Predicting folding pathways between RNA conformational structures guided by RNA stacks

Yuan Li, Shaojie Zhang*

From ACM Conference on Bioinformatics, Computational Biology and Biomedicine 2011 (ACM-BCB)
Chicago, IL, USA. 1-3 August 2011

Abstract

Background: Accurately predicting low energy barrier folding pathways between conformational secondary structures of an RNA molecule can provide valuable information for understanding its catalytic and regulatory functions. Most existing heuristic algorithms guide the construction of folding pathways by free energies of intermediate structures in the next move during the folding. However due to the size and ruggedness of RNA energy landscape, energy-guided search can become trapped in local optima.

Results: In this paper, we propose an algorithm that guides the construction of folding pathways through the formation and destruction of RNA stacks. Guiding the construction of folding pathways by coarse grained movements of RNA stacks can help reduce the search space and make it easier to jump out of local optima. RNAEAPath is able to find lower energy barrier folding pathways between secondary structures of conformational switches and outperforms the existing heuristic algorithms in most test cases.

Conclusions: RNAEAPath provides an alternate approach for predicting low-barrier folding pathways between RNA conformational secondary structures. The source code of RNAEAPath and the test data sets are available at http://genome.ucf.edu/RNAEAPath.

Introduction

RNA molecules play critical roles in the cell. The secondary structures of RNA molecules have been extensively studied because they provide insights into the functional-ity of RNAs. Native (functional) RNA secondary structures are usually thermodynamically stable and many of them are also the minimum free energy (MFE) structures. Nevertheless, at times, RNA molecules may fold into alternative secondary structures in order to participate in certain biological processes. For example, the SV-11 RNA folds into a metastable conformational structure and acts as a template for its own replication using Qβ replicase [1,2]. Further, RNA conformational switches can transform between alternative secondary structures dynamically in response to various environmental stimuli (such as heat shock and cold shock) [3-6], and carry out RNA-mediated biological activities, such as switching on or off downstream gene translation activities [7-9], regulating RNA splicing via multiple-state splicesomal conformations [10], and regulating the life cycles of virus [11].

The conformational transformations between alternative structures involve the folding of an RNA molecule into a series of sequential adjacent intermediate structures [12]. RNA folding pathways provide valuable information for understanding the catalytic and regulatory functions of RNAs (such as hok/sok of plasmid R1 [13]). RNA folding pathways may also impact subsequence biological events (such as formation of tertiary structures). Furthermore, prediction algorithms can help the design of RNA switches by providing prescribed structural alternatives.

In this paper, we present a new approach, RNAEAPath, for computing near optimal direct or indirect folding pathways between two secondary structures of an RNA molecule. We guide the search for low energy barrier folding pathways by integrating a variety of strategies for
simulating the formation and destruction of RNA stacks in a flexible framework. Benchmark tests on conformational switches show that RNAEAPath produces lower energy barrier folding pathways and outperforms the existing heuristic approaches in most test cases.

Preliminary
Consider an RNA sequence as a string \( x = x_1 \ldots x_n \) of \( n \) letters over alphabet \( \Sigma = \{A, U, G, C\} \). A pair of complementary nucleotides \( x_i \) and \( x_j \) can form hydrogen bonds and interact with each other, denoted by \( x_i \cdot x_j \). In this paper, we only consider the canonical base pairings (A·U and G·C) and the wobble base pairing (G·U). A secondary structure \( S \) of the RNA sequence \( x \) is a set of disjoint paired bases \( (i, j) \), where \( 1 \leq i < j \leq n \). \( S \) may be represented by a length \( n \) string of dots and brackets, where dots represent unpaired bases and brackets represent paired bases. An RNA structure can comprise of stacks which are lists of consecutive base pairs \( ((i_1, j_1), (i_1+1, j_1-1), \ldots, (i, j)) \) such that \( x_{i_1} \cdot x{j_1}, \ldots, x_{i=w} \cdot x_{j-w} \), and unstacking base pairs. A secondary structure is pseudoknotted if it contains two base pairs \( (i, j) \) and \( (i', j') \) with \( i < i' < j < j' \). In this paper, we only consider pseudoknot-free structures. A base pair is compatible with a secondary structure if the base pair can be added to the structure without leading to a pseudoknotted structure or pairing a base with more than one partner. A stack is compatible with \( S \) if each base pair in the stack is either in \( S \) or is compatible with \( S \).

The free energy of a secondary structure \( S \) is denoted by \( E(S) \). The set of neighboring structures of \( S \) consists of all structures that differ from \( S \) by an addition or deletion of exactly one base pair. For two secondary structures \( A \) and \( B \), the distance between \( A \) and \( B \) is the number of base pairs in \( A \) not in \( B \) plus the number of base pairs in \( B \) not in \( A \) (i.e. \( |(A - B) \cup (B - A)| \)). A folding pathway from \( A \) to \( B \) is a sequence of intermediate structures \( S_0, \ldots, S_m = B \) such that for all \( 0 \leq i < m \), intermediate structure \( S_{i+1} \) is a neighboring structure of \( S_i \). A folding pathway is direct if the intermediate structures contain only base pairs in \( A \) and \( B \) (i.e. \( S_i \subseteq A \cup B \) for \( 1 \leq i < m \)) and otherwise is indirect. The saddle point of a pathway is an intermediate structure with the highest energy, and the energy barrier of a pathway is the energy difference between its saddle point and the initial structure. Since the folding of RNA structures is thermodynamically-driven and tends to avoid high-energy intermediate structures, current computational methods aim to find RNA folding pathways with the lowest energy barriers.

