A computational proposal for designing structured RNA pools for in vitro selection of RNAs

Namhee Kim, Hin Hark Gan and Tamar Schlick

RNA 2007 13: 478-492; originally published online Feb 23, 2007;
Access the most recent version at doi:10.1261/rna.374907

A computational proposal for designing structured RNA pools for in vitro selection of RNAs

NAMHEE KIM,[1] HIN HARK GAN,[1] and TAMAR SCHLICK[1,2]

[1]Department of Chemistry, New York University, New York, New York 10003, USA
[2]Courant Institute of Mathematical Sciences, New York University, New York, New York 10012, USA

ABSTRACT

Although in vitro selection technology is a versatile experimental tool for discovering novel synthetic RNA molecules, finding complex RNA molecules is difficult because most RNAs identified from random sequence pools are simple motifs, consistent with recent computational analysis of such sequence pools. Thus, enriching in vitro selection pools with complex structures could increase the probability of discovering novel RNAs. Here we develop an approach for engineering sequence pools that links RNA sequence space regions with corresponding structural distributions via a ''mixing matrix'' approach combined with a graph theory analysis. We define five classes of mixing matrices motivated by covariance mutations in RNA; these constructs define nucleotide transition rates and are applied to chosen starting sequences to yield specific nonrandom pools. We examine the coverage of sequence space as a function of the mixing matrix and starting sequence via clustering analysis. We show that, in contrast to random sequences, which are associated only with a local region of sequence space, our designed pools, including a structured pool for GTP aptamers, can target specific motifs. It follows that experimental synthesis of designed pools can benefit from using optimized starting sequences, mixing matrices, and pool fractions associated with each of our constructed pools as a guide. Automation of our approach could provide practical tools for pool design applications for in vitro selection of RNAs and related problems.

Keywords: in vitro selection; RNA pool design; mixing matrix; sequence-structure map; graph theory

INTRODUCTION

In vitro selection is an experimental approach that allows the screening of large random-sequence libraries of nucleic acid molecules (10^{15}) for a specific function, such as binding or catalysis (Ellington and Szostak 1990; Tuerk and Gold 1990; Wilson and Szostak 1999; Jäschke 2001; Storz 2002). In recent years, numerous target-binding nucleic acid molecules (aptamers) have been identified; targets include organic molecules, antibiotics, proteins, and whole viruses (Wilson and Szostak 1999; Hermann and Patel 2000). In addition, in vitro selection experiments have led to novel RNA enzymes (ribozymes) and have ramifications for biomolecular engineering, for example, the design of allosteric ribozymes and biosensors (Soukup and Breaker 1999a,b, 2000) and aptamers for functional genomics (Famulok and Verma 2002). Other emerging applications of engineered RNAs include RNA synthetic biology, where designed RNAs are used to control cellular functions (e.g., regulate gene expression) (Isaacs et al. 2006). These exciting advances offer new investigative and application tools for molecular biology, proteomics, molecular medicine, and diagnostics (Breaker 2004). As applications of in vitro selection technology expand, the demands for efficient selection of complex RNA motifs increase in importance.

Many RNAs identified from random pools have simple structural motifs (e.g., stem–loop, stem–bulge–stem–loop) (Lee et al. 2004). Indeed, our graph-based analysis of random pools (25–100 nucleotides [nt]) showed that different RNA secondary topologies are far from uniformly distributed, with low yields for multiply branched structures, although complex structures gradually become more frequent as RNA length increases (Gevertz et al. 2005). Interestingly, recent experimental findings suggest that enhancing the structural diversity of RNA pools increases the possibility of obtaining novel RNAs with high activity (Carothers et al. 2004, 2006). Specifically, GTP aptamers with high-binding affinities are found to be more complex structurally than low-binding-affinity aptamers. The principal reason for the lack of structural diversity in

Reprint requests to: Tamar Schlick, Department of Chemistry, New York University, New York, NY 10003, USA; e-mail: schlick@nyu.edu; fax: (212) 995-4152.

Article published online ahead of print. Article and publication date are at http://www.rnajournal.org/cgi/doi/10.1261/rna.374907.

random pools is due to incomplete and inefficient sampling of the astronomical size of the sequence space; random sequence sampling is inefficient because structures are not uniformly distributed in sequence space (Gevertz et al. 2005). To overcome this problem, heuristic approaches have been used to enhance the structural diversity of RNA pools. For example, structured pools have been synthesized by maintaining a constant stem–loop (GTP aptamer selection) (Davis and Szostak 2002) and by introducing random segments in existing RNA structures (e.g., purine nucleotide synthase and domains of group I ribozymes) (Jaeger et al. 1999; Ohuchi et al. 2002, 2004; Lau et al. 2004; Yoshioka et al. 2004). Recent works have also investigated the effects of sequence length (Legiewicz et al. 2006) and nucleotide composition (Knight et al. 2005) on recovery of specific RNAs. In addition, different functional classes of single-stranded RNAs have been found to have similar nucleotide compositions, implying evolutionary convergence (Schultes et al. 1997). The success of heuristic approaches depends on the details of the introduced sequence biases and the RNA function targeted. It is thus a challenge to develop systematic pool design approaches based on deeper understanding of pool sequence and structural complexity for the discovery of novel and complex RNAs.

To enhance in vitro selection experiments, RNA pools must possess sufficient sequence and structural complexity to ensure that the target RNA property exists in the pool. Given that we know little about the distribution of active RNAs in sequence and structural space (Carothers et al. 2004), an important goal of pool design is to maximize sequence and structural diversity without synthesizing all possible sequences. Even if complete coverage of sequence space is possible, not all regions of the space are likely to be productive for finding novel RNAs. This was suggested by recent analysis showing that the properties of GTP aptamers are correlated with their sequence/structural information content (Carothers et al. 2004). Unlike sequence space, the complexity of RNA structure space is more difficult to characterize quantitatively. At the secondary structural level, structural distributions of RNA pools can be analyzed using graph theory (Gan et al. 2003; Kim et al. 2004). Such an analysis shows that random pools are not structurally diverse (Gevertz et al. 2005), suggesting that pool structural diversity depends on how the sequence space is sampled. Indeed, understanding the relationship between sequence and secondary/tertiary structure spaces is essential for the design of effective pools for in vitro selection of RNAs. Thus, developing methodologies for generating and analyzing sequence pools possessing diverse RNA sequences and structures could enhance in vitro selection technology. Ultimately, a deeper understanding of the distribution of active RNAs in sequence and structural space will emerge through productive interactions between theoretical analysis and experiment.

Here we develop a computational approach for improving pool sequence and structural diversity by sampling sequences representing diverse regions of sequence space. We show that effective sampling of sequence space regions can be performed using nucleotide base ‘‘mixing matrices’’ for nucleotide transition rates applied to chosen starting sequences. Mixing matrices applied to given sequences are essentially generators of sequence pools and can be used to guide the reactants during in vitro selection experiments. Since we show that different regions of the sequence space are associated with distinct structural distributions, designed pools with specified target secondary structures can be obtained by optimizing a set of mixing matrices and starting sequences to approximate the target structural distributions. Figure 1 illustrates the relations among pool sequence/structure analysis, mixing matrix and starting sequence, and pool synthesis.

Specifically, we define five classes of mixing matrices motivated by biological objectives, such as on covariance and random mutations, to cover diverse regions of RNA sequence space. We show that such mixing matrices can produce structural distributions that are distinct from those of random sequence pools. We further describe optimal combinations of mixing matrices for specific target structured pools, including a designed pool for GTP aptamers. This pool design approach can thus provide a systematic method for constructing structured pools that can directly guide experimental pool synthesis and in vitro selection of complex RNAs. Automation of our pool design method is presently under way.

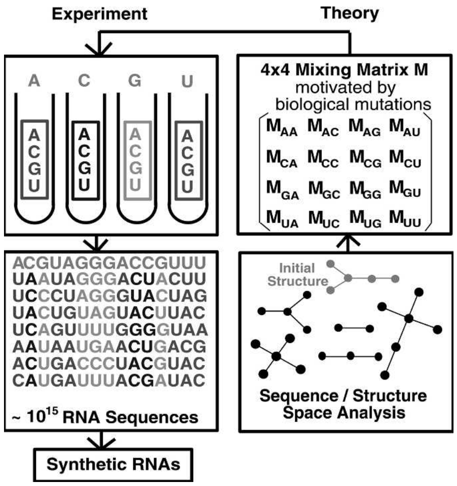


FIGURE 1. Modeling the RNA pool generation process using mixing matrices and analysis of pool structural distributions using tree graphs. The mixing matrix applied to any starting sequence specifies the mutation rates for all nucleotide bases. The matrix elements of each row represent nucleotide base (A, C, G, U) composition in a vial or synthesis port. Mixing matrices and starting sequences can be optimized to yield target structured pools.

MATERIALS AND METHODS

Defining mixing matrices for generation of nonrandom sequence pools

Understanding pool synthesis strategies provides clues for improving in vitro selection technology. The standard experimental protocol involves synthesizing DNA sequences and then transcribing them into RNA sequences by RNA polymerases. Current sequence synthesis strategies include chemical synthesis of short sequences (<150 nt); enzymatic assembly of short strands; synthesis of designed sequences with constant (double-stranded) and variable (single-stranded) regions; and synthesis of sequences around a designed sequence by random and biased mutations (Wilson and Szostak 1999). Random pools with short sequences are normally synthesized chemically, whereas longer sequence pools can be assembled enzymatically from shorter, chemically synthesized strands using techniques such as ligation or template-directed polymerization (Jäschke 2001; Stuhlmann and Jäschke 2002).

Our strategy is to define substrate pools with both random and biased sequence mutations around specific (starting) sequences to generate designed sequences with constant and variable regions. For this purpose, we introduce the mixing matrix **M**, whose elements specify mixing (or "contamination" or "doped") in the four phosphoramidite (A, C, G, and U [i.e., T in DNA synthesizer]) vials; applying mixing matrices to starting sequences leads to designed sequence pools. (Note that since successive nucleotide bases are selected independently from the vials, sequence synthesis methods correspond to the zeroth-order Markov process.) This representation of pool generation or synthesis using mixing matrices enables computational analysis of pool sequence and structural diversity. Our goal is to define, via computational analysis, an optimal set of starting sequences, mixing matrices, and associated weights for a given target structural distribution in the pool. Figure 1 illustrates pool design and synthesis via mixing matrix and analysis of sequence/structure space.

