# 2D Projections of RNA folding Landscapes

Ronny Lorenz, Christoph Flamm, Ivo L. Hofacker
{ronny, xtof, ivo}@tbi.univie.ac.at

Institute for Theoretical Chemistry
University of Vienna, Währingerstraße 17, 1090 Wien, Austria

**Abstract:** The analysis of RNA folding landscapes yields insights into the kinetic folding behavior not available from classical structure prediction methods. This is especially important for multi-stable RNAs whose function is related to structural changes, as in the case of riboswitches. However, exact methods such as *barrier tree* analysis scale exponentially with sequence length. Here we present an algorithm that computes a projection of the energy landscape into two dimensions, namely the distances to two reference structures. This yields an abstraction of the high-dimensional energy landscape that can be conveniently visualized, and can serve as the basis for estimating energy barriers and refolding pathways. With an asymptotic time complexity of $\mathcal{O}(n^7)$ the algorithm is computationally demanding. However, by exploiting the sparsity of the dynamic programming matrices and parallelization for multi-core processors, our implementation is practical for sequences of up to 400 nt, which includes most RNAs of biological interest.

## 1 Introduction

Structure formation of RNA molecules is crucial for the function of non-coding RNAs (ncRNAs) as well as for coding mRNAs with regulatory elements like riboswitches and attenuators. Some RNAs possess distinct meta-stable structures with different biological activity. A prime example are riboswitches that regulate gene expression depending on the presence or absence of a small ligand molecule. The pathogenity of viral agents like viroids is achieved by distinct meta stable structures of their 'genome'. While efficient RNA folding algorithms such as `mfold` [Zuk89] or the Vienna RNA package [HFS$^+$94] can be used to compute most equilibrium properties of an RNA molecule, they provide little information on folding dynamics. In this case one has to resort to either stochastic simulation of the folding process [FHMS$^+$01, IS00, GFW$^+$08] or analysis of the energy landscape based on enumeration or sampling. In particular the `barriers` program [FHSW02] is used to find all local minima in the energy landscape and their connecting transition states and energy barriers. The algorithm is based on a complete enumeration of all low-energy conformations [WFHS99] in the landscape and therefore scales exponentially with sequence length. In contrast, the `paRNAss` tool [GHR99] relies on sampling structures and clustering in order to detect multi-stable RNAs but gives no information on energy barriers.

The dynamic programming (DP) approach to RNA folding can also be extended to obtain

more information on the energy landscape: Cupal et al. [CHS96] proposed an algorithm that computes the density of states, i.e. the number of structures that fall into a particular energy bin, by extending the usual DP table into a third dimension corresponding to the energy bins. The `RNAbor` algorithm [FMC07], uses the base pair distance to a reference structure as the additional dimension and computes the optimal secondary structure as well as partition function for each distance class ($\delta$-neighborhood).

Both approaches can be viewed as a one-dimensional projection of the high dimensional energy landscape, which however results into a drastic loss of information. Here, we describe a related method that employs the distances to two reference structures in order to compute a 2D projection which retains enough information to predict qualitative folding behavior and is easy to visualize. In particular, we define a $\kappa, \lambda$ - *neighborhood* to be all secondary structures $s$ with $d_{\text{BP}}(s_1, s) = \kappa$ and $d_{\text{BP}}(s_2, s) = \lambda$, where $d_{\text{BP}}(s_a, s_b)$ is the base pair distance of $s_a$ and $s_b$, and proceed to compute minimum free energy (MFE) structure, as well as partition function and Boltzmann weighted structure samples for each $\kappa, \lambda$ - *neighborhood* .

## 2 Methods

### 2.1 Minimum free energy algorithm

In the following we will write $(i, j)$ to denote a base pair between the $i$th and $j$th nucleotide. A secondary structure $s$ is regarded as a set of base pairs, the *base pair distance* between two structures is defined as $d_{\text{BP}}(s_1, s_2) = |s_1 \cup s_2| - |s_1 \cap s_2|$ and equals the number of base pairs present in either but not both structures. We will write $s[i, j] = \{(p, q) \in s : i \leq p < q \leq j\}$, to identify the substructure on the sequence interval $[i, j]$. $E(s)$ denotes the free energy of structure $s$.

