Abstract folding space analysis based on helices

JIABIN HUANG,1 ROLF BACKOFEN,2 and BJÖRN VOß1,3
1Genetics & Experimental Bioinformatics, Faculty of Biology, University of Freiburg, Freiburg 79104, Germany
2Chair for Bioinformatics, Faculty of Technology, University of Freiburg, Freiburg 79110, Germany

ABSTRACT

RNA has many pivotal functions especially in the regulation of gene expression by ncRNAs. Identification of their structure is an important requirement for understanding their function. Structure prediction alone is often insufficient for this task, due to algorithmic problems, parameter inaccuracies, and biological peculiarities. Among the latter, there are base modifications, cotranscriptional folding leading to folding traps, and conformational switching as in the case of riboswitches. All these require more in-depth analysis of the folding space. The major drawback, which all methods have to cope with, is the exponential growth of the folding space. Therefore, methods are often limited in the sequence length they can analyze, or they make use of heuristics, sampling, or abstraction. Our approach adopts the abstraction strategy and remedies some problems of existing methods. We introduce a position-specific abstraction based on helices that we term helix index shapes, or hishapes for short. Utilizing a dynamic programming framework, we have implemented this abstraction in the program RNAHeliCes. Furthermore, we developed two hishape-based methods, one for energy barrier estimation, called HiPath, and one for abstract structure comparison, termed HiTed. We demonstrate the superior performance of HiPath compared to other existing methods and the competitive accuracy of HiTed. RNAHeliCes, together with HiPath and HiTed, are available for download at http://www.cyanolab.de/software/RNAHeliCes.htm.

Keywords: energy barrier; folding pathway; RNA secondary structure; structure comparison

INTRODUCTION

Recent advances in research on RNA have led to a change in perspective regarding the role of RNA. It becomes increasingly clear that RNA has many pivotal functions, especially in the regulation of gene expression by non-coding RNAs (ncRNAs) and as cis-regulatory RNA elements. Generally, the correct exertion of an ncRNA's function depends on the proper formation of its structure. This is not a big deal for the RNA in vivo, which usually finds its native conformation. But for in silico analyses, the folding process holds a lot of surprises, which renders structure prediction an error-prone task. Beyond peculiarities of the folding process, functional characteristics of an ncRNA may need more elaborate studies than predicting one minimum free energy (mfe) structure. Bistable RNAs and riboswitches, for example, can only be found when, in addition to the optimal structure, suboptimal structures are considered.

It is often useful to analyze the folding space of a ncRNA as this gives deeper insight into structural properties. Unfortunately, this does not come without a cost, which is the complexity and size of the folding space. It grows exponentially with sequence length and corresponds to a multidimensional space. Nevertheless, methods exist which can be used to carry out detailed analyses of the folding space. Suboptimal structure prediction, with the enumeration of all possible secondary structures, is available with RNAsubopt (Wuchty et al. 1999). This constitutes the most basic method for folding space analysis.

The major problem in folding space analysis is to create relations between the individual structures or shapes. These relations will then allow for inference of properties of individual structures. For example, a structure whose neighbors all have higher free energy constitutes a local minimum of the folding space. Important is the notion of neighborhood that defines the set of structures to which the candidate structure is compared. Commonly, neighboring structures are those that differ by a single base pair. Applying this idea to the set of all suboptimal structures makes it possible to compute all local minima and the saddle points connecting these. Even more important, it is possible to compute energy barriers between local minima, which is equivalent to the energy needed to transform the two structures into each other. This information is helpful, e.g., in detecting folding traps or bistability. An implementation

3 Corresponding author
E-mail bjoern.voss@biologie.uni-freiburg.de
Article published online ahead of print. Article and publication date are at http://www.rna.org/cgi/doi/10.1261/rna.033548.112.
is available with Barriers (Flamm et al. 2000, 2002). Unfortunately, due to the exponential growth of the folding space with sequence length, it is restricted to sequences up to 150 nt.

One solution to this complexity problem is the use of heuristics, which try to predict the series of intermediate structures by means of simple rules. Morgan and Higgs proposed an algorithm (Morgan and Higgs 1998) in which the structure that contains the fewest “clashing” base pairs is selected as the next intermediate structure from a set of neighboring structures. Flamm et al. (2001) extended the idea by keeping the k best candidates during the construction of a folding pathway (breadth first search, BFS). Recently, Dotu et al. (2010) proposed RNAtabupath, an algorithm in which a tabu list, storing recently visited neighboring structures, is used to rule them out in subsequent steps. A different approach to reducing the complexity in folding space analysis is shape abstraction (Giegerich et al. 2004). This method provides a means to partition the folding space into classes of similar structures. Together with features such as their probabilities, shapes provide an overview of the folding space. A major drawback of shape abstraction, as it is implemented so far, is the position independence of the abstraction mappings. A single hairpin at the 5’ end has the same “[ ]”-shape as one at the 3’ end. As a consequence, shape classes encompass structurally similar but perhaps functionally unrelated structures. This can only be overcome by a new abstraction function that we will present later. In Bogomolov et al. (2010), abstract shapes were used to guide the path heuristics toward a better folding pathway.

In addition to the energy barriers separating two structures, it is also of interest to know their structural similarity. This is, for instance, the case for riboswitches that need structurally dissimilar states to exert their function. Furthermore, a high energy barrier may imply a high structural distance, but the opposite need not hold. As a result, dissimilar structures might be kinetically connected and, thus, the equilibrium structure rather ill-defined. If this is not the case, we might speak of a well-defined structure (space), as, for example, for microRNA precursors.

The contributions in this manuscript are threefold. First, we introduce a position-specific abstraction based on helices, which we term hishapes. In addition to its usage for structure abstraction, we present its application in computing near-optimal folding pathways and show that it is superior to other methods. Last but not least, we define a distance measure for hishapes that is based on tree editing and present benchmark results showing its good performance.