For pool synthesis using four vials or ports, the corresponding mixing matrix **M** is a 4×4 matrix that specifies the molar fractions of nucleotide components A, C, G, and U (T) in the four vials. Thus, the "ij" element of **M** (i.e., $\mathbf{M}_{ij}$) denotes the molar fraction of base j in the vial "for base i." It describes how we can dope that vial for i by introducing other bases $j \neq i$ into it as well. The design problem involves selecting those doping ratios and starting sequences. For example, $\mathbf{M}_{\mathrm{AU}}$ is the fraction of U (T) nucleotides in the vial for A, $\mathbf{M}_{\mathrm{AA}}$ is the fraction of A in the vial A, and $\mathbf{M}_{\mathrm{UA}}$ is the fraction of A in the vial U (T). Thus, the elements of each row of the matrix sum to unity:

$$\sum_{\mathrm{j=A,C,G,U}} \mathbf{M}_{ij} = 1.$$

If the DNA synthesizer is to produce a fixed sequence, then a vial for base i has 100% base i and zero fraction of other bases (i.e., $\mathbf{M}_{ii}$=1 and $\mathbf{M}_{ij}$=0 for $i \neq j$). If $\mathbf{M}_{ii}$<1 and $\mathbf{M}_{ij} \neq 0$, contaminations are introduced, as specified by the off-diagonal elements of **M**. The expected number of mutations in a synthesized sequence is determined by

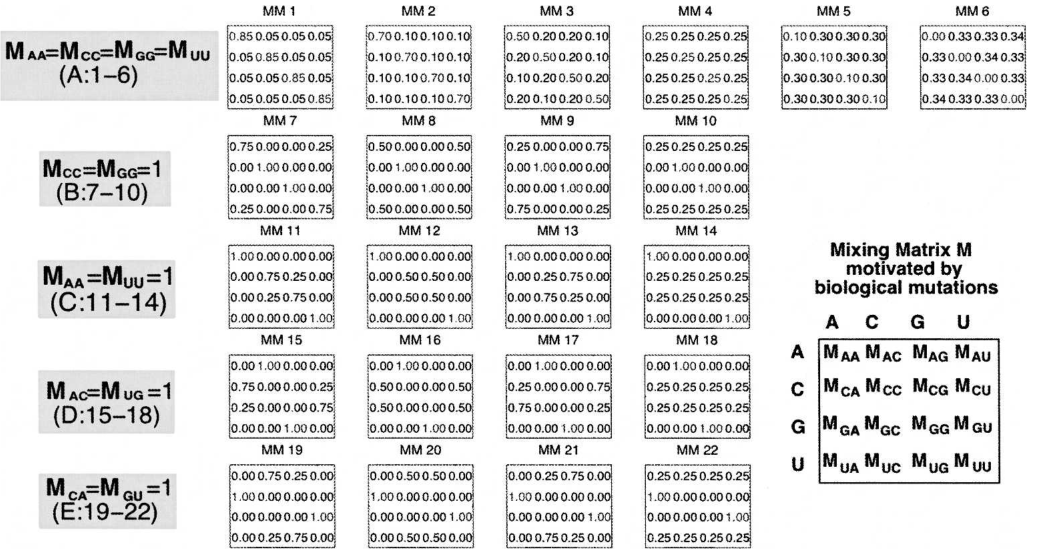
$$\sum_{j=\mathrm{A,C,G,U}} N_j(1 - \mathbf{M}_{jj}),$$

where N_j is the number of nucleotides of type j in the original sequence.

Ideally, we would like to determine the mixing matrices **M** for a target structural distribution or on the basis of specific biological-motivated contamination protocols. In practice, the inverse design problem—specify **M** and analyze the resulting structural distribution—is much easier to perform. We thus construct different mixing matrices motivated by biological covariance mutations and analyze their coverage of sequence space via a standard clustering method. Direct modeling of pools generated by a specific mixing matrix can be made by exploiting correlations between bases in folded RNAs. For example, the bases in paired and unpaired regions are correlated, allowing assignment of matrix elements for mutating bases in such regions.

Our biological motivation for choosing the mixing matrix classes is as follows. We consider mixing matrices with symmetric elements, $\mathbf{M}_{\mathrm{AU}}$=$\mathbf{M}_{\mathrm{UA}}$, $\mathbf{M}_{\mathrm{CG}}$=$\mathbf{M}_{\mathrm{GC}}$, $\mathbf{M}_{\mathrm{GU}}$=$\mathbf{M}_{\mathrm{UG}}$, to preserve base pairs. Such matrices cover the sequence subspace approximating covariance mutations (e.g., AU to UA, CG to GC, GC to UA). Covariance mutations have been used to analyze the secondary structure and sequence consensus of RNA sequence families. For example, this approach has been successfully applied to search for tRNA-related sequences and other small RNAs (Eddy and Durbin 1994). Alternatively, to disrupt stems and generate new structures, we can consider mixing matrices that do not preserve base pairs. Such matrices include asymmetric matrices without the property of covariance mutations. Noncovariance mutations, including random mutations, are commonly used to generate sequence pools for in vitro selection applications.

To sample the sequence space, we define five classes of mixing matrices motivated by biological considerations, based primarily on sequence transformations associated with covariance mutations. The mixing matrix classes are characterized by the following matrix elements: (A) varying diagonal elements $\mathbf{M}_{ii}$ with the condition $\mathbf{M}_{\mathrm{AA}}$=$\mathbf{M}_{\mathrm{CC}}$=$\mathbf{M}_{\mathrm{GG}}$=$\mathbf{M}_{\mathrm{UU}}$; (B) $\mathbf{M}_{\mathrm{CC}}$=$\mathbf{M}_{\mathrm{GG}}$=1; (C) $\mathbf{M}_{\mathrm{AA}}$=$\mathbf{M}_{\mathrm{UU}}$=1; (D) $\mathbf{M}_{\mathrm{AC}}$=$\mathbf{M}_{\mathrm{UG}}$=1; and (E) $\mathbf{M}_{\mathrm{CA}}$=$\mathbf{M}_{\mathrm{GU}}$=1. Within each class, several mixing matrices are constructed whose elements are distributed uniformly in steps of 0.25. A total of 22 mixing matrices representing the five classes are displayed in Figure 2. The matrix classes to which they belong are as



$\mathbf{M}_{AA}=\mathbf{M}_{CC}=\mathbf{M}_{GG}=\mathbf{M}_{UU}$ (A:1–6)

MM 1 $\begin{bmatrix} 0.85 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.85 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.85 \end{bmatrix}$

MM 2 $\begin{bmatrix} 0.70 & 0.10 & 0.10 & 0.10 \\ 0.10 & 0.70 & 0.10 & 0.10 \\ 0.10 & 0.10 & 0.70 & 0.10 \\ 0.10 & 0.10 & 0.10 & 0.70 \end{bmatrix}$

MM 3 $\begin{bmatrix} 0.50 & 0.20 & 0.20 & 0.10 \\ 0.20 & 0.50 & 0.20 & 0.10 \\ 0.10 & 0.20 & 0.50 & 0.20 \\ 0.20 & 0.10 & 0.20 & 0.50 \end{bmatrix}$

MM 4 $\begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$

MM 5 $\begin{bmatrix} 0.10 & 0.30 & 0.30 & 0.30 \\ 0.30 & 0.10 & 0.30 & 0.30 \\ 0.30 & 0.30 & 0.10 & 0.30 \\ 0.30 & 0.30 & 0.30 & 0.10 \end{bmatrix}$

MM 6 $\begin{bmatrix} 0.00 & 0.33 & 0.33 & 0.34 \\ 0.33 & 0.00 & 0.34 & 0.33 \\ 0.33 & 0.34 & 0.00 & 0.33 \\ 0.34 & 0.33 & 0.33 & 0.00 \end{bmatrix}$

$\mathbf{M}_{CC}=\mathbf{M}_{GG}=1$ (B:7–10)

MM 7 $\begin{bmatrix} 0.75 & 0.00 & 0.00 & 0.25 \\ 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.25 & 0.00 & 0.00 & 0.75 \end{bmatrix}$

MM 8 $\begin{bmatrix} 0.50 & 0.00 & 0.00 & 0.50 \\ 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.50 & 0.00 & 0.00 & 0.50 \end{bmatrix}$

MM 9 $\begin{bmatrix} 0.25 & 0.00 & 0.00 & 0.75 \\ 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.75 & 0.00 & 0.00 & 0.25 \end{bmatrix}$

MM 10 $\begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.00 & 1.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$

$\mathbf{M}_{AA}=\mathbf{M}_{UU}=1$ (C:11–14)

MM 11 $\begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.75 & 0.25 & 0.00 \\ 0.00 & 0.25 & 0.75 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}$

MM 12 $\begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.50 & 0.50 & 0.00 \\ 0.00 & 0.50 & 0.50 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}$

MM 13 $\begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.25 & 0.75 & 0.00 \\ 0.00 & 0.75 & 0.25 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}$

MM 14 $\begin{bmatrix} 1.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}$

$\mathbf{M}_{AC}=\mathbf{M}_{UG}=1$ (D:15–18)

MM 15 $\begin{bmatrix} 0.00 & 1.00 & 0.00 & 0.00 \\ 0.75 & 0.00 & 0.00 & 0.25 \\ 0.25 & 0.00 & 0.00 & 0.75 \\ 0.00 & 0.00 & 1.00 & 0.00 \end{bmatrix}$

MM 16 $\begin{bmatrix} 0.00 & 1.00 & 0.00 & 0.00 \\ 0.50 & 0.00 & 0.00 & 0.50 \\ 0.50 & 0.00 & 0.00 & 0.50 \\ 0.00 & 0.00 & 1.00 & 0.00 \end{bmatrix}$

MM 17 $\begin{bmatrix} 0.00 & 1.00 & 0.00 & 0.00 \\ 0.25 & 0.00 & 0.00 & 0.75 \\ 0.75 & 0.00 & 0.00 & 0.25 \\ 0.00 & 0.00 & 1.00 & 0.00 \end{bmatrix}$

MM 18 $\begin{bmatrix} 0.00 & 1.00 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.00 & 0.00 & 1.00 & 0.00 \end{bmatrix}$

$\mathbf{M}_{CA}=\mathbf{M}_{GU}=1$ (E:19–22)

MM 19 $\begin{bmatrix} 0.00 & 0.75 & 0.25 & 0.00 \\ 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.25 & 0.75 & 0.00 \end{bmatrix}$

MM 20 $\begin{bmatrix} 0.00 & 0.50 & 0.50 & 0.00 \\ 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.50 & 0.50 & 0.00 \end{bmatrix}$

MM 21 $\begin{bmatrix} 0.00 & 0.25 & 0.75 & 0.00 \\ 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \\ 0.00 & 0.75 & 0.25 & 0.00 \end{bmatrix}$

MM 22 $\begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 1.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$

Mixing Matrix M motivated by biological mutations

$$\begin{array}{c|cccc} & A & C & G & U \\ \hline A & \mathbf{M}_{AA} & \mathbf{M}_{AC} & \mathbf{M}_{AG} & \mathbf{M}_{AU} \\ C & \mathbf{M}_{CA} & \mathbf{M}_{CC} & \mathbf{M}_{CG} & \mathbf{M}_{CU} \\ G & \mathbf{M}_{GA} & \mathbf{M}_{GC} & \mathbf{M}_{GG} & \mathbf{M}_{GU} \\ U & \mathbf{M}_{UA} & \mathbf{M}_{UC} & \mathbf{M}_{UG} & \mathbf{M}_{UU} \end{array}$$

FIGURE 2. Our five classes of 22 mixing matrices (MM) for generating diverse sequence pools. The matrix classes are developed based on alteration of diagonal elements (class A) and covariance mutations (classes B–E). For pool synthesis using four vials, the mixing matrix is a 4×4 matrix specifying the molar fractions of nucleotide components A, C, G, and U in the four vials. The columns represent the molar fraction of the four bases in vial for each base denoted in each row.

follows: class A matrices 1–6, class B matrices 7–10, class C matrices 11–14, class D matrices 15–18, and class E matrices 19–22. Note that in vitro experiments effectively use random pools generated by a constant 4×4 mixing matrix, where all 16 elements are 0.25; this corresponds to our matrix 4.

Class A mixing matrices 1–6 are obtained by varying the magnitude of the diagonal elements. These matrices do not necessarily generate structure-preserving mutations. New RNA folds may be obtained from known RNAs through such noncovariance mutations. Matrix classes B–E tend to generate sequences preserving the original secondary structure, although the mixing matrices also alter bases in the unpaired regions. Specifically, matrix classes B and C tend to preserve CG and AU base pairs, respectively, by fixing bases associated with these base pairs; matrices in class D convert AU to CG base pairs; and matrices in class E transform CG to AU base pairs. Thus, our constructed matrix classes represent both covariance and noncovariance mutations to allow generation of pools with target structures and enhance pool sequence and structural diversity. Asymmetry adds additional variability; asymmetric mutation rates for base pairs can introduce defects in stems. These matrix properties are summarized in Table 1.

