraries [8-10]. Most recently, it has been ported to LINUX clusters using MPI-2.

The algorithm uses a mesh type of representation for the population, where a population element is defined to be a "maturing" RNA secondary structure. Eight neighbors surround each population element (north, south, east, west, northeast, northwest, southeast and southwest (see Figure 2). Two parents are chosen from the nine possibilities based upon a biased free energy directed selection criterion. Two children are created from these two parents by randomly mutating in stems from the stem pool (the set of all possible fully complementary stems that can be generated from the given sequence) and by recombining stems from the two parent structures. The child with the best fitness (lowest free energy) is chosen to replace the population element in the center of the eight-neighbor region. This operation can be thought of as occurring in parallel on all possible 3×3 eight neighbor toroidally wrapped regions, thus producing, with a 16K population, 16K new population elements. The algorithm iterates over several generations until a convergence criterion is satisfied which measures the relative stability of the population as a whole. Stability of the population is induced by an annealing mutation operator, which gradually reduces the number of mutations of ensuing generations.

The genetic algorithm is normally run with several different population configurations each varying by powers of two. Thus, typical runs might consist of

2K, 4K, 8K, 16K, 32K, 64K and sometimes 128K populations. Each population size is normally run 20 times to develop a consensus. A given population size can in turn be run with a power of two number of physical processors. The algorithm scales almost linearly with the number of physical processors. Thus, for example, doubling the number of physical processors will improve the speed by about a factor of 2.

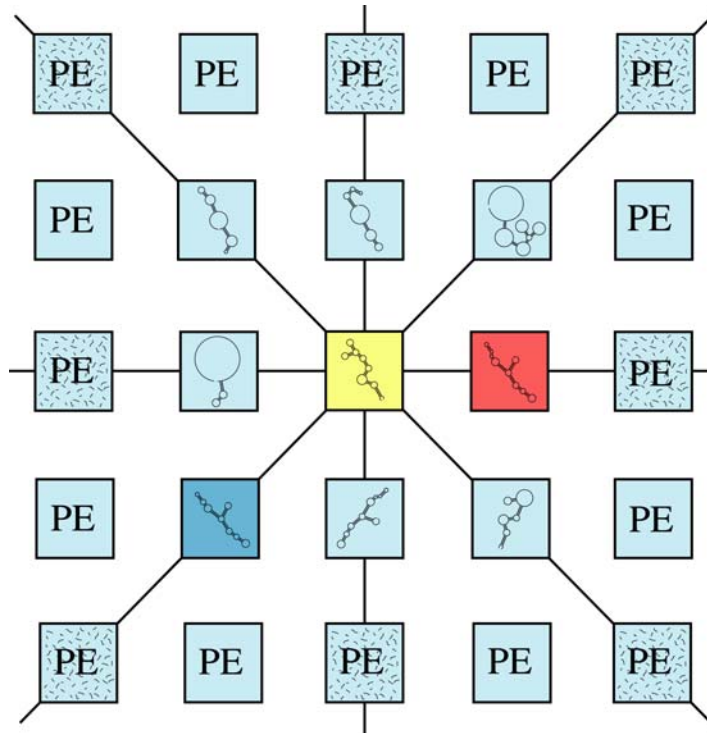


Figure 2. Representation of the population element layout showing a small window, which for example might represent a portion of a 16K population. The red and blue boxes represent chosen parents and the yellow box represents the placement for a new structure in a 3x3 neighborhood.

Population size variation has the interesting property of capturing RNA secondary structures that are representative of functional intermediates. That is, as an RNA is folding, it will sometimes form intermediate conformations that are themselves functional or are important for the folding pathway of the RNA for reaching its final state. Typically, at lower populations, the algorithm will converge to solutions, which are indicative of these functional intermediates. At higher populations, the structures will pass through these intermediates on their way to possibly more fit functional intermediates or their final state.

Many other features exist within the algorithm ranging from sequential folding to the biased use of certain motifs; the use of various sets of energy parameters, including the run-time determination of EFN2 coaxial stacking calculations; and conflict driven peelback. In addition, H-type pseudoknots, the interaction of a hairpin loop and a free base region, can be calculated. A description of some of these can be found in [5-10]. An important issue to be resolved and which is the main focus of this paper is the analysis of the significant amounts of data that are produced by the running of the algorithm. Both visual and

statistical means are used to approach this problem, most of which STRUCTURLAB, our RNA/DNA structure analysis workbench handles.

Visualization of Genetic Algorithm Runtime Population Dynamics

The Fitness, Trace and Pseudoknot Maps

Three different $n \times m$ two-dimensional dynamic graphical population maps can be generated at each generation (or chosen increment) and viewed as the genetic algorithm is running. The values of n and m are determined by the power-of-two population size. Thus, a 128×128 square region would represent a 16K population, while a 64×128 rectangular region would represent an 8K population, etc. Each pixel in a map represents the contents of a particular population element. By pointing at a particular pixel on one of the maps, a region table representing the corresponding structure will be displayed.