Previous studies
A lot of research has been done on predicting low energy barrier folding pathways. Morgan and Higgs proposed a greedy algorithm that employs the Nussinov model [14,15] for computing direct folding pathways with minimum energy barrier. They also described a heuristic that samples low energy structures from the partition function and glues them together by direct pathways [16]. The Nussinov model is simple and easy to implement, in which base stacking and loop entropies have no energetic contributions. Based on this model, Thachuk et al. [17] developed an exact algorithm, PathwayHunter, which exploits elegant properties of bipartite graphs for finding the globally optimal direct pathways. However, the Nussinov model is not as accurate as the Turner energy model [18,19] for approximating RNA thermodynamics. An exact solution based on the Turner energy model is also available. BARRIERS [20,21], exactly computes the globally optimal folding pathways between any two locally optimal secondary structures. BARRIERS reads an energy sorted list of RNA secondary structural conformations produced by RNAsubopt [22] and is able to compute both direct and indirect low energy barrier pathways.

Nevertheless, the above exact solutions are all exponential in time, because the problem itself is NP-hard [23]. Many heuristic algorithms have also been proposed following the seminal work of Morgan and Higgs. Flamm et al. [24] used breadth-first search in their heuristics (in Vienna RNA Package [25]) and kept the best \( k \) candidates at each step to bound the search. Voss et al. [26] devised a straightforward strategy for greedily searching direct pathways. Geis et al. [27] described a greedy heuristic to explore the search space of direct pathways and they also integrated look ahead techniques to diminish the search space. Recently, Dotu et al. [28] developed RNA-TabuPath, a fast heuristic that employs a TABU semi-greedy search to construct near optimal (both direct and indirect) folding trajectories. In addition, other heuristic approaches, by splitting the pathways into shorter pathways and solving each individually, have also been proposed [29,30]. There are also other formula presented for the prediction of RNA folding kinetics (see Flamm and Hofacker’s review [31] for a systematic discussion).

Many of the existing heuristic algorithms start from an initial structure \( A \), and, at each single step \( i \), walk from the intermediate structure \( S_i \) to one of its neighbors \( S_{i+1} \), until finally the end structure \( B \) is reached. The definition of neighborhood relationships as well as the fitness functions can be different. The fitness function of \( S_i \) is usually defined on the free energy of \( S_i \), or the distance from \( S_i \) to \( B \), or a function of both. In general, greedy algorithms select the ‘best’ neighbor structure that has the best fitness. In contrast, semi-greedy algorithms may select any one from the top \( k \) structures for randomization. RNA-TabuPath, which is more sophisticated and outperforms other methods [28], keeps a tabu list for saving recently
Motivations

However, using energy to guide the construction of folding pathways in the above-mentioned heuristic algorithms has its downsides. The RNA energy landscapes can be extremely large and rugged [11,32] and the ruggedness of RNA energy landscape may cause the energy-guided search to become trapped in a local optimum. Similar to using structural rearrangements for modeling RNA folding kinetics [33], we want to construct candidate folding pathways in a manner that make it easier to jump out of local optima. It has been revealed that stacking base pairs contribute significantly to the stabilization of RNA secondary structures [34,35]. The dominant RNA folding pathways involve the formation and destruction of the stacks, and the cooperative formation of a stack along with the partial melting of an incompatible stack [36]. In this paper, we propose to guide the construction of pathways by the formation and destruction of stacks (not by free energy or by distance to the end structure). We still select the constructed folding pathways according to their energy barriers. Although the construction of folding pathways is not driven by thermodynamics, the selection of folding pathways is based on energy barriers. Guiding the construction of folding pathways by coarse grained movements of RNA stacks may help reduce the search space and make it easier to jump out of local optima. In the rest of this paper, the Methods section describes the representation of folding pathways and the detailed strategies employed by RNAEAPath. The Results and Discussion section presents benchmarking results of RNAEAPath against existing methods followed by concluding remarks in the Conclusions section.

Methods

Representation of RNA folding pathways

Given an initial structure \( A \) and an end structure \( B \), we use a sequence of actions successively applied to \( A \), rather than a sequence of intermediate structures, to represent a folding pathway from \( A \) to \( B \). Representing a pathway by an action chain can avoid cyclic additions and deletions of base pairs and make it easy to simulate the formation and deletion of RNA stacks. A similar representation has also been employed in the previous work of Thachuk et al. [17].

We use two types of actions, add\(_{ij}\) and del\(_{ij}\), in the representation of RNA folding pathways. For an intermediate secondary structure \( S \) of an RNA sequence \( x \), the action add\(_{ij}\) denotes the ‘add’ition of base pair \( (i, j) \) to \( S \) (i.e. \( \text{add}_{ij}(S) = S \cup \{(i, j)\} \)) and del\(_{ij}\) denotes the ‘del’etion of base pair \( (i, j) \) from \( S \) (i.e. \( \text{del}_{ij}(S) = S - \{(i, j)\} \)). An action is direct if it concerns a base pair in \( A \cup B \) and indirect otherwise. The simplest direct pathways from \( A \) to \( B \) concern sequential deletions of all base pairs in \( A - B \) followed by additions of all base pairs in \( B - A \).