Role of graph theory in pool design

RNA graph theory aids in pool design in three ways. First, structural diversity in designed pools can be assessed quantitatively using sets of enumerated graphs, as we have done for random pools (Gevertz et al. 2005). Second, graph theory analysis suggests many RNA-like motifs that have not been observed (see RAG Web resource at http://monod.biomath.nyu.edu/rna), and thus pool design using mixing matrices can target these motifs. Third, graph motifs are intuitively cataloged in RAG as *n*-vertex families, naturally suggesting groupings to consider in pool design. Thus, RNA graphs define the space of RNA topologies or shapes for assessing and designing RNA pools. A similar representation of abstract RNA shapes using bracket notations has also been developed by Giegerich et al. (2004).

In RAG, RNA graphs are organized into *n*-vertex families, and members of a family are ordered using a topological index (i.e., Laplacian eigenvalues) (Fera et al. 2004; Gan et al. 2004). Structural complexity can be measured by the graph's vertex number (*V*) and the second smallest Laplacian eigenvalue (λ_2). For example, a linear chain has a smaller

TABLE 1. Properties of five mixing matrix classes for pool generation

Mixing matrix class	Condition	Effect	Symmetry
A: 1–6	$\mathbf{M}_{AA}=\mathbf{M}_{CC}=\mathbf{M}_{GG}=\mathbf{M}_{UU}$	Variations of random pools	1–2, 4–6
B: 7–10	$\mathbf{M}_{CC}=\mathbf{M}_{GG}=1$	Conservation of C and G	7, 10
C: 11–14	$\mathbf{M}_{AA}=\mathbf{M}_{UU}=1$	Conservation of A and U	11–12, 14
D: 15–18	$\mathbf{M}_{AC}=\mathbf{M}_{UG}=1$	Covariation of AU to CG	None
E: 19–22	$\mathbf{M}_{CA}=\mathbf{M}_{GU}=1$	Covariation of CG to AU	None

Note that symmetric mixing matrices have symmetric elements (e.g., $\mathbf{M}_{AU}=\mathbf{M}_{UA}$, $\mathbf{M}_{CG}=\mathbf{M}_{GC}$, $\mathbf{M}_{GU}=\mathbf{M}_{UG}$) that cover the sequence subspace approximating covariance mutations (e.g., AU to UA, CG to GC, GC to UA).

eigenvalue than a branched structure. The number of motifs in an n-vertex family increases with n, the number of vertices. For example, the 6-vertex tree family has six distinct trees and the 7-vertex family has 11 trees (see RAG Web resource at http://monod.biomath.nyu.edu/rna). For easy reference, each tree motif is labeled by vertex number and ordering within the family; for example, members of the 6-vertex family are labeled 6_1, 6_2, . . ., 6_6. Since vertex number (V) is related to RNA sequence length L, these are constant length pools; in fact, we found empirically that L=20(V−1) (Gan et al. 2003). Our pool design will focus on tree structures because RNA folding algorithms for tree structures are efficient; computationally demanding pseudoknot folding algorithms are also available (Rivas and Eddy 1999; Ren et al. 2005). RNA graph theory also provides a complete set of pseudoknot and nonpseudoknot motifs for more general assessment of pool structural diversity.

Starting sequences for pool generation

The six starting sequences with distinct tree structures (Fig. 3) are 70S (chain F) (80 nt), tRNA (81 nt), P5abc domain of group I intron (56 nt), GTP-binding aptamer (69 nt), modified P5abc domain (51 nt), and modified GTP-binding aptamer (54 nt). As shown in Figure 3, distinct tree structures are represented as graphs by converting stems to edges and other structural elements (e.g., loop, bulge, etc.) to vertices according to tree graph rules developed previously (Gan et al. 2003). As shown in Figure 3, the Laplacian eigenvalue (λ_2) indicates the structural complexity of starting sequences. Generally, the starting structure allows exploration of the structural neighbors of that structure via mutations. For random mutation rates (constant matrix elements of 1/4), the generated pools have no memory of the starting sequence. We generate pools with all possible combinations of 22 mixing matrices and six starting sequences for pool structured designs.

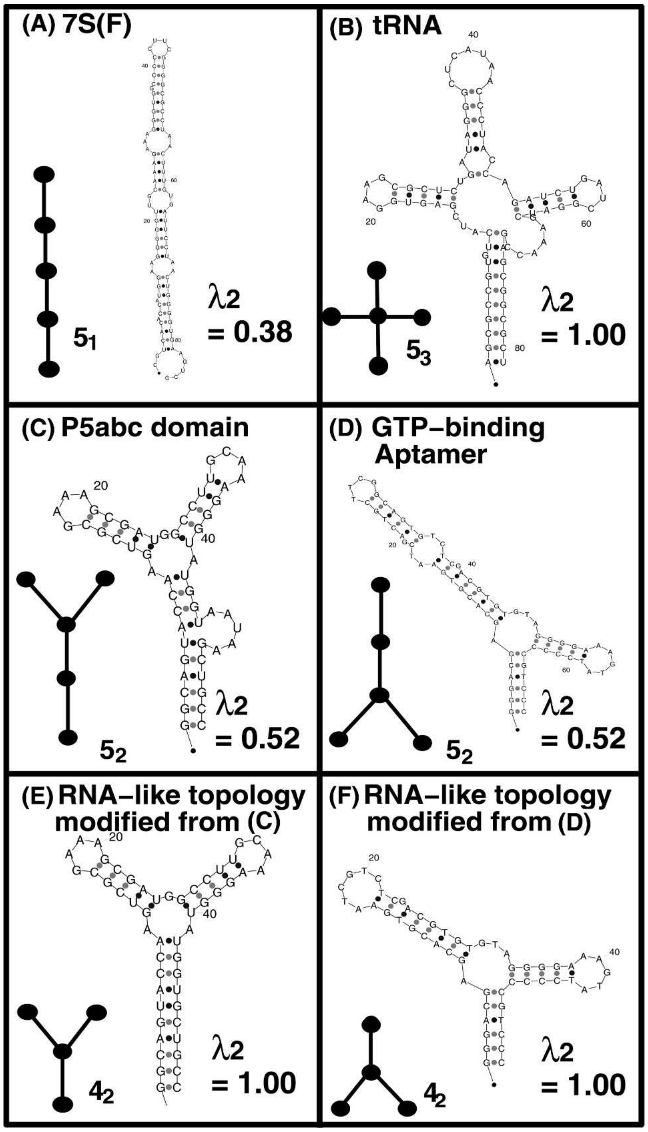


FIGURE 3. Starting sequences and their secondary structures for pool synthesis using mixing matrices. Displayed are the secondary structures and corresponding tree graphs for four existing and two modified existing RNAs. Laplacian eigenvalue (λ_2) of the tree graph indicates the structural complexity.

Mathematical relations between RNA sequence pool and structure space

Here we define the mathematical relations between the RNA sequence pool and the corresponding shape space using RNA graphs and mixing matrices (MM). Specifically, the process of generating the sequence pool using a mixing matrix **M** and a starting sequence S can be mathematically formulated. For a 4×4 mixing matrix **M** and an n-nt starting sequence $S=s_1s_2s_3...s_n$, where s_i is A, C, G, or U, the $4\times n$ probability matrix **Y** defining the effect of **M** on S is

$$\mathbf{Y} = [\mathbf{M}^{\mathrm{T}}(\mathbf{X}_1), \mathbf{M}^{\mathrm{T}}(\mathbf{X}_2), \mathbf{M}^{\mathrm{T}}(\mathbf{X}_3), \cdots, \mathbf{M}^{\mathrm{T}}(\mathbf{X}_n)], \qquad (1)$$

where the four-component vector $\mathbf{X}_i$, i=1, 2, . . ., n, represents the nucleotide base:

$$\mathbf{X}_i = \begin{cases} [1,0,0,0]^{\mathrm{T}} & \text{if } s_i = \mathrm{A}, \\ [0,1,0,0]^{\mathrm{T}} & \text{if } s_i = \mathrm{C}, \\ [0,0,1,0]^{\mathrm{T}} & \text{if } s_i = \mathrm{G}, \\ [0,0,0,1]^{\mathrm{T}} & \text{if } s_i = \mathrm{U}. \end{cases} \qquad (2)$$

The matrix **Y** represents the sequence pool generated by **M** with starting sequence S. For example, if

$$\mathbf{M} = \mathrm{MM2} = \begin{bmatrix} 0.7 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.7 \end{bmatrix}$$

and

$$S = \text{CAU, i.e.} \left(\mathbf{X}_1 = \begin{bmatrix} 0\\1\\0\\0 \end{bmatrix}, \mathbf{X}_2 = \begin{bmatrix} 1\\0\\0\\0 \end{bmatrix}, \text{ and } \mathbf{X}_3 = \begin{bmatrix} 0\\0\\0\\1 \end{bmatrix} \right),$$

then **Y** is given by

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13}\\ Y_{21} & Y_{22} & Y_{23}\\ Y_{31} & Y_{32} & Y_{33}\\ Y_{41} & Y_{42} & Y_{43} \end{bmatrix} = \begin{bmatrix} 0.1 & 0.7 & 0.1\\ 0.7 & 0.1 & 0.1\\ 0.1 & 0.1 & 0.1\\ 0.1 & 0.1 & 0.7 \end{bmatrix}.$$

The probability of finding a new sequence S' in the pool can be calculated from **Y**. For example, $P(S'\text{=ACU})=Y_{11}\bullet Y_{22}\bullet Y_{43}$=0.1•0.1•0.7 and $P(S'\text{=GUC})=Y_{31}\bullet Y_{42}\bullet Y_{23}$=0.1•0.1•0.1. Similarly, we can calculate the frequency of sequences with a specified base-pairing scheme using this mathematical formulation. However, a rigorous mapping of sequence space (**Y**) to shape space (possible RNA graphs) requires an RNA folding algorithm, as described in our previous work (Gan et al. 2003).

The challenge in computational pool design is to find an optimal set of mixing matrix (**M**), starting sequence (S), and weight (or pool fraction) for generation of target-structured pools. In principle, the mixing matrix can be calculated using statistical thermodynamics from the distribution of shapes in the designed pool. Assuming that the designed pool consists of N noninteracting RNA molecules, the probability of finding topology t in the pool is

$$P(t) = N^{-1} \sum_{i=1}^{N} \frac{\sum_{E_i} \hat{t}\rho(E_i) \exp[-\beta E(S_i)]}{\sum_{E_i} \rho(E_i) \exp[-\beta E(S_i)]}, \tag{3}$$

where $E(S_i)$ is the energy of sequence S_i, $\beta=1/kT$, $\rho(E_i)$ is the density of states, and $\hat{t}$ is an RNA topology operator defining tree or pseudoknot shapes enumerated by RNA graph theory. Equation (3) defines the relation between sequence pool $\{S_i\}$ and target structural distribution $P(t)$. Recently, we calculated $P(t)$ distributions for 25–100 nt random pools using a folding algorithm and a program for converting secondary structures into tree graphs (Gevertz et al. 2005). Thus, the goal is to determine the sequence pool $\{S_i\}$, or mixing matrices generating that pool, to produce the target distribution $P(t)$. In the Appendix, we describe a practical protocol for finding optimal mixing matrices approximating the target $P(t)$ based on analyses of sequence space and pool structural distribution. Alternative pool design methods may also be developed based on Equation (3).