Fitness Map

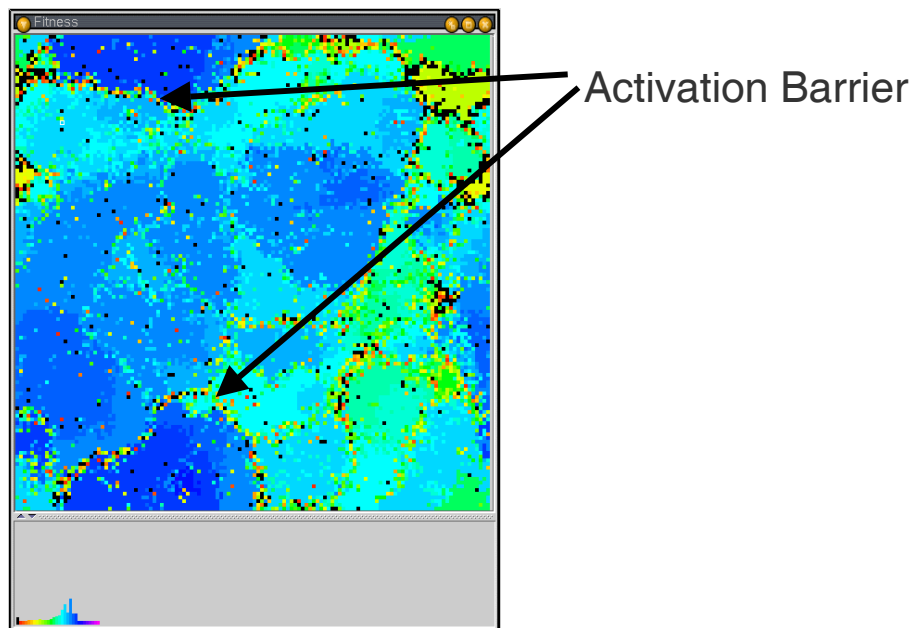


Figure 3. Fitness map representing the later stages of a GA run. Each pixel of this 16K population (128×128) is color-coded based upon the fitness value of the structure in the population element. The rugged region pointed to is representative of an “activation barrier” indicating that the structure in the areas on each side of the barrier are quite different and that the barrier represents structures that are undergoing unfolding to allow for a transition.

In the fitness map, each pixel is color-coded based on the fitness of the structure evolving in the corresponding element. Red, at one end of the color spectrum indicates poor fitness (high free energy) while purple, at the other end of the color spectrum, indicates good fitness (low free energy). Interesting characteristics of the folding landscape are sometimes discernable from the fitness map. Sometimes population clusters, represented by a uniform color, may be

surrounded by a lower fitness region outside of which is another population cluster, which has lower fitness than the original cluster but is higher than the boundary fitness (see Figure 3). This is representative of an “activation” barrier, i.e., the RNA has to unfold somewhat to transition from the higher free energy state to the lower free energy state. Other times, the transition is smoother without the existence of the intermediate barrier, indicating that relatively minor transitions are probably taking place.

Trace Map

The trace map, which has a one-to-one positional correspondence with the fitness map, allows one to follow the formation and disappearance of helical stems that one suspects or knows exists a priori. Each individual occurrence of a stem in a structure is color-coded. If more than one stem is present from the a priori list, then the pixel is shaded gray. Its gray value is determined by the percentage of stems found in a structure from the a priori list. If all the stems from the list appear, then the pixel is coded white. Thus, the trace map allows one to understand the dynamic formation of individual stems as the algorithm is running and to determine visually when many population elements acquire the depicted stems. Figure 4 illustrates the trace map following the propagation of stems for known motives in a given sequence. The illustrated trace map is derived from the same generation as the fitness map shown in Figure 3. There is a 1-1 correspondence between the pixels in the fitness map and those in the trace map.

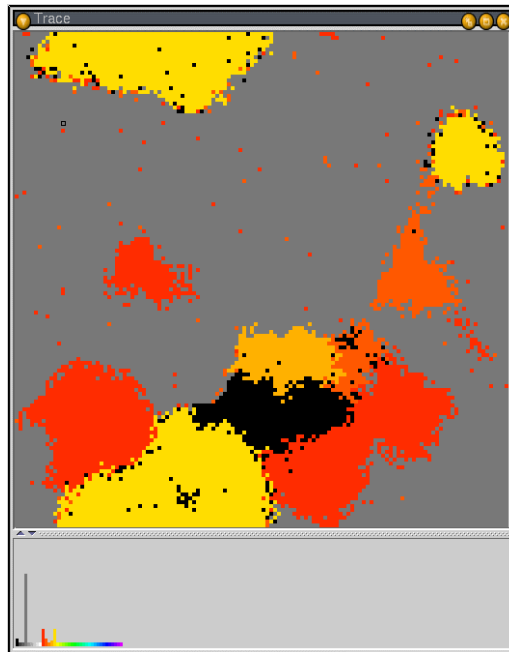


Figure 4. Trace map of a 16K population GA run at the same generation that is shown in the fitness map in Figure 3. The pixel positions in each map correspond to each other. The map allows one to follow the occurrence or disappearance of individual predetermined stems in the population. The stems are each color-coded when they appear in isolation in a structure. If several stems (but not all) appear in a structure a gray value is used to represent the number of stems that do occur. White pixels depict the occurrence of all followed stems in the population element.

Pseudoknot Map

Like the other maps, this follows the occurrence of pseudoknots in individual structures. The color-coding here indicates the number of pseudoknots that are predicted in any given structure. Because of the pixel correspondences that exist between the maps, one could, for example, follow the formation of a previously known or expected pseudoknot by visually inspecting the pseudoknot map and the trace map together. One can use the mouse pointer to point at a pixel in any of the maps to extract the corresponding underlying data.

