Computational Design of RNAs with Complex Energy Landscapes

Christian Höner zu Siederdissen[a], Stefan Hammer[a], Ingrid Abfalter[b], Ivo L. Hofacker[a,c,d], Christoph Flamm[a,*], Peter F. Stadler[e,f,g,a,d,h,*]

[a] *Department of Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria*
[b] *Research Support, Johannes Kepler University Linz, Altenberger Str. 69, 4040 Linz, Austria*
[c] *Bioinformatics and Computational Biology Research Group, University of Vienna, A-1090 Währingerstraße 17, Vienna, Austria*
[d] *Center for non-coding RNA in Technology and Health, University of Copenhagen, Grønnegårdsvej 3, DK-1870 Frederiksberg, Denmark*
[e] *Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, Härtelstraße 16-18, D-04107, Leipzig, Germany*
[f] *Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany*
[g] *RNomics Group, Fraunhofer Institut für Zelltherapie und Immunologie, Deutscher Platz 5e, D-04103 Leipzig, Germany*
[h] *Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA*

Abstract

RNA has become an integral building material in synthetic biology. Dominated by their secondary structures, which can be computed efficiently, RNA molecules are amenable not only to *in vitro* and *in vivo* selection, but also to rational, computation-based design. While the inverse folding problem of constructing an RNA sequence with a prescribed ground-state structure has received considerable attention for nearly two decades, there have been few efforts to design RNAs that can switch between distinct prescribed conformations.

We introduce a user-friendly tool for designing RNA sequences that fold into multiple target structures. The underlying algorithm makes use of a combination of graph coloring and heuristic local optimization to find sequences whose energy landscapes are dominated by the prescribed conformations. A flexible interface allows the specification of a wide range of design goals. We demonstrate that bi- and tri-stable "switches" can be designed easily with moderate computational effort for the vast majority of compatible combinations of desired target structures. `RNAdesign` is freely available under the GPL-v3 license.

Keywords: RNA sequence design; inverse folding; multi-stable structures; graph coloring

*Corresponding authors

1. Introduction

A wide variety of RNA elements requires transitions between two or more different spatial conformations. A prime example are riboswitches. These regulatory elements, which are abundant in prokaryotes, regulate mRNA transcription or translation in response to metabolite concentrations, reviewed by Serganov and Nudler (2013). Substrate binding or unbinding at the aptamer component of the riboswitch triggers a conformational change of the molecule that is propagated to the effector location, where it causes the formation or destruction of a terminator hairpin, or the exposure or sequestration of the Shine-Dalgarno sequence. RNA thermometers are a variation of this theme (Kortmann and Narberhaus, 2012). Similar mechanisms have been reported in eukaryotic genome regulation for elements in untranslated parts of mRNAs that respond to protein binding (Ray *et al.*, 2009). Major conformational changes also play a crucial role in viroid processing (Baumstark *et al.*, 1997), in the replication cycle of self-replicating RNA synthesized by Qβ-replicase (Biebricher *et al.*, 1992), the folding of rRNA after excision of self-splicing introns (Cao and Woodson, 2000), and the functioning of the hok/sok host-killing system (Gultyaev *et al.*, 1997).

Regulatory RNA elements that respond to external triggers are attractive components in synthetic biology (Wieland *et al.*, 2009; Win *et al.*, 2009; Topp and Gallivan, 2010), making the design of novel RNA components an interesting task of practical importance (Isaacs *et al.*, 2006). Recent success in designing a synthetic riboswitch acting on transcription emphasizes the feasibility and usefulness of rational design approaches for RNAs with distinct, prescribed conformations (Wachsmuth *et al.*, 2013). Similar principles have been used to construct a riboswitch based on IRES structures (Ogawa, 2011).

The task of designing an RNA sequence with a prescribed secondary structure as its ground state is known as the "inverse folding problem". Although this combinatorial problem is hard in general (Schnall-Levin *et al.*, 2008), most instances of practical interest can be solved by simple hill-climbing heuristics: An initial random seed is progressively "mutated" to approach the desired folding properties. This simple idea is the basis of `RNAinverse` (Hofacker *et al.*, 1994) and later, more efficient approaches such as `RNA-SSD` (Andronescu *et al.*, 2004; Aguirre-Hernández *et al.*, 2007), as well as the very efficient optimization algorithm (Zadeh *et al.*, 2011b) implemented in `NUPACK` (Zadeh *et al.*, 2011a). `INFO-RNA` (Busch and Backofen, 2006) uses a dynamic programming approach to compute the most stable sequence for the prescribed secondary structure as a starting point for a local search heuristic. A multi-objective optimization approach considering the trade-off between thermodynamic stability and structural similarity is used in `MODENA` (Taneda, 2011). Inverse folding problems can also be solved by an exact branch and bound algorithm (Burghardt and Hartmann, 2007). An alternative, essentially enumerative approach that covers certain classes of pseudoknots is described in (Gao *et al.*, 2010). `RNAexinv` (Avihoo *et al.*, 2011) includes some additional attributes and also the mutational robustness and the minimum free energy. As an alternative to iterative improvement, a global sampling approach was proposed in Levin *et al.* (2012).

Much less is known about the design problem for multi-stable RNAs. In this case, the design goals involve more complex properties of the energy landscape such as prescribed local optima and energy barriers. A web tool for this type of design problem is `ARDesigner` (Shu *et al.*, 2010), which implements many of the ideas discussed in Flamm *et al.* (2001). The most salient difference between the inverse folding problem for single and multiple structural constraints is that a solution need not exist in the latter case (Flamm *et al.*, 2001). Thus, computing feasible solutions as starting points for subsequent optimization steps, and —in particular— sampling these starting points so that biases can be avoided, becomes a non-trivial problem. Flamm *et al.* (2001) describe a uniform sampling procedure for two prescribed secondary structures. In this case, a non-empty set of feasible solutions always exists (Reidys *et al.*, 1997).

The general case has been discussed by Abfalter *et al.* (2003), but no corresponding software has become available. Lyngsø *et al.* (2012) implemented a much simpler, approximate sampling of initial conditions together with a genetic algorithm in their `Frnakenstein` tool. Similar ideas have been used by (Ramlan and Zauner, 2011). In the present contribution, we consolidate and expand on our earlier computational approaches to designing RNA sequences with multiple prescribed conformations that satisfy additional, complex constraints and provide with `RNAdesign` an implementation for a wide variety of RNA design tasks.

2. Theory

2.1. Notation

Let $\mathcal{A}$ denote the alphabet of monomers and let $\mathcal{B} \subset \mathcal{A} \times \mathcal{A}$ be the set of allowed base pairs. We assume that $\mathcal{B}$ is symmetric. For RNA, we have $\mathcal{A} = \{A, U, G, C\}$

and $\mathcal{B} = \{AU, UA, GC, CG, GU, UG\}$. We denote a sequence consisting of n monomers $x_i \in \mathcal{A}$ by $x = x_1 x_2 \dots x_n$.

A secondary structure Θ on x is a set of pairs (i, j), $1 \le i < j \le n$ such that for all $(i, j), (k, l) \in \Theta$ holds

1. $(x_i, x_j) \in \mathcal{B}$
2. $(i, j) = (k, l)$ or $\{i, j\} \cap \{k, l\} = \emptyset$, i.e., Θ is a matching on $\{1, 2, \dots, n\}$
3. If $i < k < j$ or $i < l < j$, then $i < k < l < j$, i.e., base pairs do not cross.

Given a secondary structure Θ, we write

$$\mathbf{C}[\Theta] = \left\{ x \in \mathcal{A}^n \middle| (x_i, x_j) \in \mathcal{B} \text{ for all } (i, j) \in \Theta \right\}$$

for the set of sequences that can form the structure Θ. We say that a sequence $x \in \mathbf{C}[\Theta]$ is compatible with Θ.

To every pair (x, Θ) of a sequence x and a secondary structure Θ compatible with x, an energy $f(x, \Theta)$ can be assigned. In practice, $f(x, \Theta)$ is computed as the sum of energy contributions of stacked base pairs and loops, which in turn are derived from a large body of accurate thermodynamic measurements (Mathews *et al.*, 2004).

The energy landscape for a fixed sequence x is defined by the function $f_x : \Theta \mapsto f(x, \Theta)$ together with an adjacency relation $\sim$ defined between secondary structures. As usual, we regard two secondary structures as adjacent if they differ by a single base pair. Later, we will use properties of f_x in the specification of the design goals. This energy landscape is a high-dimensional combinatorial object that cannot be visualized in its entirety.

Coarse-grained representations must thus be employed. Ding *et al.* (2005), proposed clustering of a Boltzmann sample. Quarta *et al.* (2009) favoured a scatter plot of folding energy versus base pair distance from the ground state. `RNA2Dfold` (Lorenz *et al.*, 2009) considers an abstracted energy surface with two anchor points. Throughout this presentation we will make use of the barrier tree of the landscapes as a comprehensive presentation (Flamm *et al.*, 2000; Wolfinger *et al.*, 2004). The leaves of the barrier tree are the local minima (metastable states) of the landscapes, which are connected by the saddle points separating them. The height of a node corresponds to the energy of the corresponding secondary structure, so that both energy differences between (meta)stable states and their separating barriers can be read off the tree immediately.