For reference, we reproduce below the classic recurrences for MFE folding, which we will extend to the $\kappa, \lambda$ - *neighborhood* in the following section. Note that the recursions employ an unambiguous decomposition of secondary structures as implemented in the Vienna RNA package [HFS$^+$94].

$$
\begin{aligned}
F_{i,j} &= \min \left\{ F_{i,j-1}, \min_{i<k\leq j} F_{ik} + C_{k+1,j} \right\} \\
C_{i,j} &= \min \left\{ \mathcal{H}(i,j), \min_{i<k<l<j} C_{kl} + \mathcal{I}(i,j;k,l), \min_{i<u<j} M_{i+1,u} + \hat{M}_{u+1,j-1} + a \right\} \\
M_{i,j} &= \min \left\{ \min_{i<u<j} (u-i-1)c + C_{u+1,j} + b, \min_{i<u<j} M_{i,u} + C_{u+1,j} + b, M_{i,j-1} + c \right\} \\
\hat{M}_{i,j} &= \min \left\{ \hat{M}_{i,j-1} + c, C_{ij} + b \right\}
\end{aligned}
\tag{1}
$$

The upper triangular matrices $F_{i,j}$, $C_{i,j}$, $M_{i,j}$ and $\hat{M}_{i,j}$ contain the optimal folding energy on the sequence interval $[i, j]$, optimal energy given that $(i, j)$ form a pair, given that $i$ and $j$

reside in a multi-loop, and for multi-loop components with exactly one stem in the interval $[i, j]$, respectively. $\mathcal{H}(i, j)$ denotes the energy of a haipin-loop closed by $(i, j)$, $\mathcal{I}(i, j, p, q)$ the energy of an interior-loop closed by $(i, j)$ and $(p, q)$. The parameters $a$, $b$, and $c$ contain the penalties for closing a multi-loop, for adding a multi-loop component, and enlarging a multi-loop by one unpaired base. We use the energy parameters as tabulated by the Turner group [MSZT99]. After filling the matrices the MFE structure is found by backtracking in the usual manner.

### 2.2 Minimum free energy $\kappa, \lambda$-neighbors

For a given RNA sequence $\mathcal{S}$ and two fixed reference structures $s_1$ and $s_2$, the MFE version of the $\kappa, \lambda$ - *neighborhood* algorithm computes energetically optimal structures $s_{\text{opt}}^{\kappa,\lambda} \in S^{\kappa,\lambda}$ where $S^{\kappa,\lambda} = \{s \mid d_{\text{BP}}(s_1, s) = \kappa \wedge d_{\text{BP}}(s, s_2) = \lambda\}$ is the $\kappa, \lambda$ - *neighborhood* of reference structure $s_1$ and $s_2$. We extend the recursions (1) such that for each entry of the energy matrices $F$,$C$,$M$ and $\hat{M}$ the optimal energy contribution of substructures $s[i, j]$ with $d_{\text{BP}}(s_1[i, j], s[i, j]) = \kappa$ and $d_{\text{BP}}(s_2[i, j], s[i, j]) = \lambda$ are computed. This leads to two additional dimensions in the energy matrices denoted by $F^{\kappa,\lambda}$,$C^{\kappa,\lambda}$,$M^{\kappa,\lambda}$ and $\hat{M}^{\kappa,\lambda}$. Since closing base pairs may lead to an increase of the base pair distance to both reference structures $s_1$ and $s_2$, additional decomposition constraints have to be introduced in the recurrences.