RESULTS

Helix-based structure abstraction

In the following, we give an informal presentation of the concept of hishapes. The formal definitions are given in Materials and Methods. Any secondary structure can be broken down into a series of five loop types that are closed by helices. These loops are hairpin, bulge, internal, stacking, and multiple loop, denoted as hl, bl, il, sl, and ml, respectively. For our purpose stacking loops are special as they only elongate helices and do not introduce new ones. Thus, a helix can only be of type hl, bl, il, or ml. The position of a helix in the sequence is defined by its innermost base pair (i, j), which is the closing base pair of its corresponding loop. Since we abstract from the length of a helix, we define the helix index to be the central position of the helix, thus (i + j)/2. Helices closed by different loops may have the same helix index. In order to represent helices in an unambiguous fashion, we mark the helix index with m, b, or i for multiple, bulge, or internal loop, respectively. In order to simplify the notation of the representation, we do not mark hairpin loop helices. Using a mapping function π, we can now map any secondary structure to a helix index shape (hishape), which is simply a list of helix indices.

In order to provide different levels of abstraction, we make use of different mapping functions. πh retains only hairpin loop helices and πh+ additionally keeps track of the nesting within multiloops. πm and πa extend πh+ through retaining multiloops and all helices, respectively. πm, πa, and πh+ preserve the nesting pattern of helices by embracing helices within multiloops by a pair of brackets (see Fig. 1B; details are given in Materials and Methods). The nesting within bulge and internal loops can be inferred from the order in the hishape, where the right helix is embedded in its left neighbor. While the number of considered structural elements increases, the level of abstraction decreases in the order πh+, πh, πm, and πa. Thus, πh is the most abstract and πa the least abstract level. Inherently, a hishape defines a class of similar structures, i.e., those with equal hishape. The class member with minimum free energy is defined as the hishape representative and termed hishrep.

Hishape space

Let \( F(s) \) be the folding space (i.e., the set of all secondary structures without pseudoknots) of RNA sequence \( s \) with length \( n \), the hishape space is defined as \( \pi(s) = \{ \pi_\alpha(y) \} | y \in F(s) \} \), where \( \pi_\alpha \in \{ \pi_h, \pi_{h+}, \pi_m, \pi_a \} \). The size of the hishape space depends on the choice of the mapping function. We did empirical measurements on random sequences for the growth of the hishape space. Based on these, we derived empirical growth asymptotics. Fitting the data to the formula \( \beta_\alpha \cdot n^{\alpha_\alpha} \) proposed in Lorenz et al. (2008) and Nebel and Scheid (2009), we derive the following numbers: \( \beta_{\pi_m} = 1.2311n, \alpha_{\pi_m} = 30.699; \beta_{\pi_a} = 1.2746n, \alpha_{\pi_a} = 12.179; \beta_{\pi_h} = 1.3183n, \alpha_{\pi_h} = 5.0103 \) and
\( \beta_{\pi_x} = 1.3931n, \alpha_{\pi_x} = 3.4114. \) For structure (disallowing lonely base pairs) and shape space (level 5), we derive \( \beta_{str} = 1.4282n, \alpha_{str} = 4.2366 \) and \( \beta_{shape} = 1.1331n, \alpha_{shape} = 20.700, \) respectively. As expected, the size of the \( \text{hishape} \) space is in between those of structure and shape space. We did not examine the growth behavior for different nucleotide distributions, e.g., GC content, but we expect only minor effects since the major effect of the GC content will be on the free energy. Particularly, these differences can be neglected when comparing \( \text{hishape} \), shape, and structure space.

**Spliced leader RNA from Leptomonas collosoma**

The spliced leader RNA (SL) from *Leptomonas collosoma* (LeCuyer and Crothers 1993) has two alternating structures differing by only 1.7 kcal/mol in free energy. Figure 2 shows the results of shape and \( \text{hishape} \) analysis. While the two \( \pi_m \) \( \text{hishapes} \) ([27] and [38]) reflect the two experimentally verified conformations of the spliced leader RNA, RNAs shapes yields the same abstract shape “[]” for both conformations. The probability of the “[]”-shape is 0.961912, and contributions to this come from both conformations. This example shows that for certain applications, \( \text{hishape} \), shape, and structure space.

**Energy barrier estimation**

Important features of the folding space are pathways connecting alternative structures. For these pathways, commonly the most interesting features are the saddle structure and its energy, from which the energy barrier can be calculated. Computation of these folding pathways within our program HiPath follows the idea of a guided path. Guide points are provided by \( \text{hishapes} \), and, in order to achieve a reasonably fast method, the paths between guide points are computed heuristically. For this, we chose breadth first search, which has already been used for pathway approximation in Flamm et al. (2001). Unlike Morgan-

![FIGURE 1. (A) Example secondary structure, (B) properties of its helices, and the resulting \( \text{hishapes} \). hl, bl, il, and ml refer to hairpin, bulge, internal loop, and multiple loop, respectively. The letters m, b, and i appended to helix indices within \( \text{hishapes} \) indicate the loop type (multiple, bulge, and internal loop, respectively). Helix indices without suffix represent hairpin loops. Pairs of brackets in a \( \text{hishape} \) provide nesting information within multiloops. The structure plot in A was created using VARNA (Darty et al. 2009).](image)

<table>
<thead>
<tr>
<th>helix name</th>
<th>( h_L(i,j) )</th>
<th>( \text{hishape}(h_L) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>( h_{hl}(13,21) )</td>
<td>17</td>
</tr>
<tr>
<td>h2</td>
<td>( h_{hl}(33,39) )</td>
<td>36</td>
</tr>
<tr>
<td>h3</td>
<td>( h_{hl}(30,41) )</td>
<td>35.56</td>
</tr>
<tr>
<td>h4</td>
<td>( h_{hl}(26,45) )</td>
<td>35.5</td>
</tr>
<tr>
<td>h5</td>
<td>( h_{hl}(52,60) )</td>
<td>56</td>
</tr>
<tr>
<td>h6</td>
<td>( h_{ml}(6,66) )</td>
<td>36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>( \text{hishape} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{hishape}_{\pi_m} )</td>
<td>[36m, (( 17, 35.5 )), (( 35.5, 26, 36, 56, ))]</td>
</tr>
<tr>
<td>( \text{hishape}_{\pi_m} )</td>
<td>[36m, (( 17, 36, 56, ))]</td>
</tr>
<tr>
<td>( \text{hishape}<em>{\pi</em>{bh}} )</td>
<td>([( 17, 36, 56, ])]</td>
</tr>
<tr>
<td>( \text{hishape}<em>{\pi</em>{bh}} )</td>
<td>[( 17, 36, 56, )]</td>
</tr>
</tbody>
</table>