For the examples in this contribution we use exhaustive enumeration with the programs `RNAsubopt`

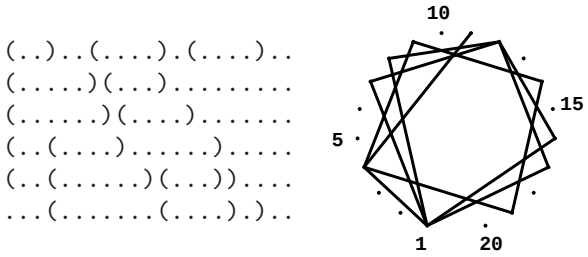

Figure 1: The $M = 6$ secondary structures on the l.h.s. give rise to the dependency graph G in which each edge corresponds to a base pair in at least one of the input structures. Each edge is thus constrains the set of possible sequences: the endpoints of each edge must be different nucleotides that can pair with each other. To make the example smaller, the minimum number of unpaired positions in a hairpin is reduced to 2 here. (Adapted from (Abfalter *et al.*, 2003))

(Wuchty *et al.*, 1999) and `barriers` to construct exact barrier trees. Approximations could be obtained by folding algorithms that directly address metastable structures (Waldispühl and Clote, 2007; Li and Zhang, 2011) and heuristics to estimate saddle points (Morgan and Higgs, 1998). Using the barrier trees enables a wide variety of design goals to be expressed in a concise manner.

Now consider a collection $\{\Theta_1, \Theta_2, \dots, \Theta_M\}$ of M distinct secondary structure of the same length n. Is there a sequence x that is simultaneously consistent with all the Θ_i? If so, our task is to determine x such that all the prescribed Θ_i features as prominently as possible among the structures formed by x. We first address the existence question.

2.2. The Search Space $\mathscr{C}$

Given $\{\Theta_1, \Theta_2, \dots, \Theta_M\}$, the set of sequences simultaneously consistent with all these secondary structures is

$$\mathscr{C} = \mathbf{C}[\Theta_1] \cap \mathbf{C}[\Theta_2] \cap \dots \cap \mathbf{C}[\Theta_M] \tag{1}$$

Hence, the design problem is solvable if $\mathscr{C} \neq \emptyset$. This question is addressed in Flamm *et al.* (2001).

The *dependency graph* $G = G(\Theta_1, \Theta_2, \dots, \Theta_M)$ has n vertices corresponding to the sequence positions of x. There is an edge connecting $k \in V(G)$ with $l \in V(G)$ if and only if (k, l) is a base pair in at least one of the secondary structures Θ_i, i.e.,

$$E(G) = \bigcup_{i=1}^{M} \Theta_i \tag{2}$$

see Fig. 1.

Generalized Intersection Theorem. (Flamm *et al.*, 2001) Suppose $\mathcal{B} \subseteq \mathcal{A} \times \mathcal{A}$ contains at least one symmetric

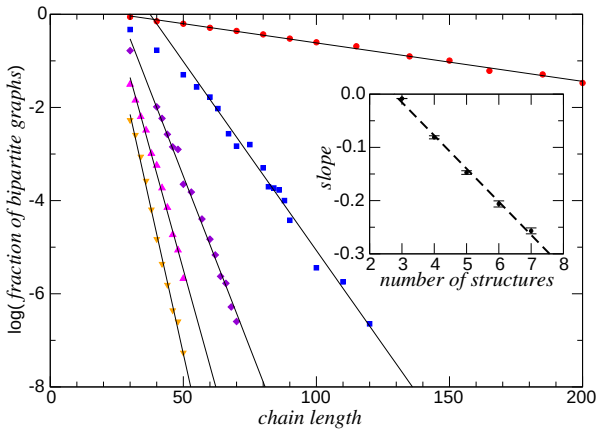

Figure 2: Statistics of the fraction of bipartite graphs versus sequence length with different numbers M of prescribed structures generated with uniform distribution for the set of all secondary structures of fixed chain length. • $M = 3$, ■ $M = 4$, ♦ $M = 5$, ▲ $M = 6$, and ▼ $M = 7$.

base pair, and G is the dependency graph of a set of secondary structures. Then:

1. $\mathscr{C} \neq \emptyset$ if G is bipartite. For the RNA alphabet, bipartiteness of G is also a necessary condition.
2. There are $|\mathscr{C}| = \prod_{\text{components } \psi \text{ of } G} F(\psi)$ sequences in $\mathscr{C}$, where $F(\psi)$ is the number of sequences that are compatible with a connected component ψ of G.

The proof of the intersection theorem makes use of two ingredients. (1) Base-pairing in the natural alphabet divides the letters into two subsets $V_1 = \{G, A\}$ and $V_2 = \{C, U\}$ with base-pairs allowed only between the subsets but not within them. (2) Since edges in the dependency graph are base pairs and must have a nucleotide from V_1 at one end and a nucleotide from V_2 on the other end, it must be possible to color the vertices of the dependency graph with V_1 and V_2. A simple breadth-first-search coloring algorithm can be used to test whether G is bipartite.

Even for $M = 3$ different secondary structures, it is simple to construct triples of structures with conflicting base pairs that lead to a triangle in $G(\Theta_1, \Theta_2, \Theta_3)$. In order to estimate the probability of a non-empty $\mathscr{C}$, we sampled secondary structures with uniform probability as described by Tacker *et al.* (1996) and checked whether the dependency graph of M-tuples of structures is bipartite. For $M \geq 3$, we find an exponential decrease with sequence length, see Fig. 2. However, the exponent is very small for $M = 3$, indicating that tri-stable switches in particular should not be uncommon.

The exponential decrease with length n can be explained as follows: The obstructions to bipartiteness can be small, i.e., triangles corresponding to just three incompatible base pairs in three sequences. It appears with some finite probability in a triple of positions. Hence the chance to avoid such configurations in long sequences decreases exponentially in the case of random, unrelated input structures. If the mutual structure distances are bounded, however, so is the chance to find inconsistent configurations.

2.3. Sequence Design as Graph Coloring

In this section, we outline a dynamic programming approach that can be used to enumerate and uniformly sample from $\mathscr{C}$. To this end, we consider sequences as $\mathcal{A}$-colorings of the dependency graph G, that is, as maps $c : V(G) \to \mathcal{A}$ which obey the pairing rules, i.e., $(c(k), c(l)) \in \mathcal{B}$ for all $(k, l) \in E(G)$.

The important observation for our purposes is that colorings can be obtained by combining partial colorings: Let H be a subgraph of G, and consider two vertex sets $U, W \subseteq V(H)$. A partial coloring of U in H is a map $c_U : U \to \mathcal{A}$ such that $(c(u), c(v)) \in \mathcal{B}$ for all $u, v \in U$ with $(u, v) \in E(H)$. Partial colorings c_U and c_W on U and W, respectively, are compatible if (i) $c_U(y) = c_W(y)$ for all $y \in U \cap W$ and (ii) $(c_U(u), c_W(v)) \in \mathcal{B}$ for $u \in U$ and $v \in W$ with $(u, v) \in E(H)$. Denote by $\partial(U, W)$ the set of vertices in which U and W overlap or are adjacent. Denote by $\mathbf{c}(U, a)$ and $\mathbf{c}(W, b)$ the sets of all those colorings on U and W that are fixed to some assignments a and b on $\partial(U, W)$. Then the set of colorings in $U \cup W$ consists exactly of the combinations of colorings on U and W for which a and b are consistent, i.e., identical on $U \cap W$ and satisfying the color constraints on adjacent vertices. For simplicity, write $\mathbf{c}(U \cup W) = \bigcup_{a,b} \mathbf{c}(U) \circ \mathbf{c}(U \cup W)$

The idea is to use this type of composition of the set of all conflict-free colorings for the step wise construction of $\mathbf{c}(G)$. Graph coloring is a well-known NP-complete problem (Jensen and Toft, 1994). Of course, our approach cannot overcome this in general. We can, however, search for a decomposition of G that allows us to concatenate partial colorings with as little resource consumption as possible.

It is particularly easy to compose colorings at cut vertices. In the first step, we therefore decompose (each connected component of) G into its blocks, i.e., the two-connected components and those edges that are not contained in a cycle, Fig. 3. The blocks and cut-vertices, i.e., the vertices common to two or more blocks, can be determined in linear time. For each block B, we then determine the sets of colorings $\mathbf{c}(B, q)$ with fixed colors

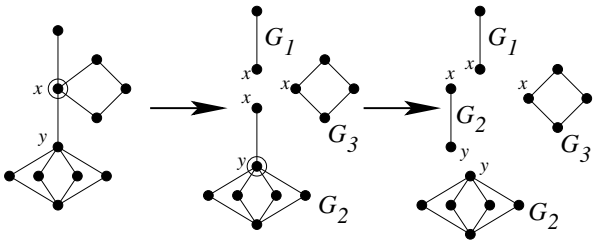

Figure 3: Decomposition of the dependency graph of Fig. 1 into its blocks. Colors need to be constrained only at the cut vertices x and y. The number of colorings in this example is
$|\mathbf{c}(G)| = \sum_{c_y} |\mathbf{c}(G_2'', c_y)| \cdot \sum_{c_x} |\mathbf{c}(G_2', c_x c_y)| \cdot |\mathbf{c}(G_1, c_x)| \cdot |\mathbf{c}(G_3, c_x)|$.
(Adapted from (Abfalter *et al.*, 2003))

q assigned to cut vertices of G in B. The two-connected components are arranged in a tree. Choosing an arbitrary root, we can compose the colorings recursively by traversing from the leafs up.

