Computational Approaches Towards the Design of Pools for the In Vitro Selection of Complex Aptamers

Xuemei Luo†, Maureen McKeague‡, Sylvain Pitre, Michel Dumontier, James Green, Ashkan Golshani, Maria C. DeRosa*, Frank Dehne*

Address: ¹School of Computer Science, ²Department of Chemistry, ³Department of Biology, ⁴Department of Systems and Computer Engineering. Carleton University, Ottawa, Canada.

* To whom all correspondence should be addressed: frank@dehne.net (www.dehne.net); maria_derosa@carleton.ca

† These two authors contributed equally.

Citation: Luo et.al, “Computational Approaches Towards the Design of Pools for the In Vitro Selection of Complex Aptamers”

Shortened title: Computational Design of Complex Aptamer Pools

Keywords: in vitro selection, random pool, aptamer pool design, RNA/DNA secondary structure, genetic algorithm, ATP aptamer

Luo et al.
Abstract

It is well known that using random RNA/DNA sequences for SELEX experiments will generally yield low complexity structures. Early experimental results suggest that having a structurally diverse library which, for instance, includes high-order junctions, may prove useful in finding new functional motifs. Here, we develop two computational methods to generate sequences that exhibit higher structural complexity and can be used to increase the overall structural diversity of initial pools for in vitro selection experiments. Random Filtering selectively increases the number of five-way junctions in RNA/DNA pools, and Genetic Filtering designs RNA/DNA pools to a specified structure distribution, whether uniform or otherwise. We show that using our computationally designed DNA pool greatly improves access to highly complex sequence structures for SELEX experiments (without losing our ability to select for common one-way and two-way junction sequences).

Introduction

Background

Aptamers are target-binding nucleic acid molecules that can bind with high affinity and selectivity to a range of molecules including antibiotics, proteins, viruses, catalytic RNA molecules and most recently whole cells (Wilson & Szostak, 1999; Hermann & Patel, 2000; Hodgson & Suga, 2004; Shangguan et al., 2006). Methods for the evolution of aptamers were first established by Gold and Szostak (Ellington & Szostak, 1990; Tuerk et al. 2007).
& Gold, 1990). To date, over 3,400 aptamers that bind to a wide variety of ligands have been selected using a biochemical process known as Systematic Evolution of Ligands by Exponential Enrichment “SELEX” (Lee et al., 2004; Thodima et al., 2006). SELEX involves the repetitive partitioning of a library of random nucleic acid sequences on the basis of selective binding to the desired target, followed by amplification by polymerase chain reaction.

Recent in vitro selection experiments have led to the identification of novel aptamers for biotechnological purposes such as functional genomics (Famulok & Verma, 2002) and biosensing (Soukup & Breaker, 1999). Other recent applications of aptamers include control of gene expression and medical diagnosis (Breaker, 2004; Isaacs et al., 2006). As the applications of aptamers grow, the need for design, identification and selection of novel and improved aptamers becomes even more apparent.

Problem

In RNA and DNA, junctions are common secondary structural elements that connect different helical segments (Lilley et al., 1995). RNA junctions are important for the structural and catalytic properties of RNAs. It has been shown that they are involved in a variety of different functional roles in nucleic acids, including the self-cleaving catalytic properties of the hammerhead ribozyme (Scott et al., 1996), promotion of functional folded states of the hairpin ribozyme (Wilson et al., 2005), recognition of the binding pocket domain by purine riboswitches (Batey et al., 2004; Serganov et al., 2004), and translation initiation of the hepatitis C virus at the internal ribosome entry site (Kieft et al., 2002).
It is well known that in vitro random RNA/DNA pools are not structurally diverse and heavily favour simple topological structures such as stem-loops structures due to incomplete and insufficient random sequence sampling (Gevertz et al., 2005). Analysis of existing aptamers has revealed that the majority of oligonucleotide sequences generated from random in vitro selection experiments have simple structures with low degrees of complexity and that highly complex structures are far less abundant (Khoo et al., 2002; Zinnen et al., 2002; Laserson et al., 2004). This lack of structural diversity in random pools may explain why complex structure motifs such as high-order junctions are rare in known synthetic aptamers. To further demonstrate this, we folded the 2,793 known aptamers from the Ellington Lab aptamer database (Lee et al., 2004) by ViennaRNA (Hofacker, 2003), and then counted the number of junctions of these known aptamers using the CountJunctions algorithm described in the Supplemental Material. The results showed that among these known aptamers, less than 1% of them contain four-way and five-way junction structures (see Figure S1 in the Supplemental Material). However, natural aptamers (e.g. riboswitches) more commonly contain high-order junctions. For example, the lysine riboswitch involves three-helical and two-helical bundles joined by a five-way junction (Serganov et al., 2008). Almost all other riboswitches can be classified into one of two types based on their high-junction number. Type I riboswitches contain a fundamental three-way junction architecture. In Type II riboswitches, such as the SAM-I binding riboswitch, the aptamer domain is arranged in a four-way junction structure (Schwalbe et al., 2007).

It has recently been shown that increasing the structural diversity of the starting oligonucleotide pool can enhance the possibility of finding novel aptamers with improved
activity (Carothers et al., 2004; Carothers et al., 2006). In a recent investigation, five-way junctions were found to provide the scaffold for the formation of functionally more active catalytic deoxyribozymes (Chiuman & Li, 2006). It was shown that the structures with 5Js were evolved from pre-existing and less complex structures with three-way junctions, and that the more complex structures had a significantly higher rate of activity.

Consequently, novel approaches have been employed that aim to increase the structural complexity of starting pools. It has been shown that structural complexity is largely dependent on sequence length (Sabeti et al., 1997). However, RNA/DNA pools for SELEX are usually restricted to sequences of at most 100 nucleotides (100nt) in length due to the difficulty involved in the synthesis above this length. This 100nt starting sequence typically includes 15-nucleotide primer sites at both the 3’ and 5’ends, which reduces the total number of variable positions from 100 to 70. Thus, effective methods should make use of primer sites as part of their strategy for predicting high complexity structures. One approach to increase the structural complexity is to keep the structure constant and introduce random segments in the proximity of the existing structure (Jaeger et al., 1999; Ohuchi et al., 2002; Lau et al., 2004; Yoshioka et al., 2004). Other approaches involve changing sequence length and composition (Knight et al., 2005; Legiewicz et al., 2006). Davis and Szostak (Davis & Szostak, 2002) isolated a high-affinity GTP aptamer from a partially structured RNA library, one half of which consisted of fully random sequences and the other half of a random region with a designed stem-loop in the middle. Subsequent analysis of the obtained aptamers revealed that most of those with highest affinity originated from the partially structured portion of their pool. They also observed that the highest affinity aptamers had fairly complex
structures (Carothers et al., 2004) and concluded from their experiments that more complex RNA structures are required for greater GTP-binding activity.