A hairpin loop closed by $(i, j)$, for example, contributes to $C_{i,j}^{\kappa,\lambda}$ only if the substructure $s[i, j]$ consisting of the single pair $\{(i, j)\}$ only has distances $\kappa$ and $\lambda$ to the two substructures $s_1[i, j]$ and $s_2[i, j]$, respectively. Thus, we introduce the shorthand

$$\mathfrak{H}(i, j, \kappa, \lambda) = \begin{cases} \mathcal{H}(i, j) & \text{if } d_{\text{BP}}(s_1[i, j], \{(i, j)\}) = \kappa,\ d_{\text{BP}}(s_2[i, j], \{(i, j)\}) = \lambda \\ \infty & \text{else} \end{cases} \tag{2}$$

For non-hairpin loops, we introduce five terms $\delta_1^x - \delta_5^x$, where the superscript $x$ is either 1 or 2, denoting the reference structure.

$$\delta_1^x(i, j) = d_{\text{BP}}(s_x[i, j], s_x[i, j-1]) \tag{3}$$
$$\delta_2^x(i, j, u) = d_{\text{BP}}(s_x[i, j], s_x[i, u-1] \cup s_x[u, j]) \tag{4}$$
$$\delta_3^x(i, j, p, q) = d_{\text{BP}}(s_x[i, j], \{(i, j)\} \cup s_x[p, q]) \tag{5}$$
$$\delta_4^x(i, j, u) = d_{\text{BP}}(s_x[i, j], \{(i, j)\} \cup s_x[i+1, u] \cup s_x[u+1, j-1]) \tag{6}$$
$$\delta_5^x(i, j, u) = d_{\text{BP}}(s_x[i, j], s_x[u, j]) \tag{7}$$

Each of the $\delta$ in eqs. (3-7) covers a distinct case in the energy minimization recursions, and denotes the minimal distance to the reference structure incurred when decomposing a substructure into two parts (since base pairs in the reference structure crossing the decomposition splitting positions $j$, $u$, $p$ and $q$ must be opened). For example $\delta_1^x(i, j)$ equals 1, if $j$ is paired in the structure interval $s_x[i, j]$ of the reference structure $s_x$ and 0 otherwise.

Decompositions into more than one substructure lead to additional combinatorial possibilities. They are taken into account by minimizing over $(\omega, \hat{\omega})$ pairs, where the sum $(\omega + \hat{\omega})$

reflects the residual of the base pair distance between the substructures and the references. Thus, the recursions to compute $E(s_{\text{opt}}^{\kappa,\lambda}) = F_{1,n}^{\kappa,\lambda}$ are:

$$
\begin{aligned}
F_{i,j}^{\kappa,\lambda} &= \min \begin{cases} F_{i,j-1}^{\kappa-\delta_1^1(i,j),\lambda-\delta_1^2(i,j)}, \\ \min\limits_{i\le u<j} \min\limits_{\substack{\omega_1+\hat{\omega}_1=\kappa-\delta_2^1(i,j,u) \\ \omega_2+\hat{\omega}_2=\lambda-\delta_2^2(i,j,u)}} F_{i,u-1}^{\omega_1,\omega_2} + C_{u,j}^{\hat{\omega}_1,\hat{\omega}_2} \end{cases} \\
C_{i,j}^{\kappa,\lambda} &= \min \begin{cases} \mathfrak{H}(i,j,\kappa,\lambda), \\ \min\limits_{i<p<q<j} \left\{ C_{p,q}^{\kappa-\delta_3^1(i,j,p,q),\lambda-\delta_3^2(i,j,p,q)} + \mathcal{I}(i,j,p,q) \right\}, \\ \min\limits_{i<u<j} \min\limits_{\substack{\omega_1+\hat{\omega}_1=\kappa-\delta_4^1(i,j,u) \\ \omega_2+\hat{\omega}_2=\lambda-\delta_4^2(i,j,u)}} \left\{ M_{i+1,u}^{\omega_1,\omega_2} + \hat{M}_{u+1,j-1}^{\hat{\omega}_1,\hat{\omega}_2} + a \right\} \end{cases} \\
M_{i,j}^{\kappa,\lambda} &= \min \begin{cases} M_{i,j}^{\kappa-\delta_1^1(i,j-1),\lambda-\delta_1^2(i,j)} + c \\ \min\limits_{i\le u<j} \left\{ (u-i)\cdot c + C_{u,j}^{\kappa-\delta_5^1(i,j,u),\lambda-\delta_5^2(i,j,u)} + b \right\}, \\ \min\limits_{i\le u<j} \min\limits_{\substack{\omega_1+\hat{\omega}_1=\kappa-\delta_2^1(i,j,u) \\ \omega_2+\hat{\omega}_2=\lambda-\delta_2^2(i,j,u)}} \left\{ M_{i,u-1}^{\omega_1,\omega_2} + C_{u,j}^{\hat{\omega}_1,\hat{\omega}_2} + b \right\}, \end{cases} \\
\hat{M}_{i,j}^{\kappa,\lambda} &= \min \begin{cases} C_{i,j}^{\kappa,\lambda} + b \\ \hat{M}_{i,j-1}^{\kappa-\delta_1^1(i,j),\lambda-\delta_1^2(i,j)} + c, \end{cases}
\end{aligned} \tag{8}
$$