Consider an example sequence \( x = \text{GGGGAAAA CCCCUUUU} \) with initial and final structures shown in Figure 1. This simple pathway is obtained by first deleting all GC pairs from \( A \) until the RNA is single stranded, and then adding all AU pairs until \( B \) is obtained. Note that each intermediate structure \( S \) differs from both its successor and predecessor by exactly one base pair. The actions in the example are all direct actions and the energy barrier is 5.50 - (-6.60) = 12.10 kcal/mol.

An addition action \( \text{add}_{ij}(S) \) conflicts with \( S \) if either \( x_i \) or \( x_j \) is already paired in \( S \), and it clashes with \( S \) if there exists a base pair \( \{(x'_{i'}, x'_{j'})\} \in S \) if \( i' < i < j' \) or \( i' < i' < j' < j \). A deletion action \( \text{del}_{ij}(S) \) conflicts with \( S \) if \( (x_i, x_j) \notin S \). An addition or deletion action is valid and can be applied to \( S \) properly if it neither conflicts with nor clashes with \( S \).

A pathway from \( A \) to \( B \) can be represented by an action chain, which is a sequence of valid actions \( a_1, \ldots, a_m \) such that \( S_0 = A, S_t = a_t(S_{t-1}) \) for \( 1 \leq t \leq m \) and \( S_m = B \). Note that an action chain for \( A \) to \( B \) implies a sequence of valid actions that can be successively applied to \( A \) without introducing conflicts or clashes and produce \( B \). We use the term “action chain” when the sequence is certified to be valid, and the term “sequence of actions” if its validity is not guaranteed.

This representation of a pathway \( p \) from \( A \) to \( B \) has the following important properties. First, every folding pathway can be represented by a unique action chain and every action chain represents a unique folding pathway (note that it is not necessarily true for a sequence of actions). Second, rearranging the order of actions in \( p \) results in a new sequence of actions which represents a new folding pathway from \( A \) to \( B \) when it is valid. (It is an action chain that can be successively applied to \( A \) properly and obtain \( B \).) Third, introducing a pair of complementary actions (e.g. \( \text{add}_{ij} \) and \( \text{del}_{ij} \)) to \( p \) results in a new sequence of actions which also represents a new folding pathway from \( A \) to \( B \) if it is valid.

In RNAEAPath, folding pathways are represented in the form of action chains, instead of a sequence of intermediate structures. This representation makes the life cycle of a folding pathway transparent to the algorithm.

from a set of neighboring structures that have the top lowest free energy or have the top shortest distance to \( B \) (or the combination of both).

In the rest of this paper, the Methods section describes benchmarking results of RNAEAPath against existing methods followed by concluding remarks in the Conclusions section.
and also makes it easier for us to simulate the cooperative formation and destruction of RNA stacks by rearranging the order of actions or introducing multiple pairs of complementary actions.

Predicting low energy barrier folding pathways

Given an RNA sequence $x$, an initial structure $A$ and a final structure $B$, RNAEAPath computes a near optimal low energy barrier folding pathway from $A$ to $B$ in an evolutionary algorithm framework [37]. Figure 2 elucidates the overall paradigm for RNAEAPath. In this algorithm, the population of each generation is comprised of folding pathways ordered by their fitness. The functions $M_y(p)$ are mutation strategies, each of which takes in a pathway $p$ and produces a set of offspring pathways. These mutation strategies are central to the effectiveness of RNAEAPath and will be discussed in the Mutation strategies subsection.

$\ell_1$, $\ell_2$, $\ell_3$, MAX and $\gamma$ are positive integer control parameters.

The initial population of RNAEAPath, $P_0$, is filled with a set of simple pathways. Then, the algorithm goes through several iterations. $P_{k+1}$ is the population of the $k - 1^{st}$ iteration. In the $k^{th}$ iteration, the algorithm produces $O_k$ (an ordered list of pathways) and $P_k$ (the population of the $k^{th}$ iteration) from $P_{k-1}$. $O_k$ stores the best $\ell_1$ pathways in $P_{k-1}$ and the best $\ell_2$ pathways produced by each $p \in P_{k-1}$. More specifically, each pathway $p \in P_{k-1}$ produces $\ell_3^p$ offsprings through every mutation strategy $M_y(1 \leq y \leq Y)$. The resulting offsprings produced by $p$ are stored in a temporary list $T$, and the top $\ell_2$ pathways are added to $O_k$. Finally, the best solution of the $k^{th}$ iteration, termed as OPT$_k$, is the best pathway in $O_k$. And, $P_k$ (the population of the $k^{th}$ iteration) is composed of the best $\ell_3$ pathways of $O_k$ and will be used in the next iteration to produce $P_{k+1}$.

This helps keep the diversity of the population large, since $P_k$ contains at most $\ell_2$ offsprings produced by each $p \in P_{k-1}$, no matter how many high-qualified offsprings are produced by each pathway. The algorithm terminates when a stopping condition is met, and it returns the best solution of the last iteration. Since $O_k$ retains the best $\ell_1$ pathways from $P_{k-1}$ in each iteration, the best one ever encountered by the algorithm is retained in lists $O_k$ and $P_k$, and stored in OPT$_k$. So, OPT$_k$ has no worse fitness when compared to OPT$_{k-1}$, and RNAEAPath always returns the best action chain it ever discovered.