However, the above observations are based on ad-hoc pool designs for individual experiments and targets. Furthermore, these ad-hoc pools with more complex structures were generated by inserting particular stem-loops that were known \textit{a priori} to be relevant for the given target (Davis & Szostak, 2002). Such prior knowledge is not available in many cases. Hence, it is necessary to develop more general pool design approaches that can be used for targets without such prior knowledge. Recently, Schlick and coworkers (Kim et al., 2007a; Kim et al., 2007b; Kim et al., 2010) developed a computational approach for designing structured RNA pools by modeling the pool generation process using mixing matrices. However, the scope of this approach was limited and the approach is not computationally efficient (see Discussion).

**Contributions**

We have studied the design of RNA/DNA pools for \textit{in vitro} selection experiments and developed two new methods to improve their structural complexity and diversity:

1. a computational method called \textit{Random Filtering} to selectively increase the number of five-way junctions in RNA/DNA pools, and

2. a computational method called \textit{Genetic Filtering} to design RNA/DNA pools with any desired structure distribution, including a uniform structure distribution (i.e. 20 percent 1-way (1J), 2-way (2J), 3-way (3J), 4-way (4J) and 5-way (5J) junctions each).

Our first aim was to increase the structural complexity of RNA/DNA pools. Since RNA/DNA secondary structures can be described using unlabeled tree graphs (Gan et al.,

\textit{Luo et al.}
To appear in RNA (rnajournal.org).

2003), we used the degree of branching, i.e. the number of junctions, to measure their structural complexity. In this approach, a tree graph with a higher order junction has a greater structural complexity. Random Filtering led to a significant enrichment in the number of highly complex structures in DNA pools. For example, pools were generated where more than 10% of the sequences are five-way junctions and more than 15% of the sequences are four-way junctions as compared to nearly zero five-way junctions and around 0.2% four-way junctions in a typical random DNA pool. We found very similar results for 100nt DNA pools where 30nt are fixed primer-binding regions.

Our second aim was to engineer a RNA/DNA pool with a uniform distribution over a range of possible structural complexities, to maximize the diversity of sequences available in the starting pool for in vitro selection (Gevertz et al., 2005). For a given set of primer sites determined by the respective SELEX experiment, our Genetic Filtering method can provide a roadmap for creating a RNA/DNA pool with nearly uniform structure distribution (i.e. 20% of each 1J, 2J, 3J, 4J and 5J structures).

We tested the hypothesis that our designed pools can increase the chance of sampling more complex oligonucleotide structures during in vitro selection. It has already been observed by Davis and Szostak (Davis & Szostak, 2002), for the case of GTP aptamers, that pools based on customized starting sequences may increase the likelihood of finding better aptamer sequences. To further test this hypothesis, we used a DNA sequence designed with our Genetic Filtering method to generate a uniform structure distribution for the starting pool of a SELEX experiment to select for ATP-binding DNA aptamers. We found that after 8 rounds of selection cycles, complex structures such as five-way junctions accounted for 20% of the sampled sequences. These experiments
confirmed that our methods greatly improved the likelihood of generating sequences of increased structural complexity for SELEX experiments.

**Results**

**Increasing the Number of Complex Structures in SELEX Starting Pools**

*(Random Filtering Method)*

Our first goal was to increase the percentage of complex structures in starting pools for *in vitro* selection experiments. We devised a computational method termed *Random Filtering*, presented in the Materials and Methods section, which selectively increases the number of five-way junctions in DNA pools for SELEX. In very simplified terms, it starts from a random RNA or DNA pool and computes the number of junctions for each sequence in the pool. Each 5J sequence is then mutated at every non-primer single stranded site 1 million times so as to calculate the structure distribution of the respective pool design. The pool with the largest percentage of five-way junctions is selected.

Using Vienna RNA to fold one million 100nt random sequences (the choice of Vienna RNA is discussed in the Supplemental Material), we identified 76 5J sequences. These sequences were subjected to *Random Filtering* to generate a 5J enhanced pool. **Figure 1A** shows an increase in the structural complexity (using junction order) of a pool generated from *Random Filtering* (using the 76 5Js as a starting point) as compared to a pool of 1 million 100nt random sequences. The most frequent structures in the *Random Filtering* pools are 3Js (48.65%) while the most frequent structures in the random pools are 2Js (64.35%). In the *Random Filtering* pools, 10.23% of the sequences are 5Js and
15.45% of the sequences are 4Js, compared with 0.01% 5Js and 0.95% 4Js in the random pools.

Since primer sites (short-fixed sequences at the 5'- and 3'- ends required for amplification) are generally used in SELEX experiments, we considered the effect that including them would have on structural diversity. We used a 15 nt 5’ primer site “GGAAGAGATGGCGAC” and a 15 nt 3’ primer site “AGCTGATCCTGATGG”, thereby occupying 30 fixed positions in our starting sequence. This leaves only 70 nt that can be modified for the design of the starting pool, decreasing the sequence length from 100, making it considerably more difficult to find complex structures (Sabeti et al., 1997).

From a 100nt pool of 20 million random sequences containing these primer sites, we obtained only 35 sequences exhibiting 5Js. These sequences were used to generate a pool using Random Filtering. Compared with seventy six 5Js found in a 1 million 100nt random pool without primer sites, 5Js are 40 fold less abundant in a 100nt pool containing 30nt fixed primer sites. According to Sabeti et al (Sabeti et al., 1997), a 5J motif in a 70nt random pool is about 100 to 200 fold less abundant than a 5J motif in a 100nt random pool. However, primer sites can participate in the formation of junctions even though the effective pool length has been shortened. Therefore, the 5Js abundances in a 100nt pool containing 30nt primer sites should be higher than a simple 70nt random pool. More discussion of the relationship between sequence length and structure distribution can be found in the Supplemental Material.
Similar to the case without primer sites, Random Filtering produces a more structurally complex pool (Figure 1B). The most frequent structures in our Random Filtering pools are 3J sequences (45.76%) while the most frequent structures in random pools are 2J (71.72%). In Random Filtering pools, 6.65% of the sequences are 5Js and 9.58% of the sequences are 4Js, compared with 0.00% 5Js and 0.19% 4Js in random pools. Therefore, Random Filtering increases the percentage of highly complex structures available for selection in the SELEX experiments. The sequence shown in Figure 2 called RFPool A has 19 single stranded positions (indicated as “N”) excluding primer sites and resulted in the largest number of 5Js (31.13%) (Table 1).

The RFPool A was synthesized, amplified, cloned and sequenced and these sample sequences were folded using ViennaRNA. Of the 17 sequences recovered, 11.76% of them formed five-way junctions (see Table 1, sequences available in the Supplemental Material, Table S1).

**Designing a Diverse SELEX Starting Pool with Uniform Structural Distribution (Genetic Filtering Method)**

Recent findings show that increased structural diversity of the starting oligonucleotide pool can enhance the possibility of finding novel aptamers with improved activity (Carothers et al., 2004; Carothers et al., 2006). However, random pools are not structurally diverse and heavily favor simple structures such as hairpin loops (Gevertz et al., 2005). In the previous section we showed how the Random Filtering method enriches
a pool with complex structures such as five-way junctions. In this section, we introduce a new method, termed *Genetic Filtering*, to design a *diverse* starting pool for *in vitro* selection with a *uniform* structure distribution, i.e. a pool with 20 percent of each 1J, 2J, 3J, 4J and 5J structures.