**FIGURE 2.** The alternating structures of the spliced leader RNA from *L. collosoma* with their free energy (\( E \) in kcal/mol), \( \text{hishapes} \), \( \text{hishape} \) probabilities (\( P \)), and their abstract shapes (level 5).
Higgs, BFS keeps the $l$ best intermediate results at each iteration step, which significantly increases prediction accuracy. We provide two methods—one, called HiPath-full, for predicting energy barriers for all pairs of $hisheps$, and the other, named HiPath-pair, for predicting the energy barrier for a given pair of structures/$hisheps$.

**Full method for barrier estimation**

For a complete folding space analysis, we start with computing all $hisheps$ of interest (e.g., all or the $k = 100$ best). For all possible pairs, we compute a near-optimal pathway using BFS (e.g., $l = 10$) on the $hisreps$ and store its saddle structure and energy in a matrix. For vicinal $hisheps$, i.e., $hisheps$ with similar $hisreps$, the BFS results can be expected to be good, but for distant $hisheps$, they can very likely be improved. We do this by applying Dijkstra’s algorithm (Dijkstra 1959) to compute optimal paths based on the initial results. The improvement is a result of combining short, more accurate paths into long ones. We applied the HiPath-full procedure to all ($N = 3535$) $p$ $hisheps$ of the SL RNA (56 nt). Results for the 10 best $hisheps$ are summarized in Table 2. For example, the pathway from $[27]$ to $[38]$ has an energy barrier of 11.8 kcal/mol. Compared to the exact value from Barriers—11.1 kcal/mol, our method is 0.7 kcal/mol or $\pm 6\%$ wrong.

Another interesting feature can be figured out when looking at the rows for $hisheps$ $[27]$ and $[38]$. Whenever $hishep$ $[27]$ is compared to a $hishep$ containing helix 38, the energy barrier is equally high (11.8 kcal/mol), while it is lower for those $hisheps$ containing helix 27 (at most, 6.7 kcal/mol). For $hishep$ $[38]$, it is vice versa. Thus, helices 27 and 38 are kinetically incompatible, which nicely reflects the bistable character of this RNA.

**Pairwise barrier approximation**

In our results, we empirically found that the number of $hisheps$ grows exponentially with sequence length. Thus, in general the full procedure is computationally very expensive. So, how can we improve? Consider the case that we are only interested in computing the energy barrier for a certain start and target structure. How can we reasonably restrict the number of $hisheps$ without losing analysis depth? Here, we make use of related $hisheps$ as defined by Equation 5. Related $hisheps$ are those $hisheps$ where each hairpin loop helix index is present in the $hishep$ of start and/or the target structure. They can be computed using a modified grammar in which a syntactic filter is applied to the productions for hairpin loops. This filter checks to see if the hairpin loop helix index is present in the $hishep$ of start and/or the target structure. With this approach, we compute rigorously all related $hisheps$ in a very lean and, thus, fast way. For the set of related $hisheps$, we apply the same procedure as described above, with the

### Table 1. Comparison of run time (RT) and memory consumption (M) of RNAHeliCes, RNAshapes, and RNAfold on random sequences of length 240 to 1200 nt

<table>
<thead>
<tr>
<th>Length (nt)</th>
<th>RNAHeliCes</th>
<th>RNAshapes</th>
<th>RNAfold</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT (sec)</td>
<td>M (kB)</td>
<td>RT (sec)</td>
</tr>
<tr>
<td>240</td>
<td>4.20</td>
<td>30,214</td>
<td>0.13</td>
</tr>
<tr>
<td>480</td>
<td>40.11</td>
<td>119,428</td>
<td>0.94</td>
</tr>
<tr>
<td>720</td>
<td>137.28</td>
<td>269,912</td>
<td>4.42</td>
</tr>
<tr>
<td>960</td>
<td>380.09</td>
<td>486,929</td>
<td>12.80</td>
</tr>
<tr>
<td>1200</td>
<td>687.98</td>
<td>762,645</td>
<td>25.47</td>
</tr>
</tbody>
</table>

Final values represent averages of three independent measurements. (n.m.) Memory consumption could not be measured since the run time was too short.

### Table 2. HiPath-full energy barriers in kcal/mol for the 10 best $hisheps$ of SL RNA based on complete enumeration of $\pi_m$ $hisheps$