Since two-connected components can be large, we need to decompose them further. While it might seem natural to use successive higher-order connectivities for this purpose, we explore here an alternative approach.

Let G be a two-connected graph. An ear decomposition of G is a sequence $\mathcal{E} = (P_0, P_1, \dots)$ of paths where $G_0 = P_0$ is a single vertex,

$$G_k = \bigcup_{i=0}^{k} P_i \tag{3}$$

and $G_\mu = G$, with $\mu = |E| - |V| + 1$ being the dimension of the cycle space of G. An ear decomposition is "open" if P_i, $i \geq 1$, has two distinct end points in G, see Fig. 4 for an example. An open ear decomposition exists exactly for two-connected graphs (Whitney, 1932). We use the EDS algorithm by Maon *et al.* (1986) to produce the decomposition. For details, see Sec. 3.1.

With the ear decomposition $\mathcal{E}$ of G we associate a sequence of subgraphs of G for which we construct the colorings:

$$\overline{G}_k = \bigcup_{i=k+1}^{\mu} P_i \tag{4}$$

By definition $\overline{G}_0 = G$ and $\overline{G}_\mu = \varnothing$, the empty graph. Further, we have

$$\overline{G}_k = P_{k+1} \cup \overline{G}_{k+1} \tag{5}$$

The intersection $A_k := G_k \cap \overline{G}_k$ is completely disconnected for each k and by construction forms a cut in G. We call these vertex sets the *attachment points* of $\overline{G}_k$ on G_k.

Our task is now to construct and evaluate the sets $\mathbf{c}(\overline{G}_k, a_k)$ of colorings of the graph $\overline{G}_k$ with colors a_k fixed on the set A_k of its attachment points. To this end, we start from the outer-most path $\overline{G}_{\mu-1}$, for which the colorings are easily constructed and counted, and proceed inwards until we reach $\overline{G}_0 = G$.

These sets $\mathbf{c}(\overline{G}_k, a_k)$ can be computed by combining, in the above sense, colorings of the path P_{k+1} with colorings of the subgraph $\overline{G}_{k+1}$, again with prescribed assignments a_{k+1} at its attachment points A_{k+1}.

$$\mathbf{c}(\overline{G}_k, a_k) = \mathbf{c}(P_{k+1}, b) \circ \mathbf{c}(\overline{G}_{k+1}, a_{k+1}) \tag{6}$$

Since A_k, A_{k+1}, and P_{k+1} are not disjoint in general, the colorings a_k, b, and a_{k+1} at the sets of attachment points must, of course, coincide at their intersections. However, as P_{k+1} and $\overline{G}_{k+1}$ are not connected by any other edges in G, the concatenation $\circ$ of the coloring sets is constrained only by the common vertices. In particular, the end points of P_k are attachment points in A_k, and the attachment points of $\overline{G}_{k+1}$ are either contained in the interior of P_{k+1} (b and a_{k+1} coincide on $A_{k+1} \setminus A_k$), or they are attachment points of $\overline{G}_{k+1}$, and thus $a_k = a_{k+1}$ on these vertices.

The path P_{k+1} is subdivided by the interior attachment points into $|A_{k+1} \setminus A_k| + 1$ sub-paths. For any coloring condition $\mathcal{B}$, it is straightforward to compute the set of colorings on a path of length ℓ with fixed colors at its endpoints, see e.g. Flamm *et al.* (2001). From these, colorings of longer paths with fixed colors at the attachment points are easily obtained.

This decomposition of the sets of colorings forms the basis for the recursive enumeration of colorings by dynamic programming. In each decomposition step k, we need to store the number of colorings $|\mathbf{c}(\overline{G}_k, a_k)|$ of $\overline{G}_k$ given a fixed coloring of the attachment vertices, i.e., $|\mathcal{A}|^{|A_k|}$ values. The maximum $\alpha = \max_k |A_k|$ over the steps of the ear decomposition thus determines the memory requirements.

The CPU time required to compute one entry in this matrix is determined by the set $A_{k+1} \setminus A_k$ of attachment points of $\overline{G}_{k+1}$ that are attached to the interior of P_{k+1}. The total effort to count all colorings is therefore $|\mathcal{A}|^\beta$ with $\beta = \max_k(|A_k| + |A_{k+1} \setminus A_k|)$. The exponents α and β depend explicitly on the spanning tree of G used in the construction of the ear decomposition. Fig. 5 shows that α and β can vary dramatically, depending on the choice of the the spanning tree. Note that α and β are strongly correlated. The data suggest that $|A_{k+1} \setminus A_k| \in \{0, 1, 2\}$, i.e., in each step at most two earlier attachment points are consumed.

2.4. Generating Colorings

For small connected components of $\mathscr{C}$ it is possible (and efficient) to explicitly enumerate all colorings. For

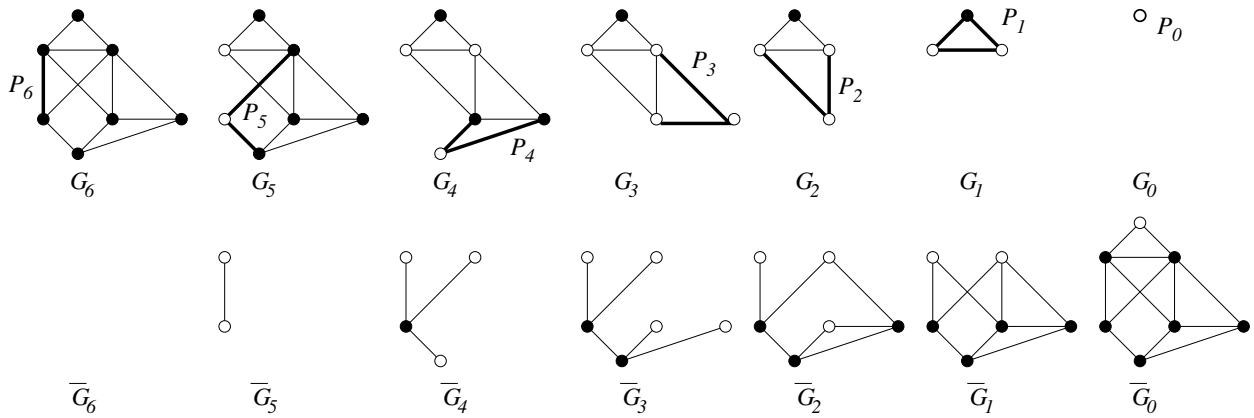

Figure 4: Graphs associated with an ear-decomposition: (top) Ear-decomposition of a block: In each step from G_6 to G_0, a path (ear) is removed until a central cycle is left. (bottom) The corresponding $\overline{G}_k$ of each step is shown. The attachment points of the ears are depicted by unfilled vertices. For more compact illustration, a non-bipartite graph is shown. (Adapted from (Abfalter *et al.*, 2003))

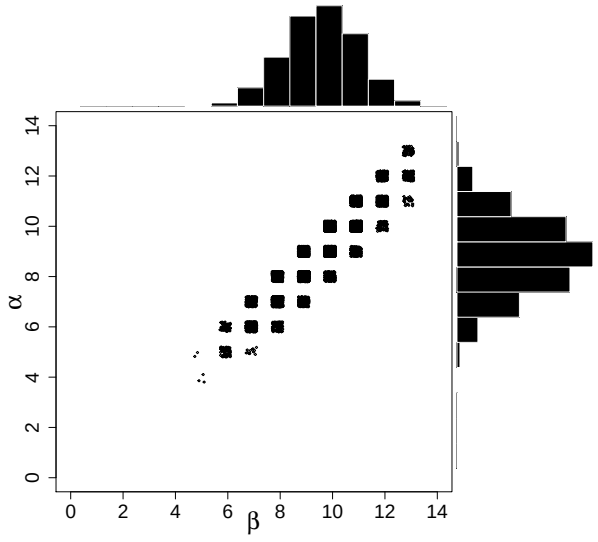

Figure 5: Distribution of the exponents α (memory requirements) and β (CPU runtime) for a fixed graph with $\mu = 13$. 1000 spanning trees were generated by replacing randomly selected tree edges with non-tree-edges. Each vertex was used as root for the Maon/Schieber algorithm to compute the ear decomposition. The (rare) best spanning trees yield $\alpha = 4$ and $\beta = 5$ in this example.

large components, however, this becomes inefficient, and we must resort to a sampling technique. To this end, we use the generic idea of stochastic backtracing in dynamic programming, which is used in a similar context, for instance, to generate samples of secondary structures (Tacker *et al.*, 1996; Ding and Lawrence, 2001; Waldispühl and Clote, 2007). Here, we set for each k from 1 to μ the colors a_k for the attachment points with probabilities proportional to $|\mathbf{c}(\overline{G}_k, a_k)|$. In the second step, we sample the colorings for the connecting paths, whose end points now have fixed colors, as described in Flamm *et al.* (2001). This simple procedure ensures uniform sampling from $\mathscr{C}$ and hence unbiased generation of feasible solutions. Biased samples could, of course, be generated with less effort, for example, by depth first search search with a random vertex order. In many cases, however, it is desirable to minimize *a priori* sequence biases.