For a given pair of primer sites, *Genetic Filtering* first creates an initial ‘generation’ of oligonucleotide pool designs. See Figure 2 for an example of a pool design. Each pool design is assigned a fitness score which indicates how close a pool derived from that design is to the desired uniform structure distribution. New generations of pool designs are created by selecting designs from previous generations with better (i.e. smaller) fitness scores and applying three types of operations: *mutation*, *copy* and *crossover*. *Genetic Filtering* creates several generations of pool designs (typically between 500 and 3000) until either a pool design with uniform structure distribution is obtained or the fitness score of the best pool design remains unchanged for a number of generations.

We further modified the fitness score for pool designs in *Genetic Filtering* such that designs without a desired number of single stranded (random) positions were penalized. In our case, we chose 24 as the ideal number of single stranded positions. The reason for this lies in the SELEX experiment. Most SELEX experiments begin with a starting pool consisting of nanomoles of sequences, as this is a reasonable amount of DNA that can be synthesized. This translates to approximately $10^{14}$ to $10^{15}$ sequences. Given that 4 possible bases can be inserted at any random sequence position and $10^{15} \approx 4^{25}$, this implies a target number of random positions in the pool design of approximately 25 is sufficient for the entire sequence space to be sampled once in a given
experiment. Here, we chose to include 24 random positions in our pool design to ensure complete sequence coverage in our starting pool (i.e. allowing four expected copies for every different sequence).

We ran our Genetic Filtering method on a 100nt pool design with primer sites “ATACCAGCTTATTCAATT” and “AGATAGTAAGTGCAATCT” for 571 generations with a runtime of 2hrs 10min on a 200 processor cluster. The best fitness score decreases steadily with each generation until it remains steady after approximately 500 generations (Figure 3). The distribution is close to a uniform distribution (Table 2) with a distance from the optimal of 2.95. The best pool design, GFPool1, is also shown in Figure 3. The results of other Genetic Filtering pools with different primer sites or number of variable positions are described in the Supplemental Material (Table S2 and Figures S3 and S4).

Result of SELEX Experiment with Improved Pool Design.

SELEX experiments were performed using our uniform structure pool, GFPool1, generated by our Genetic Filtering method, in an effort to confirm that our design approach yielded a pool that allowed for an increased range of complex structures to be sampled during the selection process. Selections were performed to find DNA sequences that showed affinity for ATP (Huizenga & Szostak, 1995). In the original ATP aptamer SELEX experiments, the random DNA pool used to screen for ATP binders consisted of ~2×10^{14} different sequences made up of a 72 nucleotide random region flanked by
defined primer binding sites. After 8 rounds of selection, the random region of the sequence with the best binding affinity was determined to be 5'- GACTG GGCTT GTGCT TGGGG GAGTA TTGCG GAGGA AAGCG GCCCT GCTGA AGTGG GATAC ATGTG GATAC CC -3' (personal communication, J.W. Szostak). This sequence, with its flanking primer sites, is of low complexity, predicted to form a 2J. Given the extensive study of this system, we chose ATP as the target for testing our uniform structure pool, GFPool1, generated by our Genetic Filtering method. A total of eight rounds of SELEX were performed, equivalent to the original procedure by Szostak. The recovery of DNA after each round, corresponding to the amount of DNA binding to the ATP target, was monitored using UV-vis absorbance and fluorescence measurements. With each round, an increased percentage of the pool DNA bound to the target was observed (see Figure 4). DNA from GFPool1 prior to selections as well from the round 8 enriched pool was cloned and sequences were obtained. The secondary structures were analysed using Vienna RNA. Table 2 shows the distribution of junction structures obtained for both the starting and enriched pools (sequences can be found in the Supplemental Material Tables S3 and S4). Of the 19 sequences obtained experimentally from the non-enriched starting pool, 1Js, 2Js, 3Js, 4Js and 5Js were found. Additionally, 2Js, 3Js and 5Js were found in the enriched pool, confirming not only the structural diversity of this designed pool, but also that complex structures were retained after several rounds of enrichment.

Two of the 5J sequences, ATPF4 and ATPF8 (see Supplemental Material, Table S4), were tested for binding affinity to ATP, yielding dissociation constant ($K_d$) values of 24.4 µM and 3.7µM, respectively (see Supplemental Material, Figure S5).
comparison, a $K_d$ of 6 µM is reported for the published ATP aptamer sequence (Huizenga & Szostak, 1995). It is important to note that the published ATP aptamer sequence underwent extensive optimization and minimization prior to binding affinity experiments. The fact that one 5J generated from our SELEX experiments shows improved binding over that optimized system, while the other 5J does not, suggests that complexity alone does not guarantee better target binding but higher complexity structures do have the potential to yield better aptamers. This clearly underscores the need for structural diversity, not just structural complexity, in the initial starting pool.

Insert Figure 4 and Table 2

**Discussion**

Analysis of existing aptamers have revealed that the majority of RNA sequences, which are selected from random *in vitro* selection experiments, have simple structures with low degrees of complexity and that the highly complex structures are far less abundant (Khoo et al., 2002; Zinnen et al., 2002; Laserson et al., 2004). For example, aptamers that bind to ATP, chloramphenicol, neomycin B and streptomycin all have linear stem-loop or slightly branched structures (Laserson et al., 2004). In a recent computational analysis of *in vitro* RNA random pools (Gevertz et al., 2005), it was shown that random pools are not structurally diverse. They heavily favor simple topological structures due to incomplete and insufficient random sequence sampling. The structure distribution for random pools shown in **Figure 1**, which is consistent with the analytical results in (Sabeti et al., 1997), illustrates that 99.80% of structures in a random pool are expected to be 1J, 2J, or 3J
structures. This lack of structural diversity in random pools suggests that complex structure motifs with high-order junctions would be rare in selected aptamers.

Recent experimental findings show that enhancing the structural diversity of RNA/DNA pools increases the chance of finding novel aptamers with higher binding affinity (Carothers et al., 2004; Carothers et al., 2006). To increase the number of functional RNA/DNA sequences available in the starting pool for in vitro selection, the starting pool should be engineered to have a uniform distribution over all possible structures (Gevertz et al., 2005).

In this paper, we present two new systematic computational pool design approaches that increase the covered search space for in vitro selection of complex aptamers: Random Filtering and Genetic Filtering. Compared with close to zero five-way junctions and around 0.2% four-way junctions in a typical random DNA pool with $10^6$ structures of 100 nucleotides each, Random Filtering generated pools of the same size where more than 10% of the sequences are five-way junctions and more than 15% of the sequences are four-way junctions. Very similar results were found for 100nt DNA pools where 30nt are fixed primer sites. Our RFPool A (designed via Random Filtering) was synthesized and a few sequences were determined and folded using ViennaRNA. In a subset of only 17 sequences, 11.76% were found to be five-way and 11.76% were four-way junctions. Compared to a completely random pool, our Random Filtering method can dramatically increase the likelihood of obtaining high-order junctions experimentally.