Pool sizes

For practical reasons, our computations used relatively small pools of 10,000 sequences. To show the effect of pool size, Figure 4 plots the frequency of several tree motifs (4_1, 4_2, 5_1, 5_2, 5_3, and 6_1) for pool sizes of 5,000–60,000 sequences using mixing matrix 4 (MM4) and the initial tRNA sequence. We see that the pool fractions for distinct tree motifs saturate beyond 5000 sequences, indicating that the error due to sample size is small. The rapid saturation of pool fraction stems from mapping secondary structures using simple graphs. If detailed motif features (size of loops, stems, etc.) are incorporated into the mapping, larger pool sizes will certainly be required.

Measures of sequence and structure similarity

RNA graphs allow global analysis of RNA secondary structures. To analyze sequence and structure space of designed pools at the base level, we use two standard measures of distance between any two RNAs: Hamming distance and tree edit distance. The Hamming distance is the number of differing letters between two equal-length RNA sequences aligned end to end (Hamming 1987). The tree edit distance between two (full) tree secondary structures measures the minimum sum of the cost (insertion, deletion, and replacement of nodes) along an edit path for converting one tree into another (Hofacker 2003). We use the tree edit distance measure as implemented in RNAdistance of the Vienna RNA package available at http://rna.tbi.univie.ac.at. Other distance measures, such as string edit distance or base-pair distance implemented in RNAdistance (Hofacker 2003), can also be used to compare two RNA structures; also available are the more sophisticated sequence/structure alignment algorithms Foldalign (Havgaard et al. 2005) and Dynalign (Mathews and Turner

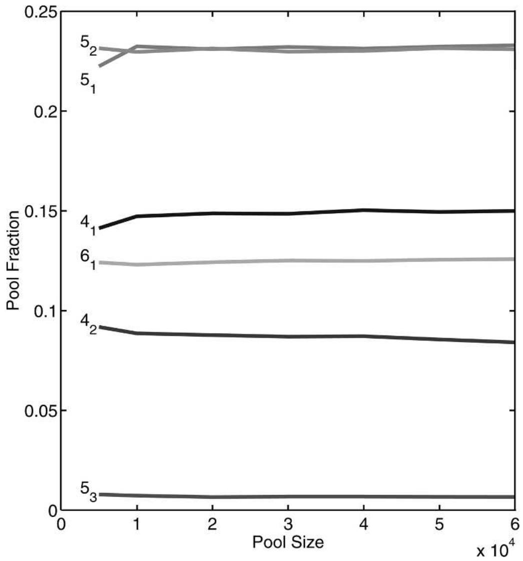


FIGURE 4. Effect of pool size on pool fractions of selected tree motifs. The pools are generated using random mixing matrix 4 and starting tRNA sequence in Figure 3B.

2002). Here we use Hamming and tree edit distances together with a clustering technique—the multidimensional scaling (MDS) method (Cox and Cox 1994) implemented in the R statistical package (http://www.r-project.org)—to map the RNA sequence/structure space.

RESULTS

Coverage of sequence space regions generated by mixing matrix classes and starting sequences is distinct from random pools

We consider our set of starting sequences (Fig. 3) and 22 mixing matrices (Fig. 2) to explore the sequence/structure space of sequence pools and to optimize target pools.

To analyze the clustering patterns in sequence space, we cluster all sequences generated by the mixing matrices using a standard clustering technique (e.g., MDS), allowing visualization of sequence similarity/dissimilarity properties (Cox and Cox 1994). A similar procedure is commonly used for investigating the diversity of chemical compound libraries (Xie et al. 2000). Such analysis helps establish the relation between each mixing matrix and the generated sequence space. Given a pool of sequences, we define Hamming distances (number of dissimilar bases) between all pairs of sequences (see Materials and Methods), allowing data projections in 2D, 3D, and higher dimensions.

Figure 5, A and B, shows the 2D and 3D clustering of sequences generated by 22 mixing matrices using starting sequences for the modified P5abc (Fig. 3E) and 70S (Fig. 3A) RNAs, respectively. In Figure 5A, we see that the sequences generated by the five mixing matrix classes and the P5abc starting sequence cover distinct regions of the sequence space, especially the boundary and central regions. The boundaries are spanned by matrix classes B–E, and the central region by matrix class A. Intriguingly, the random MM4 produces sequences that are localized in sequence space, showing that the standard approach does

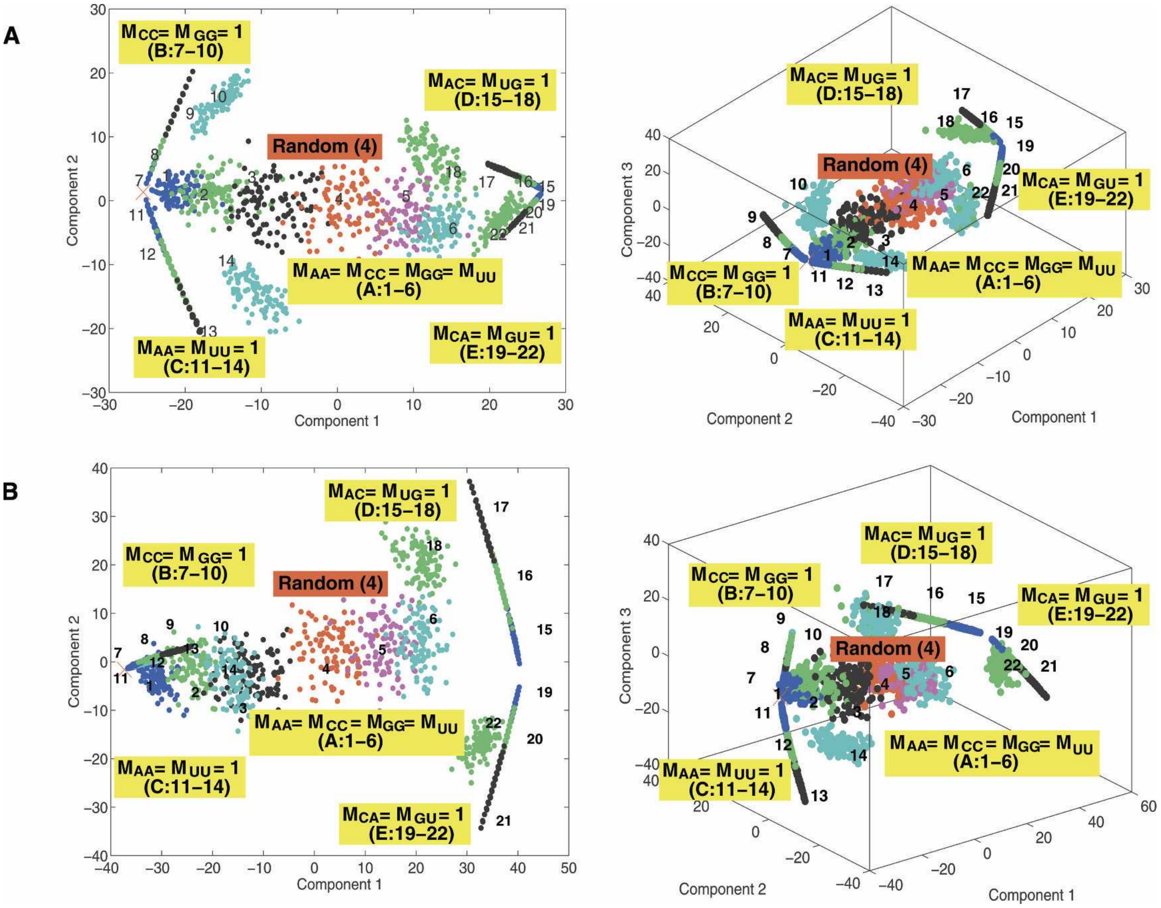


FIGURE 5. Two- and three-dimensional clustering plots using the MDS transformation for sequences generated from 22 mixing matrices (labeled 1–22) (*A*) starting with a modified P5abc domain (Fig. 3E) and (*B*) with 70S (Chain F) (Fig. 3A). The distance between any sequence pair is the Hamming distance, a measure of the number of dissimilar nucleotide bases. Axes represent two or three largest components of the projection. Each color represents a sequence pool generated by one of the 22 mixing matrices; the "×" mark on the *left* represents result for an invariant sequence transformation corresponding to diagonal matrix $\mathbf{M}_{ii}$=1. The mixing matrices are grouped into five classes (A–E) according to their matrix properties (Table 1).

not provide an efficient sampling of diverse regions of sequence space in agreement with observations. More adequate sampling of sequence space is provided by the nonrandom mixing matrices. The 70S starting sequence yields similar global 2D and 3D clustering patterns: the five matrix classes yield clusters in distinct sequence regions (Fig. 5B). Although the 22 mixing matrices of Figure 2 provide a comprehensive coverage of the sequence space, some regions remain sparsely populated, indicating that the matrix classes must be expanded for more complete coverage. Still, the chosen matrix set is diverse enough for initial assessment of our structured pool design concept.

RNA motif distributions depend on generating mixing matrices

By "folding" the resulting pool sequences using Vienna RNAfold and converting motifs into tree graphs, we can assess each pool's structural distribution (Gan et al. 2003; Gevertz et al. 2005). Figure 6 shows the frequencies of various tree motifs in pools generated by the 22 mixing matrices starting with the 4_2 tree motif from the modified P5abc (Fig. 3E). Corresponding distributions for all six starting sequences in Figure 3 are shown in Table 2. In Table 2, populations of <0.5% are reported as 0.

First, it is evident that motif distributions in our designed pools vary significantly from those in random pools (MM4; see arrow and dot-filled histograms). For example, with the Figure 3E starting sequence, the 4_2 tree motif has a yield of ~8%, for the random mixing matrix, versus 79%, 46%, and 34% in matrix pools 7–9, which do not mutate C and G bases, respectively. At the other extreme, matrices 20–22 produce small proportions (5%–6%) of 4_2 tree structure. For the 3_1 tree motif, mixing matrices 5, 6, 14, 15, 19, 20, 21, and 22 generate higher pool fractions than the random pool 4, whereas matrices 7–13 yield considerably lower numbers. Thus, motif distributions depend on both the mixing matrix and the starting sequence because different sequence space regions result. Because the overlap of sequence regions of our 22 pools is weak, the motif distribution is very different in each case (Fig. 5).

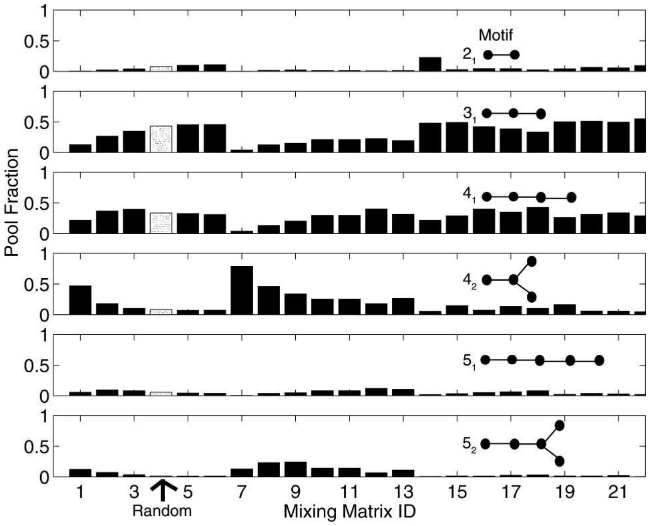


FIGURE 6. Pool fraction distributions for six tree motifs in pools generated from 22 mixing matrices (labeled 1–22), starting with a modified P5abc domain (Fig. 3E), which has a 4_2 tree motif. The results for the random pool 4 (marked with arrow) are displayed as dot-filled histograms.