<table>
<thead>
<tr>
<th></th>
<th>27</th>
<th>38</th>
<th>[10.5,38]</th>
<th>[13,38]</th>
<th>[27,49.5]</th>
<th>[27,52.5]</th>
<th>[11.5,38]</th>
<th>[11,38]</th>
<th>[27,47.5]</th>
<th>[16,5,38]</th>
</tr>
</thead>
<tbody>
<tr>
<td>[27]</td>
<td>11.8</td>
<td>11.8</td>
<td>11.8</td>
<td>4.9</td>
<td>4.0</td>
<td>11.8</td>
<td>11.8</td>
<td>6.7</td>
<td>11.8</td>
<td></td>
</tr>
<tr>
<td>[38]</td>
<td>10.1</td>
<td>5.2</td>
<td>2.5</td>
<td>10.1</td>
<td>3.1</td>
<td>4.9</td>
<td>10.1</td>
<td>4.3</td>
<td>10.1</td>
<td></td>
</tr>
<tr>
<td>[10.5,38]</td>
<td>8.8</td>
<td>3.9</td>
<td>3.9</td>
<td>8.8</td>
<td>3.9</td>
<td>3.9</td>
<td>8.8</td>
<td>3.9</td>
<td>8.8</td>
<td></td>
</tr>
<tr>
<td>[13,38]</td>
<td>8.5</td>
<td>0.9</td>
<td>3.6</td>
<td>8.5</td>
<td>2.1</td>
<td>3.3</td>
<td>8.5</td>
<td>2.7</td>
<td>8.2</td>
<td></td>
</tr>
<tr>
<td>[27,49.5]</td>
<td>1.3</td>
<td>8.2</td>
<td>8.2</td>
<td>8.2</td>
<td>1.3</td>
<td>8.2</td>
<td>8.2</td>
<td>3.1</td>
<td>8.2</td>
<td></td>
</tr>
<tr>
<td>[27,52.5]</td>
<td>0.0</td>
<td>7.8</td>
<td>7.8</td>
<td>7.8</td>
<td>0.9</td>
<td>7.8</td>
<td>7.8</td>
<td>2.7</td>
<td>7.8</td>
<td></td>
</tr>
<tr>
<td>[11.5,38]</td>
<td>7.8</td>
<td>1.4</td>
<td>2.9</td>
<td>1.41</td>
<td>7.8</td>
<td>7.8</td>
<td>2.6</td>
<td>7.8</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td>[11,38]</td>
<td>7.7</td>
<td>2.5</td>
<td>2.8</td>
<td>2.5</td>
<td>7.7</td>
<td>7.7</td>
<td>2.5</td>
<td>7.7</td>
<td>2.5</td>
<td></td>
</tr>
<tr>
<td>[27,47.5]</td>
<td>2.5</td>
<td>7.6</td>
<td>7.6</td>
<td>2.5</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>[16,5,38]</td>
<td>7.3</td>
<td>1.5</td>
<td>2.4</td>
<td>1.5</td>
<td>7.3</td>
<td>7.3</td>
<td>1.5</td>
<td>2.1</td>
<td>7.3</td>
<td></td>
</tr>
</tbody>
</table>
difference that we use Dijkstra’s algorithm to compute the optimal path for the user-defined start and target structure/hisshape only.

In order to assess the accuracy of our approximation, we used data set “short” (see Materials and Methods) to compare our algorithm to four other methods, namely Barriers, Morgan-Higgs (MH), BFS with 10 best candidates, and RNAtabupath (Dotu et al. 2010). The results in Table 3 show that the hishape-based pathway approximation performs best compared to the other heuristics and achieves reasonable accuracy when compared with the exact results from Barriers. The error ranges from 0.78 to 1.72 kcal/mol and is, in all cases, ~12%. HiPath was, in all cases, 4–30 times faster than Barriers.

In order to get an impression of the source of the inaccuracy of our method, we compared the saddle structures and hishapes predicted by our method with the “native” ones computed by Barriers (data not shown). Interestingly, in all cases, the native saddle structure comprises helices (hairpin loops) that were not present in the start or target structure. Thus, the hishape of the native saddle does not belong to the set of related hishapes. Furthermore, the BFS step of our procedure does not compensate for this error, at least for the cases shown. Overall, it seems that the energetically most favorable pathway is more complex than expected. We performed a second benchmark using the data set “ribswitches” (see Materials and Methods). The results in Table 4 show that our algorithm is able to compute better folding pathways in all cases. We can improve the estimated energy barrier by 0.4–3.6 kcal/mol or by ~1.6 kcal/mol on average.

Performance

The efficiency of our algorithm strongly depends on related hishape calculation that is carried out by RNAHeliCes. On a typical sequence, e.g., the lysine riboswitch of lycC from Bacillus subtilis (233 nt), the run time remains <50 sec. In order to get a general picture, we measured run time and memory consumption of HiPath, BFS, and Barriers. The results are given in Table 5. Barriers produced results within a reasonable time only for sequences up to 100 nt in length. BFS is the fastest and least expensive method. HiPath performs quite well and computes the energy barrier for two structures of length 500 nt in ~6 min, consuming ~340 MB of memory.

Abstract structure comparison

Another feature of interest when analyzing the folding space of an RNA is structural diversity. For hishapes, i.e., hishreps, we can, of course, use existing methods for structure comparison, and they would benefit from the reduced number of entities that need to be compared. But why not design a comparative approach solely based on hishapes? They are inherently tree-like, and the positional information provides reasonable resolution for comparison.

We introduce the hishape-based tree edit distance (HiTed), which is an extension of the tree edit distance (Shapiro 1988). Our method extends the tree edit distance for abstract trees of RNA secondary structures (Shapiro and Zhang 1990). This representation abstracts from the size of structural elements and is, thus, closely related to the idea of abstract shapes. In Shapiro and Zhang (1990), the edit operations—relabeling, delete, and insert—and a corresponding cost function for edit operations on abstract trees are defined.

HiTed is based on this scoring scheme for loop/helix editing and extends it by the positional distance of helices. The latter is the absolute difference of the helix indices, e.g., d(35i, 45i) = |35 – 45| = 10. The two distance measures are combined using a weighting factor λ as shown in Definition 6 (Materials and Methods). The initial intention in the design of HiTed was to have a distance measure for alternative hishapes of the same sequence. Nevertheless, we think that HiTed is also suitable for comparing structures/hishapes of different sequences. In order to assess this and to analyze the influence of the weighting factor λ, we provide two benchmarks. First, we compare the results for different λ-values, and, second, we compare HiTed with other structure comparison methods, namely RNAdistance (Hofacker et al. 1994) and RNAforester (Höchsmann et al. 2003). We do this using the Brasero data set and protocol (see Materials and Methods). We take the area under the curve (AUC) values of the ROC plots to visualize the results. Figure 3 shows how the AUC changes with 0 ≤ λ ≤ 32.

**Abstract structure comparison**

Another feature of interest when analyzing the folding space of an RNA is structural diversity. For **hisshapes**, i.e., **hishreps**, we can, of course, use existing methods for structure comparison, and they would benefit from the reduced number of entities that need to be compared. But why not design a comparative approach solely based on **hisshapes**? They are inherently tree-like, and the positional information provides reasonable resolution for comparison.