Population Histogram

As the genetic algorithm is running it is possible to activate the display of a population histogram, which for each generation (or chosen increment) will display a color-coded histogram representing the fitness, trace or pseudoknot distributions for the entire population. The fitness and trace histogram distributions are visible in Figures 3 and 4 respectively. As an example, when the genetic algorithm starts most of the population, as expected, is distributed in the poorer fitness region. As the algorithm proceeds one can see the gradual redistribution of the histogram showing more and more of the population having improving fitness.

At times, the population histogram will indicate the presence of multiple significant peaks. This has been correlated with the existence of competing populations of structures. In the case of the HIV-1, we have shown that two metastable conformational states appear to be possible where both of the conformations have similar energies. These results are consistent with experimental studies indicating that the HIV-1 leader sequence can form two different functional conformations [16-19].

Data Generated by the Genetic Algorithm

The genetic algorithm is capable of generating several different types of data files, which can later be used for analyzing results. Setting various parameters in the input to the genetic algorithm can control the type and number of files generated. The types of files are enumerated below:

- 1) Solution files – As the GA is running, for each generation (or specified stride) a file is generated which represents the best current structure in the population or the current consensus structure for the generation. Considering that there may be as many as 700-to 800 generations per run (this depends on the sequence size) there may be as many as 16,000 files generated if one captures all the structures for each generation. To reduce the amount of data output, it is also possible to only generate results at the end of a run. In this case only the final best or consensus structure will be produced.
- 2) Pseudoknot files – these files are associated with each solution file (see 1 above). They indicate those stems that will form pseudoknots.
- 3) Efn2 files – these files are associated with each solution file (see 1 above). They indicate the coaxial stacking present (if any) between stems in multibranch loops. These files are generated only if the enf2 energy rule set is used for the folding.

- 4) Solution files representing the top “n” structures in the population – this permits the evaluation of several terminating structures at the end of a run.
- 5) Histogram file representing the energy distribution in the population – this file will contain the population counts for each energy level found in a given population.
- 6) Max, Min and Avg files – these files contain data on the minimum, maximum and average energies for each generation of each run.
- 7) Motif and Stemtable – these files can be generated after initialization and contain all the stems that are used by the genetic algorithm’s operators (the alphabet). The motif table specifically contains those stems that makeup additional constructs that are considered to be motifs. This currently includes coaxially stacked stems. These tables may be modified and preloaded into the GA for experimental purposes bypassing the initialization procedures that originally generated these files.
- 8) Trace –a file containing for each generation the number of occurrences of a pre-specified stem or group of stems in the population is produced. This file contains information that is similar in concept to the visual trace map mentioned above.

Visual Data Mining with STRUCTURELAB

STRUCTURELAB [14] is an extensive RNA/DNA structure analysis workbench, which controls the activation and analysis of results produced by various algorithms running on a multitude of computer platforms with different operating systems. It functions in a client/server mode where, in some cases a tight coupling exists between the running program and STRUCTURELAB, while in other cases the program may be run like a batch job where STRUCTURELAB can do other things while waiting for the batch job to complete. The remainder of this paper will focus on how STRUCTURELAB can be used to analyze the large amounts of data that are produced by the massively parallel genetic algorithm for RNA structure prediction. A significant amount of other functionality exists within STRUCTURELAB, but will not be discussed here (See Figure 5).

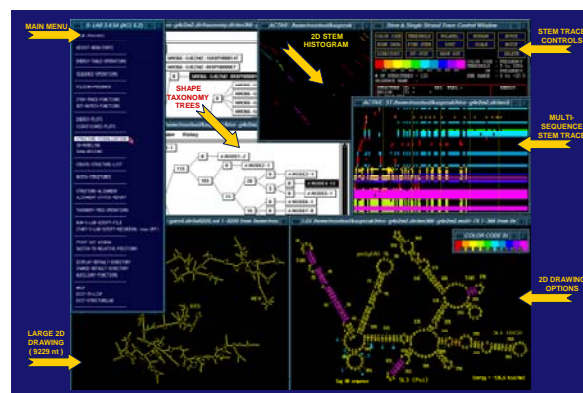


Figure 5. Depiction of some of the capabilities of STRUCTURELAB. Depicted in a clockwise fashion is the main starting menu, RNA secondary structure cluster trees, dot-matrix histogram, Stem Trace control window, a Stem Trace, secondary structure drawing and annotation and the ability to draw very complex structures.

Stem Trace

Stem Trace [15] is a multifaceted visualization interface that is part of STRUCTURELAB. It permits the depiction of results from GA runs that may be obtained from various sources. Several of its capabilities are enumerated below:

- 1) Following the maturation process of structures from single runs of the GA.
- 2) Depiction of the intermediate structures from all runs of the GA for a given sequence. This permits the visualization of the occurrence of common stem formation across multiple runs.
- 3) Depiction of the final structures that are generated by the GA. These structures may indicate those that are representative of the majority of the population of structures or those that are representative of those that are the best final structures in the population.
- 4) Depiction of the final structures generated by the GA from several different sequences that comprise a family. A multiple sequence alignment is required in this case to present the stems of the structures in their proper positions.
- 5) Depiction of population variation runs. This permits the portrayal, for example, of populations that are 2K, 4K, 8K, 16K, 32K, 64K, etc., all in one presentation. This allows structural comparisons for detecting metastable states that may transition to more stable states over several generations.