Second, we note a pattern in the correlation between 4_2 and 5_2 trees and between 3_1 and 4_1 trees. With the Figure 3E starting sequence, Figure 6 and Table 2 show that sequence pools from matrices 7–13 have a large proportion of 4_2 and 5_2 tree structures compared with the random pool 4. Similarly pools from matrices 14–22 possess >30% 3_1 and 4_1 tree structures. This pattern emerges because the 4_2/5_2 and 3_1/4_1 tree-motif pairs are related by an internal loop or bulge, which can be created by a few mismatched base pairs.

Third, the structural distributions generated by a tRNA sequence (5_3 tree motif; Fig. 3B) differ from those for the modified P5abc domain (4_2 tree motif; Fig. 3E) in one important respect (Table 2). The most likely motifs are the simpler 5_1 and 5_2 trees rather than the starting 5_3 motif. MM1, for example, generates only 5% 5_3 motif, but 26% 5_2 tree. In contrast, MM7, which preserves C and G bases, generates 23% 5_3 trees, while other combinations of mixing matrices and starting structures yield almost no 5_3 trees (Table 2). The mean mutation rate for MM7 from the starting tRNA sequence (Fig. 3B) is ~0.1 (~8.5 positions among 81 nt). Thus, the 5_3 motif populations produced by matrices 1 and 7–11 are much higher than in random pools (1.3%). The difficulty of generating significant populations with the tRNA-like 5_3 tree motif likely stems from the lower thermodynamic stability of 5_3 compared to 5_1 and 5_2 trees. Our analysis shows that matrices 7–9 generate sequences that are favorable for stabilizing the 5_3 motif because these matrices preserve energetically favorable CG base pairs.

To increase the population of complex folds like the tRNA-like 5_3 tree motif, we consider refining the mixing matrices 7–9. Since class B matrices 7–9 produce a higher frequency of 5_3 tree, we search for matrices in the neighborhood of this class by exhaustively varying the elements in each row with $\Delta \mathbf{M}_{ij}$=0.2, yielding 56 possible cases. Assuming that each row is independent, the total number of mixing matrices around the class B matrix region is 56^2 or 3136, since two rows (second and third) are identical and the other two rows (first and last) have a total of 56 cases each. We filter the 3136 trial mixing matrices, yielding better than a 23% tRNA-like 5_3 tree structure. Remarkably, 12 of the 3136 mixing matrices for tRNA-like topology fulfill our requirement forming 5_3 motifs. We use these

TABLE 2. Structural distributions of pools generated by 22 mixing matrices in Figure 2 starting with the six sequences in Figure 3, A–F

Starting sequence/structure	Result: Motif ID	Pool fraction (%)																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
(Fig. 3A), 5_1	4_1	**23**	21	18	21	**22**	**22**	**25**	**25**	**25**	**25**	18	14	10	**32**	**33**	**31**	**22**	17	**35**	**38**	**39**	**40**
	4_2	7	8	9	11	11	**12**	3	3	3	3	6	9	9	**14**	**17**	**14**	**25**	7	8	6	6	5
	5_1	**35**	**28**	**26**	24	24	24	**48**	**39**	**39**	**39**	**36**	23	**25**	18	22	19	22	**26**	**28**	**26**	**25**	**28**
	5_2	11	19	**22**	21	21	21	5	9	9	9	18	**24**	**26**	12	12	20	19	22	13	14	16	8
	6_1	**14**	**12**	**12**	9	8	8	**14**	**17**	**17**	**17**	**15**	**13**	**16**	4	3	2	2	**12**	5	4	4	5
(Fig. 3B), 5_3	4_1	9	9	11	15	**16**	**18**	7	9	10	9	11	11	12	**24**	**29**	**33**	**31**	15	7	9	11	**17**
	4_2	6	7	6	9	9	9	1	3	6	5	5	6	7	**15**	6	7	5	7	6	6	4	**10**
	5_1	18	20	22	23	**25**	23	15	13	17	16	23	22	22	19	**24**	**25**	**26**	**25**	20	18	23	**24**
	5_2	**26**	**27**	**24**	23	22	23	12	17	14	**28**	20	21	20	20	12	9	11	22	23	23	19	22
	5_3	**5**	1	1	1	1	1	**23**	**12**	**13**	**3**	**3**	1	1	0	0	0	0	0	**3**	**3**	**2**	1
	6_1	10	12	**14**	12	12	11	7	8	11	10	**15**	**16**	**14**	6	10	8	9	**13**	12	12	**14**	11
(Fig. 3C), 5_2	2_1	0	1	2	4	**5**	**5**	0	0	0	1	0	0	1	**20**	**6**	**6**	3	0	4	4	**5**	**7**
	3_1	8	19	24	31	**33**	**34**	1	5	7	12	13	19	14	**47**	**34**	**38**	**32**	22	25	30	**32**	**39**
	4_1	21	34	39	39	38	38	5	18	28	31	19	29	24	22	26	28	23	**40**	29	35	33	36
	4_2	**30**	**18**	13	11	10	10	**31**	**22**	9	**18**	**24**	**19**	**22**	8	**16**	**18**	**26**	**13**	**21**	**12**	9	8
	5_1	**12**	**15**	**14**	10	10	9	3	9	10	**13**	**18**	**15**	**14**	2	6	3	4	**15**	**11**	**14**	**15**	8
	5_2	**27**	**11**	**7**	4	4	4	**60**	**44**	**47**	**22**	**22**	**15**	**23**	1	**12**	**8**	**11**	**9**	**10**	4	**6**	2
(Fig. 3D), 5_2	3_1	2	5	7	11	**12**	**12**	0	1	1	5	3	5	4	**23**	4	9	8	6	8	**13**	10	**21**
	4_1	**30**	23	25	29	**30**	**30**	2	6	8	19	16	22	17	32	28	**30**	26	27	**32**	**34**	**34**	**43**
	4_2	**13**	11	11	12	**13**	**13**	9	11	**13**	**14**	11	**14**	**16**	**14**	8	12	12	11	7	9	10	8
	5_1	**24**	22	**25**	23	22	22	1	2	2	12	17	21	14	15	**26**	22	18	**25**	19	20	21	17
	5_2	16	**24**	**19**	16	15	15	**52**	**46**	**48**	**33**	**38**	**26**	**35**	9	**28**	**20**	**25**	**20**	**28**	**17**	**20**	7
	6_1	**5**	**6**	**7**	5	4	4	0	0	0	2	4	5	3	2	2	3	3	**6**	3	3	3	2
(Fig. 3E), 4_2	2_1	1	2	4	8	**10**	**11**	0	2	2	1	1	1	1	**23**	3	5	4	2	4	6	6	**10**
	3_1	13	26	35	43	**45**	**46**	4	12	15	21	21	23	19	**48**	**49**	42	38	33	**50**	**51**	**50**	**55**
	4_1	22	**37**	**40**	34	33	31	4	13	20	30	30	**40**	31	22	29	**40**	**35**	**43**	26	31	34	29
	4_2	**47**	**18**	**10**	8	7	7	**79**	**46**	**34**	**25**	**25**	**18**	**26**	5	**15**	7	**13**	**10**	**16**	6	5	5
	5_1	6	**9**	**8**	6	4	4	0	4	5	**8**	**8**	**11**	**10**	2	3	5	6	**8**	2	4	3	2
	5_2	**12**	**7**	**3**	2	1	1	**13**	**23**	**24**	**14**	**14**	**7**	**11**	0	1	1	**3**	**3**	1	1	2	0
(Fig. 3F), 4_2	3_1	19	26	29	37	**40**	**39**	12	16	18	28	22	28	28	**46**	**32**	**41**	**41**	30	33	**40**	**45**	**52**
	4_1	24	36	**40**	37	36	36	5	12	11	27	32	35	28	25	**41**	37	33	**42**	32	34	33	29
	4_2	**37**	**19**	**12**	10	9	10	**67**	**54**	**53**	**29**	**31**	**22**	**32**	8	**21**	**13**	**15**	**13**	**25**	**16**	**14**	7
	5_1	5	**10**	**11**	8	6	6	0	0	0	5	6	7	4	3	3	4	4	**9**	3	3	3	3
	5_2	**13**	**7**	**5**	3	2	2	**13**	**15**	**13**	**11**	**8**	**6**	**6**	1	1	1	1	**4**	**4**	3	2	1

Each pool has 10,000 sequences. Bold fonts represent frequencies greater than those in the random pool (MM4).

"MMT" matrices (Fig. 7) to generate graph-structural distributions with tRNA shapes, as shown in Table 3. For example, MMT6 generates a 51% tRNA-like 5_3 tree motif with 15 mutations out of 81 bases.

Note that each pool generated by the 12 mixing matrices has 5000 sequences. Compared with the random pool, these refined matrices generate complex structures (e.g., 5_3 and also 6_4 and 6_5) routinely. This search demonstrates the feasibility of improving yields of specific structures using appropriate mixing matrices and starting sequences.

Sequence/structure correlations exist in designed pools

The above survey of tree structural distributions provides an analysis of RNA shapes in designed pools. We now analyze sequence/structure distributions at the nucleotide base level generated by the 22 mixing matrices starting with a 51-nt P5abc domain (Fig. 3E). In Figure 8, we use sequence Hamming and tree edit distances to quantify sequence and structure distances, respectively, as defined in Materials and Methods. Recall that the Hamming distance is the number of differing letters between two equal-length RNA sequences aligned end to end (Hamming 1987). The tree edit distance between two (full) tree secondary structures measures the minimum sum of the cost (insertion, deletion, and replacement of nodes) along an edit path for converting one tree into another (Hofacker 2003).

All mixing matrices give rise to localized distributions as measured from the initial sequence/structure. As the matrix diagonal elements decrease from 0.85 to 0 (class A matrices 1–6), both sequence and structure distances increase. The sequence distance is determined by the strength of the nondiagonal elements, with matrices 1 and 6 yielding the smallest and largest Hamming distances, respectively. As expected, classes B (7–10) and C (11–14) with fixed C, G

$$\text{MMT 1}\ \begin{bmatrix}0.8&0.2&0.0&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.2&0.0&0.0&0.8\end{bmatrix}\quad \text{MMT 2}\ \begin{bmatrix}0.8&0.2&0.0&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.2&0.2&0.0&0.6\end{bmatrix}\quad \text{MMT 3}\ \begin{bmatrix}1.0&0.0&0.0&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.0&0.2&0.0&0.8\end{bmatrix}$$

$$\text{MMT 4}\ \begin{bmatrix}1.0&0.0&0.0&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.2&0.0&0.0&0.8\end{bmatrix}\quad \text{MMT 5}\ \begin{bmatrix}1.0&0.0&0.0&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.4&0.0&0.0&0.6\end{bmatrix}\quad \text{MMT 6}\ \begin{bmatrix}1.0&0.0&0.0&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.6&0.4&0.0&0.0\end{bmatrix}$$

$$\text{MMT 7}\ \begin{bmatrix}0.8&0.0&0.2&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.0&0.0&0.0&1.0\end{bmatrix}\quad \text{MMT 8}\ \begin{bmatrix}0.8&0.0&0.2&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.0&0.2&0.0&0.8\end{bmatrix}\quad \text{MMT 9}\ \begin{bmatrix}0.8&0.0&0.2&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.2&0.0&0.0&0.8\end{bmatrix}$$

$$\text{MMT 10}\ \begin{bmatrix}0.8&0.2&0.0&0.0\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.0&0.0&0.0&1.0\end{bmatrix}\quad \text{MMT 11}\ \begin{bmatrix}0.8&0.0&0.0&0.2\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.0&0.0&0.0&1.0\end{bmatrix}\quad \text{MMT 12}\ \begin{bmatrix}0.8&0.0&0.0&0.2\\0.0&1.0&0.0&0.0\\0.0&0.0&1.0&0.0\\0.0&0.2&0.0&0.8\end{bmatrix}$$

FIGURE 7. Twelve refined or variants of class B mixing matrices for enhancing pools with the tRNA-like (5_3) structure (MMT); they generate pools with at least 23% 5_3 tree motif.

and A, U, respectively, produce distributions with small Hamming distances. In contrast, classes D (15–17) and E (19–21) produce sequences with the maximum Hamming distance because no identity base transition is allowed.