In the remaining of this section, we discuss details regarding fitness evaluation, initialization of the population, stopping conditions and mutation strategies of RNAEAPath.

Fitness of action chains

The order of folding pathways (valid action chains) is primarily determined by their energy barriers. In case of a tie, the order is determined by the average of energy differences between the initial structure $A$ and intermediate structures. Note that lower energies are preferred in the previous two methods of ordering. If a tie

| Structures | Energy (kcal/mol) | Actions |
|------------|------------------|---------|
| A | (((((....))))... | -6.60 | $a_1$ del1,12 |
| $S_1$ | (((((....))))... | -2.90 | $a_2$ del2,11 |
| $S_2$ | ....((....)).... | 0.40 | $a_3$ del3,10 |
| $S_3$ | ....((....)).... | 3.70 | $a_4$ del4,9 |
| $S_4$ | ....((....)).... | 0.00 | $a_5$ add8,13 |
| $S_5$ | ....((....)).... | 5.50 | $a_6$ add7,14 |
| $S_6$ | ....((....)).... | 4.60 | $a_7$ add6,15 |
| $S_7$ | ....((....)).... | 3.70 | $a_8$ add5,16 |
| B | ....((....)).... | 2.80 | |

Figure 1 A simple folding pathway that converts an RNA sequence from structure $A$ to $B$. A simple folding pathway that converts an RNA sequence from structure $A$ to $B$. The leftmost column shows a simple direct pathway from $A$ to $B$, the center column shows the free energies (in kcal/mol) of the intermediate structures, and the rightmost column presents the action chain $a_1, \ldots, a_8$ for this pathway.
still exists, then shorter action chains are preferred. Action chains are ordered arbitrarily if their relative order cannot be determined based on these three criteria.

The initial population of folding pathways

The initial population, \( \mathbb{P}_0 \), contains 4 simple pathways from \( A \) to \( B \) formed by first deleting all base pairs in \( A - B \) and then adding those in \( B - A \), similar to the pathway shown in Figure 1. Although we can also arrange base pair deletions and additions in an arbitrary order, we tailor them in a manner that simulates successive degradation and formation of RNA stacks. This is because random deletions and additions of base pairs tend to form additional unpaired loop regions that introduce entropic penalties (see Figure 3 for an illustration). We can degrade or form each stack either from the outermost base pair to the innermost base pair and vice versa. Usually, it yields a lower energy barrier if we degrade a stack from the outermost base pair to the innermost base pair and form a stack from the innermost base pair to the outermost base pair. However, for the sake of simplicity and generosity, we construct 4 simple pathways in \( \mathbb{P}_0 \), which degrade all the stacks from the same direction and form all the stacks from the same direction. These simple pathways constitute a diversified and unbiased initial population for the algorithm to start from.

The number of offsprings produced by each mutation strategy

In each generation, the expected total number of offsprings produced by each individual is a constant positive integer \( L \). The number of offsprings that each individual produces using mutation strategy \( M_y \) in the \( k \)-th generation, is denoted by \( \ell_{k}^{M_y} \). In the initial generation, \( \ell_{0}^{M_y} \) is equivalent to \( g/y \) for all the mutation strategies. In the \( k \)-th generation, \( \ell_{k}^{M_y} \) is determined adaptively according to the quality of the offsprings produced using \( M_y \) in the \( k - 1 \)-st iteration. Let \( b_{k-1}^{M_y} \) be the number of offsprings that are both produced through \( M_y \) and selected to construct \( \mathbb{P}_k \), the population of the \( k - 1 \)-st generation. Then, \( \ell_{k}^{M_y} \) in the \( k \)-th generation is computed as follows.

\[
\ell_{k}^{M_y} = \max \left\{ \ell_{k-1}^{M_y}, \min \left( \frac{\sum_{j=1}^{y} (b_{k-1}^{M_y})^{j-1} \ell_{j-1}^{M_y}}{\sum_{j=1}^{y} (b_{k-1}^{M_y})^{j-1} \ell_{j-1}^{M_y}}, \ell_{k-1}^{M_y} \right) \right\}
\]
Mutation strategies that have produced more high quality offsprings in the \((k - 1)\)st iteration are allowed to generate more offsprings in the \(k\)th generation. In contrast, mutation strategies that perform poorly in the \(k - 1\)st generation, are only allowed to generate a small number (\(\nu_{\text{min}}\) with default value 3) of offsprings. Note that the sum of \(\ell_{i}^{k}\) for \(1 \leq y \leq Y\) may be greater than \(\ell\).

**Stopping conditions**

The algorithm terminates when (1) the current best solution achieves the lowest possible value \(|E(B) - E(A)|\), or (2) when no improvement has been found over \(\gamma\) consecutive iterations (a plateau), or (3) when \(\text{MAX}\) number of iterations have passed and successive iterations do not discover better results. Note that the algorithm may simulate further than \(\text{MAX}\) iterations if improvements are made in the very last iteration and it stops immediately if no improvement is made between successive iterations. More specifically, the algorithm stops when any of the following conditions is satisfied:

1. the energy barrier of \(\text{OPT}_{k}\) is equivalent to \(|E(B) - E(A)|\).
2. \(k > \gamma\) and the fitness of \(\text{OPT}_{k}\) is equivalent to that of \(\text{OPT}_{k-1}\).
3. \(k \geq \text{MAX}\) and the fitness of \(\text{OPT}_{k}\) is equivalent to that of \(\text{OPT}_{k-1}\).