### 2.3 Time and memory complexity

Regarding the time complexity of the algorithm, a contribution of $\mathcal{O}(n^3)$, where $n$ denotes RNA sequence length is implicit due to the underlying MFE folding algorithm. The additional degrees of freedom of the multi-loop decompositions in $C_{i,j}^{\kappa,\lambda}$ and $M_{i,j}^{\kappa,\lambda}$ increase the complexity by a factor of $\kappa \cdot \lambda$. The extension of the dynamic programming matrices by two further dimensions $\kappa$ and $\lambda$ additionally requires quadratically more effort. If the maximum distance values of $\kappa$ and $\lambda$ is limited to $\kappa \le d_1$ and $\lambda \le d_2$, the time complexity becomes $\mathcal{O}(n^3 \cdot d_1^2 \cdot d_2^2)$. Since the maximum number of base pairs on a sequence of length $n$ is $\sim \frac{n}{2}$, the maximum achievable base pair distance between any two structures is bounded by $n$. Thus, the total asymptotic time complexity of the $\kappa, \lambda$ - *neighborhood* algorithm results in $\mathcal{O}(n^7)$ for any distance boundaries $d_1$ and $d_2$.

A similar argument holds for the memory complexity which is $\mathcal{O}(n^2 \cdot d_1 \cdot d_2) = \mathcal{O}(n^4)$. Thus, the memory increase compared to regular MFE folding is $d_1 \cdot d_2 \le n^2$.

### 2.4 Partition function of the $\kappa, \lambda$ - *neighborhood*

A modification of algorithm (8) to compute the partition function

$$Q^{\kappa,\lambda} = \sum_{s_x \in S^{\kappa,\lambda}} e^{-E(s_x)/kT} \tag{9}$$

for each $\kappa, \lambda$ - *neighborhood* according to the algorithm of McCaskill et al. [McC90] is straight forward. The energy contributions are Boltzmann weighted and all sums/minimizations are replaced by products/sums. This can be done, as the recursions (8) perform unique decompositions and therefore already constitute a partitioning.

Since clustering of the complete secondary structure space into $\kappa, \lambda$ - *neighborhoods* is a partitioning too, $\sum_{\kappa,\lambda} Q^{\kappa,\lambda} = Q$, where $Q = \sum_s e^{-E(s)/kT}$ is the partition function of the complete ensemble of all secondary structures.

The Boltzmann probabilities of a $\kappa, \lambda$ - *neighborhood* in the complete ensemble and for a structure $s_x \in S^{\kappa,\lambda}$ inside a $\kappa, \lambda$ - *neighborhood* become

$$P(S^{\kappa,\lambda}) = \frac{Q^{\kappa,\lambda}}{Q} \quad \text{and} \quad P(s_x \in S^{\kappa.\lambda}) = \frac{e^{-E(s_x)/kT}}{Q^{\kappa,\lambda}} \tag{10}$$

Stochastic backtracking yields a Boltzmann weighted sample of representative structures.