A Stem Trace is defined as a 2-dimensional graph. Each position along the X-axis represents a generation from the GA. Each position along the Y-axis represents a unique stem from a secondary structure. Thus, the set of points intersected by a vertical line in the graph represents the stems that determine a secondary structure. The ordering of the stems along the Y-axis can be altered depending on the mode of visualization desired. For example, raw output from a GA run shows the maturation of a structure over several generations. The lower generation numbers show few points (stems) while higher generations, usually contain more points indicating more mature complex structures. Thus, stems that have a high propensity for formation will usually appear early in the graph.

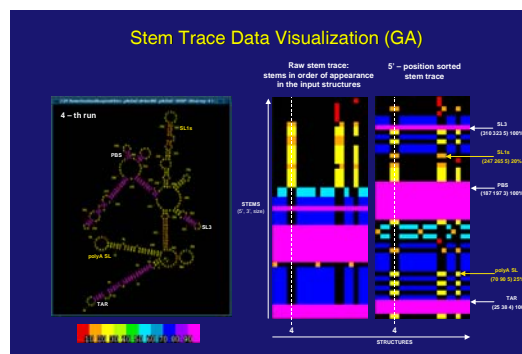


Figure 6. Example Stem Trace and structure. The two right-hand panels represent the same Stem Trace but one is unsorted (order of occurrence of stems) and the other is 5' sorted respectively. Structure 4 is indicated by the vertical dotted line and is also shown in the secondary structure drawing. Occurrence frequency of stems is color-coded (red means low occurrence and purple is high occurrence).

The stems in the Stem Trace plot can also be sorted based upon their 5' positions. This presentation has the advantage of depicting stems that are physically close together as opposed to a more dispersed presentation generated based on order of appearance. Figure 6 illustrates the final consensus structures for the unsorted and sorted Stem Traces. Also shown is a color-coded drawing of structure 4.

In all cases stems are color-coded based upon frequency of occurrence, red being low frequency (less than 10%) and purple being high frequency (greater than 90%). Sets of structures can also be placed in bins (delineated by light vertical lines) when showing either population variation results (each bin is a different population) or multiple sequence family results (each bin is a the results from a given sequence).

As mentioned above, Stem Trace could be used for comparison of the results from variable population runs. Shown in Figure 7a, is combined presentation of 2K, 4K, 8K, 16K, 32K and 64K runs for the control region of HIV-1 MN. Illustrated here is the transition from a “branched” conformation, at lower populations, to a more “linear” conformation at higher populations [16-19]. However, certain stems are maintained in both conformations (blue and purple). This type of representation is very useful for the detection of metastable states as is the case that is presented in this example. Figure 7b shows the two dominant conformations (branched and linear) with the enclosed dotted regions indicating constant motifs.

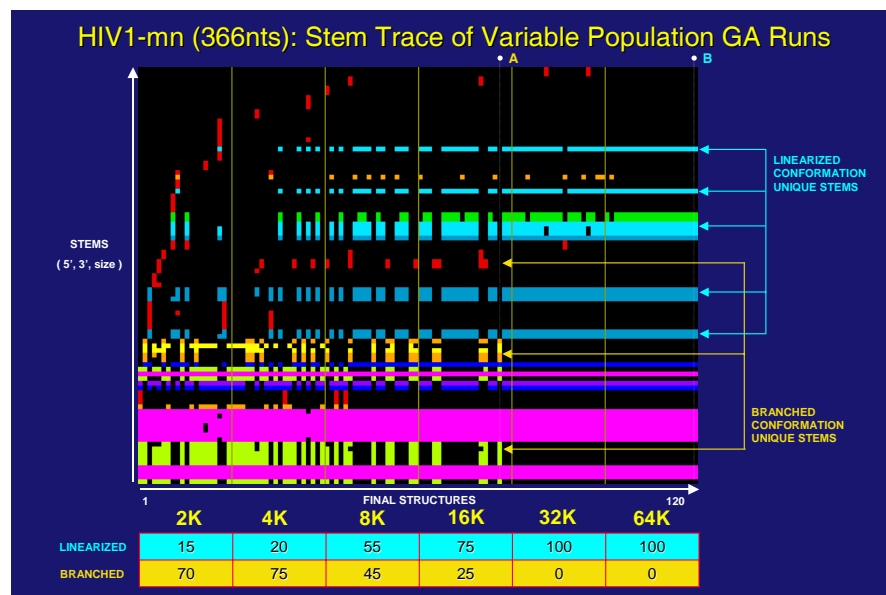


Figure 7a. Example of merged Stem Traces from multi-population GA runs. Each block represents 20 runs of the GA for the control region of HIV-MN for populations ranging from 2K to 64K. Structural transitions are observable with increasing population sizes. Indicated are the percentage of branched vs. linear structures that occur at different population sizes.