**Mutation strategies**

In RNAEAPath, the mutation strategies employed to evolve folding pathways can be categorized into three types: (1) rearranging the order of actions, (2) introducing indirect pathways and (3) formation of a single stack or cooperative conversion of a pair of incompatible stacks. In this section, let \(\mathcal{M}_{1}, \ldots, \mathcal{M}_{Y}\) denote the mutation strategies and let \(p = a_{1}, \ldots, a_{m}\) denote the input pathway \(A = S_{0}, \ldots, S_{m} = B\). For each mutation strategy \(\mathcal{M}_{y}(p)\), we describe the process for generating one new pathway \(q\) using each mutation strategy when given \(p\).

**Type 1: reordering of actions**

As described in the subsection of representation of RNA folding pathways, shuffling the order of actions of the input pathway \(p\) can result in a new pathway from \(A\) to \(B\). In RNAEAPath, two mutation strategies of this type are employed. \(\mathcal{M}_{1}\) changes the position of an arbitrary action, and \(\mathcal{M}_{2}\) swaps the positions of two arbitrary actions.

- **\(M_{1}\)**: Let \(\mathcal{M}^{t_{1}}_{1}(p)\) denote the sequence of actions obtained by first removing an action \(a_{t_{1}}\) (\(1 \leq t_{1} \leq m\)) from \(p\) and then inserting it after \(a_{t_{2}}\), for all \(t_{2} \in \{0, \ldots, t_{1} - 1, t_{1} + 1, \ldots, m\}\). Note that the resulting sequence of actions may not necessarily be a valid action chain. For instance, in Figure 1, \(\mathcal{M}^{t_{1}}_{1}(p) = a_{2}, a_{3}, a_{4}, a_{1}, a_{5}, \ldots, a_{8}\) and \(\mathcal{M}^{t_{1}}_{1}(p) = p\) are valid action chains, while \(\mathcal{M}^{t_{1}}_{1}(p) = a_{1}, a_{6}, a_{2}, \ldots, a_{7}\) is not.

The procedure for computing \(\mathcal{M}^{t_{1}}_{1}(p)\) is described in the following.

1. Choose \(t_{1}\) uniformly at random from the interval \([1, m]\).
2. Compute the interval \([l, u]\), \((t_{1} < l < u < m)\), where \(l\) is the minimum and \(u\) is the maximum such that for all \(t_{2} \in [l, u]\) and \(t_{2} \neq t_{1}\), \(\mathcal{M}^{t_{1}}_{1}(p)\) is a valid action chain.
3. Choose \(t_{2}\) from the interval \([l, u]\).

   1. If \(a_{t_{2}}\) is an addition operation, for all \(l \leq t < t' \leq u\) and \(t \neq t' \neq t_{1}\), the probability of choosing \(t\) is greater than that of \(t'\).
   2. Otherwise (a deletion operation), for all \(l \leq t \leq t' \leq u\) and \(t \neq t' \neq t_{1}\), the probability of choosing \(t\) is less than that of \(t'\).

We do not choose \(t_{2}\) \((t_{2} \neq t_{1})\) uniformly at random in \([l, u]\), instead, we tend to place addition operations in...
the front part of $p$, and deletion operations in the later part of $p$. This is because adding base pairs early and deleting them later during the folding may help stabilize the intermediate secondary structures. (Please see Additional file 1 for the detailed description of the discrete probability.)

$\mathcal{M}_2$: Let $\mathcal{M}_2^{\text{add}}(p)$ denote the sequence of actions obtained by swapping $a_i$ with $a_{i'}$. If the resulting sequence of actions is a valid action chain, let it be $q$; otherwise, restart the process. For example, in Figure 1, $\mathcal{M}_2^{\text{add}}(p)$ is not a valid action chain, while $\mathcal{M}_2^{\text{add}}(p) = a_1, a_4, a_3, a_2, a_5, \ldots, a_8$ is. $t_1$ and $t_2$ are chosen uniformly at random from $[l(t_1, t_2): 1 \leq t_1 < t_2 \leq n]$. $t_2 - t_1$ is chosen uniformly at random from $[l(t_1, t_2): 1 \leq t_1 < t_2 \leq n]$.

Mutation strategies of type 1 provide methods for shuffling the order of actions of an input pathway and generating slightly different new pathways. However, these strategies are not capable of introducing additional (indirect) base pairs, and the offsprings of a direct pathway produced through type 1 strategies are not capable of introducing additional pathways. Similarly, temporary deletion and reformation of a base pair also can create an indirect pathway. Therefore, strategies in Type 3: formation of a single stack and simultaneous formation and destruction of a pair of incompatible stacks (Type 3: formation of a single stack and simultaneous formation and destruction of a pair of incompatible stacks ($\mathcal{M}_3$)).

$\mathcal{M}_4$: Let $\mathcal{M}_4^{\text{add}}(p)$ denote the sequence of actions obtained by forcing the formation of a stack $h$ in $\mathcal{M}_4$ after action $a_i$, where $\mathcal{M}_4$ is compatible with $S_t$. The following describes the procedure for computing $\mathcal{M}_4^{\text{add}}(p)$.