2.5. Local Moves and Optimization

All heuristics for RNA design use "local moves" to navigate $\mathscr{C}$ in an attempt to further improve the sequence. The most obvious move, i.e., changing the color of a single vertex of G, however, will typically not be feasible as it destroys compatibility. Instead we need to always replace all vertices belonging to one connected component of the dependency graph. For designs with a single target this reduces to mutating a single unpaired base or replacing a base pair with another one. This type of structure-dependent moves is also used, for instance, to explore neutral networks of sequences folding into the same secondary structures (Schuster *et al.*, 1994). As the number of target structures grows, the dependency graph will have larger but fewer connected components. This means that the fraction of the sequence changed in a single "local move" becomes larger and larger.

In the context of sequence design with design goals specified in terms of the energy landscape, locality in terms of sequence is highly desirable. The RNA energy model has the interesting property that the difference in minimum free energy between two sequences that differ by a single nucleotide is bounded by a constant c (Fontana *et al.*, 1993). This is a consequence of the additivity of the energy model, which limits the effect of

a mutation to the maximum energy difference between two adjacent loops upon removing their separating base pair, in practice twice the maximum stacking energy. As an immediate consequence

$$|f(x,\Theta) - f(x',\Theta)| \leq c d_H(x,x') \tag{7}$$

where $d_H(\,.\,,\,.\,)$ denotes the Hamming distance. Small changes in the sequence therefore cause only moderate changes in the Boltzmann distribution of structures and are thus less prone to destroying achievements of past optimization steps.

The design goals are represented by an objective function $\Xi : \mathscr{C} \to \mathbb{R}$ that assigns a "fitness" to each sequence x, i.e., a feasible coloring of G. We use a simple, Simulated Annealing-like strategy to optimize Ξ. In each step, a candidate x' is generated by a local move in one of the components of G. We accept x' if

$$\Xi(x') \leq \Xi(x) + t, \quad t \sim \exp(\lambda) \tag{8}$$

The new candidate sequence x' is always accepted if it is better according to the optimization criterion Ξ than its parent x. To avoid locally optimal traps, a candidate sequence is also accepted if the energy difference is less than an exponentially distributed random variate (drawn new each time). The parameter λ controls the speed with which local optima are left again.

2.6. Design Goals

This fitness function Ξ can combine many features of the energy landscape of x that can be expressed in terms of the secondary structure model. Examples of such building blocks are properties of the Boltzmann ensemble of secondary structures of x such as its partition function $Z(x)$, the ensemble free energy $g(x) = -RT \ln Z(c)$, the minimum free energy $f(x) = \min_\Theta f(x,\Theta)$, the base-pairing probability matrix $\mathbf{P}(x)$, and the energy of a given structure $f(x,\Theta)$. All these properties are readily computed by RNA folding algorithms as implemented, for instance, in the `Vienna RNA Package` (Hofacker *et al.*, 1994; Lorenz *et al.*, 2011).

A basic design task, on which we focus here, is to construct RNA sequences for which the prescribed structures Θ_i have nearly the same folding energy and which together dominate the Boltzmann ensemble. The Θ_i will thus correspond to the ground state and the most important metastable states in the fitness landscape. The simplest fitness function for this task aims at simultaneously minimizing the energy of the Θ_i, for instance

$$\Xi(x) = \max_{i=1\ldots M} f(x,\Theta_i) \tag{9}$$

Since optimization of equ.(9) forces an increase in the fraction of the most under-represented target structure, it leads to comparable abundances of all prescribed structures. The advantage of this ansatz is that it can be evaluated very efficiently, requiring only the determination of the energy of M individual secondary structures and avoids the use of the computationally demanding RNA folding algorithm. The effort to evaluate $\Xi(x)$ is only $\mathbf{O}(Mn)$, compared to the cubic in n runtime of RNA folding. A disadvantage, however, is the lack of direct control over the ground state and hence over the ensemble in which the Θ_i are embedded.

Zadeh *et al.* (2011b) argued that design fitness functions should not only contain the positive design goals but also encapsulate negative design goals, i.e., they should explicitly penalize unwanted structures in the Boltzmann ensemble. A good example is the ensemble defect $d(x,\Theta)$, defined as the expected base pair distance of a random structure picked from the Boltzmann ensemble of the target structure Θ. It can be computed in quadratic time from $\mathbf{P}(x)$ (Zadeh *et al.*, 2011b). The sum of the ensemble defects is one of several conceivable generalizations of the multi-target design problem.

Flamm *et al.* (2001) used a different form of the objective for bistable structures, aiming directly at minimizing the difference between the energies of the individual structures and the ensemble free energy. For M structures, this approach yields

$$\begin{aligned}\Xi(x) = &\left(\sum_{k=1}^{K} f(x,\Theta_k) - g(x)\right) \\ &+ \gamma\left(\sum_{k<l}(f(x,\Theta_k) - f(x,\Theta_l))^2\right)\end{aligned} \tag{10}$$

The first part of equ. 10 minimizes the difference between the energies of the target structures and the Gibbs free energy of the ensemble, while the second part yields targets that have approximately the same energy. The weight γ allows us to favour one goal over the other. Fitness functions based on RNA folding are expensive to evaluate but promise better designs. An appealing approach is thus to first find a sequence using equ. 9, which is then used as the initial seed for further optimization using equ. 10 or another scheme.

Additional design goals can easily be included in Ξ. For instance, a prescribed sequence composition can be approached by suitable penalty terms for sequence bias. In particular, a log-multinomial function is available that allows penalizing mono-nucleotide distributions that deviate from a user-selected probability vector. More elaborate features of the fitness landscape,

such as minimum heights of energy barriers, could also be included as discussed in Flamm *et al.* (2001), albeit at high computational cost.

2.7. Summary of the Design Algorithm

The complete design algorithm consists of the following steps:

1. INPUT: a set of secondary structures $\{\Theta_i | 1 \leq i \leq M\}$ and the objective function Ξ.
2. Construct the dependency graph $G(\Theta_1, \ldots, \Theta_M)$.
3. If G is not bipartite, stop since the design problem is unsolvable.
4. Decompose the graph first into its connected components, then further into the biconnected components, and finally construct an open ear decomposition for each block.
5. Compute the numbers $|\mathbf{c}(H, a)|$ of colorings for the various subgraphs in the decomposition with fixed color assignments at their attachment and cut points.
6. Using these tables, generate sequences with uniform distribution on the set of compatible sequences.
7. Optimize these start sequences by local search with respect to the desired cost function Ξ for the design problem at hand.
8. OUTPUT: Optimized nucleic acid sequence compatible with all predefined structures.

3. Results

3.1. Implementation

We opted to implement the algorithm described above in the functional programming language Haskell (The GHC Team, 1989–2013; Hudak *et al.*, 2007). Haskell promotes a high-level style of programming and makes it easy to separate the logically distinct facets of an algorithm. In terms of implementation, the functional style of programming sometimes requires expressing an algorithm differently than known from the imperative world (Okasaki, 1999; Bird, 2010). Here, this concerns in particular the graph decomposition algorithm and the evaluation of candidate sequences.

The ear decomposition algorithm of Maon *et al.* (1986), which we use to handle complex components of the dependency graph, is implemented using the functional graph library (Erwig, 1997). The decomposition algorithm by Maon *et al.* (1986) adapts well to a functional description as it is not described in terms of an explicit graph coloring, but rather as a decomposition of the original graph into a spanning tree, tree edges, and non-tree edges. The resulting ears are then colored by legal assignments of base pairs. The laziness properties of algorithms implemented in Haskell make it possible to handle assignments with a large number of legal assignments without having to explicitly store them.

The evaluation of candidate sequences is a potential performance bottleneck, as it requires evaluation of the energy of sequence candidates given the structure constraints. We make use of fusion (Hinze *et al.*, 2011), a compiler optimization technique aimed at removing intermediate data structures in functional programs, which often yields executables with a runtime performance comparable to that of `C` implementations. In particular, we use stream fusion (Coutts *et al.*, 2007; Leshchinskiy, 2009) during sequence sampling. Energy evaluations are performed in a functional version of the Vienna RNA folding algorithms (Lorenz *et al.*, 2011), which are also fused (Höner zu Siederdissen, 2012).

In order to facilitate the exploration of different objective functions, the user can supply Ξ on the command line as a function of the primitive features outlined in Sect. 2.6. It is easy to extend both, the design algorithm and the command line parser to include additional terms if necessary. The current implementation of `RNAdesign` uses equ.(10) as the default objective function.