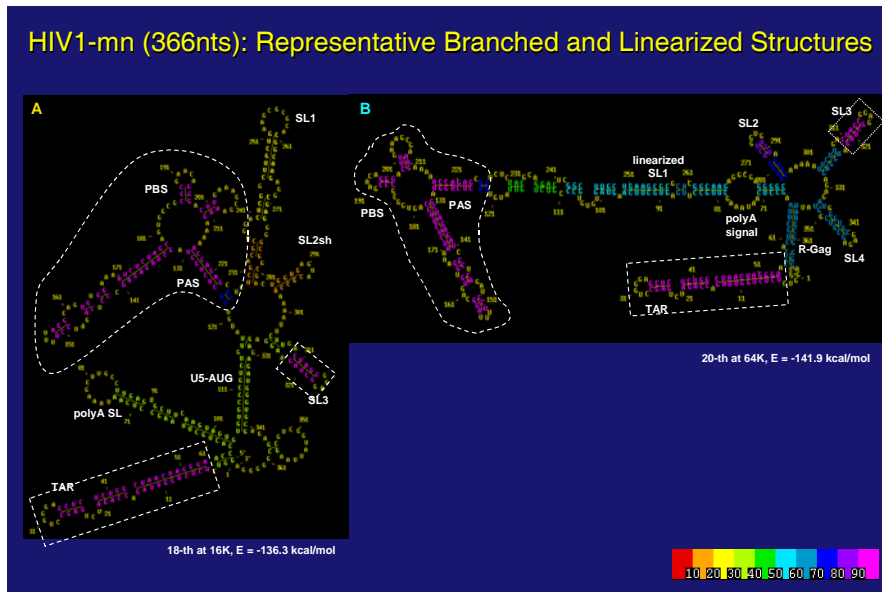


Figure 7b. The dominant branched and linear structures that occur as a result of the variable population runs shown in Figure 7a. The enclosed dotted regions represent the “scaffold” motifs that do not undergo any structural changes.

Stem Trace Control Window

The stem trace control window permits the activation of several functions and the display of information related to Stem Trace. A mouse pointer may be moved over the Stem Trace plot causing specific information associated with individual stems pointed to, to be displayed. Several other functions can be activated from buttons associated with the window. Some are described below:

- 1) *Draw/Label* - if the draw button is pressed and then the middle mouse button is pressed on a given structure in the plot, the secondary structure drawing for that structure will be shown. It is also possible to point at individual stems in the plot to have them color-coded in the structure drawing, or the entire structure can be color-coded based on the selected structure. The color-coding of the structure drawing therefore allows one to associate the frequency of occurrence of stems in a structure based upon the results obtained from the genetic algorithm. Figure 6 shows a typical Stem Trace plot with a selected color-coded drawn structure.
- 2) *Movie* - A movie consisting of the structures present within the Stem Trace can be produced. Each structure is shown in sequence. A fixed stride can also be defined. If the Stem Trace represents a single run from the genetic algorithm, the movie will portray the developing structure from its immature state to its final state. On the other hand if the Stem Trace depicts the final results of several runs or the final results of a population variation run, the variability of the final results will be portrayed. The structures can be drawn in a standard form or as circle diagrams.
- 3) *Threshold* – The Stem Traces can be thresholded, reducing the data clutter by indicating only those stems that have a frequency of occurrence that lies within a threshold range. When the draw option is selected with the thresholded Stem

Trace, only those stems that survive the thresholding will be shown in the drawn structure (Figure 10 illustrates a similar functionality for the Stem Histogram operator).

- 4) *Mine Data* – This option, which contains several sub-options, permits the determination and the examination of occurrences of the unique structures in a Stem Trace. Thus for example, in a single genetic algorithm run, one can determine the number of different unique structures that form and their occurrence rate over their development. This type of data is useful for determining the existence of folding pathway states within a run.
- 5) *Motif* - Motifs can be constructed that can then be used to search for sequences that can be folded into the motifs that are represented. This is similar to the RNAMOT program [] except that the motifs can be generated relatively easily and dynamically. The topology of the structure is built by selecting individual stems present in the Stem Trace plot with a mouse. This topology can also be generated with a fuzzy characteristic allowing stems or single stranded regions to be of varying sizes.
- 6) *Single Strand Stem Trace* – A complementary view of the Stem Trace may be depicted which shows the single stranded regions occurrence rate. In some sense this is the opposite of the regular Stem Trace, which depicts the occurrence rate of helical base paired regions. This is especially useful for depiction of stable loop regions from the GA solutions that may have associated stems that are somewhat variable in length. Figure 8 illustrates the single strand stem trace being used for a comparison between two different strains of HIV-1 (mn and lai) for the “linear” structure. A threshold of greater than 50% is used to eliminate some noisy loops. Sequence alignment procedures are used whenever more than one sequence is being compared to adjust for insertions and deletions.

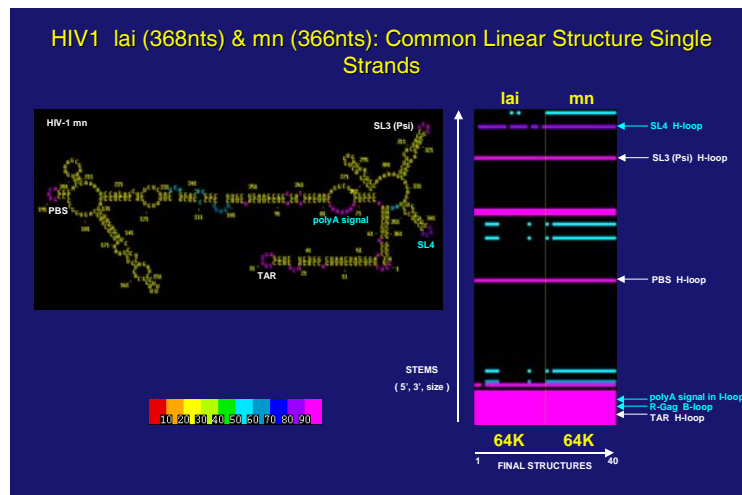


Figure 8. Illustration of Single Strand Stem Trace applied to the linear HIV-MN and HIV-LAI structures. 64K population results are shown for 20 runs of each sequence. The results have been filtered for those single-stranded regions that occur more than 50% of the time. Merging the stem traces for both sequences permits a structural comparison for the conservation, in this case, of single-stranded regions. Appropriate color-coding is shown in the associated secondary structure drawing.