Our second method, Genetic Filtering can generate DNA pool designs of 100 nucleotides each, where 30nt are fixed primer sites, with distributions in silico that are very close to a uniform structure distribution, i.e. highly diverse. Analysis of a small set
of the sequences obtained from the *GFPool1* (designed via *Genetic Filtering*) *in vitro* showed significant structural diversity, with each of the 1J, 2J, 3J, 4J, and 5J structures present at levels of at least 10%. Complex 5J structures accounted for 26% of the sequences prior to enrichment. After *GFPool1* was subjected to 8 rounds of selection against the target ATP, complex 5Js accounted for 20% of the sampled sequences. The likelihood of observing this concentration of 5J structures in a random pool, where we find only 35 of 20 million sequences resulting in 5J structures, is approximately $10^{-10}$. These results confirm that our suggested approach not only leads to better access to more complex structures for *in vitro* selection experiments but also that SELEX is capable of preserving highly complex structures if they are present in the initial DNA pool. When the binding affinities of two 5J sequences from this enriched pool were tested, one sequence (ATPF8) displayed stronger affinity for ATP than Huizenga and Szostak’s low complexity aptamer, while the other (ATPF4) showed weaker affinity. This suggests that higher complexity sequences have the potential to be better aptamers, but will not always lead to improved binding. Thus, structural *diversity*, not just structural complexity, is an important characteristic of an ideal starting pool for SELEX. Our *Genetic Filtering* method is particularly aimed at designing such pools.

It may initially appear that the structural diversity of a pool resulting from our filtering approaches will be overly constrained given the stretches of fixed positions in the stems. However, an examination of a number of structures resulting from the same pool design demonstrates that while the sequence in these regions is constrained, the corresponding structure is not. Examples of different junction structures formed from the pool in Figure 2 are shown in the Supplemental Material, Figure S6. These sample
structures illustrate that the same fixed sequence region may participate in a stem in one structure while forming a loop in another. For example, in the 5J structure shown in Figure S6A, the sequence GCGT starting at base 61 participates in a stem while the same fixed sequence forms a loop in the corresponding 4J structure illustrated in Figure S6B.

While Inverse Folding associated with the ViennaRNA package was developed to find sequences that fold into a predefined structure (Hofacker, 2003), it cannot be applied for the design of starting pools for in vitro selection. This is primarily due to the fact that Inverse Folding cannot generate sequences containing primer sites directly, which are required for these in vitro experiments. As shown in Figure S7, adding primers to complex structures post-design can have a dramatic effect. From 20 million 5Js of length 70nt generated by Inverse Folding, less than 1% of engineered sequences retained their structure upon the addition of two primer sites of length 15nt. In addition, we compared the Inverse Folding method and the Random Filtering method, and found that Random Filtering generates more complex structures than Inverse Folding, making our method more suitable to generate structurally diverse pool for in vitro experiments (a description of the methods and results are found in the Supplemental Material, see Figure S8).

Schlick and coworkers (Kim et al., 2007a; Kim et al., 2007b; Kim et al., 2010) developed a computational approach for designing structured RNA pools by modeling the pool generation process using mixing matrices. We compared our method with the mixing matrices approach. Applying the mixing matrix approach to create a pool with a uniform distribution of 12 tree structures (see Supplemental Material) resulted in error rates of 60% (using 1 mixing matrix) and 38.9% (using 2 mixing matrices). In contrast,
Genetic Filtering reduced the error to 35.23% (with one single pool which is equivalent to 1 mixing matrix). The structural distribution and pool design obtained by this experiment is shown in Figure S9 in the Supplemental Material. For in vitro experiments, our method requires only one synthesis, whereas the mixing matrices approach needs one synthesis for each mixing matrix. Therefore, our method is also less costly and less time consuming compared with the mixing matrices approach.

Our results complement the mixing matrices approach in several ways. Random Filtering and Genetic Filtering are able to add more complex structures to the pool. For example, the most complex structures created via mixing matrices are 4J structures whereas Random Filtering and Genetic Filtering can increase and control, respectively, the number of 5J structures in the pool. The mixing matrices approach also does not cover the inclusion of primer sites. However, primer-binding regions are required in essentially all in vitro selection experiments. Since the primer regions are considered in our approach, our designed pools can be directly implemented into in vitro selection experiments. While we showed how Genetic Filtering generates pools with uniform structure distributions, the method can be used to design any given target structure distribution, not only for junction structures, but also for tree structures. The approach can also be applied to design pools including specific sub-structures such as common structural components of aptamers.

In conclusion, we have developed two methods for designing improved starting pools for functional nucleic acid selection experiments: Random Filtering and Genetic Filtering. Random Filtering yields a DNA pool (with and without primer-binding regions) with significant enrichment in the number of highly complex structures present.
Genetic Filtering allows us to engineer diverse DNA pools with a nearly uniform
distribution over all possible structures from 1Js to 5Js. A uniform pool design created
with Genetic Filtering was synthesized and subjected to a SELEX experiment for ATP
binders. We found that after 8 rounds of selection, complex 5J structures still accounted
for a sizeable percentage of the pool, confirming that our methods greatly improved
generation of high complexity structures and that these structures were preserved during
the selection process. The disparate binding affinity of two 5J sequences from this
enriched pool confirmed that higher complexity sequences have the potential to lead to
better aptamers, but certainly do not guarantee improved binding. This suggests that
SELEX pool designs should aim to increase structural complexity while still preserving a
diversity of both low and high complexity structures. Future work will examine in more
depth whether these designed pools lead more generally to aptamers with improved
binding characteristics.

Materials and Methods

Computational Method: Random Filtering

Random Filtering is a computational method to selectively increase the number of
five-way junctions in DNA pools for SELEX. Random Filtering proceeds as follows: we
start from a random RNA or DNA pool and use ViennaRNA (see discussion below) to
generate their secondary structures. We then compute the number of junctions for each
sequence with the CountJunctions algorithm outlined below. Each 5J sequence is then
mutated at every single stranded position by randomly choosing one of four bases
(ACGT) to substitute for the original. This process is repeated 1 million times for each 5J
sequence so as to calculate the structure distribution of each respective pool design. Sequences are only mutated at non-primer positions. The pool with the largest percentage of five-way junctions is selected. It was initially unclear whether using a million rounds of mutations was sufficient to determine the distribution with a reasonable degree of accuracy. Table S5 shows a typical example for a pool’s structure distribution determined with different numbers of mutations. Each experiment was repeated 100 times, and the numbers shown represent the average and variance for each value. The averages are very stable for different numbers of mutations. However, the variance is clearly reduced with additional experiments, reaching very close to zero when the number of mutations reaches 1 million. Hence, 1 million mutations are expected to be sufficient to reliably estimate the structure distribution.