In many cases it is important to enforce sequence constraints. For example, the Shine-Dalgarno sequence, the start codon, and the sequence of the ligand-binding aptamer are typically fixed in design problems for riboswitches. We therefore provide an option to restrict the set of nucleotides that may be varied during the design process. The current implementation allows the user to specify, for each sequence position, the set of allowed nucleotides. It is important to note that sequence constraints further shrink $\mathscr{C}$ and may render a design problem infeasible even if the prescribed target structures are consistent. `RNAdesign` of course detects such cases.

In addition to the functional implementation, we are developing a memory-efficient implementation in C++ to extend the range of applications to large complex problems, i.e, very long sequences and $M \gg 3$ independent target structures.

3.2. Artificial SV11-like Bistable Riboswitches

SV11, a 115 nt long RNA, is a recombinant of the plus and minus strands of the phage-derived MNV-11 RNA. Both molecules are efficient substrates for

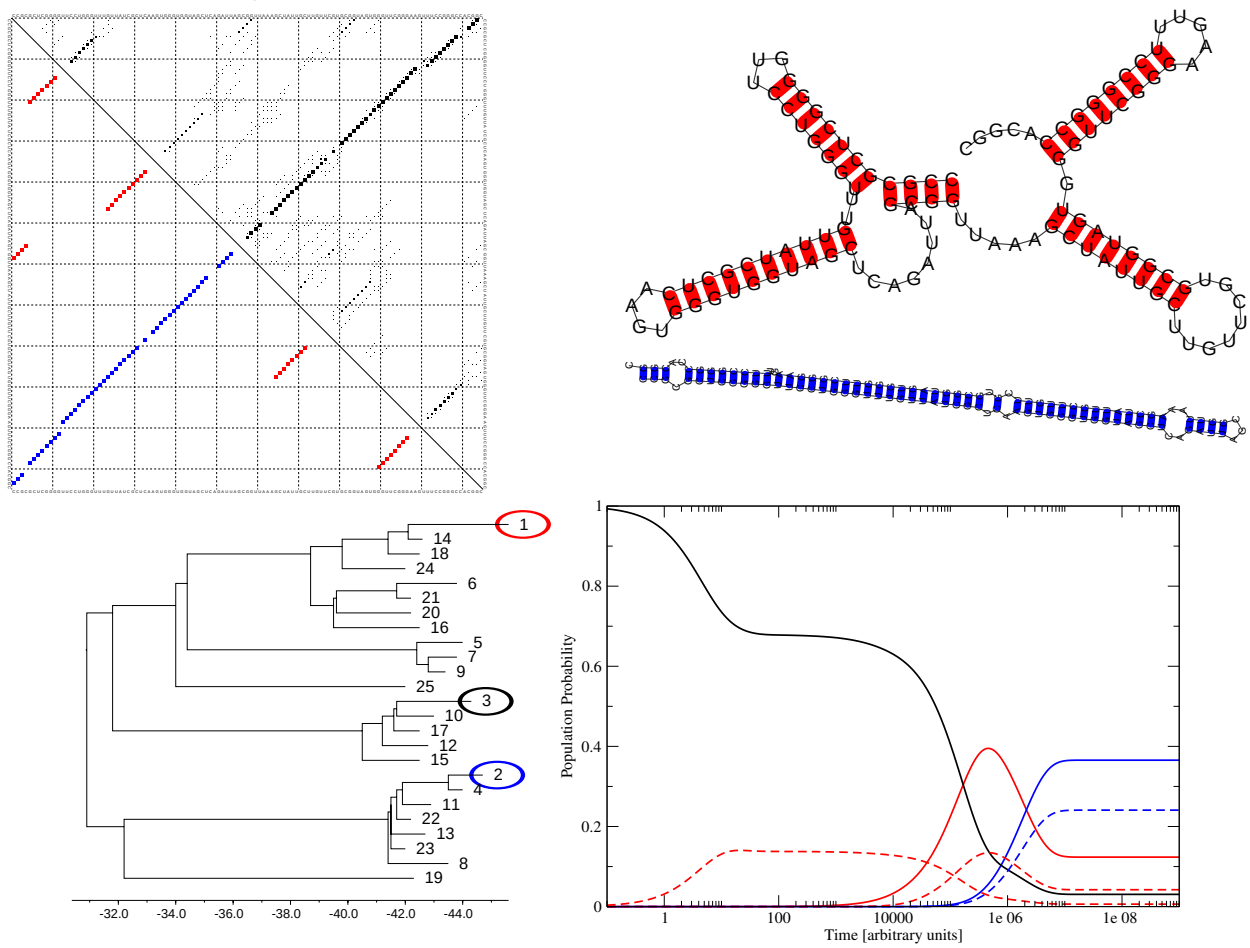

Figure 6: A designed sequence for the SV11 riboswitch. **(Dotplot)** The upper triangular matrix plots the probability for each individual nucleotide to be paired. The lower triangular plot shows the two structural constraints. **(Structures)** In red, the lower-energy structure that forms a Y-shaped multi-branched loop and two additional exterior loops, and in blue the single long helix structure. The red structure is almost equivalent to the minimum-free energy structure, which forms a single additional `AU` base pair, resulting in a three-nucleotide hairpin instead of a five-nucleotide hairpin with an additional gain of 0.6 kcal/mol. Accordingly, the second structure in the suboptimal ensemble is the red structure, followed by the blue structure at the third position in the ensemble with a difference of 0.9 kcal/mol. **(Barrier tree (left bottom) and folding kinetics (right bottom))** The red and blue curves correspond to the target structures and are dominant in the kinetics. The dashed lines are structures that are very similar (base pair distance of five or less) to the target structures. As the energy distance to the open chain is too large to be included in the barrier and kinetics calculations, we started from a structure (colored black) that is somewhat related to the red target.

Qβ replicase and arise consistently in artificial selection experiments (Biebricher *et al.*, 1992). SV11 is frequently used as an extreme example of an RNA whose properties are determined by folding intermediates rather than its thermodynamic ground state alone. Co-transcriptional folding results in a metastable conformation consisting of a Y-shaped multibranched structure and two additional exterior hairpin loops, which is replicated by Qβ replicase. The ground state, in contrast, is a single long helix structure with a hairpin which no longer serves as a template for the Qβ replicase. The metastable structure can spontaneously rearrange to the ground state. This transition is effectively irreversible because of an energy difference exceeding 30 kcal/mol, as computed by `RNAeval` (Lorenz *et al.*, 2011). For the same reason, the base pair probabilities in the equilibrium ensemble give no indication of important structural alternatives.

Because of its extreme properties, the SV11 structure pair has been used repeatedly as an example, including for design tasks (Lyngsø *et al.*, 2012) whose goal is to find a sequence that realizes the two conformations with nearly equal energy. In Fig. 6, we show that our software readily solves this computational problem.