- 7) *Fuzzy Stem Trace* – When depicting the results from multiple runs of the genetic algorithm whether it is with multiple sequences from a family of sequences, or multiple results from one sequence, the Stem Trace may have to be adjusted to accommodate for slight differences in stem positions. Thus, the fuzzy Stem Trace allows the user to pick individual stems in the Stem Trace, similar to the way motifs are built (see above) and the Fuzzy Stem Trace will search the space of stem solutions for those that differ by a position of 1 in their start or stop position and, in addition, differ in size from the selected set of stems. In this way, stems that are not automatically aligned in the Stem Trace can be re-aligned.

Stem Histogram

There are times when one would like to view a database of secondary structures in a manner that is essentially orthogonal to that which is available with Stem Trace. A Stem Histogram is a two-dimensional plot of possible base pair interactions from such a database. The plot is constructed by assuming that a given base sequence is represented along the X-axis. The reverse complement of the sequence is represented along the Y-axis. Thus, every potential base pair can be represented by a dot at any position in the matrix where there is a matching base. To reduce the complexity of such plots, usually a threshold of at least 2 consecutive base pairs is needed to permit the presentation of the base pairs. Thus, a Stem Histogram consists of many possible diagonal runs representing the formation of helical stems. A database of possible structures can be loaded into a Stem Histogram where, if several structures have the same stems, they are color-coded indicating the frequency of occurrence of such stems. Again purple represents a high frequency of occurrence and red represents a low frequency of occurrence. Because some stems may not start at exactly the same position or may not be of the same length, some diagonal runs may be somewhat offset from each other showing multiple colors within what appears to be one diagonal (see Figure 9).

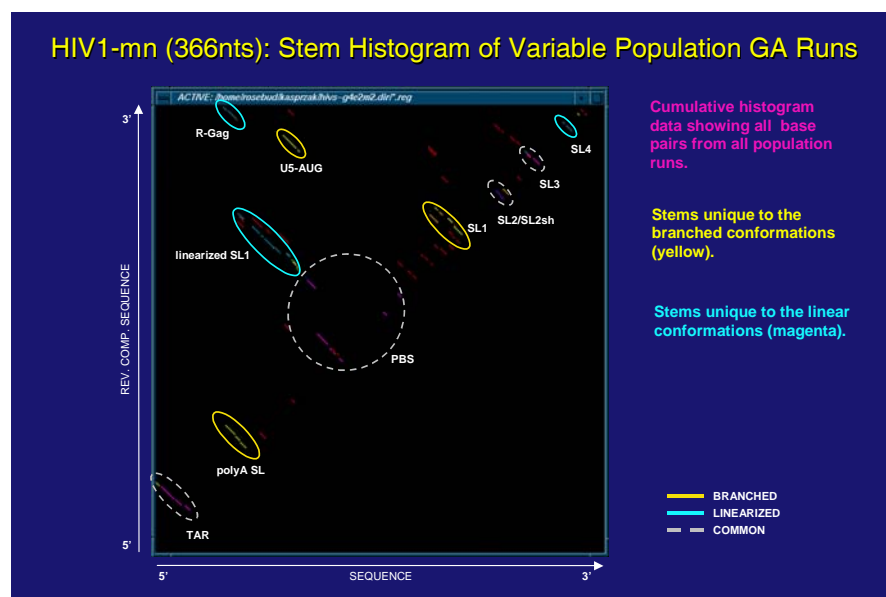


Figure 9. Stem Histogram of all populations runs of HIV –1 MN from the GA. Those stems that are constant as well as those stems that form the mutually exclusive branched and linear structures are circled.

A threshold may be applied to the Stem Histogram permitting the survival of those stems whose frequency of occurrence in the database of structures surpasses the given threshold. This has an additional benefit if the threshold chosen is above 50%. In this case a structure may be drawn from all the surviving stems since under these circumstances all the stems are compatible and therefore non-conflicting. This can be quite useful for depicting structural elements derived from a genetic algorithm run where fluctuations may occur in some portions of a conformation but other portions remain constant. This is illustrated in Figure 10 depicting two conformational states of the HIV-1 MN non-coding region where a “scaffold” remains constant but a portion of the structure is variable.