1. Choose $t$ uniformly at random from the interval $[1, m]$, and obtain the associated intermediate structure $S_t$.
2. Find a set of stacks that neither conflict with nor clash with $S_t$, and pick up a stack $stack_h$ uniformly at random from the set.
3. Ensure that each base pair $(i, j)$ in $[stack_h - S_t]$ is sequentially (from the innermost base pair to the outermost base pair) formed after $a_i$.
   3.1. If an action $add_{ij}$ appears in $[a_{i+1}, \ldots, a_m]$, move it up and place it after $a_i$ using strategy $\mathcal{M}_1$.
   3.2. Otherwise, introduce a pair of complementary actions $add_{ij}$ and $del_{ij}$ to $p$ after $a_i$ using strategy $\mathcal{M}_3$.

We can introduce additional stacks that are compatible with $S_t$ using $\mathcal{M}_4$ by forcing a sequence of addition actions successively forming base pairs in $[stack_h - S_t]$, after $a_i$.

$\mathcal{M}_5$: Let $\mathcal{M}_5^{\text{add}}(p)$ denote the sequence of actions obtained by forcing the formation of a stack $stack_h \in STA$ which
is incompatible with $S_t$ after action $a_t$. Shown on the right side of Figure 4 is a folding pathway which simultaneously destructs and forms a pair of incompatible stacks. Shown on the left side is a simple folding pathway which has exactly the same start and end structures, while it folds into a single stranded structure during the folding. Usually, the pathway on the right has lower energy barrier than the one on the left because it never folds into a single stranded structure. The folding pathway on the right side of Figure 4 can be introduced using strategy $M_1$. And, the procedure for computing $M_1^{\text{inh}}(p)$ is as follows:

1. Choose an arbitrary deletion action $a_t = \text{del}_{i,j}$ from $p$, and obtain the associated intermediate structure $S_t$.
2. Find a set of stacks which either conflicts with or clashes with $S_t$, and choose a stack $\text{stack}_h$ uniformly at random from the set.
3. For each base pair $(i', j')$ in $\text{stack}_h - S_t$ that is compatible with $S_t$, place $\text{add}_{i',j'}$ to $p$ after $a_t$ using strategy $M_4$.
4. For each base pair $(i', j')$ in $\text{stack}_h - S_t$ that is incompatible with $S_t$,
   4.1. Find all the base pairs $(i^*, j^*)$ in $S_t$ that are incompatible with $(i', j')$, and ensure that each

| Folding pathway 1 | Folding pathway 2 |
|-------------------|-------------------|
| (((((((((.............)))))))))) | (((((((((.............)))))))))) |
| (((((((.................))))))) | (((((((.................))))))) |
| (((((........................)))))) | (((((........................)))))) |
| (((..................................)))) | (((..................................)))) |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |
| ........................................... | ........................................... |

**Figure 4** Two different folding pathways with the identical initial and final secondary structures. Two different folding pathways with the identical initial and final secondary structures. Left: Folding pathway 1 destroys a stack completely before an incompatible stack is formed. Right: Folding pathway 2 destructs a stack and forms an incompatible stack simultaneously.
base pair \((i^*, j^*)\) is deleted before the action \(a \overline{d}_{i^*, j^*}\).

4.2. If a action \(d_{i^*, j^*}\) appears in \(\{a_1, \ldots, a_m\}\), move it up before \(a \overline{d}_{i^*, j^*}\) using strategy \(M_1\).

4.3. Otherwise, introduce a pair of complementary actions \(d_{i^*, j^*}\) and \(a \overline{d}_{i^*, j^*}\) using strategy \(M_3\).

Using \(M_5\), we can introduce the simultaneous formation of a stack \(\text{stack}_h\) which is incompatible with \(S_p\) and destruction of existent stacks (or base pairs) that hamper the formation of \(\text{stack}_h\). Since cooperative formation and destruction of stacks may contribute additional stacking energies for stabilizing the intermediate structures, better folding pathways with lower energy barriers may be rendered.

**Results and discussion**

**Benchmark tests**

We benchmarked RNAEAPath against existing methods (BARRIERS [20,21], PathwayHunter [17], Find-path [24], and RNATabuPath [28]) by predicting low energy barrier folding pathways between two designated RNA secondary structures of 18 conformational switches. All the conformational switches were taken from the work of Dotu et al. [28]. Five of them are riboswitches, including rb1, rb2, rb3, rb4, and rb5. The metastable structures of these riboswitches have been experimentally determined by inline probing [9,39]. The thirteen remaining cases concern conformational switches, including hok, SL (Spliced leader RNA), s15, s-box leader, thiM leader, ms2, HDV, dsrA, ribD leader, amv, alpha operon and HIV-1 leader. Sequences of these conformational switches can also be obtained from paRNAss web site [40], and some of the metastable secondary structures were computationally determined using RNAbor [41].

We summarize the results computed by PathwayHunter, the results computed by BARRIERS, the results computed by Findpath (with the look ahead parameter \(k = 10\), the best results over 1000 runs found by RNATabuPath, and the best results over 1 run and 5 runs found by RNAEAPath in Table 1 respectively. And we use '-' to mark test cases that methods fail to apply to in the table.