### 2.5 Sparse matrix approach and Parallelization

Some properties of the $\kappa, \lambda$ - *neighborhood* can be used to improve the runtime and reduce memory requirements. Due to the definition of the $\kappa, \lambda$ - *neighborhood* of two structures $s_1$ and $s_2$, there exist combinations of $\kappa, \lambda$ distance pairs which do not contribute to any solution. For example, there is no $\kappa, \lambda$-neighbor with $\kappa + \lambda < d_{\text{BP}}(s_1, s_2)$. An increase or decrease of the base pair distance of a structure $s$ to one of the reference structures implicitly changes the base pair distance to the other reference. In particular, if $d_{\text{BP}}(s_1, s_2) =$ even (resp. odd), then $\kappa + \lambda =$ even (resp. odd). This checkerboard-like pattern of the $\kappa, \lambda$ - *neighborhood* roughly halves the number of entries in the extended dimensions $\kappa$ and $\lambda$ actually needed for the calculations. Furthermore, the maximum distance $d_{\text{max}}$ to any reference structure in any substructure $s[i, j]$ of length $m = j - i + 1$ is constrained to $d_{\text{max}} < m$. These observations introduce sparsity in the dynamic programming matrices. Hence, two-dimensional matrices $F, C, M, \hat{M}$ with lists of triples, containing energy $E$, distance $\kappa$ and distance $\lambda$ at each matrix entry can be used. By iterating over the list instead of all $\kappa, \lambda$ combinations, impossible structure formations are avoided.

Further runtime improvements can be obtained through parallelization by noting that all entries of the matrices $F$, $C$, $M$ and $\hat{M}$ with $j - i =$ const. can be computed concurrently if the matrices are filled in diagonal order, see [FHS00].

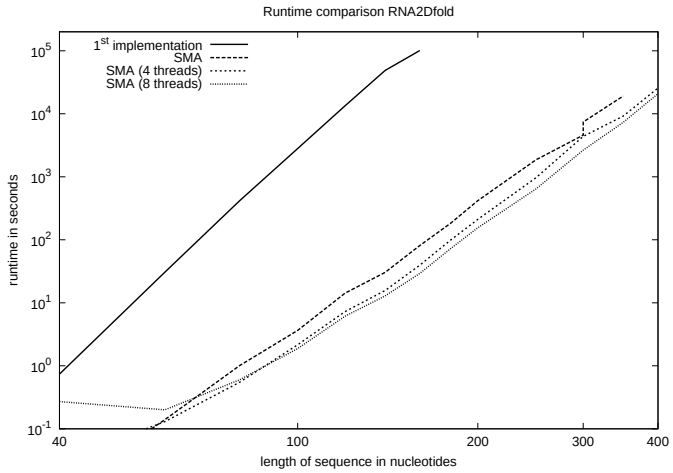


Figure 1: Runtimes of MFE calculation for the complete $\kappa, \lambda$ - *neighborhood* . Timings are given for naïve approach ($1^{st}$ implementation) and the sparse matrix approach (SMA) using $1$, $4$ and $8$ threads on a dual quad-core Intel® Xeon® E5450 @3.00GHz with 32GB RAM. Runtimes are means of $15$ random sequences. Reference structures used were the MFE structure and the open chain. With $8$ processor cores, a sequence of 400 nt can be processed in about $5.8h$.

# 3 Results

## 3.1 Implementation

The partition function as well as the MFE version of the $\kappa, \lambda$ - *neighborhood* algorithm was implemented in `ISO C` and will be available as a stand-alone program `RNA2Dfold` in one of the next releases of the Vienna RNA Package. A release candidate is available from `http://www.tbi.univie.ac.at/~ronny/RNA/`. The implementation provides most of the command line options of `RNAfold` such as different dangling end models and temperature. Given an RNA sequence $\mathcal{S}$ and two reference structures $s_1$ and $s_2$, `RNA2Dfold` computes for each $\kappa, \lambda$ - *neighborhood* the MFE structure $s_{\text{opt}}^{\kappa,\lambda}$ and its free energy, the probabilities $P(S^{\kappa,\lambda})$, $P(s_{opt}^{\kappa,\lambda} \in S^{\kappa.\lambda})$, the probability of $s_{opt}^{\kappa,\lambda}$ in the complete ensemble and the Gibbs free energy $\Delta G^{\kappa,\lambda}$. The maximum values $d_1$ and $d_2$ with $\kappa \leq d_1$ and $\lambda \leq d_2$ can be specified by the user.