As the mixing matrices are altered, the distribution of the tree edit distances also changes. Generally, tree edit distance increases with mutation rate. For example, Figure 8 shows that tree edit distances become larger as diagonal elements of matrix classes A, B, and C decrease. Changing C and G bases (matrix class C) has a larger effect on the starting structure than changing A and U bases (matrix class B), as evident from the pool distances from the origin. This is due to lower free energies associated with GC base pairs compared to AU base pairs. Thus, Figure 8 indicates that the distribution of sequence/structure distances from the initial sequence is controlled by the elements of the mixing matrices. Although the patterns of sequence/structure distributions are not sensitive to the starting sequences (data not shown), the densities within the localized regions are markedly changed. Figure 8 shows that the pools generated by most mixing matrices (except for 1, 8, and 9) and starting sequences of a modified P5abc domain produce a single cluster. We find that contour plots with string edit distance or base-pair distance (data not shown) show somewhat less information about pool structural properties than those with tree edit distance. Other secondary structure measures may also be used to capture structural differences among folds in the same vertex or tree class. For example, it is informative to know the distribution of stem, loop, and bulge sizes (Fontana et al. 1993).

TABLE 3. Structural distributions of pools generated by 12 refined class B mixing matrices in Figure 7 starting with the tRNA sequence in Figure 3B

	Result:		Pool fraction (%)											
Starting sequence/structure	Motif ID	Random	1	2	3	4	5	6	7	8	9	10	11	12
(Fig. 3B), 5_3	4_1	15	8	1	0	1	0	1	2	**19**	10	6	2	7
	4_2	9	3	0	0	2	**13**	1	1	2	1	1	3	3
	5_1	23	18	1	5	2	0	14	8	20	20	16	16	19
	5_2	23	13	12	5	9	16	15	18	7	7	10	**27**	19
	5_3	1	**24**	**50**	**50**	**35**	**39**	**51**	**30**	**38**	**44**	**29**	**27**	**24**
	6_1	12	9	2	1	2	1	1	5	4	5	6	3	7
	6_2	8	5	3	5	2	4	6	**9**	4	4	6	8	**9**
	6_4	1	**9**	**19**	**15**	**20**	**22**	1	**9**	**2**	**6**	**12**	**2**	**3**
	6_5	0	**8**	**10**	**15**	**21**	**5**	**8**	**11**	**2**	**1**	**9**	**9**	**6**

Bold fonts represent frequencies greater than those in the random pool.

Parameter optimization can lead to design of structured RNA pools

The preceding analysis of sequence space and assessment of structural distributions generated by nonrandom mixing matrices allow design of target structured pools. Here we use the pool design algorithm (Appendix) to develop several structured pools by selecting an optimal combination of starting sequences, mixing matrices, and associated weights $\{(S_i, \mathbf{M}_i, \alpha_i)\}$. The best combination for a target pool is dictated by the frequency data (Fig. 6; Tables 2, 3).

To illustrate, Table 4 shows four examples of designed pools that are rich in specific tree structures (e.g., 4_1, 5_1, 5_2); also displayed are their pool characteristics (mixing matrix weights and tree motif frequencies). Specifically, our target pools are: Pool T_A with 4_1 and 4_2 structures; Pool T_B with 5_1, 5_2, and 5_3 structures; Pool T_C with 4_2, 5_2, and 5_3 structures; and Pool T_D with 4_1, 4_2, and 5_3 structures. Pools T_A and T_B are 4- and 5-vertex pools, respectively, and Pools T_C and T_D are pools with mixed n-vertex structures. Each designed pool represents an optimal combination of starting sequences, mixing matrices, and associated weights derived using Step 5 of our design algorithm (see Appendix). Briefly, we initially choose pool fractions $T_1, T_2, \ldots, T_n$ for target motifs and the number of mixing matrices to approximate the target pool. We then use Equation (6) in the Appendix to calculate the weight α_1, which depends on T_1, starting sequence S_1, and mixing matrix $\mathbf{M}_1$. Next, we minimize the error between the target and estimated target pool fractions, Equation (8) in the Appendix, over all pools generated by starting sequence/mixing matrix pairs $\{(S_i, \mathbf{M}_i)\}$. This procedure yields optimized starting sequences, mixing

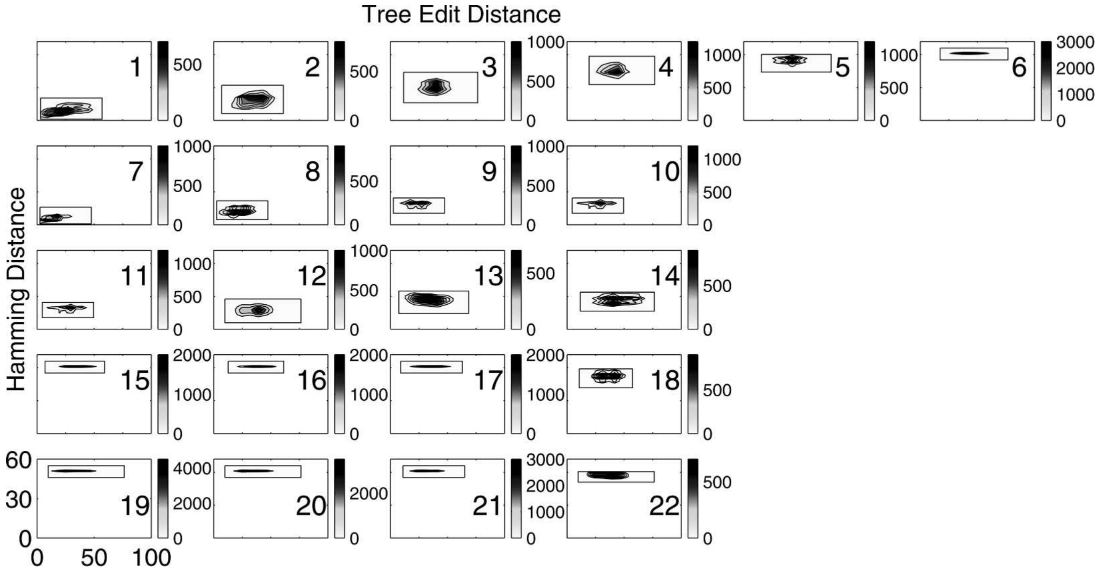


FIGURE 8. Contour plots of sequence/structure relationships using Hamming distance versus tree edit distance for pools generated by 22 mixing matrices, starting from a modified P5abc domain (Fig. 3E). Note that the *X* and the *Y* axes are always 0–100 and 0–60, respectively, and that each intensity bar indicates the frequency of joint distance distributions (the frequency outside the box is 0). There are 10,000 sequences in each pool.

matrices, and weights; the mean mutation rate is calculated based on these sequences, mixing matrices, and their weights.

As shown in Table 4, the optimized Pool T_A for a 30% 4_2 tree (T_1) and for a 25% 4_1 structure (T_2) is constructed using the Figure 3E starting sequence for matrices 8 and 3 with weights of 0.556 and 0.444, respectively. The mean mutation rate is 0.337 compared to the random base mutation rate of 0.75. Correspondingly, our designed pool for 4_1 and 4_2 motifs contains 25% and 30% 4_1 and 4_2 trees, respectively, compared with 29% and 12% for the random pool (MM4). The increase of 4_2 species is accompanied by the decrease of the 5_1 structure to 6% compared with 23% for the random Pool T_F. The next highest species in Pool T_A is 3_1 (22%).

Pool T_B, targeting 20% each of the 5_1, 5_2, and 5_3 structures, is generated from the Figure 3A sequence for 5_1 with matrix 13 at a weight of 0.18 and the Figure 3B sequence for 5_3 with matrix T12 at a weight of 0.82. For this pool optimization, we expanded our mixing matrix/starting sequence repertoire to include those in Table 3 (the 12 mixing matrices for generating pools with a high frequency of 5_3 motifs, which are extremely rare in random pools). Thus, this optimization was performed over the set of 144 (22×6+12×1) mixing matrix/starting sequence pairs.

Resulting Pool T_B contains 20% each 5_1, 5_2, and 5_3 tree motifs, compared with 23%, 16%, and 0% for the random pool (MM4), matching the target exactly. We found that using the 12 MMT matrices dramatically increases the population of the 5_3 motif (at a cost of decrease of 3_1, 4_1, and 4_2 motifs). The 6_1 structure (9%) is the next highest species in Pool T_B.

Target Pools T_C and T_D are mixed pools with both 4- and 5-vertex tree structures, designed from our 144 mixing matrix/starting sequence pairs. The targets for Pool T_C are 4_2, 5_2, and 5_3 tree motifs, and those for Pool T_D are 4_1, 4_2, and 5_3 tree motifs (20% for each). Pool T_C is generated by the Figure 3E sequence (MM9, 0.60) and the Figure 3B sequence (MMT2, 0.40), and Pool T_D is produced by the Figure 3B sequence (MMT6, 0.329) and the Figure 3F sequence (MM13, 0.608). The results are as expected: Pool T_C has frequencies for 4_2, 5_2, and 5_3 motifs of between 19% and 20%; Pool T_D has frequencies for 4_1, 4_2, and 5_3 trees of 17%, 20%, and 20%, respectively, all within 3% of the target.

Our designed pools above, involving three of the 22 mixing matrices and two of the 12 MMT matrices, only touched the surface of possibilities. Still, in practice, it might be preferable to approximate a target pool using a small number of mixing matrices. Once our algorithm is automated (Appendix), exploration of pool design can be routinely performed.

A designed pool improves the selection of GTP aptamers

We now apply our pool design approach for enhancing GTP-binding aptamers. Szostak's group recently found that the GTP aptamer's binding affinity is correlated with the informational complexity (Carothers et al. 2004, 2006). Informational complexity is correlated with structural complexity (e.g., number of stems, vertex number of tree graph). As the information content and binding affinity decrease (Carothers et al. 2004, see their Fig. 1, panels A and

TABLE 4. Five designed structured pools (T_A–T_E) and their characteristics

Target motifs in designed pool (% in pool)	Weights of mixing matrices (starting sequence in Fig. 3)	Frequency of tree motifs (%)										Mean mutation rate
		3_1	4_1	4_2	5_1	5_2	5_3	6_1	6_2	6_4	6_5	
T_A: 4_1, 4_2 (25%, 30%)	55.6% MM8 (Fig. 3E), 44.4% MM3 (Fig. 3E)	22	**25**	**30**	6	14	0	0	0	0	0	0.337
T_B: 5_1, 5_2, 5_3 (20%, 20%, 20%)	18% MM13 (Fig. 3A), 82% MMT12 (Fig. 3B)	0	8	4	**20**	**20**	**20**	9	7	2	5	0.147
T_C: 4_2, 5_2, 5_3 (20%, 20%, 20%)	60% MM9 (Fig. 3E), 40% MMT2 (Fig. 3B)	9	12	**20**	3	**19**	**20**	1	1	8	4	0.234
T_D: 4_1, 4_2, 5_3 (20%, 20%, 20%)	39.2% MMT6 (Fig. 3B), 60.8% MM13 (Fig. 3F)	17	**17**	**20**	8	10	**20**	0	2	0	3	0.343
T_E: GTP 4_2, 5_2 (20%, 26%)	62.5% MM13 (Fig. 3D), 37.5% MM10 (Fig. 3F)	13	21	**21**	11	**26**	0	1	0	0	0	0.349
T_F: Random	100% MM4 (Fig. 3D)	11	**29**	12	**23**	16	0	5	2	0	0	0.750

Each designed pool is specified by a set of mixing matrix/starting sequence/weight (in percent). The optimal set of mixing matrix/starting sequence/weight for a target pool is determined by our pool design algorithm in the Appendix. The mean mutation rate is calculated using starting sequences and mixing matrices and their weights. The frequencies of targeted structures in designed pools are highlighted in bold.