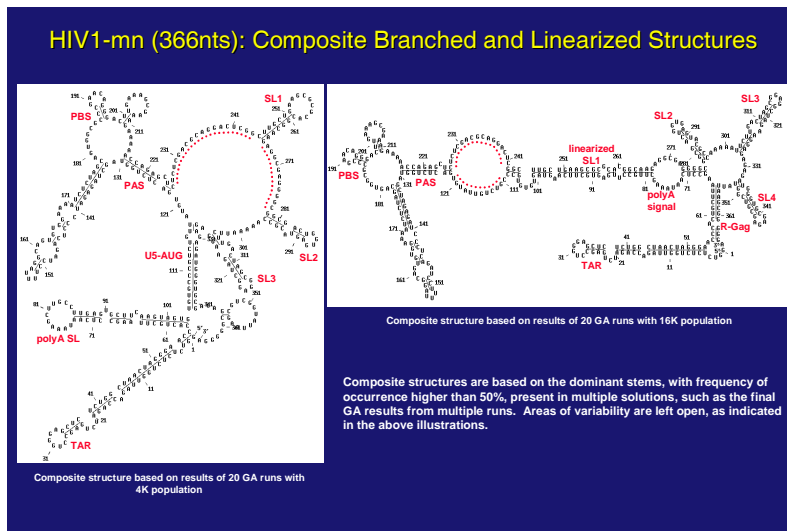


Figure 10. Illustration of the composite structure capability of STRUCTURELAB. Here the two dominant structures are shown (branched and linear) that contain stems that occur more than 50% of the time for the 4K population run and for the 16K population run. These two drawings capture the essence of the two conformations. Areas of variability that contain stems with less than a 50% occurrence rate are left open and are depicted by red dots.

The Stem Histogram allows one to view the stems present in the space of all solutions generated by multiple runs. Its presentation is less sensitive to slight positional differences in stems when compared with Stem Trace. In Stem Trace, two stems may appear to be close together but structurally might be quite distant. The Stem Histogram will indicate this separation more vividly since its positions are based on the absolute scale of sequence length. On the other hand the Stem Histogram does not separate out which structures contain individual stems. Stem Trace is essentially an orthogonal view to the Stem Histogram, allowing one to differentiate and compare the stems that appear in the different structures.

Taxonomy Trees

Since RNA secondary structures can be represented by a tree data structure (See Figure 11) [20], a tree-matching algorithm [21] can be applied to a database of structures generated by the genetic algorithm. The matching process applies scored editing operations to the trees. Ultimately, a matrix of scores is produced which can be presented to a clustering algorithm to place together those secondary structures that are similar, while those that are dissimilar far apart.

Three different levels of abstraction can be used to measure the degrees of dissimilarity. The first level simply measures the similarity based upon the topology of the structures. Thus, the edit distances are determined by the existence or lack of existence of the topological features such as bulge loops, hairpin loops, internal loops and multibranch loops. The next level of abstraction considers the size of the loops and stems and adds the absolute size differences into the edit distance calculation. The third level takes into account the size differences inherent in the component parts of loops. Thus, the sum of the absolute differences of the component parts of the loop structures is factored into the cost. If the number of components differs, the best sum is chosen. Figure 12 shows a taxonomy tree for the data presented in Figures 7a. Identical leaf nodes are collapsed into single representative nodes to compress the size of the cluster trees. The upper set of nodes is indicative of structures that are “branched” while the lower set of nodes represent those structures that are “linear”.

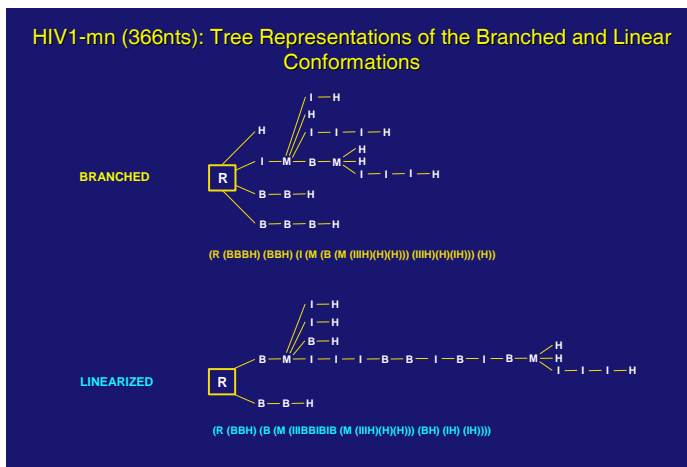


Figure 11. Tree representations of the topologies of the branched and linear structures for HIV-MN. The trees could also be represented by the parenthesis notation. R is the root node that ties together the entire tree.

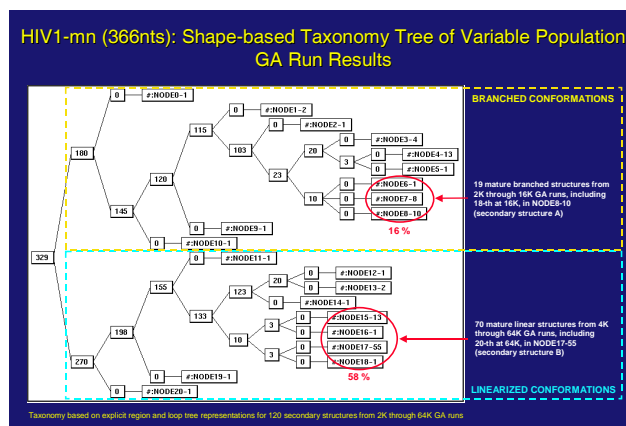


Figure 12. Cluster tree derived from a tree-matching algorithm using topological trees similar to those shown in Figure 11. The topological trees were generated from the results of all the GA population runs for HIV-MN. The terminal nodes, as indicated, are shown in a condensed mode depicting only the number of individual structures represented by that node rather than the names of each of the individual structures. This feature is used to conserve display space. It should be noted that two distinct clusters form, one representing the branched conformation and the other representing the linear conformation.