For parallelization we used *OpenMP* which allows efficient use of modern multi-core systems while requiring only small changes to the serial version of the source code. The performance gain from exploiting sparsity as well as parallelization is demonstrated in Fig. 1. The resulting speedups for 4 and 8 cores were 2.0 and 2.9, respectively. On modern multi-core systems `RNA2Dfold` can easily compute the MFE structures and partition functions for all $\kappa, \lambda$ - *neighborhoods* for RNA sequences up to about 400 nt. This length range covers functional RNAs such as riboswitches and viroids.

### 3.2 2D Projection of the energy landscape

The probability densities and partition functions calculated by `RNA2Dfold` can be used for several secondary structure space analysis. One of the possible applications of the $\kappa, \lambda$ - *neighborhood* algorithm is the prediction of metastable structure states and the detection of bi-stable RNA switches. Typically, the MFE structure is used as the first reference structure $s_1$. A meta-stable state, suitable as second reference structure $s_2$, can be obtained e.g. from a first run of `RNA2Dfold` using the open chain as second reference, and selecting $s_2$ from the $s_{\text{opt}}^{\kappa,\lambda}$. We note that this provides an alternative to the `paRNAss` approach for detecting RNA switches that avoids sampling errors. Computing the $\kappa, \lambda$ - *neighborhood* of $s_1$ and $s_2$ and plotting the MFE values, probability densities and/or the Gibbs free energy of the partitioned landscape as a two dimensional height map reveals a qualitative picture of the roughness of the landscape. In the examples of Fig. 2 both RNAs can be clearly recognized as bi-stable switches. Molecules with more than 2 long-lived meta-stable states should exhibit additional minima in the interior of the height map. Furthermore, the height map yields a lower bound on the energy barrier between $s_1$ and $s_2$, and indicates the difficulty of refolding from $s_1$ to $s_2$ and vice versa.

### 3.3 A heuristic for finding non direct refolding paths

Most existing approaches utilize heuristics that consider only direct (minimal length) refolding paths between two states [MH98, FHMS$^+$01]. Since direct paths allow no detours, potentially stabilizing base pairs which are not in either of the two ground states cannot be formed. This can lead to intermediate structures with energetically unfavorable loop motifs and thus unnecessarily high energy barriers. In contrast, a refolding path with guaranteed minimal barrier can be obtained from the `barriers` program [FHSW02]. Since the approach is based of exhaustive enumeration of the energy landscape, it is limited to short sequences, typically less than a 100 nt.

The $\kappa, \lambda$ - *neighborhood* can be used as base for various heuristics estimating the refolding path and energy barrier. Note however that taking the representatives $s_{\text{opt}}^{\kappa,\lambda}$ from a series of adjacent $\kappa, \lambda$-neighbors does usually not yield a continuous path of adjacent structures. Nevertheless, the height map already provides a lower bound for the height of the transistion state and therefore for the energy barrier too. Direct path heuristics perform poorly when the two structures are far apart. Therefore, a natural extension is to construct an intermediate structure $s_m$, termed *mesh-point*, thus splitting the path construction problem between the two reference structures $s_1$ to $s_2$ into two path constructions from $s_1$ to $s_m$ and from $s_m$ to $s_2$. The problem is of course to find suitable mesh points, and the $\kappa, \lambda$ - *neighborhoods* turn out to be an excellent starting point for this. The `Pathfinder` algorithm given below (3.1) connects such mesh points using the direct path heuristic from [FHMS$^+$01]. The method produces indirect paths, since the mesh points need not lie on a shortest path between $s_1$ and $s_2$.