B), the aptamers have simple structures such as 2_1 or 3_1 tree motifs. Specifically, a high-affinity GTP aptamer with high informational complexity (Carothers et al. 2004, see their Fig.1, panel C) has the 5_2 tree structure (Fig. 3D). Interestingly, no GTP aptamer with a 4_2 tree structure (Fig. 3F) has been reported, although it is structurally similar to the 5_2 tree. Because the frequency of the 4_2 motif is only 12% in the random Pool T_F (Table 4), we propose designing a GTP aptamer pool by enriching the pool with 5_2 and 4_2 motifs. Our target pool fractions (T_i) are 20% for 4_2 and 26% for 5_2. Our optimization yields Pool T_E (Table 4) as a combination of two subpools: the Figure 3D sequence (MM13, 0.625) and the Figure 3F sequence (MM10, 0.375). The frequencies of 4_2 and 5_2 trees in the designed pool are 21% and 26%, respectively, nearly as desired and very different for the 12% and 16% distributions of these motifs in the random Pool T_F. The sequence/structure contour plots in Figure 9 show differences between the designed and random pools; the designed pool has a relatively high mean mutation rate of 0.349.

DISCUSSION

Following our previous analysis that random RNA pools are not structurally diverse (Gevertz et al. 2005), we have proposed computational tools for designing RNA pools for enhancing in vitro selection based on sequence/structure relationships. We represent pool synthesis experiments as mixing matrices applied to starting sequences; this approach can be likened to considering mutations around given sequences. Such mutations are then optimized to target specific structures and increase structural diversity. By constructing five classes of mixing matrices based mainly on conservation of base pairs, we have developed 22 representative mixing matrices covering diverse regions of the sequence space. We showed that sequence diversity represented by the mixing matrices leads to greater structural diversity, allowing the design of pools with target structural characteristics through optimization of starting sequence/mixing matrix pairs and associated weights (pool fraction for each pair). The optimized mixing matrix/starting sequence pairs and weights provide sufficient information for pool synthesis.

Thus, our work suggests that designing pools for enhancing in vitro selection can follow several research avenues. Maximizing sequence and structural diversity broadly can increase the probability of finding a given RNA property using nonrandom mixing matrix/starting sequence pairs. An advantage of this approach is that designed pools can be directly implemented in pool synthesis. Alternatively, we can target a specific structural distribution by determining optimal mixing matrices and starting sequences without explicit sequence/structure

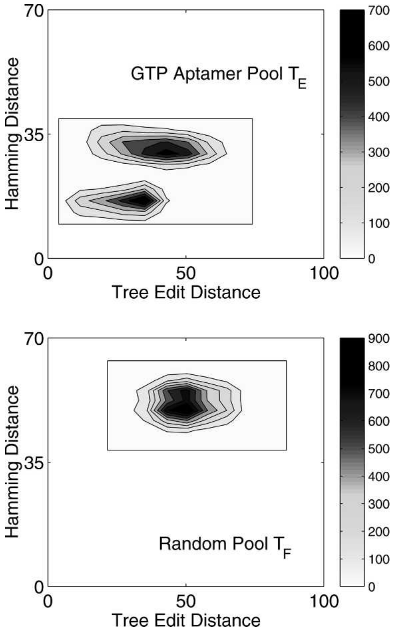


FIGURE 9. Comparison of designed GTP (*upper*) and random (*lower*) pools using contour plots of Hamming distance versus tree edit distance. The GTP pool is generated by 62.5% MM13 starting with the 5_2 motif (Fig. 3D) and 37.5% MM10 starting with the 4_2 motif (Fig. 3F). The random pool is generated using the starting sequence in Figure 3D.

mapping. Our targeted pool design can be applied to known structures, novel motifs (Kim et al. 2004), complete sets like *n*-vertex pools, or perhaps submotifs of RNAs (Zorn et al. 2004). Of course, a more comprehensive set of mixing matrices covering wider regions of the sequence space should be sought systematically. For example, matrices conserving noncanonical base pairs (AC, CA, GA, AG, etc.) can complement our current set, which conserves canonical base pairs; there are 12 such classes from a total of 16 possible base pairs.

Another design theme is enrichment of pools with structures resembling a target-active molecule. We illustrated this approach using GTP-binding aptamers. The conventional approach – designing pools in the sequence neighborhood of a target molecule (Lau et al. 2004; Ohuchi et al. 2004; Yoshioka et al. 2004), however, does not ensure that the designed pools will cover the structural neighbors of the target molecule, unless sequence mutations are made to localized sequence segments, as is commonly done in many experiments. In contrast, our optimized pool design approach (Appendix), allows enrichment of pools with specific RNA topologies or structures (e.g., tRNA-like 5_3 tree). In addition, novel tree topologies in the neighborhood of the target molecule, as suggested by structural enumeration (Kim et al. 2004), could be similarly engineered.

Clearly, further developments of sequence/structure analysis techniques are needed to improve the pool design and overcome specific limitations. Understanding the sequence/structure relationship is one of the most fundamental biological problems not only for RNA but also for proteins. In our analysis, we are limited by the usage of numerical secondary structure folding algorithms, which are still imperfect and inefficient for predicting pseudoknot structures. However, our risk has been reduced here by "folding" of small RNAs (<100 nt) only and focusing on statistical properties (e.g., frequencies of topologies). A general strategy for improving structure prediction is to consider many suboptimal structures using, for example, the Boltzmann sampling method (Ding and Lawrence 2003).

Ultimately, RNA tertiary and higher-order folding is essential to understand RNA function. Perhaps progress on this problem will be realized in the near future. For now, we offer our sequence pools possessing diverse RNA secondary structures as an approach to enhance in vitro selection technology.

Our pool design algorithm can be fully automated given target RNA shapes (and possibly starting sequences). We are developing a publicly available Web server to allow experimentation of pool design and analysis of RNA pool properties (e.g., base composition, size distribution of stems, bulges, etc.), and to define optimal mixing matrices for pool synthesis. Experimental synthesis of designed pools (specific structural motifs and their frequency) can be performed by using optimized starting sequences, mixing matrices, and associated weights. When available, location of this server will be noted on our group Web site (http://monod.biomath.nyu.edu). We hope that this tool will help stimulate the productive interaction between theoretical and experimental efforts.

ACKNOWLEDGMENTS

We are grateful to Andres Jäschke and Peter Unrau for stimulating discussions on pool synthesis and in vitro selection and thank the reviewers for constructive comments. This work was supported by the Human Frontier Science Program (HFSP) and by a Joint NSF/NIGMS Initiative in Mathematical Biology (DMS-0201160).

APPENDIX

An algorithm for designing structured RNA pools

Our pool design algorithm is based on analyses of sequence and structure spaces to allow design of specific structures, including novel RNA-like motifs identified using graph theory analysis (Kim et al. 2004). The algorithm below assumes that we have available reference data such as shown in Tables 2 and 3 that relate mixing matrices and starting sequences to motif distributions in resulting pools. The sequence space regions are mapped via various mixing matrices using a standard clustering method; the structural distribution is computed by converting secondary structures into tree graphs. By knowing the structural distributions of various sequence space regions, we then can optimize the choice of starting sequences and mixing matrices to approximate the target structured pool for future work.

Our pool design algorithm involves the following steps:

1. Specify a target distribution of topologies/shapes.
2. Define candidates for starting sequences and mixing matrices that aim to cover the sequence space. The mixing matrices have been constructed, for example, based on covariance mutations. The mixing matrices and starting sequences may remain the same for different structured pool designs. We "visualize" the diversity of a set of RNA sequences using a standard sequence similarity/dissimilarity clustering based on Hamming distance (number of dissimilar bases) between any pair of aligned sequences. In this study, we used mainly six starting sequences and constructed 22 mixing matrices to cover the sequence space (see Results).
3. Compute shape frequency distributions corresponding to all starting sequence/mixing matrix pairs, as discussed below and detailed in our previous study (Gevertz et al. 2005). This step analyzes pool structural diversity.
4. Choose the number of mixing matrices to approximate the designed pool.
5. Find an optimal combination of starting sequences (S_i) and mixing matrices ($\mathbf{M}_i$) and associated weights (α_i) to approximate the target RNA shape distribution. The mathematical procedures for this step are detailed below.

The designed pool is composed of k smaller subpools defined by the set $\{(S_i, \mathbf{M}_i, \alpha_i)\}$, i=1, 2, . . ., k. The above pool design algorithm can be fully automated given target RNA shapes (and possibly starting sequences). We are planning to make publicly available a Web server to allow experimentation of pool design and analysis of RNA pool properties, and to obtain mixing matrices for pool synthesis. Experimental synthesis of designed pools can be performed by using trial S_i, $\mathbf{M}_i$, and α_i.

In Step 3, the pool structural distribution is calculated by mapping RNA secondary structures into graph space. This is done by predicting secondary structures of all sequences using the Vienna RNAfold package and then converting them into tree graphs, as described elsewhere (Gevertz et al. 2005). It is known that 73% of known base pairs are predicted by free-energy minimization algorithms such as RNAfold for sequences with <700 nt (Mathews and Turner 2006). For greater accuracy, the Boltzmann sampling method can be used to generate a set of 1000 suboptimal structures (Ding and Lawrence 2003), although at a higher computational cost (1000 times pool size). Specifically, base-pairing information in the .ct file generated by the RNAfold program is used to convert a secondary fold into a tree graph. The topologies of the folds are determined using Laplacian eigenvalues of tree graphs as implemented in our RNA Matrix Program (Gan et al. 2004) (server available at http://monod.biomath.nyu.edu/rna). Specifying tree topologies using eigenvalues is inexact because different topologies can have the same spectrum; the assignment error rate is a few percent for small tree topologies (<10 vertices). This step is similar to the RNAshapes program, which uses bracket notations for representing secondary structures (Giegerich et al. 2004; Steffen et al. 2006). Unless stated otherwise, each sequence pool has 10,000 sequences, which is adequate for assessing structural distributions using simple tree graphs. Structure prediction and conversion to tree graphs for 10,000 80-nt sequences require ~1 h on an SGI 300 MHz MIPS R12000 IP27 processor.