We introduce the **hisshape**-based tree edit distance (HiTed), which is an extension of the tree edit distance (Shapiro 1988). Our method extends the tree edit distance for abstract trees of RNA secondary structures (Shapiro and Zhang 1990). This representation abstracts from the size of structural elements and is, thus, closely related to the idea of abstract shapes. In Shapiro and Zhang (1990), the edit operations—relabeling, delete, and insert—and a corresponding cost function for edit operations on abstract trees are defined.

HiTed is based on this scoring scheme for loop/helix editing and extends it by the positional distance of helices. The latter is the absolute difference of the helix indices, e.g., $d(35i, 45i) = |35 – 45| = 10$. The two distance measures are combined using a weighting factor $\lambda$ as shown in Definition 6 (Materials and Methods). The initial intention in the design of HiTed was to have a distance measure for alternative **hishapes** of the same sequence. Nevertheless, we think that HiTed is also suitable for comparing structures/ **hishapes** of different sequences. In order to assess this and to analyze the influence of the weighting factor $\lambda$, we provide two benchmarks. First, we compare the results for different $\lambda$-values, and, second, we compare HiTed with other structure comparison methods, namely RNAdistance (Hofacker et al. 1994) and RNAforester (Höchsmann et al. 2003). We do this using the Brasero data set and protocol (see Materials and Methods). We take the area under the curve (AUC) values of the ROC plots to visualize the results. Figure 3 shows how the AUC changes with $0 \leq \lambda \leq 32$.  

**TABLE 3. Energy barriers for different sequences, given in kcal/mol**

<table>
<thead>
<tr>
<th>Instance</th>
<th>Length</th>
<th>MH</th>
<th>BFS</th>
<th>Tabu</th>
<th>HiPath</th>
<th>Barriers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spliced leader RNA (L. collasoma)</td>
<td>56 nt</td>
<td>18.2</td>
<td>13.0</td>
<td>12.9</td>
<td>12.4 (0.3 sec)</td>
<td>11.1 (1.3 sec)</td>
</tr>
<tr>
<td>pheST Attenuator (E. coli)</td>
<td>73 nt</td>
<td>19.7</td>
<td>14.95</td>
<td>15.66</td>
<td>14.15 (2.6 sec)</td>
<td>13.37 (16.5 sec)</td>
</tr>
<tr>
<td>S15 mRNA leader (E. coli)</td>
<td>74 nt</td>
<td>24.97</td>
<td>19.57</td>
<td>18.2</td>
<td>17.0 (2.6 sec)</td>
<td>15.28 (61.5 sec)</td>
</tr>
<tr>
<td>5' UTR of MS2 RNA genome</td>
<td>73 nt</td>
<td>31.9</td>
<td>24.88</td>
<td>24.88</td>
<td>23.3 (9.2 sec)</td>
<td>22.0 (291.9 sec)</td>
</tr>
</tbody>
</table>

(BFS) Breadth first search, (MH) Morgan-Higgs. For HiPath and Barriers, the run time is given in brackets. sRNAtabupath was used with 500 iterations. The maxkeep value was 10 for BFS alone, as well as the BFS used within HiPath. HiPath was used with a minimum of 40 **hisshapes** and at the most abstraction level $\pi_\text{m}$. For Barriers, we used suboptimal structures generated with RNAsubopt in an energy range of 12, 15, 17, and 24 kcal/mol, respectively, and “RNA-noLP” as the move set.
Overall, there seems to be no general optimal $\lambda$-value, but a range of 0 to 5 seems to be reasonable to achieve reliable results. For SRP and sRNA, the additional positional distance introduced by HiTed improves prediction accuracy, and we achieve the best results with values of 17 and 2, respectively. Conversely, for the miRNA and tRNA data sets, the performance decreases with increasing $\lambda$, and the optimum is $\lambda = 0$ for these data sets. One reason for this might be that the helices occur at quite diverse positions within the sequences, thus the penalization of the positional difference decreases performance.

We compare HiTed with RNAdistance and RNAforester. For HiTed, we chose two different values for $\lambda$. $\lambda = 5$ resembles a consensus value based on the previous results, and $\lambda = 0$ switches off the positional distance contribution. The latter is similar to using RNAdistance in the coarse-grained tree editing mode, the distance measure introduced by Shapiro and Zhang (1990). We refer to this as RNAdistanceSZ, compared to RNAdistance using default parameters, i.e., tree editing on full structure representation. Finally, RNAforester was used once with default parameters and once without scoring sequence homology (RNAforesterNOSEQ). Results are shown in Figure 4. Interestingly, our abstract and fast comparative analysis of RNA achieves the second best accuracy for one data set (SRP) and comparable results for the other data sets.

### Table 5. Comparison of run time (RT) and memory consumption (M) of HiPath, BFS ($k = 10$) and Barriers on pairs of structures from random sequences of length 100–500 nt

<table>
<thead>
<tr>
<th>Length (nt)</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT (sec)</td>
<td>M (kB)</td>
<td>RT (sec)</td>
<td>M (kB)</td>
<td>RT (sec)</td>
</tr>
<tr>
<td>HiPath</td>
<td>0.85</td>
<td>7788</td>
<td>8.07</td>
<td>30,374</td>
<td>65.32</td>
</tr>
<tr>
<td>BFS</td>
<td>0.02</td>
<td>n.m.</td>
<td>0.10</td>
<td>n.m.</td>
<td>0.18</td>
</tr>
<tr>
<td>Barriers</td>
<td>1439.39</td>
<td>3,366,569</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

HiPath was used with the same parameters as described in Table 4. For Barriers, we used suboptimal structures below the barrier energy estimated by HiPath plus 1 kcal/mol. Final values represent averages of three independent measurements. (n.m.) Memory consumption could not be measured since the run time was too short, (—) computation did not finish within 3 d.

Overall, there seems to be no general optimal $\lambda$-value, but a range of 0 to 5 seems to be reasonable to achieve reliable results. For SRP and sRNA, the additional positional distance introduced by HiTed improves prediction accuracy, and we achieve the best results with values of 17 and 2, respectively. Conversely, for the miRNA and tRNA data sets, the performance decreases with increasing $\lambda$, and the optimum is $\lambda = 0$ for these data sets. One reason for this might be that the helices occur at quite diverse positions within the sequences, thus the penalization of the positional difference decreases performance.