After computing the $\kappa, \lambda$ - *neighborhood* of the start ($s_1$) and target ($s_2$) structure, we test

**Algorithm 3.1** Pseudo-code of the Pathfinder($s_a$, $s_b$, iter) algorithm where $s_a$ is the start structure, $s_b$ is the stop structure, iter is the maximal number of iterations

```
MeshpointHeap ← ∅ /* initialize min-order mesh-point heap */
bestpath ← DirectPath(s_a, s_b) /* get best refolding path so far */
while Meshpoints available ∧ |MeshpointHeap| < m do
    s ← Meshpoint structure /* sample a mesh point structure */
    path ← DirectPath(s_a, s) + DirectPath(s, s_b)
    if Barrier(path) < Barrier(bestpath) then
        insert(MeshpointHeap, (s, path, Barrier(path)))
    end if
end while
if iter > 0 then
    for 0 . . . m do
        (s, path) ← pop(MeshpointHeap)
        path ← Pathfinder(s_a, s, iter − 1) + Pathfinder(s, s_b, iter − 1)
        if Barrier(path) < Barrier(bestpath) then
            bestpath ← path
        end if
    end for
else
    (s, path) ← pop(MeshpointHeap)
    bestpath ← path
end if
return bestpath
```

as mesh-points all MFE structures $s_{\text{opt}}^{\kappa,\lambda}$ where $\kappa+\lambda \leq \gamma$ with a constant $\gamma$. This constraint limits the maximal deviation from a direct path and allows an adjustable exploration of the underlying energy landscape. Clearly, it is possible to recursively subdivide the problem further if required (see Pseudocode). With this simple approach the `Pathfinder` algorithm is able to find refolding paths with energy barriers very close or identical to those of an exhaustive search using the `barriers` program [FHSW02]. Results for an artificially designed RNA switch of 45 nt length revealed a barrier height of 10.7 kcal/mol (see Fig. 2A) which is the same as found with a barrier tree analysis. In contrast to that, a direct path generated according to [FHMS$^+$01] predicts an energy barrier of 13.33 kcal/mol. As mentioned before, the heightmap already provides a lower bound for the energy barrier. Here, direct paths are bounded by at least 13.3 kcal/mol, while indirect paths are bounded by 10.0 kcal/mol. Refolding between the aptamer- and non-aptamer fold of the *add*-riboswitch [RLGM07] (Fig. 2B) also shows the same energy barrier of 6.77 kcal/mol for both, Pathfinder and barrier tree analysis, while a direct path exhibits 7.28 kcal/mol.

## 4 Conclusion

We introduced a method for a unique partitioning of the RNA secondary structure space, in which structures are lumped together according to their base pair distances to two reference structures. In effect, this provides a 2D projection of the high-dimensional folding space. To overcome the high time complexity of $\mathcal{O}(n^7)$ our implementation exploits the sparseness of the dynamic programming matrices as well as *OpenMP* parallelization. The