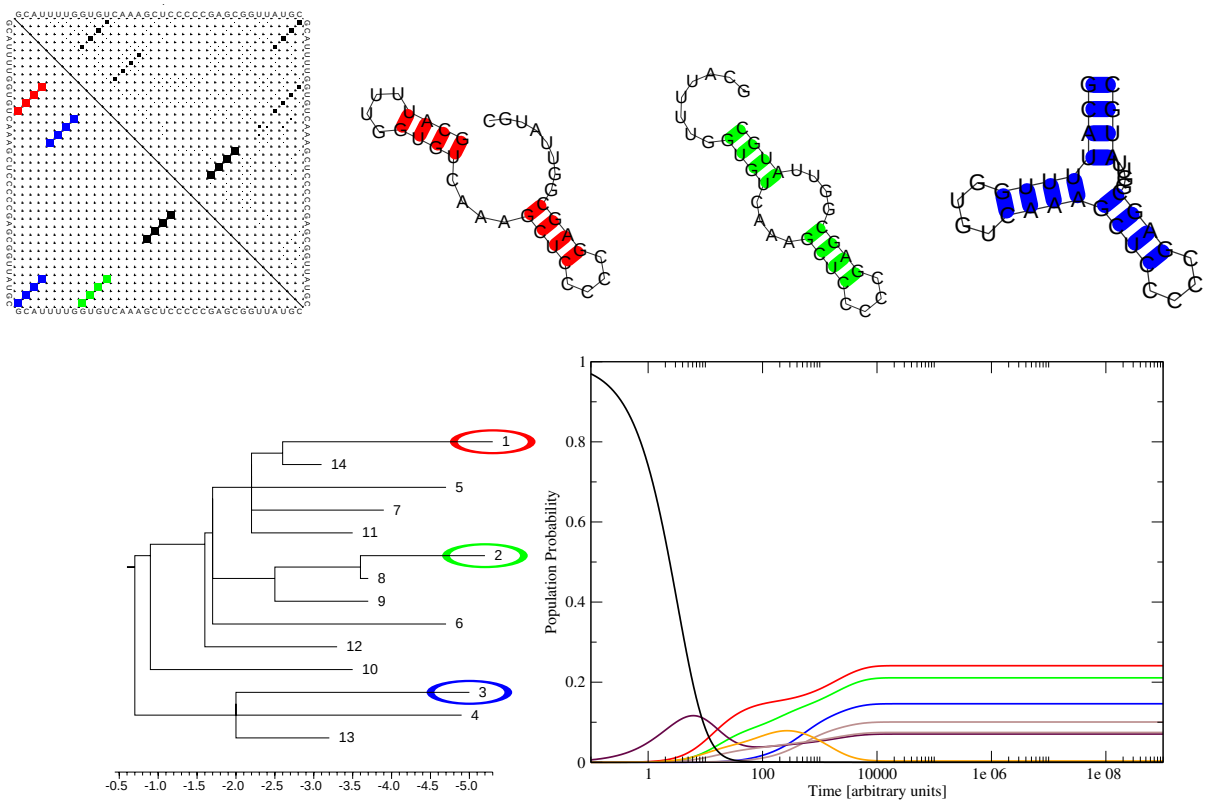

Figure 7: Example of a tri-stable switch molecule. **(Dotplot)** All structures have great statistical weight within the thermodynamic equilibrium. Base pairs belonging to one structure are colored red, green, or blue, respectively. All structures share four base pairs, which are colored black. **(Structures)** The three structures correspond to the three lowest energy structures of the RNA. The red (top-left) structure is the mfe structure, the green and blue structures have free energies 0.1 and 0.3 kcal/mol above the ground state, respectively. **(Barrier tree (left, bottom) and folding kinetics (right, bottom))** The desired structures of the tri-stable switch form the most prominent local minima in the folding landscape. Kinetic curves in brownish colors correspond to mixed conformations where compatible structural features of different local minima structures have been blended into a single structure.

The sequence proposed by our algorithm using the default optimization criterion of equ. 10 is almost optimal. Both structures differ by only 0.9 kcal/mol. The minimum-free energy structure differs from the complex (red) structure by a single `AU` base pair.

Using only the criterion in equ.(10), the algorithm required 120 seconds on an `Intel Core i5-3570K`. In total, 50 candidates were created, with a thinning of 200 (i.e., only every 200^{th} candidate is retained) with an initial burn-in period of 100 candidates. Of the 50 candidates that are returned, only the top-most was selected. Other, suboptimal, solutions are returned to provide alternatives that can be evaluated before running the algorithm again.

3.3. A Tri-stable Riboswitch

In Fig. 7 we present a small, artificial example of a tri-stable system with three prescribed target structures (red, green, and blue). The computational design problem is solved by our tool using the default fitness function equ.(10) within 10 000 optimization steps, amounting to 45 seconds on standard PC hardware. The designed sequence readily folds into exactly these three structures. The red structure is the minimum-free energy structure, the green and the blue ones are the first two sub-optimal local minima in the energy landscape. Alternative structures with non-negligible probability have a small base pair distance from one of the targets. As indicated by the partition function dotplot, the targets are very well represented in the structural ensemble. Base pairs that are not part of one of the three design goals are very rare. There are, however, some "mixed" structures that facilitate the transition between the three local optima. We use a barrier tree to visualize the landscape of the designed sequence. Simulated folding kinetics, starting from the open chain, shows that the

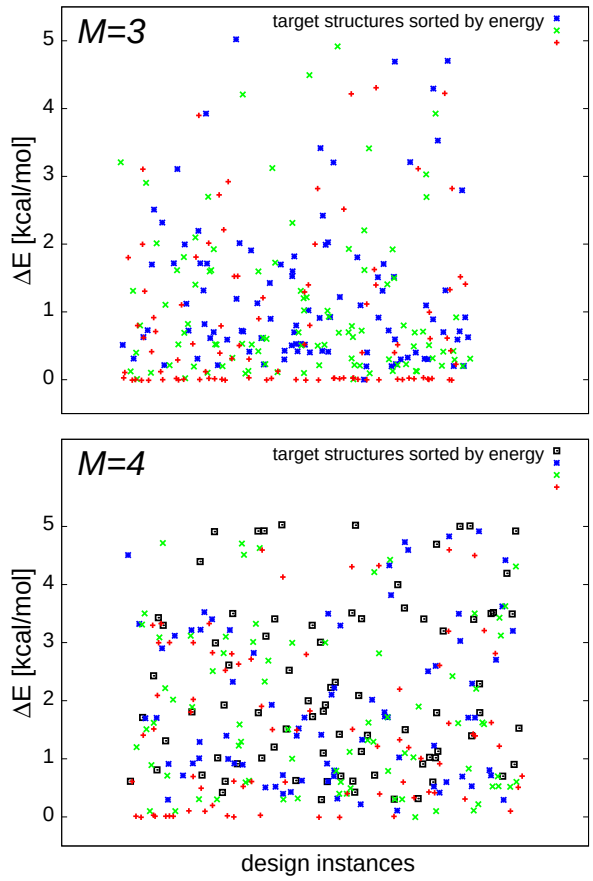

Figure 8: Performance of the sequence design algorithm on 100 randomly generated instances. **(Top)** Tri-stable targets. Of the 300 structures, the 288 that appear within the 5 kcal/mol window are shown. The remaining 12 structures have $1 \leq d_{bp} \leq 14$. The lowest-energy structure (red) is typically very close (mean 0.93, median 0.4 kcal/mol difference) to the mfe structure, the second (mean 1.17, median 0.7 kcal/mol; green) and the third (mean 1.41, median 0.9 kcal/mol; blue) structure are more distant. (A small amount of jitter was added to better separate data points). **(Below)** Four target structures. Only 24 of the 400 target structures lie outside the 5 kcal/mol window. The energy differences to the ground state for the energy-sorted targets are red (mean 1.85, median 1.6 kcal/mol), green (mean 2.17, median 1.9 kcal/mol), blue (mean 2.40, median 2.05 kcal/mol), and black (mean 2.62, median 2.3 kcal/mol).

three target structures are, again, the three most prominent structures.

3.4. A Large-scale Set of Multi-stable Targets

The example above shows the effectiveness of our algorithm in designing sequences for tri-stable targets. To demonstrate that our method scales well to larger design problems, we generated an ensemble of 100 design problems, each produced from a random sequence of length 100 as follows: We first used `RNAshapes` (Reeder and Giegerich, 2005) to extract the three most stable coarse-grained structures and their most stable fine-grained representatives ("shreps"). The SV11 sequence, for example, has (at least) two shapes: the low-energy rod-like structures with shape `[]` and the high-energy complex structure `[[][]][][]` with a Y-shaped multi-branched loop and two additional external stems. This way we ensure that the design problem is feasible. We solve the design problems using the fitness function equ.(10), perform a single optimization run for each problem, and retain a single top-scoring sequence, as output.

In order to evaluate the quality of the designed sequences we investigated their energy landscapes in more detail. Using `RNAsubopt` (Lorenz *et al.*, 2011) we produced all suboptimal structures within 5 kcal/mol of the ground state, and determined the suboptimal structure Θ_i' within this energy band that is closest to the design goal Θ_i. Ideally, the base-pairing distance $d_{bp}(\Theta_i, \Theta_i')$ should vanish and $f(\Theta_i)$ should be very close to the ground state for all three design targets. In 96% of the design problems, all three target structures were indeed contained within the 5 kcal/mol energy band, and even in the remaining few cases, a very similar structure was observed within this range. Fig. 8 shows the energy differences for the 100 designs as a scatter plot. In most of the designs, one of the three targets is the ground state. The mean and median energy differences between the ground state and the worst of the three target structures are only 1.0 and 1.5 kcal/mol, respectively. Overall, these data show that our approach produces close-to-optimal tri-stable designs reliably and efficiently.

To assess the increase in difficulty of designing sequences for more than three target structures simultaneously, we repeated the large-scale experiment, but this time using four target structures instead of three. The results shown in the lower panel of Fig. 8 lead us to believe that our approach does indeed scale to very complex multi-stable targets. As the quality of the generated sequences also degenerated slightly with median differences from 1.6 to 2.3 kcal/mol, further investigation into complex multi-structure targets will be required. Nevertheless, these differences only amount to roughly 1 to 3 stacked base pairs. Again, we automatically selected the top-scoring sequence for each target instead of trying the, say, best five sequences for each structure.

4. Discussion and Conclusions

We have shown that multi-stable RNA sequences with prescribed alternative secondary structures can be

constructed efficiently by means of a generic computational approach. With `RNAdesign` we provide an efficient implementation that combines an exact solution of a graph coloring problem with the heuristic optimization of feasible solutions by local search. When more than two target structures are prescribed, a combinatorial consistency condition must be satisfied. For triples of targets and moderate sequence length, the design problem frequently has feasible solutions, although the probability decreases exponentially with chain length. Randomly generated sets of four or more target structures, however, typically cannot be realized by the same sequence. Since very few multi-stable RNAs have been described in the literature, we resorted to artificial test cases to verify that our approach solves the computational problem at hand.