We compare HiTed with RNAdistance and RNAforester. For HiTed, we chose two different values for $\lambda$. $\lambda = 5$ resembles a consensus value based on the previous results, and $\lambda = 0$ switches off the positional distance contribution. The latter is similar to using RNAdistance in the coarse-grained tree editing mode, the distance measure introduced by Shapiro and Zhang (1990). We refer to this as RNAdistanceSZ, compared to RNAdistance using default parameters, i.e., tree editing on full structure representation. Finally, RNAforester was used once with default parameters and once without scoring sequence homology (RNAforesterNOSEQ). Results are shown in Figure 4. Interestingly, our abstract and fast comparative analysis of RNA achieves the second best accuracy for one data set (SRP) and comparable results for the other data sets. The
SRP data set seems to be tricky, since, for all three tools, the variant that takes more information into account performs better. The opposite situation appears for the miRNA data set. For this data set, only RNAforester shows good performance and the other methods perform rather poorly. Here, one reason may lie in the fact that a single hairpin structure, being the structure of miRNA precursors, is quite likely to occur in random sequences, thus increasing the false positive rate. Looking into the noise part of the miRNA data set shows that this is actually the case. Additionally, the diversity of the T2 set is large as it contains sequences with up to 188 nt, while the sequences in R and F2 are at most 87 and 88 nt long, respectively. Albeit HiTed with \( l = 0 \) and RNAdistance SZ seem conceptually similar, their performance differs reasonably. The coarse-grained tree representation differs from the hishape tree in that it also models stacking regions and the external loop, which presumably is the reason for these differences.

Performance

We compared the run time and memory consumption of HiTed (\( \lambda = 5 \)) with RNAdistance and RNAforester. The results in Table 6 show that HiTed is the fastest but also the most memory-consuming method. The latter fact is an implementation issue and can thus be resolved by optimizing the code.

DISCUSSION

In the present paper, we introduce the concept of hishapes, which is closely related to the idea of abstract shapes. Briefly, we provide new mapping functions and preserve all functionality of shape analysis. Among these are search space reduction by (hi)shape filtering and probabilistic analysis based on (hi)shape classes. Compared to abstract shapes, the major advantage of the new abstraction is its position-specificity, which provides a better resolution, especially for short RNAs. The cost for this is a slightly increased search space, which is still much smaller than the structure space. Nevertheless, the abstraction keeps significant features of the structure space. Although the mathematical proof is not yet provided, based on our preliminary empirical analyses, we are convinced that hishapes comprise all, or at least a significant subset, depending on the abstraction level, of local minima of the folding space. Important features when analyzing the folding space of RNA are the energy barriers separating local minima. Their exact computation is expensive, and thus, several heuristic methods have been developed to allow for the analysis of long sequences and also to greater depth. HiPath, our hishape-guided energy barrier calculation, belongs to these methods and outperforms all heuristic methods compared in this manuscript. Comparing HiPath predictions for short sequences with exact values computed with Barriers shows that the inaccuracy is, in general, \( \sim 10\% \). Taking into account the inaccuracies of the thermodynamic parameters and that we neglect kinetic effects, the pathways and hence, the energy barriers predicted by RNAHeliCes provide reasonable alternatives.

The kinetics of RNA folding will be a major aspect in our future work on RNAHeliCes. In all abstraction levels we present, hairpin loops and their associated helices play a major role. For a helix closed by a hairpin loop, the helix index corresponds to the center of the closing base pair of the hairpin loop. The nucleation of the helix, formation of a hairpin loop, is the rate-limiting step in helix formation where, according to the master equation, the rate is dependent on the gain/loss in free energy. Thus, the thermodynamically most favorable hairpin loop is also the most likely to be formed first. Such a helix-based approach for predicting folding kinetics has been presented by Zhao et al. (2010). Their move set comprises helix addition, helix deletion, arm-by-arm exchange, and two-arm by two-arm exchange, which all fit perfectly into the hishape model. Together with the method for fast energy barrier computation, hishapes provide a promising candidate for kinetic studies.

So far, we have discussed the thermodynamic and perhaps kinetic capabilities of RNAHeliCes. Commonly,
also structural characteristics of the folding space are of interest. For this purpose, we introduced the *hishape*-based tree edit distance (HiTed). The performance of HiTed is comparable to those of other commonly used methods. This is somewhat surprising, taking into account the abstract tree representation on which it is based and the somewhat arbitrary scoring scheme it uses. Probably, the abstraction helps, at least in some cases, to circumvent pitfalls that the algorithms working on the full tree representation face. The scoring scheme likely needs some refinements, which may also be accompanied by changing the representation of *helix indices*. The currently used absolute position will disturb results when common structures are shifted within related sequences by insertions or elongated 5’ UTRs. Here, for example, relative positional values might be better suited. Additionally, choosing a reasonable value for parameter $\lambda$ is important in achieving optimal results. A good choice depends on various factors, such as sequence similarity, expected structural diversity, and the aim of the analysis. Therefore, it is difficult to provide a rule of thumb, but

**FIGURE 4.** ROC plots comparing HiTed, RNAdistance, and RNAforester on data sets SRP (A), sRNA (B), miRNA (C), and tRNA (D). The noise data used in all four cases are random genome segments from viral genomes. RNAdistance was used with default parameters (tree edit distance on full structure representation) and with coarse-grained tree edit distance (Shapiro and Zhang 1990) RNAdistance$_{SZ}$. RNAforester was used with default parameters and without scoring sequence similarity (RNAforesternoseq). HiTed was used with the indicated $\lambda$-values.
Any RNA secondary structure can be transformed into a list of helix indices which we term \( \text{hishape} \) which is discussed in Materials and Methods. For \( \pi_b \), all abstractions preserve the nesting pattern of helices by embracing helices within multiloops by a pair of brackets. Note that the mapping defined in Equation 4 does not ensure the correct nesting of helices by itself. This has to be achieved via the correct evaluation order within the algorithm, which is discussed in Materials and Methods. For \( \pi_b \) and \( \pi_a \), \( \text{hishapes} \) may be ambiguous since multiloop and symmetric internal loop helices can have \( \text{hishapes} \) equal to their enclosed helices. Therefore, the letter “\( m \)” is attached to the end of \( \text{hishapes} \) derived from \( h_{ml}(i,j) \) in \( \pi_m \) as well as in \( \pi_a \), while the letter “\( b \)” denotes \( \text{hishapes} \) derived from \( h_{bl}(i,j) \), and the letter “\( i \)” denotes \( \text{hishapes} \) derived from \( h_{il}(i,j) \) in \( \pi_a \).