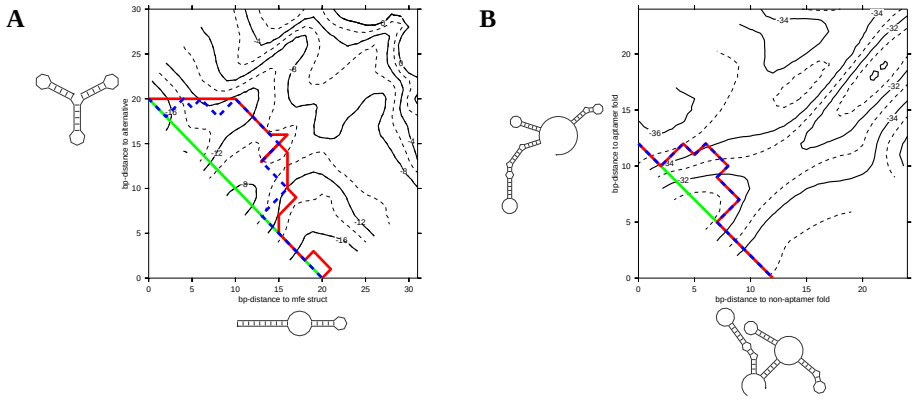


Figure 2: Gibbs free energy height map of all $\kappa, \lambda$ - *neighborhoods* and projection of the refolding paths generated by `barriers` (red line) and the `Pathfinder` (blue dashed line) without recursive refinement. Mesh-points are taken from the $\kappa, \lambda$ - *neighborhood* of both reference structures. **A**: MFE- and alternative structure of an artificial RNA switch with sequence `GGGCGCGGUUCGCCCUCCGCUAAAUGCGGAAGAUAAAUUGUGUCU` and meta stable structure conformations `(((((.....)))))((((((.....))))))((((((.....))))))` *(MFE structure)* and `((((((((((.....(((((.....)))))).....))))))))))` *(alternative structure)*. **B**: Aptamer- and non-aptamer fold of an *add*-riboswitch [RLGM07]. In contrast to direct paths (straight diagonal green line), the `Pathfinder` solution is as good as the (optimal) solution generated by `barriers` in both cases. In **B**, identical refolding paths are obtained for `Pathfinder` and `barriers` analysis.

resulting program is fast enough to treat RNA molecules up to 400 nt which covers most biologically interesting cases such as riboswitches and viroids.

The $\kappa, \lambda$ - *neighborhoods* provide both a qualitative picture of the energy landscape, as well as a convenient starting point for more detailed exploration. As an example we show that it can be used to suggest excellent intermediate nodes for the construction of refolding paths, resulting in a fast heuristic that often gives optimal results. Such heuristics are needed e.g. for kinetic folding strategies like `Kinwalker` [GFW+08]. Furthermore, the height maps could provide the starting point for methods that recognize RNA switches or for coarse grained folding simulations.

## Acknowledgments

This work has been funded, in part, by the Austrian GEN-AU projects “bioinformatics integration network III” and “non coding RNA”.

## References


[CHS96] J. Cupal, I.L. Hofacker, and P.F. Stadler. Dynamic programming algorithm for the density of states of RNA secondary structures. *Computer Science and Biology*, 96(96):184–186, 1996.

[FHMS+01] C. Flamm, I.L. Hofacker, S. Maurer-Stroh, F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7:254–265, 2001.

[FHS00] Martin Fekete, Ivo L. Hofacker, and Peter F. Stadler. Prediction of RNA base pairing probabilities using massively parallel computers. *J. Comp. Biol.*, 7:171–182, 2000.

[FHSW02] Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. Barrier Trees of Degenerate Landscapes. *Z. Phys. Chem.*, 216:155–173, 2002.

[FMC07] E. Freyhult, V. Moulton, and P. Clote. Boltzmann probability of RNA structural neighbors and riboswitch detection. *Bioinformatics*, 23(16):2054, 2007.

[GFW+08] M. Geis, C. Flamm, M.T. Wolfinger, A. Tanzer, I.L. Hofacker, M. Middendorf, C. Mandl, P.F. Stadler, and C. Thurner. Folding kinetics of large RNAs. *Journal of Molecular Biology*, 379(1):160–173, 2008.

[GHR99] R. Giegerich, D. Haase, and M. Rehmsmeier. Prediction and visualization of structural switches in RNA. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, page 126, 1999.

[HFS+94] I.L. Hofacker, W. Fontana, P.F. Stadler, S. Bonhoeffer, M Tacker, and P. Schuster. Fast Folding and Comparison of RNA Secondary Structures. *Monatshefte f. Chemie*, 125:167–188, 1994.

[IS00] H Isambert and E D Siggia. Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme. *Proc Natl Acad Sci U S A*, 97(12):6515–20, Jun 2000.

[McC90] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990.

[MH98] S.R. Morgan and P.G. Higgs. Barrier heights between ground states in a model of RNA secondary structure. *Journal of Physics A-Mathematical and General*, 31(14):3153–3170, 1998.

[MSZT99] D.H. Mathews, J. Sabina, M. Zuker, and D.H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5):911–940, 1999.

[RLGM07] R. Rieder, K. Lang, D. Graber, and R. Micura. Ligand-induced folding of the adenosine deaminase A-riboswitch and implications on riboswitch translational control. *Chembiochem*, 8(8), 2007.

[WFHS99] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–165, February 1999.

[Zuk89] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244(4900):48–52, April 1989.