For all methods, free energies of the intermediate structures of the folding pathways (including PathwayHunter) are evaluated based on the Turner model using RNAeval (with -d1 option) from the Vienna RNA Package [25]. The default configuration parameters of RNAEAPath are as follows. \(\text{MAX}\) is 10, \(\gamma\) is 5, \(\xi_1\) is 100, \(\ell_1\) is 10, \(\ell_2\) is 5 and \(\ell_3\) is 100. Due to the stochastic nature of the evolutionary algorithm, we report the best energy barrier of RNAEAPath found over both 1 run and 5 runs.

BARRIERS is the only exact solution that produces indirect pathways based on the Turner model. BARRIERS has already been compared with existing heuristic algorithms on the same test cases in the work of Dotu et al. [28]. We put the results of BARRIERS in the table.

**Table 1** Benchmarks of BARRIERS, PathwayHunter, Findpath, RNATabuPath, and RNAEAPath for predicting folding pathways between conformational switches on the 18 test cases

| Instance       | BARRIERS | PathwayHunter | FindPath | RNAATabuPath | RNAEAPath |
|----------------|----------|---------------|----------|--------------|-----------|
|                |          |               |          | \((n = 1000)\) | \((n = 1)\) | \((n = 5)\) |
| rb1            | -        | -             | 24.04    | 24.04        | 22.2      | 22.0      |
| rb2            | -        | 10            | 8.2      | 7.25         | 6.5       | 6.5       |
| rb3            | -        | -             | 22.4     | 17.9         | 17.5      | 16.7      |
| rb4            | -        | -             | 16.9     | 16.9         | 16.9      | 16.9      |
| rb5            | -        | -             | 24.54    | 24.54        | 21.44     | 21.44     |
| hok            | -        | -             | 28.5     | 29.66        | 20.7      | 20.1      |
| SL             | 11.80    | -             | 13       | 12.9         | 13.0      | 12.9      |
| attenuator     | 8.3      | -             | 8.7      | 8.6          | 8.7       | 8.5       |
| s15            | 6.60     | -             | 7.1      | 6.6          | 7.1       | 7.1       |
| sbox leader    | -        | 7.9           | 5.2      | 5.2          | 5.2       | 5.2       |
| thiM leader    | -        | -             | 16.13    | 14.84        | 12.3      | 12.3      |
| ms2            | -        | 11.6          | 6.6      | 6.6          | 6.6       | 6.6       |
| HDV            | -        | 23.53         | 174      | 170          | 16.8      | 16.8      |
| dsrA           | 8.0      | -             | 8.3      | 8.2          | 8.0       | 8.0       |
| ribD leader    | -        | -             | 10.71    | 9.5          | 9.5       | 9.5       |
| amv            | -        | 12.2          | 5.8      | 5.8          | 5.74      | 5.74      |
| alpha operon   | -        | 11.8          | 6.5      | 6.5          | 6.1       | 6.1       |
| HIV-1 leader   | -        | 14.3          | 9.3      | 11.3         | 8.9       | 8.9       |

Benchmarks of BARRIERS, PathwayHunter, Findpath, RNATabuPath, and RNAEAPath for predicting folding pathways between conformational switches. Energy barriers (measured in kcal/mol) of the best folding pathways over \(n\) runs are shown. Boldface numbers are the best energy barriers found by the heuristic algorithms.
just for the sake of comparison. It has been pointed out that BARRIERS gives provably globally optimal pathways in 4 out of 18 cases (i.e. SL, attenuator, s15 and dsrA). BARRIERS cannot be directly applied to 5 cases because either the initial or the end structure is not locally optimal (i.e. rb2, sbox leader, ms2, amv and alpha operon), and cannot converge in the remaining cases. Possibly due to the fact that both the number of RNA secondary conformations to consider and the computational resources required increase exponentially with the growing length of the RNA sequence and the growing range of energy barrier. PathwayHunter is an exact algorithm capable of producing the optimal direct folding pathways based on the Nussinov model. PathwayHunter can be directly applied to 10 cases, because it requires the pair of input structures being able to form a 'pairwise-optimal' bipartite conflicting graph (see the work of Thachuk et al. [17] for details). It is not surprising that the performance of the exact algorithm, PathwayHunter, evaluated by free energy (in kcal/mol), is worse than the heuristic algorithms. This is because PathwayHunter is optimized based on the Nussinov model and only produces direct pathways, while the optimal direct pathways predicted based on the Nussinov model may not be the optimal pathways (considering both direct and indirect pathways) based on the Turner model. All the remaining three methods are heuristics capable of producing both direct and indirect pathways based on the Turner model. The energy barriers predicted by RNAEAPath over 5 runs are exactly the same as RNATabuPath in 5 cases, worse in 1 case, and better in all the remaining 12 cases.

By analyzing the best folding pathways produced by RNAEAPath, we found that most high-quality pathways involve the melting of stacks in the initial structure, the (possibly simultaneous) construction of stacks in the final structure, and the formation of auxiliary temporary stacks for obtaining folding pathways with lower energy barriers. We may take the lowest energy barrier folding pathway of rb2 found by RNAEAPath, shown in Figure 5.
as an example. The stack colored in red is an auxiliary temporary stack introducing intermediate structures with lower free energies (which is constructed using $M_4$). Some of the stacks in the initial structure (in blue) are gradually melting, while at the same time, an incompatible stack (in green) is being formed (which is constructed using $M_5$). The stack colored in red is an auxiliary temporary stack introducing intermediate structures with lower free energies. This example convinces us that the advantages of RNAEAPath mainly come from employing mutation strategies that guide the construction of folding pathways by the formation and destruction of stacks and introducing additional stacking interactions that are important for stabilizing the intermediate structures.