In Step 5, we approximate a target structural distribution by optimizing a set of starting sequence/mixing matrix pairs based on pool structural frequency data. Generally, we consider a designed pool composed of k subpools, each generated with a mixing matrix/starting sequence pair and associated with a weight α_i: $p(S_1, \mathbf{M}_1, \alpha_1)$, $p(S_2, \mathbf{M}_2, \alpha_2)$, …, $p(S_k, \mathbf{M}_k, \alpha_k)$, where $\alpha_1+\alpha_2+\ldots+\alpha_k=1$ and $p(S_i, \mathbf{M}_i, \alpha_i)$ denotes synthesizing the α_i fraction of the pool sequences using starting sequence S_i and mixing matrix $\mathbf{M}_i$. Optimization of the three pool parameters S_i, $\mathbf{M}_i$, and α_i can be formulated as follows: If the $n\times1$ matrix $\mathbf{T}$ is the target distribution with T_i fractions of structures 1, 2, …, n and $F_l(S_i, \mathbf{M}_i)$ is the pool fraction of structure l generated by starting sequence S_i and mixing matrix $\mathbf{M}_i$ in Tables 2 and 3, the pool parameters $(S_i, \mathbf{M}_i, \alpha_i)$ can be optimized by the following equation:

$$\mathbf{T}=\begin{pmatrix}T_1\\T_2\\\vdots\\T_n\end{pmatrix}=\alpha_1\begin{pmatrix}F_1(S_1,\mathbf{M}_1)\\F_2(S_1,\mathbf{M}_1)\\\vdots\\F_n(S_1,\mathbf{M}_1)\end{pmatrix}+\cdots+\alpha_k\begin{pmatrix}F_1(S_k,\mathbf{M}_k)\\F_2(S_k,\mathbf{M}_k)\\\vdots\\F_n(S_k,\mathbf{M}_k)\end{pmatrix}, \quad (4)$$

where $\alpha=(\alpha_1, \alpha_2, \ldots, \alpha_k)$ subject to the conditions $\alpha_1+\alpha_2+\ldots+\alpha_k=1$ and $\alpha_i\geq0$. Since experimental implementation of pool synthesis is simpler with fewer mixing matrices, we consider the solution of α for k=2 below; the optimization procedure can be generalized. Formula (4) with only two mixing matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ reduces to

$$\mathbf{T}=\begin{pmatrix}T_1\\T_2\\\vdots\\T_n\end{pmatrix}=\alpha_1\begin{pmatrix}F_1(S_1,\mathbf{M}_1)\\F_2(S_1,\mathbf{M}_1)\\\vdots\\F_n(S_1,\mathbf{M}_1)\end{pmatrix}+(1-\alpha_1)\begin{pmatrix}F_1(S_2,\mathbf{M}_2)\\F_2(S_2,\mathbf{M}_2)\\\vdots\\F_n(S_2,\mathbf{M}_2)\end{pmatrix}. \quad (5)$$

The solution for the only weight is

$$\alpha_1=\frac{T_1-F_1(S_2,\mathbf{M}_2)}{F_1(S_1,\mathbf{M}_1)-F_1(S_2,\mathbf{M}_2)}. \quad (6)$$

The estimated pool fractions for the other shapes or topologies 2, 3, …, n are derived from the known α_1, $F_1(S_1, \mathbf{M}_1)$, and $F_1(S_2, \mathbf{M}_2)$ as follows:

$$\begin{aligned}\overline{T}_2&=\alpha_1F_2(S_1,\mathbf{M}_1)+(1-\alpha_1)F_2(S_2,\mathbf{M}_2),\\\overline{T}_3&=\alpha_1F_3(S_1,\mathbf{M}_1)+(1-\alpha_1)F_3(S_2,\mathbf{M}_2),\\&\vdots\\\overline{T}_n&=\alpha_1F_n(S_1,\mathbf{M}_1)+(1-\alpha_1)F_n(S_2,\mathbf{M}_2).\end{aligned} \quad (7)$$

We then optimize $(S_1, \mathbf{M}_1)$, and $(S_2, \mathbf{M}_2)$ by minimizing the error

$$\sum_{l=1}^{n}|T_l-\bar{T}_l|. \quad (8)$$

The above procedure will allow us to obtain the optimized parameters α_1, $(S_1, \mathbf{M}_1)$, and $(S_2, \mathbf{M}_2)$ for a target distribution $\mathbf{T}$. The convergence of the procedure depends on the number of mixing matrices and starting sequences, or coverage of the sequence/structure space.

Received August 17, 2006; accepted January 17, 2007.

REFERENCES

Breaker, R.R. 2004. Natural and engineered nucleic acids as tools to explore biology. *Nature* **432:** 838–845.

Carothers, J.M., Oestreich, S.C., Davis, J.H., and Szostak, J.W. 2004. Informational complexity and functional activity of RNA structures. *J. Am. Chem. Soc.* **126:** 5130–5137.

Carothers, J.M., Davis, J.H., Chou, J.J., and Szostak, J.W. 2006. Solution structure of an informationally complex high-affinity RNA aptamer to GTP. *RNA* **12:** 567–579.

Cox, T.F. and Cox, M.A.A. 1994. *Multidimensional scaling*. Chapman & Hall, Boca Raton, FL.

Davis, J.H. and Szostak, J.W. 2002. Isolation of high-affinity GTP aptamers from partially structured RNA libraries. *Proc. Natl. Acad. Sci.* **99:** 11616–11621.

Ding, Y. and Lawrence, C.E. 2003. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Res.* **31:** 7280–7301.

Eddy, S.R. and Durbin, R. 1994. RNA sequence analysis using covariance models. *Nucleic Acids Res.* **22:** 2079–2088.

Ellington, A.D. and Szostak, J.W. 1990. In vitro selection of RNA molecules that bind specific ligands. *Nature* **346:** 818–822.

Famulok, M. and Verma, S. 2002. In vivo applied functional RNAs as tools in proteomics and genomics research. *Trends Biotechnol.* **20:** 462–466.

Fera, D., Kim, N., Shiffeldrim, N., Zorn, J., Laserson, U., Gan, H.H., and Schlick, T. 2004. RAG: RNA-As-Graphs web resource. *BMC Bioinformatics* **5:** 88–96.

Fontana, W., Konings, D.A.M., Stadler, P.F., and Schuster, P. 1993. Statistics of RNA secondary structures. *Biopolymers* **33:** 1389–1404.

Gan, H.H., Pasquali, S., and Schlick, T. 2003. Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. *Nucleic Acids Res.* **31:** 2926–2943.

Gan, H.H., Fera, D., Zorn, J., Shiffeldrim, N., Tang, M., Laserson, U., Kim, N., and Schlick, T. 2004. RAG: RNA-As-Graphs database—Concepts, analysis, and features. *Bioinformatics* **20:** 1285–1291.

Gevertz, J., Gan, H.H., and Schlick, T. 2005. In vitro RNA random pools are not structurally diverse: A computational analysis. *RNA* **11:** 853–863.

Giegerich, R., Voss, B., and Rehmsmeier, M. 2004. Abstract shapes of RNA. *Nucleic Acids Res.* **32:** 4843–4851.

Hamming, R.W. 1987. *Coding and information theory*. Prentice-Hall, Englewood Cliffs, NJ.

Havgaard, J.H., Lyngso, R.B., and Gorodkin, J. 2005. The FOLDALIGN web server for pairwise structural RNA alignment and mutual motif search. *Nucleic Acids Res.* **33:** W650–W653.

Hermann, T. and Patel, D.J. 2000. Biochemistry—Adaptive recognition by nucleic acid aptamers. *Science* **287:** 820–825.

Hofacker, I.L. 2003. Vienna RNA secondary structure server. *Nucleic Acids Res.* **31:** 3429–3431.

Isaacs, F.J., Dwyer, D.J., and Collins, J.J. 2006. RNA synthetic biology. *Nat. Biotechnol.* **24:** 545–554.

Jaeger, L., Wright, M.C., and Joyce, G.F. 1999. A complex ligase ribozyme evolved in vitro from a group I ribozyme domain. *Proc. Natl. Acad. Sci.* **96:** 14712–14717.

Jäschke, A. 2001. Artificial ribozymes and deoxyribozymes. *Curr. Opin. Struct. Biol.* **11:** 321–326.

Kim, N., Shiffeldrim, N., Gan, H.H., and Schlick, T. 2004. Candidates for novel RNA topologies. *J. Mol. Biol.* **341:** 1129–1144.

Knight, R., De Sterck, H., Markel, R., Smit, S., Oshmyansky, A., and Yarus, M. 2005. Abundance of correctly folded RNA motifs in sequence space, calculated on computational grids. *Nucleic Acids Res.* **33:** 5924–5935.

Lau, M.W., Cadieux, K.E., and Unrau, P.J. 2004. Isolation of fast purine nucleotide synthase ribozymes. *J. Am. Chem. Soc.* **126:** 15686–15693.

Lee, J.F., Hesselberth, J.R., Meyers, L.A., and Ellington, A.D. 2004. Aptamer database. *Nucleic Acids Res.* **32:** D95–D100.

Legiewicz, M., Lozupone, C., Knight, R., and Yarus, M. 2006. Size, constant sequences, and optimal selection. *RNA* **11:** 1701–1709.

Mathews, D.H. and Turner, D.H. 2002. Dynalign: An algorithm for finding the secondary structure common to two RNA sequences. *J. Mol. Biol.* **317:** 191–203.

Mathews, D.H. and Turner, D.H. 2006. Prediction of RNA secondary structure by free-energy minimization. *Curr. Opin. Struct. Biol.* **16:** 270–278.

Ohuchi, S.J., Ikawa, Y., Shiraishi, H., and Inoue, T. 2002. Modular engineering of a Group I intron ribozyme. *Nucleic Acids Res.* **30:** 3473–3480.

Ohuchi, S.J., Ikawa, Y., Shiraishi, H., and Inoue, T. 2004. Artificial modules for enhancing rate constants of a Group I intron ribozyme without a P4-P6 core element. *J. Biol. Chem.* **279:** 540–546.

Ren, J.H., Rastegari, B., Condon, A., and Hoos, H.H. 2005. HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots. *RNA* **11:** 1494–1504.

Rivas, E. and Eddy, S.R. 1999. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.* **285:** 2053–2068.

Schultes, E., Hraber, P.T., and LaBean, T.H. 1997. Global similarities in nucleotide base composition among disparate functional classes of single-stranded RNA imply adaptive evolutionary convergence. *RNA* **3:** 792–806.

Soukup, G.A. and Breaker, R.R. 1999a. Engineering precision RNA molecular switches. *Proc. Natl. Acad. Sci.* **96:** 3584–3589.

Soukup, G.A. and Breaker, R.R. 1999b. Nucleic acid molecular switches. *Trends Biotechnol.* **17:** 469–476.

Soukup, G.A. and Breaker, R.R. 2000. Allosteric nucleic acid catalysts. *Curr. Opin. Struct. Biol.* **10:** 318–325.

Steffen, P., Voss, B., Rehmsmeier, M., Reeder, J., and Giegerich, R. 2006. RNAshapes: An integrated RNA analysis package based on abstract shapes. *Bioinformatics* **22:** 500–503.

Storz, G. 2002. An expanding universe of noncoding RNAs. *Science* **296:** 1260–1263.

Stuhlmann, F. and Jäschke, A. 2002. Characterization of an RNA active site: Interactions between a Diels–Alderase ribozyme and its substrates and products. *J. Am. Chem. Soc.* **124:** 3238–3244.

Tuerk, C. and Gold, L. 1990. Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase. *Science* **249:** 505–510.

Wilson, D.S. and Szostak, J.W. 1999. In vitro selection of functional nucleic acids. *Annu. Rev. Biochem.* **68:** 611–647.

Xie, D.X., Tropsha, A., and Schlick, T. 2000. An efficient projection protocol for chemical databases: Singular value decomposition combined with truncated-Newton minimization. *J. Chem. Inf. Comput. Sci.* **40:** 167–177.

Yoshioka, W., Ikawa, Y., Jaeger, L., Shiraishi, H., and Inoue, T. 2004. Generation of a catalytic module on a self-folding RNA. *RNA* **10:** 1900–1906.

Zorn, J., Gan, H.H., Shiffeldrim, N., and Schlick, T. 2004. Structural motifs in ribosomal RNAs: Implications for RNA design and genomics. *Biopolymers* **73:** 340–347.