Secondary structure prediction was performed using the ViennaRNA 1.6.5 (Hofacker, 2003) software package in which RNAfold generates a single RNA/DNA secondary structure prediction through energy minimization based on a dynamic programming algorithm originally developed by Zuker and Stiegler (Zuker & Stiegler, 1981). Our choice of ViennaRNA for secondary structure prediction is discussed in the Supplemental Material. ViennaRNA (Hofacker, 2003) provides as output a nucleotide pairing list indicating for each nucleotide whether or not it is paired with another nucleotide, and if so to which. From this information we then determine the number of junctions in the RNA/DNA secondary structure using the CountJunctions algorithm presented in detail in the Supplemental Material.
Computational Method: Genetic Filtering

Genetic Filtering is a computational method to design diverse DNA pools for SELEX with uniform structure distribution, i.e. 20% 1J, 2J, 3J, 4J and 5J structures each. A graphical overview of our method is given in Figure 5 and a more detailed Genetic Filtering pseudo-code is provided in the Supplemental Material. For a given pair of primer sites, Genetic Filtering first generates an initial generation of pool designs. Each pool design is assigned a fitness score which indicates its closeness to the desired uniform structure distribution. New generations of pool designs are then obtained by selecting designs from previous generations with better (i.e. smaller) fitness scores and then applying three types of operations: mutation, copy and crossover. The initial generation of pool designs is based on a set of five-way junctions which contain the given primer sites. The reason for selecting a set of five-way junctions as the initial pool designs is that it is possible to obtain low complexity structures through the mutation of high complexity structures (see Figure 1) but the opposite is highly unlikely. This was also reported by a recent investigation (Chiuman & Li, 2006) where 5Js were derived from the preexisting and less complex structures with 3Js through evolutionary pathways, but it was observed that the possibility is very small. A similar experiment is shown in Table S6. Mutating 50 different 3J sequences 1 million times each, for a total of 50 million sequences, generated only 162 5J structures.

Given an initial seed population of pool designs based on 5J structures, our genetic algorithm proceeds as shown in Figure 5. Note that, all sequences (i.e. pool designs) always contain the given primer sites. Each sequence (pool design) is assigned a fitness score which indicates its closeness to the desired structure distribution. For a
sequence $S$, let $jct_i$ be the percentage of $i$-way junctions counted after mutating its single stranded positions $10^4$ times, and let $djct_i$ be the desired percentage of $i$-way junctions ($i = 1, \ldots, 5$). The fitness score for $S$ is calculated as $\sum_{i=1}^{5} |jct_i - djct_i|$. The desired percentage of $i$-way junctions is the uniform structure distribution and each $djct_i$ is set to 20%. Note that, we perform $10^4$ instead of $10^6$ mutations of single stranded positions for the fitness score calculations. The number of mutations is reduced for fitness score estimation to save computation time. As discussed in Results, our Genetic Filtering method takes hours on a 200 processor PC cluster. As shown in Table S5, the structure distribution estimated with $10^4$ mutations is close to that estimated with $10^6$ mutations. Furthermore, we confirm our final result by re-calculating the fitness score with $10^6$ mutations.

Another consideration for the calculation of the fitness score is that the maximum size of a RNA/DNA pool that can be synthesized and manipulated is typically $10^{14}$-$10^{15}$, which is approximately $4^{24}$-$4^{25}$. We chose our pool design to have 24 single stranded positions excluding primer sites in order to ensure complete sequence space coverage. Hence, when each sequence (pool design) is assigned a fitness score, we penalize sequences whose number of single stranded positions ($num_{ss}$) is not equal to the desired number of single stranded positions ($dnum_{ss}$), and less penalty is given to pool designs with fewer single-stranded positions. For details see Algorithm Fitness_function_with_penalty in the Supplemental Material.

After the fitness scores have been calculated for the current population of pool designs, biased selection (based on fitness scores) is applied to select a new intermediate population of pool designs with better fitness. Here, we use a roulette wheel selection
method to select pool designs from the current population where each pool design is chosen with a probability that corresponds to its relative fitness. By repeatedly spinning the roulette wheel, a new intermediate population is selected and fitter individuals have a greater chance to be selected than weaker ones. Crossover, mutation and copy are then applied to the intermediate population to create the next generation. For the crossover method, we first choose two sequences $S_1$ and $S_2$ from the intermediate population of pool designs, and then generate a random number called $randomPosition$ between two parameters $Low$ and $High$. To create the next generation, we switch the sub-sequences of $S_1$ and $S_2$ from position $randomPosition$ to position $length - randomPosition$, where $length$ is the total length of each sequence. The crossover method is designed to potentially switch sub-structures between two sequences. This is clearly not always the case, but when it happens, new pool designs of interest can be generated. When a sequence in the intermediate population is selected for mutation, we generate a random number between 0 and 1 for each position of the sequence excluding primer sites. If the number is less than a parameter $P_{position}$, we mutate the position by choosing with equal likelihood one of the other three nucleotides. The mutated sequence will then enter the next generation. When a sequence in the intermediate population is selected for a copy operation, the sequence is simply copied unchanged to the next generation. The above crossover, mutation and copy methods have parameters $P_{crossover}$, $P_{mutation}$ and $P_{copy}$, respectively, which represent the likelihood of applying the respective method to a given sequence. Furthermore, for every 10 generations we re-add the best sequence found so far to the new generation of pool designs.
Genetic Filtering generates a number of generations of pool designs (typically between 500 and 3000) until either a pool design with uniform structure distribution is found, or the score of the best pool design remains unchanged for a number of generations, or a maximum number $NG$ of generations is reached.

DNA Experiments

Reagents for DNA synthesis were purchased from Glen Research. PCR and acrylamide-gel components were purchased from BioShop. All other chemicals were purchased from Sigma-Aldrich.

Oligonucleotide Library and Primer Synthesis

The oligonucleotide pools were synthesized on a 1µmol scale using a MerMade 6 Oligonucleotide synthesizer (BioAutomation Corporation, USA). The sequence, GFPool1 was obtained from the Genetic Filtering procedure. This library consisted of approximately $10^{14}$-$10^{15}$ single stranded DNA fragments comprising two primer binding sites necessary for PCR (shown in bold): 5’-

ATACCAGCTTATTCAATTGCNNNGCAATNNNGTCNGGACNNNGTTCNG ACNNTCGGCGNNCGCCGANCNTATCTNNNNAGATAGTAAGTGCAATCT. A small amount of this pool was set aside prior to SELEX for cloning and sequencing experiments.
The following primers used for amplification and cloning of the selected oligonucleotides were synthesized: Primer1: 5’- ATACCAGCTTATTCAATT-3’ and Primer2: 5’- AGATTGCACCTACTATCT-3’. In order to isolate the single-stranded aptamers from the double stranded PCR product, the primers were synthesized with the following modifications: ModPrimer 1: 5’-fluorescein-ATACCAGCTTATTCAATT and ModPrimer 2 5’-poly-dA$_{20}$-HEG- AGATTGCACCTACTATCT-3’. Similarly, the RFPool A and the primers required for PCR amplification were synthesized with the following sequences:

**RFPool A 5’-**

GGAAGAGATGGCGACGCTACTCCCATCNGGTACCGTNNNNACGGGTATTNNNGA  
CGCNNNNNGCGTCNNNACACNNNNGTGTGGATCGCTGATCCTGATGG  
**RF Primer 1 5’-**GGAAGAGATGGCGACGCT and **RFPrimer 2 5’-**  
CCATCAGGATCAGCTGAT

All synthesized DNA was purified by 12% polyacrylamide gel electrophoresis (PAGE) followed by clean-up using Microcon YM-3 Centrifugal Filter Devices.