Detailed low energy barrier folding pathways for all the test cases are available on RNAEAPath web site.

Control parameters and performance
In order to evaluate the performance of RNAEAPath with different parameter configurations, we played with several other control parameters, including $\ell_1$, the number of top offsprings preserved in the next generation, varying from 1 to 16, $\ell_3$, the size of population in each generation, varying from 80 to 120 and $L$, the total number of offsprings each individual is expected to produce, varying from 80 to 120. The detailed results are shown in Additional file 1. In general, RNAEAPath produces pathways of roughly the same quality for most
test cases with different control parameters, among which the default parameter setting is the best.

We explored the relationship between the performance of RNAEAPath and the number of generations completed by plotting energy barriers of the best folding pathways produced by RNAEAPath with the default parameters in each generation, as shown in Figure 6. In general, the energy barriers decrease dramatically in the first one or two generations, and then the decrements slow down and finally plateau within 10 generations. For instance, in the case of rb3, the predicted energy barriers of folding pathways in the initial population is 27.3 kcal/mol. It decreases by 7.2 kcal/mol (24.9%) through the first two generations and decreases by 2.5 kcal/mol (9.2%) through the next three generations. Through all the remaining generations, no further improvement is made.

We also evaluated the execution time for each run of RNAEAPath. All the tests were performed on a 32 bit PC with 2.4 GHz Quad-processor and 3.2 GB memory, running Fedora 11. With the default control parameters, RNAEAPath terminates in 1 minute in the best case (rb4), 445 minutes in the worst case (hok), and 43 minutes on average. The detailed running times are shown in Additional file 1. We did not perform direct comparisons between the running time of RNATabuPath and that of RNAEAPath, since RNATabuPath is only accessible via web server.

Conclusions
In conclusion, we have presented a new algorithm, RNAEAPath, for predicting low energy barrier folding pathways between conformational structures. RNAEAPath guides the construction of folding pathways through the destruction and formation of RNA stacks using various types of mutation strategies, and integrates them in a well-established computational framework of evolutionary algorithm. These mutation strategies can help reduce the search space and make it easier to jump out of local optima. By analyzing the results, we confirmed that most of the best folding pathways involve the formation of auxiliary stacks, or involve the cooperative formation and disruption of incompatible stacks. The benchmarking results show that RNAEAPath outperforms the existing heuristics on most test cases. We believe that this is because the construction of folding pathways in RNAEAPath captures important biological findings.

Additional material

Additional file 1: Supplementary data for predicting folding pathways between RNA conformational structures guided by RNA stacks. Supplementary data for predicting folding pathways between RNA conformational structures guided by RNA stacks in a PDF file.
24. Flamm C, Hofacker IL, Maurer-Stroh S, Stadler PF, Zehl M. Design of multistable RNA molecules. *RNA* 2001, 7:254-265.
25. Hofacker IL. Vienna RNA secondary structure server. *Nucleic Acids Res* 2003, 31:3429-3431.
26. Voss B, Meyer C, Geigerch R. Evaluating the predictability of conformational switching in RNA. *Bioinformatics* 2004, 20:1573-1582.
27. Geis M, Flamm C, Wolfinger MT, Tanzer A, Hofacker IL, Middendorf M, Mandl C, Stadler PF, Thumer C. Folding kinetics of large RNAs. *J Mol Biol* 2008, 379:160-173.
28. Dotu I, Lorenz WA, Van Hentenryck P, Clote P. Computing folding pathways between RNA secondary structures. *Nucleic Acids Res* 2010, 38:1711-1722.
29. Bogomolov S, Mann M, Voß B, Podelski A, Backofen R. Shape-based barrier estimation for RNAs. German Conference on Bioinformatics 2010, 41-50.
30. Lorenz R, Flamm C, Hofacker IL. Beyond energy minimization: Approaches to the kinetic folding of RNA. *Monatsh Chem* 2008, 139:447-457.
31. Flamm C, Hofacker IL. A complex adaptive systems approach to the kinetic folding of RNA. *Biosystems* 2005, 82:257-265.
32. Sponer J, Leszczynski J, Hobza P. Nature of nucleic acid-base stacking: non-empirical ab initio and empirical potential characterization of 10 stacked base dimers. Comparison of stacked and H-bonded base pairs. *J Phys Chem* 1996, 100(13):5590-5596.
33. Yakovchuk P, Protozanova E, Frank-Kamenetski MD. Base-stacking and base-pairing contributions into thermal stability of the DNA double helix. *Nucleic Acids Res* 2006, 34:564-574.
34. Zhao P, Zhang WB, Chen SJ. Predicting secondary structural folding kinetics for nucleic acids. *Biphot* 2010, 98:1617-1625.
35. Freyhult E, Moulton V, Clote P. Boltzmann probability of RNA structural neighbors and riboswitch detection. *Bioinformatics* 2007, 23:2054-2062.