**Definition 3 (Related hishapes)**

Given two hishapes \( H_1 \) and \( H_2 \) in an arbitrary abstraction type, and let \( \phi \) be a function extracting hairpin loop helix indices, related hishapes \( H_1 \) are those satisfying

\[
\phi(H_1) \subseteq \phi(H_1) \cup \phi(H_2)
\]

**Implementation**

In order to circumvent implementation-specific problems, e.g., index errors, and to take advantage of already existing code, we implemented the algorithms using Bellman’s GAP (Giegerich and Sauthoff 2011; Sauthoff et al. 2011). Here, a DP algorithm is split into a grammar and several algebras. The grammar ensures the correct nesting and juxtaposition of structural elements and, thus, describes the candidates of the search space, while the algebras evaluate these candidates. In the case of RNA structure analysis, algebras for energy minimization, partition function (McCaskill 1990) calculation, and pretty printing of the structure in dot-bracket-format and others exist. Algebras can be combined using product operations, which allow complex analyses to be built in comparing the results for different values of \( \lambda \) should help in this process.

A combination of the methods presented might include the idea of predicting conformational switching followed by paRNAss. Here, a conformational switch is characterized by the existence of two local minima that are structurally dissimilar and separated by a reasonable energy barrier. Furthermore, RNAHeliCes and HiTed might be used for the identification of common structures of two or more RNAs. This would provide an alternative to the prediction of consensus shapes (Reeder and Giegerich 2005).

Altogether, we believe that \( \text{hishape-based} \) abstraction provides a valuable tool for various applications in RNA secondary structure analysis. Improvements may be achieved for run time and memory usage by manual optimizations of the code generated by the GAP compiler. On the conceptual side, modified abstractions based on other helix features, e.g., outermost base pair, may be useful and extend the range of applications for our method.

**MATERIALS AND METHODS**

**Defining helix index shapes**

In the following, we provide formal definitions for the new abstraction based on \( \text{hishapes} \). We consider unknotted secondary structures as defined, for example, in Hofacker et al. (1994).

**Definition 1 (helix and helix index)**

A helix is a series of stacking base pairs starting with the closing base pair of a hairpin, bulge, internal, or multiple loop (\( h_{bl} \), \( h_{il} \), or \( h_{ml} \), respectively). Thus, a helix can be denoted by \( h_{L}(i,j) \) where \( i \) and \( j \) are the bases of the innermost base pair and \( L \) is the loop type (\( L \in \{h_{bl}, h_{il}, h_{ml}\} \)). The helix index of a helix \( h_{L}(i,j) \) is its central position, thus

\[
\text{helix index}(h_{L}(i,j)) = hi(i,j)
\]

\[
hi(i,j) = (i + j)/2
\]

**Definition 2 (hishop, hishrep, and hishape space)**

Any RNA secondary structure can be transformed into a list of helices \( H \). Using mapping functions \( \pi_{bl} \), \( \pi_{hL} \), \( \pi_{ml} \), or \( \pi_{aL} \) we can map \( H \) to a list of \( \text{hishapes} \) which we term \( \text{hishape} \) (helix index shape).

\[
H = \{h_{L}(i,j), h_{L}(k,l), \ldots \}, \text{ where } L \in \{h_{bl}, h_{il}, h_{ml}\}
\]

\[
\text{hishape}_{bl}(H) = \{\pi_b(h_{L}(i,j)), \pi_b(h_{L}(k,l)), \ldots\},
\]

where \( \pi_b \in \{\pi_b, \pi_{hL}, \pi_{ml}, \pi_{aL}\} \).
Approximating folding pathways

Full pathway analysis

The method for computing all pairwise energy barriers for a set of hishapes of an RNA sequence is as follows: First, compute all pairwise paths using BFS, and store saddle structure in matrix MBFS. Second, use Dijkstra’s algorithm to find the shortest path in MBFS for all pairs of hishapes. The procedure is given in algorithm 1.

We use findpath.h from the Vienna RNA package v1.7.2 for BFS computation. This algorithm computes the full matrix, and as a result, this matrix holds two saddle structures for each pair of hishapes/structures. In many cases, they will be the same, but especially for longer sequences, they may be different, corresponding to the fact that different pathways have been predicted for the forward and backward reaction. When computing energy barriers for individual pairs we take the saddle with lower free energy.

Definition 4 (HiPath energy barrier)

For a given start structure S and target structure T and given that the function HiPath(i, j) computes the saddle structure of a pathway from structure i to structure j and ΔG(X) is the free energy of structure X

\[ E_{HiPath} = -\Delta G(S) + \min(\Delta G(HiPath(S, T), \Delta G(HiPath(T, S))) \tag{6} \]

Pairwise pathway approximation

For the computation of a single pathway between a given start and target structure, we restrict the search space to related hishapes as defined by Equation 5. Additionally, only the shortest path from the start to the target structure is computed. An outline is shown in algorithm 2.

The number of (related) hishapes has a large impact on the speed of the procedure, and thus we provide means to reasonably restrict it. The calculation of (related) hishapes always starts at the most abstract level. If, in this level the number of hishapes does not reach a user defined threshold n, the next less-abstract level is used. This is done either until the threshold n or a user-defined, lowest abstraction level t is reached.