**SELEX Experiments**

ATP-agarose and unmodified-agarose columns, 0.25mL in volume, were washed with approximately 10mL of column buffer (300mM NaCl, 5mM MgCl$_2$, 20mM Tris HCl, pH 7.6). The oligonucleotide library, suspended in 1 mL of column buffer was heated for 5 minutes at 75°C then cooled to room temperature over 20 minutes. The pool was then immediately loaded into the unmodified-agarose column and left to incubate with mild shaking at room temperature for 30 minutes. The column was washed with 4mL of column buffer, collecting the DNA that did not bind to the unmodified agarose material.
to be used for the selection process. After this pre-selection, 8 rounds of positive selection were performed as follows. The DNA library was subjected to 5 minutes of heating and a 20 minute incubation at room temperature followed by immediate incubation with the ATP-modified agarose column, as described above. After 30 minutes, sequences that did not have an affinity for ATP were washed away with 5mL of column buffer. To obtain the sequences bound to the ATP, the column was incubated for 10 minutes with 0.5mL elution buffer (40mM Tris HCl, 3.5M urea, pH 8) at 80°C and removed using centrifugation. Five elution fractions were collected and the DNA was purified using an ultrafiltration stirred cell (Millipore). This DNA was quantified using UV-visible spectroscopy and fluorescence.

The entire selected oligonucleotide pool was amplified in 5-15 parallel PCR reactions. Each reaction consisted of 0.1M Tris HCl pH 9, 50mM KCl, 1% Triton X-100, 1.9mM MgCl$_2$, 0.3mM dNTP mix, 1µM each primer and 5 units of Taq DNA polymerase. The DNA was initially melted for 10 minutes at 94.0°C, followed by 25 cycles of 94.0°C (1 min), 47.0°C (1 min) and 72.0°C (1 min). Final extension occurred at 72.0°C for 10 minutes after the last cycle. PCR products were dried down, heated at 55°C for 5 minutes in the presence of formamide and run on a 12% denaturing PAGE to separate the double stranded product. The fluorescein labelled DNA strand (the selected sequences) could be identified using an Alpha Imager UV-illuminator. The corresponding DNA bands were cut from the gel and extracted using the freeze/rapidthaw method described by (Chen & Ruffner, 1996) in 10mM Tris HCl buffer, pH 7.4. After purifying the DNA on the stirred cell and re-suspending it in column buffer, the DNA could be used for the next selection round.
Cloning and Sequencing

The selected oligonucleotides from SELEX round 8 and a small amount of the starting library were amplified using the unmodified primers (Primer 1 and 2) and cloned using a StrataClone PCR Cloning Kit (Agilent Technologies). The colonies were grown on LB-ampicillin agar plates for 16 hours, at 37°C and colonies of interest were selected via blue-white screening. Each colony was removed carefully from the agar and vortexed in 50uL of deionized water. These samples were sent for direct sequencing at the University of Calgary University Core DNA Services using the T3 promoter primer and T7 promoter primer. Sequencing results were analyzed and vector sequences were deleted from the total sequence so as to retain only the data that represent the selected random sequences.

Dissociation constant \((K_d)\) experiments

1mg of ATP agarose (approximately 23 nanomoles of ATP) in a microcentrifuge filter tube was washed with column buffer and then exposed to varying concentrations of 5’-fluorescein labelled sequences (from 1 nM to 20 µM) in 100 µL of column buffer. After vortexing, the tubes were centrifuged at 10000g for 10 minutes. 100 µL of 90°C column buffer was added and vortexed briefly, followed again by centrifugation at 10000g for 10 minutes. The fluorescence of the eluted DNA was recorded and the dissociation constants were evaluated by minimizing the residuals values between calculated and observed experimental \(\Delta\) fluorescence data using the solver feature of Microsoft Excel (Fylstra et al., 1998; Nenov & Fylstra, 2003).
Acknowledgements

We would like to thank Yingfu Li (Biochemistry, McMaster Univ.) for initially pointing out the problem. Thanks to Yasmine Miguel for assisting with binding studies. This work was made possible through support from the Natural Sciences and Engineering Council of Canada (NSERC), the Canadian Foundation for Innovation (CFI), the Ontario Research Fund (ORF), and Carleton University.
References


Tables

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In silico</strong></td>
<td>3.28%</td>
<td>25.68%</td>
<td>24.68%</td>
<td>15.22%</td>
<td>31.13%</td>
<td>19</td>
</tr>
<tr>
<td><strong>In vitro</strong></td>
<td>0%</td>
<td>47.06%</td>
<td>29.41%</td>
<td>11.76%</td>
<td>11.76%</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: *In silico* and *in vitro* structural distribution for the RFPool A DNA pool with primer sites “GGAAGAGATGGCGAC” and “AGCTGATCTGATGG” shown in Figure 2. The *in vitro* values were generated by cloning and sequencing of the pool after chemical synthesis (17 sequences were obtained). The sequences can be found in the Supplemental Material.

<table>
<thead>
<tr>
<th></th>
<th>1Js</th>
<th>2Js</th>
<th>3Js</th>
<th>4Js</th>
<th>5Js</th>
<th>ss positions</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In silico</strong></td>
<td>19.6%</td>
<td>20.4%</td>
<td>18.9%</td>
<td>20.4%</td>
<td>20.7%</td>
<td>24</td>
<td>2.95</td>
</tr>
<tr>
<td><strong>In vitro (before enrichment)</strong></td>
<td>15.8%</td>
<td>26.3%</td>
<td>10.5%</td>
<td>21.1%</td>
<td>26.3%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>In vitro (after 8 rounds of SELEX)</strong></td>
<td>0%</td>
<td>50%</td>
<td>30%</td>
<td>0%</td>
<td>20%</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: *In silico* structural distribution, distance from the optimal, and number of single stranded positions present for DNA pools with primer sites as well as experimental structural distributions prior to starting SELEX and after 8 rounds of selection using ATP as the target. *GFPool1*: Pool shown in Figure 4 using primer sites ATACCAGCTTATTCAATT" and “AGATAGTAAGTGCAATCT”.