`RNAdesign` can accommodate a wide range of design goals. Although our test cases focus on nearly equal enrichment of target structures in the Boltzmann ensemble, more complex features of the fitness landscapes can easily be incorporated. As discussed by Flamm *et al.* (2001) for the case of bi-stable structures, it is feasible (albeit computationally demanding) to estimate for a candidate sequence x the energy barrier $f^{\neq}(x, \Theta_i, \Theta_j)$ between target structures Θ_i and Θ_j. For moderate sequence lengths, this can be computed exactly by using `RNAsubopt` and `barriers`, and for longer sequences a path-based heuristic provides at least an upper bound (Morgan and Higgs, 1998; Flamm *et al.*, 2001). On this basis, it is even possible to estimate kinetic parameters such as first passage times to target structures (Wolfinger *et al.*, 2004). It will be easy to extend the `RNAdesign` so that kinetic parameters of this type can be included into the design fitness function Ξ.

The current version of `RNAdesign` already supports inclusion of prescribed energy differences between the target conformations. This is desirable, for instance, for the rational design of riboswitches that are triggered by ligand binding. In this case, the fitness landscape is distorted by the binding energy of the ligand in certain structures. This causes a re-folding of the molecule in which conformational changes in the ligand binding domain are used to change adjacent structural domains. Our recent construction of a transcriptional riboswitch based on the theophylline aptamer domain (Wachsmuth *et al.*, 2013) shows that the RNA energy model is sufficiently accurate to capture such effects.

We can, therefore, argue that the relative ease with which multistable structures can be designed reflects the evolutionary accessibility of such molecules. Our data suggest, in particular, that RNA sequences with three or four disparate local optima with energies close to the ground state are abundant and can readily be optimized by a local search in sequence space. A similar observation has been made by Ramlan and Zauner (2011). If such structures provide a selective advantage, evolution should therefore be able to evolve them *de novo* in different contexts. This immediately raises the question of whether multi-stable RNAs have arisen in the history of life and how abundant they are in nature.

For the case of two alternative structures the answer is, of course, affirmative, as demonstrated by a diverse set of riboswitches for a wide variety of ligands (Serganov and Nudler, 2013) and several classes of RNA thermometers (Kortmann and Narberhaus, 2012). Self-induced conformational switches (Nagel and Pleij, 2002) act as a kind of timing device. Here, the molecule is trapped in a metastable structure that either allows or blocks the RNA's function. Decay to the ground state then flips the switch. Molecules undergoing such conformational changes have also been observed as the outcomes of artificial selection experiments, for instance, selecting for suitability as a template for Qβ replicase (Biebricher and Luce, 1982; Biebricher *et al.*, 1992).

For more than two structural alternatives, the situation is less obvious. No self-induced or small metabolite-triggered RNA switch with three or more structural alternatives has so far been characterized. Complex conformational changes, however, play a role in splicing and the action of ribozymes, including self-splicing introns and other allosteric nucleic acid catalysts (Jose *et al.*, 2001; Soukup and Breaker, 2000). A well-understood system that comes at least close to a self-induced tri-stable RNA switch is the *Hok/Sok* system of plasmid R1 in *E. coli* (Gultyaev *et al.*, 1997; Møller-Jensen *et al.*, 2001). Allosteric nucleic acid catalysts (Jose *et al.*, 2001; Soukup and Breaker, 2000). Here, the binding of one effector causes a change in the structure of the ribozyme molecule, which in turn allows the binding of a second effector necessary for the final activation of the enzymatic function. An artificial catalytic system consisting of two RNAs that catalyze their ligation with the help of a transient hammerhead ribozyme structure relies on several coordinated structural rearrangements (Gwiazda *et al.*, 2012).

The possibility that multi-stable conformational switches are a common element beyond simple ON/OFF switches in RNA-based regulation leads to the question of whether RNA-based circuits provide a compact – and hard to disentangle – implementation of complex regulatory programs. Beyond such intriguing perspectives on RNA biology, we encountered also several non-trivial computational problems that provide interesting avenues for future research on rational RNA de-

sign.

Instead of a "fair" starting sequence sampled uniformly from $\mathscr{C}$, one might want to "stack the deck" as much as possible in favour of a successful design. This invites the question of whether there are any efficient dynamic programming algorithms that compute the sequence that minimizes the sum of free energies on a prescribed set of structures. A promising way to address this question is a generalization of the `intaRNA` approach of Busch and Backofen (2006) to multiple structures. Another obvious challenge is to improve the coloring step using an explicit construction for ear decompositions that guarantee small values of α and β.

Since the design goals for more than two sequences are not feasible in general, one may be interested in a slight relaxation of the structure in $\{\Theta_i\}$, i.e., in a set $\{\Theta'_i\}$ that is as close as possible to the original and for which the design is feasible. A natural objective function for this task is, for instance, $\sum_i d_G(\Theta'_i, \Theta_i)$ for some graph edit distance $d_G(\,.\,,\,.\,)$. A simpler, but maybe less natural, approach is to directly edit the dependency graph G, i.e., by removing a minimal number of edges.

An alternative approach to relaxing the structural constraints is to allow a small number of non-canonical base pairs. The `CONTRAfold` algorithm by Do *et al.* (2006) considers all 16 possible base pairings instead of just the canonical six. Another solution is to use the space of extended secondary structures (Höner zu Siederdissen *et al.*, 2011), which also considers all 16 possible base pairs, and, in addition, explicitly annotates nucleotide pairings with the nucleotide edge engaged in pairing. As both of these models have basically no constraints, the space of candidate sequences is unrestricted. However, since canonical base pairs are more likely than non-canonical base pairs, it makes sense to always constrain the search space to those sequences for which canonical pairings predominate. Formally, this equates to allowing some –but not too many– color conflicts.

Finally, it is desirable to impose more sophisticated conditions on sequence composition. We currently allow penalizing candidate sequences according to a mono-nucleotide model. It seems feasible to explore di-nucleotide distributions instead of the current mono-nucleotide model. Such models have already been used in a gene prediction context (Gesell and Washietl, 2008), and their impact on sequence design will be interesting to explore.

The `RNAdesign` tool opens the door to the largely unexplored realm of tri-stable and even higher-level multistable structures, which is of utmost interest for synthetic biology. With small modifications of the energy model our approch can easily be extended to interacting multistable RNA molecules, a topic that is of particular interest for the design of small trans-acting and multistable self-assembling RNAs.

Acknowledgements

This work has been funded, in part, by the Austrian *Fonds zur Förderung der wissenschaftlichen Forschung (FWF) SFB F43* "RNA regulation of the transcriptome" and the *Deutsche Forschungsgemeinschaft (STA 850/15-10)*.

Availability

The `RNAdesign` software can downloaded from `http://www.bioinf.uni-leipzig.de/Software/RNAdesign/`

References

Abfalter I, Flamm C, Stadler PF, 2003. Design of multi-stable nucleic acid sequences. In: Mewes HW, Heun V, Frishman D, Kramer S, editors, Proceedings of the German Conference on Bioinformatics. GCB 2003, vol. 1, (pp. 1–7). München, D: belleville Verlag Michael Farin.

Aguirre-Hernández R, Hoos HH, Condon A, 2007. Computational RNA secondary structure design: empirical complexity and improved methods. BMC Bioinformatics 8:34.

Andronescu M, Fejes AP, Hutter F, Hoos HH, Condon A, 2004. A new algorithm for RNA secondary structure design. J Mol Biol 336:607–624.

Avihoo A, Churkin A, Barash D, 2011. `RNAexinv`: An extended inverse RNA folding from shape and physical attributes to sequences. BMC Bioinformatics 12:319.

Baumstark T, Schroder AR, Riesner D, 1997. Viroid processing: Switch from cleavage to ligation is driven by a change from a tetraloop to a loop E conformation. EMBO J 16:599–610.

Biebricher CK, Diekmann S, Luce R, 1992. In vitro recombination and terminal elongation of RNA by Qβ replicase. EMBO J 11:51129–5135.

Biebricher CK, Luce R, 1982. Structural analysis of self-replicating RNA synthesized by Qβ replicase. J Mol Biol 154:629–648.

Bird R, 2010. Pearls of Functional Algorithm Design. Cambridge University Press.

Burghardt B, Hartmann AK, 2007. RNA secondary structure design. Phys Rev E Stat Nonlin Soft Matter Phys 75:021920.

Busch A, Backofen R, 2006. INFO-RNA–a fast approach to inverse RNA folding. Bioinformatics 22:1823–1831.

Cao Y, Woodson S, 2000. Refolding of rRNA exons enhances dissociation of the tetrahymena intron. RNA 6(9):1248–1256.

Coutts D, Leshchinskiy R, Stewart D, 2007. Stream Fusion: From Lists to Streams to Nothing at All. In: Proceedings of the 12th ACM SIGPLAN international conference on Functional programming, ICFP'07, (pp. 315–326). ACM.

Ding Y, Chan CY, Lawrence CE, 2005. RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. RNA 11:1157–1166.

Ding Y, Lawrence CE, 2001. Statistical prediction of single-stranded regions in RNA secondary structure and application to predicting effective antisense target sites and beyond. Nucleic Acids Res 29:1034–1046.