Abstract structure comparison

Any RNA secondary structure can be represented as a node-labeled tree (Zuker and Sankoff 1984; Shapiro 1988; Shapiro and Zhang 1990), and this representation was shown to be especially useful for comparative purposes, such as distance computation and alignment. Hishapes are also inherently tree-like and can thus be represented as trees, too. An example is shown in Figure 5, and the definition is as follows.

Definition 5 (hishape tree)

A hishape tree T consists of a set of helix nodes N that are connected by edges. Each N is a tuple (c, t) where c is the helix index.
and \( t \) the helix type, \( t \in \{ h, m, i, b \} \). If the hishape does not provide the helix index for a certain node, as is the case for multiloop helices in abstraction \( \pi_{h\theta} \), \( \theta \) is set to \(-1\). An edge represents a parent-child relationship according to the nesting of helices. Any \( T \) has a root node \((0, e)\) that corresponds to the external loop. The hishape tree of the empty structure consists of only the root node.

For the comparison of two hishape trees, we follow the idea of tree editing (Zhang and Shasha 1989), which was already applied to abstract trees of RNA secondary structures, where only the loops/helices are represented (Shapiro and Zhang 1990). Here, edit operations act on complete loops/helices which perfectly suits the idea of hishapes. In order to make use of the additional positional information provided by hishapes, we extend the scoring and define the hishape-based tree edit distance (HiTed).

**Definition 6 (hishape-based tree edit distance [HiTed])**

The three edit operations—relabelling, delete, and insert—can be represented as pairs \((a,b)\), \((a,-)\), \((-b)\), respectively. Given two hishape trees \( T_1 \) and \( T_2 \) and a sequence \( S = s_1, \ldots, s_k \) of edit operations

\[
\text{HiTed}(T_1, T_2) = \sum_{i=1}^{k} (D_{TED}(s_i))
\]

\[
D_{TED}(N_1, N_2) = \begin{cases} 
\sigma_{\text{conversion}}(N_1, \text{Null}), & N_2 = \text{Null} \\
\sigma_{\text{conversion}}(\text{Null}, N_2), & N_1 = \text{Null} \\
\lambda \times \sigma_{\text{distance}}(N_1, N_2) + \sigma_{\text{conversion}}(N_1, N_2), & \text{otherwise}
\end{cases}
\]

\[
\sigma_{\text{distance}}(t_1, t_2) = \begin{cases} 
0, & c_1 = -1 \lor c_2 = -1 \\
|c_1 - c_2|, & \text{otherwise}
\end{cases}
\]

\[
\sigma_{\text{conversion}}(t_1, t_2, t_3) = |L[t_1, t_2, t_3]|
\]

The first alternative in Equation 9 considers the case of a multiloop helix, for which no helix index is given in abstraction type \( \pi_{h\theta} \). Values in Equation 11 are taken from Shapiro and Zhang (1990).

In order to find the set of edit operations that minimizes HiTed, we make use of dynamic programming as has been presented by Zhang and Shasha (1989). The implementation makes use of functions from RNA StrAT (Guignon et al. 2005).

### Benchmarking data sets and procedures

**Data set “short”**

Data set “short” is based on the four shortest sequences from the parRNAss evaluation (Voß et al. 2004), namely spliced leader RNA \((L. collosoma)\), pheST Attenuator \((Escherichia coli)\), S15 mRNA leader \((E. coli)\), and the 5’ UTR of MS2 RNA genome. We selected these sequences in order to facilitate a reasonable run time of Barriers, which grows exponential with sequence length and is limited to \( \sim 150 \) nt. For each sequence, the minimum free energy structure was taken as structure A, and a target structure B was determined using the following procedure: predict hishapes using \( \pi_{h\theta} \), scan the energy sorted hishape list for a hishape for which each helix index differs by more than 5 from each helix index of structure A.

**Data set “riboswitches”**

This data set is taken from Li and Zhang (2011) and provides a compilation of seven riboswitch sequences, namely the adenine riboswitch of \( ydhl \) gene from \( B. subtilis \) (Mandal and Breaker 2004), the adenine riboswitch of \( add \) gene from \( Vibrio vulnificus \) (Lemay et al. 2011), the guanine riboswitch of \( xpt-phuX \) operon from \( B. subtilis \) (Mandal et al. 2003), the S-adenosylmethionine riboswitch of \( metI \) from \( Thermoanaerobacter tengcongensis \) (Epshtein et al. 2003), the c-di-GMP riboswitch of the \( tfbX \) from \( Candidatus desulfurococcus \) (Smith et al. 2009), the lysine riboswitch of the \( lysC \) from \( B. subtilis \) (Blouin et al. 2010), and the thiamine pyrophosphate riboswitch of \( thiamin \) from \( B. subtilis \) (Mironov et al. 2002; Rentmeister et al. 2007). Most important about this data set is that it provides the native “on” and “off” conformations of the riboswitches. This allows benchmarking the methods in a realistic scenario.

### Data set and benchmark procedure for structure comparison

For the structure-comparative benchmarks, we took four different data sets from the Brasero (Allali et al. 2012) collection, namely
Measuring run time and memory consumption

For memory usage measurements, we monitored the VmHWM ("high water mark") value in the /proc file system. The run time is the CPU time (sum of user and system time) as measured using GNU time. All measurements were carried out on an 8x AMD Opteron 8378 machine with 128 GB RAM under openSUSE 11.2.

ACKNOWLEDGMENTS

We thank Georg Sauthoff for his work to modify Bellman’s GAP for our purpose. This work was supported by the Deutsche Forschungsgemeinschaft (grant Vo 1450/2-1 to B.V.).

REFERENCES


Abstract folding space analysis based on helices

Jiabin Huang, Rolf Backofen and Björn Voß

RNA 2012 18: 2135-2147 originally published online October 25, 2012
Access the most recent version at doi:10.1261/rna.033548.112

This article cites 39 articles, 15 of which can be accessed free at:
http://rnajournal.cshlp.org/content/18/12/2135.full.html#ref-list-1

Receive free email alerts when new articles cite this article - sign up in the box at the top right corner of the article or click here.