Luo et al.
Figures

**Figure 1**

A

<table>
<thead>
<tr>
<th>Junction structure</th>
<th>Percentage of sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1J</td>
<td>9.63%</td>
</tr>
<tr>
<td>2J</td>
<td>64.35%</td>
</tr>
<tr>
<td>3J</td>
<td>22.25%</td>
</tr>
<tr>
<td>4J</td>
<td>0.95%</td>
</tr>
<tr>
<td>5J</td>
<td>10.23%</td>
</tr>
</tbody>
</table>

B

<table>
<thead>
<tr>
<th>Junction structure</th>
<th>Percentage of sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1J</td>
<td>14.07%</td>
</tr>
<tr>
<td>2J</td>
<td>71.72%</td>
</tr>
<tr>
<td>3J</td>
<td>33.33%</td>
</tr>
<tr>
<td>4J</td>
<td>14.01%</td>
</tr>
<tr>
<td>5J</td>
<td>6.65%</td>
</tr>
</tbody>
</table>

Legend:
- Random pools
- Pools generated by Random Filtering
Figure 2
Figure 3
Figure 4
For every ten generations, the best sequence so far is put into the new population.

Figure 5
Figure Legends

Figure 1: Comparison of average structural distribution (percent of sequences displaying a given junction order: either one-way, two-way, three-way, four-way or five-way junction structures) of 1 million sequences for random RNA/DNA pools (dark grey) and RNA/DNA pools generated by our Random Filtering process (light grey). A: without primer sites, Random Filtering has greatly increased the percentage of high complex structures such as 5Js (by 10%) and 4Js (by 15%). B: with 15nt primer sites at both ends, “GGAAGAGATGGCGAC” and 15 “AGCTGATCCTGATGG”, (light grey). Random Filtering has greatly increased the percentage of high complex structures such as 5Js (by 6%) and 4Js (by 10%).

Figure 2: Sample secondary structure of the RFPool A DNA sequence including primer sites “GGAAGAGATGGCGAC” and “AGCTGATCCTGATGG” designed to generate the highest percentage of 5J sequences (31.13%). Image generated by Mfold. The sequence has 19 single stranded positions excluding primer sites (noted as ‘N’). As a starting pool for SELEX experiments, these 19 single stranded positions are considered as random (N) and the other positions are considered as fixed.

Figure 3: The best fitness value observed so far at each generation of Genetic Filtering (primer sites “ATACCAGCTTATTCAATT” and “AGATAGTAAGTGCAATCT”) Parameters: NG=10000  NoChargeCutoff=500  Pcrossover=0.0  Pmutation=0.9  Pposition=0.0156  Pcopy=0.1 (see Materials and Methods). The best fitness score (ie. smallest value),
which indicates its closeness to the desired uniform structure distribution, decreases steadily with each generation until it remains steady after approximately 500 generations.

Inset: Sample secondary structure of the DNA pool sequence (GFPool1). This 100nt in length pool, containing 24 random nucleotide positions, was synthesized and used in selection experiments against ATP. Image generated by Mfold.

**Figure 4:** Percent recovery of DNA eluted from the ATP selection column versus selection round. With each round, an increased percentage of the pool DNA bound to the target was observed (recovery determined using UV-Vis and fluorescence measurements).

**Figure 5:** Graphical overview of the Genetic Filtering algorithm as described in the text.
Supplemental Material

Contents:

• Choice of secondary structure folding algorithm (ViennaRNA)
• The CountJunction algorithm for determining the number of junctions in a RNA/DNA secondary structure
• Relationship between structure distribution and sequence length
• Comparison of Inverse Folding with Random Filtering
• Genetic filtering pools using different primer sites and different number of single stranded positions
• Comparison of Genetic Filtering with the Mixing Matrices approach
• Supplemental figures and tables (ordered by occurrence in the main text)
  o Figures S1 - S12
  o Tables S1 - S6
• Genetic Filtering algorithm (pseudo-code)
• Supplemental References

Choice of secondary structure folding algorithm (ViennaRNA)

Currently, both our approaches for pool design are based on RNA/DNA secondary structure folding, using RNAfold from the ViennaRNA package. Consequently, our approaches are subjected to the limits of these folding algorithms. As an example, RNA sequences that adopt pseudoknot structures are not amenable by our approaches since RNAfold in ViennaRNA package cannot fold a sequence into a pseudoknot. Mfold is another RNA/DNA folding package using thermodynamic methods (Mathews et al., 1999; Zuker, 2003). Hofacker and coworkers compared the performance of RNAfold and Mfold, and found that RNAfold is considerably faster than Mfold (Hofacker et al., 1994). Recently, other folding programs such as GTfold (Mathuriya et al., 2009) and MC-Fold (Parisien & Major, 2008) have been developed. Figure S10 shows the comparison of structure distributions of the same 1000 random sequences predicted by RNAfold, GTfold and MC-Fold. The structure distributions generated by
RNAfold and GTfold are almost identical. However, MC-Fold predicted considerably more complex structures. Upon closer inspection of MC-Fold, we observe that this method starts by enumerating all possible junction configurations and therefore has a tendency to overestimate the number of complex structures. We also observed a considerable difference in runtime between these three methods. On a single machine, the run time required for a single sequence by RNAfold and GTfold are very similar (Mathuriya et al., 2009). However, MC-Fold runs several orders of magnitude slower. On a PC using an Intel Core 2 Quad Q6600 processor with 2.4GHz and 8GB DDR2 RAM, the run time of RNAfold for folding one sequence was about 0.008029 seconds, while it took about 845 seconds for MC-Fold, which is approximately 100,000 times slower than RNAfold. Thus, it would take 27 years for MC-Fold to predict the secondary structure for one million sequences. It is therefore not feasible to use MC-Fold to predict secondary structures in our methods despite its capacity in predicting pseudoknot structures.

Although there are more recent methods available for secondary structure prediction, ViennaRNA is faster and sufficiently precise for our purpose of computing structure distributions for large numbers of sequences.

The CountJunction algorithm for determining the number of junctions in a RNA/DNA secondary structure

ViennaRNA (Hofacker, 2003) provides as output a nucleotide pairing list indicating for each nucleotide whether or not it is paired with another nucleotide, and if so to which. From this information we then determine the number of junctions in the RNA/DNA secondary structure using the CountJunctions algorithm described in this section.

Luo et al.
The CountJunctions algorithm traverses the sequence from 5' to 3'. For each stem (at least four contiguous base pairs), there are two strands. Starting from one strand, if there is a stem along the path to another strand, we increase the junction number by 1.

For example, in Figure S12, stem $S_1$ has two strands, $S_1'$ and $S_1''$. Along the path from $S_1'$ to $S_1''$ in the 5' to 3' direction, each stem $S_2$, $S_3$, $S_4$ and $S_5$ forms one junction, so the number of junctions for stem $S_1$ is 5. For each stem, the number of junctions is at least 1.

The number of junctions for the RNA secondary structure is the maximum number of junctions of all stems. For example, if a secondary structure contains a one-way and a three-way junction (see Figure S6c), the secondary structure is defined as a three-way junction structure.