Conclusion

The issue of RNA structure/function determination is a difficult and complex problem. The basic principle that is applied is that given an RNA sequence, the three-dimensional structure that is ultimately formed is totally determined by its sequence and its surrounding environment. Environmental factors such as solvent, ions and proteins are quite important for structure/function determination. Thus, the ability to incorporate information about the external environment is quite important. The genetic algorithm allows some of this information to be used to guide the folding. For example, a set of “sticky stems” can be defined which act as hints, during a run of the GA, biasing the formation of specific structural elements. RNA secondary structures are important initial building blocks for the final three-dimensional structures. Also, it is important to remember that the final structure is not necessarily the only determinate of an RNA’s function. Functional intermediates or multiple stable states may occur that impart multiple functionality to these molecules. Many examples exist in the literature, which demonstrate the diversity of RNA. The genetic algorithm described in this paper is an example of a way to determine those states that may be important for RNA functionality.

STRUCTURELAB used in conjunction with the genetic algorithm’s results has proven to be a very valuable tool for determining these functional states. A vast amount of information is obtained from the GA from both individual and multiple runs, including variable population runs. The various visualization tools that are part of STRUCTURELAB and the GA accomplish analysis of these results. Each one of these tools views the data from a somewhat different perspective. Ultimately these perspectives are combined to obtain an understanding of the folding patterns of the RNAs in question. Future research requires the integration of the visual data mining environment with sophisticated statistical tools to enable the extraction of subtle information such as correlated folding events and those specific structural elements that are important for folding pathways.

References

- [1] Zuker, M. (1989): On finding all suboptimal foldings of an RNA molecule. *Science*. 244, 48-52.
- [2] Rivas, E. and Eddy, S.R. (1999): A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* 285, 2053-2068.
- [3] Mathews, D.H., Sabina, J., Zuker, M., Turner D.H. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.* 288, 911-940.
- [4] Shapiro, B.A. and Navetta, J. (1994) A massively parallel genetic algorithm for RNA secondary structure prediction. *The Journal of Supercomputing*. 8, 195-207.
- [5] Shapiro, B.A. and Wu, J.C. (1996): An annealing mutation operator in the genetic algorithm for RNA folding. *CABIOS*, 12 (3), 171-180.
- [6] Shapiro, B.A. and Wu, J.C. (1997): Predicting RNA H-type pseudoknots with the massively parallel genetic algorithm. *Bioinformatics*. 13 (4), 459-471.

- [7] Wu, J.C. and Shapiro, B.A. (1999): A Boltzmann filter improves the prediction of RNA folding pathways in a massively parallel genetic algorithm. *J. Bio. Struct. Dynam.* 17 (3), 581-595.
- [8] Shapiro, B.A., Wu, J.C., Bengali, D., and Potts, M.J. (2001): The massively parallel genetic algorithm for RNA folding: MIMD implementation and population variation. *Bioinformatics.* 17 (2), 137-148.
- [9] Shapiro, B.A., Bengali, D., Kasprzak, W., and Wu, J.C. (2001): RNA Folding Pathway Functional Intermediates: Their Prediction and Analysis. *J. Mol. Biol.* 312(1), 27-44.
- [10] Shapiro, B.A., Bengali, D., Kasprzak, W., and Wu, J.C. (2001): Computational insights into RNA folding pathways: Getting from here to there. *Proceedings of the Atlantic Symposium on Computational Biology and Genome Systems and Technology.*
- [11] Holland, J.H. (1975): *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI.
- [12] Holland, J.H. (1992): *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence (Complex A).* MIT Press, Cambridge, MA.
- [13] Goldberg, D.E. (1989): *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Wesley, Reading, MA.
- [14] Shapiro, B.A. and Kasprzak, W. (1996): STRUCTURELAB: A heterogeneous bioinformatics system for RNA structure analysis. *Journal of Molecular Graphics.* 14, 194-205.
- [15] Kasprzak, W. and Shapiro, B.A. (1999): Stem Trace: An interactive visual tool for comparative RNA structure analysis. *Bioinformatics.* 15 (1), 16-31.
- [16] Berkhout, B., Ooms, M., Beerens, N., Huthoff, H., Southern, E. and Verhoef, K (2002): In Vitro evidence that the untranslated leader of HIV-1 genome is an RNA checkpoint that regulates multiple functions through conformational changes. *J. of Biol. Chem.* 277 (22), 19967-19975.
- [17] Huthoff H. and Berkhout, B. (2002): Multiple secondary structure rearrangements during HIV-1 RNA dimerization. *Biochem.,* 41, 10439-10445.
- [18] Huthoff, H. and Berkhout, B. (2001): Two alternating structures of the HIV-1 leader RNA. *RNA.* 7, 153-157.
- [19] Kasprzak, W. and Shapiro, B.A. (2002): Structural dependencies of the HIV-1 dimer initiation site as determined by the massively parallel genetic algorithm. *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences,* CSREA Press, Vol. I, 48-54.
- [20] Shapiro, B.A. (1988): An algorithm for comparing multiple RNA secondary structures. *CABIOS.* 4 (3), 387-393.

[21] Shapiro, B.A. and Zhang, K. (1990): Comparing multiple RNA secondary structures using tree comparisons. *CABIOS*. 6 (4), 309-318.

This publication has been funded in part with Federal funds from the National Cancer Institute, National Institutes of Health under contract NO1-CO-12400.