Do CB, Woods DA, Batzoglou S, 2006. CONTRAfold: RNA secondary structure prediction without physics-based models. Bioinformatics 22:e90.

Erwig M, 1997. Functional programming with graphs. ACM SIGPLAN Notices 32:52–65.

Flamm C, Fontana W, Hofacker I, Schuster P, 2000. RNA folding kinetics at elementary step resolution. RNA 6:325–338.

Flamm C, Hofacker IL, Maurer-Stroh S, Stadler PF, Zehl M, 2001. Design of multi-stable RNA molecules. RNA 7:254–265.

Fontana W, Stadler PF, Bornberg-Bauer EG, Griesmacher T, Hofacker IL, Tacker M, Tarazona P, Weinberger ED, Schuster P, 1993. RNA folding and combinatory landscapes. Phys Rev E 47:2083–2099.

Gao JZ, Li LY, Reidys CM, 2010. Inverse folding of RNA pseudoknot structures. Alg Mol Biol 5:27.

Gesell T, Washietl S, 2008. Dinucleotide controlled null models for comparative rna gene prediction. BMC bioinformatics 9:248.

Gultyaev A, Franch T, Gerdes K, 1997. Programmed cell death by hok/sok of plasmid R1: coupled nucleotide covariations reveal a phylogenetically conserved folding pathway in the hok family of mRNAs. J Mol Biol 273(1):26–37.

Gwiazda S, Salomon K, Appel B, Müller S, 2012. RNA self-ligation: From oligonucleotides to full length ribozymes. Biochimie 94:1457–1463.

Hinze R, Harper T, James DW, 2011. Theory and Practice of Fusion. Implementation and Application of Functional Languages (pp. 19–37).

Hofacker IL, Fontana W, Stadler PF, Bonhoeffer LS, Tacker M, Schuster P, 1994. Fast folding and comparison of RNA secondary structures. Monatsh Chem 125:167–188.

Höner zu Siederdissen C, 2012. Sneaking Around concatMap: Efficient Combinators for Dynamic Programming. In: Proceedings of the 17th ACM SIGPLAN international conference on Functional programming, ICFP '12, (pp. 215–226). New York, NY, USA: ACM. URL `http://doi.acm.org/10.1145/2364527.2364559`.

Höner zu Siederdissen C, Bernhart SH, Stadler PF, Hofacker IL, 2011. A folding algorithm for extended RNA secondary structures. Bioinformatics 27:129–136.

Hudak P, Hughes J, Peyton Jones S, Wadler P, 2007. A History of Haskell: Being Lazy with Class. In: Proceedings of the third ACM SIGPLAN conference on History of programming languages, HOPL III, (pp. 1–55). ACM.

Isaacs FJ, Dwyer D, Collins JJ, 2006. RNA synthetic biology. Nature Biotech 24:545–554.

Jensen TR, Toft B, 1994. Graph Coloring Problems. New York: John Wiley & Sons.

Jose A, Soukup G, Breaker R, 2001. Cooperative binding of effectors by an allosteric ribozyme. Nucleic Acids Res 29(70:1631–1637.

Kortmann J, Narberhaus F, 2012. Bacterial RNA thermometers: molecular zippers and switches. Nat Rev Microbiol 10:255–265.

Leshchinskiy R, 2009. Recycle Your Arrays! Practical Aspects of Declarative Languages (pp. 209–223).

Levin A, Lis M, Ponty Y, O'Donnell CW, Devadas S, Berger B, Waldispühl J, 2012. A global sampling approach to designing and reengineering RNA secondary structures. Nucleic Acids Res 40:10041–10052.

Li Y, Zhang S, 2011. Finding stable local optimal RNA secondary structures. Bioinformatics 27:2994–3001.

Lorenz R, Bernhart SH, Höner zu Siederdissen C, Tafer H, Flamm C, Stadler PF, Hofacker IL, 2011. ViennaRNA Package 2.0. Algorithms for Molecular Biology 6.

Lorenz R, Flamm C, Hofacker IL, 2009. 2D projections of RNA folding landscapes. In: German Conference on Bioinformatics, vol. 157 of *Lecture Notes in Informatics*, (pp. 11–20). GI.

Lyngsø RB, Anderson JW, Sizikova E, Badugu A, Hyland T, Hein J, 2012. `Frnakenstein`: multiple target inverse RNA folding. BMC Bioinformatics 13:260.

Maon Y, Schieber B, Vishkin U, 1986. Parallel ear decomposition search (EDS) and ST-numbering in graphs. Theor Comp Sci 47:277–298.

Mathews DH, Disney MD, Childs JL, Schroeder SJ, Zuker M, Turner DH, 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction ofRNA secondary structure. Proc Natl Acad Sci USA 101:7287–7292.

Møller-Jensen J, Franch T, Gerdes K, 2001. Temporal translational control by a metastable RNA structure. J Biol Chem 276:35707–35713.

Morgan SR, Higgs PG, 1998. Barrier heights between groundstates in a model of RNA secondary structure. J Phys A 31:3153–3170.

Nagel JH, Pleij CW, 2002. Self-induced structural switches in RNA. Biochimie 84(9):913–23.

Ogawa A, 2011. Rational design of artificial riboswitches based on ligand-dependent modulation of internal ribosome entry in wheat germ extract and their applications as label-free biosensors. RNA 17:478–488.

Okasaki C, 1999. Purely functional data structures. Cambridge University Press.

Quarta G, Kim N, Izzo JA, Schlick T, 2009. Analysis of riboswitch structure and function by an energy landscape framework. J Mol Biol 393:993–1003.

Ramlan EI, Zauner KP, 2011. Design of interacting multi-stable nucleic acids for molecular information processing. Biosystems 105:14–24.

Ray PS, Jia J, Yao P, Majumder M, Hatzoglou M, Fox PL, 2009. A stress-responsive RNA switch regulates VEGFA expression. Nature 457:915–919.

Reeder J, Giegerich R, 2005. Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction. Bioinformatics 21:3516–3523.

Reidys C, Stadler PF, Schuster P, 1997. Generic properties of combinatory maps: Neutral networks of RNA secondary structures. Bull Math Biol 59:339–397.

Schnall-Levin M, Chindelevitch L, Berger B, 2008. Inverting the Viterbi algorithm: an abstract framework for structure design. In: Proceedings of the 25th international conference on Machine learning, ICML '08, (pp. 904–911). New York, NY, USA: ACM.

Schuster P, Fontana W, Stadler PF, Hofacker IL, 1994. From sequences to shapes and back: A case study in RNA secondary structures. Proc Roy Soc Lond B 255:279–284.

Serganov A, Nudler E, 2013. A decade of riboswitches. Cell 152:17–24.

Shu W, Liu M, Chen H, Bo X, Wang S, 2010. `ARDesigner`: a web-based system for allosteric RNA design. J Biotechnol 150:466–473.

Soukup G, Breaker R, 2000. Allosteric nucleic acid catalysts. Curr Opin Struct Biol 10(3):318–325.

Tacker M, Stadler PF, Bornberg-Bauer EG, Hofacker IL, Schuster P, 1996. Algorithm independent properties of RNA structure prediction. Eur Biophy J 25:115–130.

Taneda A, 2011. MODENA: a multi-objective RNA inverse folding. Adv Appl Bioinform Chem 4:1–12.

The GHC Team, 1989–2013. The Glasgow Haskell Compiler (GHC). `http://www.haskell.org/ghc/`.

Topp S, Gallivan JP, 2010. Emerging applications of riboswitches in chemical biology. ACS Chem Biol 5:139–148.

Wachsmuth M, Findeiß S, Weissheimer N, Stadler PF, Mörl M, 2013. *De novo* design of a synthetic riboswitch that regulates transcription termination. Nucleic Acids Res 41:2541–2551.

Waldispühl J, Clote P, 2007. Computing the partition function and sampling for saturated secondary structures of RNA, with respect

to the Turner energy model. J Comput Biol 14:190–215.
Whitney H, 1932. Non-separable and planar graphs. Trans Amer Math Soc 34:339–362.
Wieland M, Benz A, Klauser B, Hartig JS, 2009. Artificial ribozyme switches containing natural riboswitch aptamer domains. Angew Chem Int Ed 48:2715–2718.
Win MN, Liang JC, Smolke CD, 2009. Frameworks for programming biological function through RNA parts and devices. Chem Biol 16:298–310.
Wolfinger MT, Svrcek-Seiler WA, Flamm C, Hofacker IL, Stadler PF, 2004. Efficient computation of RNA folding dynamics. Journal of Physics A Mathematical and General 37:4731.
Wuchty S, Fontana W, Hofacker IL, Schuster P, 1999. Complete suboptimal folding of RNA and the stability of secondary structures. Biopolymers 49:145–165.
Zadeh JN, Steenberg CD, Bois JS, Wolfe BR, Pierce MB, Khan AR, Dirks RM, Pierce NA, 2011a. NUPACK: Analysis and design of nucleic acid systems. J Comput Chem 32:170–173.
Zadeh JN, Wolfe BR, Pierce NA, 2011b. Nucleic acid sequence design via efficient ensemble defect optimization. J Comput Chem 32:439–452.