Algorithm CountJunctions:
1. Scan the nucleotide pairing list created by ViennaRNA and determine the list $[S_1, \ldots, S_n]$ of all stems. We define a stem as a maximal list of at least four contiguous pairs.
2. Consider the list $L$ of nucleotides from the 5' end to the 3' end. Each stem $S$ consists of two contiguous subsequences $S'$ and $S''$ of $L$. Sort the list $L' = [S_1', S_1'', S_2', S_2'', \ldots, S_n', S_n'']$ according to their location in $L$.
3. For each stem, calculate its number of junctions. A $k$-way junction consists of a sequence $S_1', S_2', S_2'', S_3', S_3'', \ldots, S_k', S_k'', S_1''$.
4. Find the maximum number of junctions.

In Step 1 of Algorithm CountJunctions we define a stem as a maximal list of at least four contiguous pairs. We chose the most strict setting of minimum stem length = 4 in accordance with (Bindewald et al., 2008). The relationship between junction distribution count generated by Algorithm CountJunctions and minimum stem length is shown in Figure S11. We observe that selecting a smaller minimum stem length results in a larger number of predicted high complexity structures. However, the difference is not substantial.
Relationship between structure distribution and sequence length

For our Random Filtering method to work, several five-way junctions need to be initially found by repeatedly examining random sequences; these 5Js serve as the starting point for the Random Filtering method. The ability to generate 5Js depends primarily on the sequence length (Sabeti et al., 1997), whereas the number of 5Js obtained also depends on the size of the pool. The relationship between sequence length and the number of 5Js that can be found for one million random sequences with and without primer regions can be seen in Figure S2. The numbers shown are consistent with, but not equal to, the estimates given in Sabeti et al. (Sabeti et al., 1997); see discussion below. As the pool sequence length decreases, the number of 5Js in the random pool decreases as well. For the random pool with primer regions, only 1 and 2 5Js were found in the pool with sequence length 100 and 98, respectively, and no 5Js were found in the pool with sequence length 96. This suggests that 98nt is the minimum sequence length required in order to generate 5Js in a 1-million random pool with two 15nt primer regions.

In our CountJunctions algorithm, the number of junctions for an RNA secondary structure is the maximum number of junctions of all stems. For example, if a secondary structure contains a 1J and a 3J junction (see Figure S6c), the secondary structure is defined as a 3J junction structure. In contrast, Sabeti et al. (Sabeti et al., 1997) calculated the probability of finding a particular motif within a random sequence. That is, their method would count the sequence in Figure S6c as having a 1J and 3J motif. Therefore, the percentages of n-way junction in a random pool calculated by our CountJunction method are different from those of Sabeti et al. Since 5J is the most complex structure
found for 100nt random pools in our computation, a sequence containing a 5J structure is less likely to contain other motifs. Therefore, the 5J abundance calculated by our CountJunction method should be close to those found by Sabeti et al. For example, 78 5Js are found in a 100nt random pool without the primer regions (Figure S2). Based on this starting point, 63, 50 and 39 5Js can be found for 98, 96 and 94nt pools, respectively, according to Equation 2 in Sabeti et al. (Sabeti et al., 1997), given n=~56 (Figure S2). The “without primer” curve in Figure S2 is in direct agreement with this analytical calculation.

Comparison of Inverse Folding with Random Filtering

We randomly chose 46 5J structures generated by Inverse Folding in Figure S7 and a random pool, respectively, and mutated their single stranded positions 1 million times. Random Filtering generates more complex structures than Inverse Folding (Figure S8) and would be more suitable to generate structurally diverse starting pools for in vitro selection experiments.

Genetic Filtering pools using different primer sites and different number of single stranded positions

The effect of primer site and number of single stranded positions was investigated for the Genetic Filtering approach. Using a 5' primer site “ATACCAGCTTATTCAATT" and 3' primer site “AGATTGCACCTTACTATCT", Genetic Filtering stopped at generation 1626, and the runtime on a 200 processor PC cluster was 6hrs 17min. Figure S3 shows

Luo et al.
the best (smallest) fitness score of the pool designs observed so far at each generation. The best pool design, GFPool2 (Figure S4) delivered a structure distribution (Table S2) close to the uniform structure distribution, having a deviation of 4.93. The number of single stranded positions in the best pool design was 25, despite having a penalty in the fitness function for deviations from the ideal 24 positions. In order to obtain a pool design with uniform structural distribution and exactly 24 random positions, we performed the following additional computation. For each single stranded position, excluding primer sites, we kept that single stranded position fixed and mutated the other 24 single stranded positions $10^6$ times, calculating the structural distribution for those $10^6$ mutations. The best distribution was obtained when the 20th single stranded position was kept fixed, in which case the distribution is nearly equal to the optimal uniform structural distribution (see Table S2). The pool design, GFPool2A, obtained by Genetic Filtering with very close to uniform structural distribution and exactly 24 random positions is “ATACCAGCTTATTTCAATTAAGCNNNNNGTCCACNNNNNNGTGGATNGCCC CNNNNGGNNGGGCNNTGGCTNNNGCCAGGTTAGATTGCACCTTACTATCT”.

**Comparison of Genetic Filtering with the Mixing Matrices approach**

We developed a program to convert RNA/DNA secondary structures into the tree structures as defined in (Gan et al., 2003; Gevertz et al., 2005). To design a pool with a uniform distribution of 12 tree structures from $2_1$ to $6_5$ in (Gevertz et al., 2005), similar to the method for designing the uniform junction distribution pool, we started from a random pool and used ViennaRNA to generate the secondary structures, then converted the secondary structures to the tree structures. To be consistent with the mixing matrices
approach, the minimum length of a stem structure was defined as at least two contiguous pairs, and pools of 10,000 sequences were used for computation. Each tree structure is classified into different structure categories. Given an initial seed population of pool designs based on a number of tree structures such as $5_3$, $6_1$, $6_2$, $6_3$, $6_4$ and $6_5$, we used Genetic Filtering to create a single design that would yield a startup pool with a uniform distribution of 12 tree structures.
Figure S1: Structural distribution of the 2793 known aptamers from the Ellington Database (Lee et al., 2004).
Figure S2: Relationship between 5J population and sequence lengths for a random pool of 1 million sequences with (light grey) and without (grey) primer sites. When 5J sequences are needed in a 1-million random pool without primer regions, the sequence length should be over 76. In order to generate 5Js in a 1-million random pool with two 15nt primer regions, 98nt is the minimum sequence length required. Analytical calculation (black) is based on Equation 2 in Sabeti et al. (Sabeti et al., 1997), given \( n \approx 56 \). The “without primer” curve is in direct agreement with this analytical calculation.
Figure S3: The best fitness value observed at each generation of *Genetic Filtering* (primer sites “ATACCAGCTTATTCAATT” and “AGATTGCACTTACTATCT”) Parameters: NG=10000 NoChargeCutoff=500 Pcrossover=0.0 Pmutation=0.9 Pposition=0.0156 Pcopy=0.1 (see Materials and Methods). The best fitness score decreases steadily with each generation and stopped at generation 1626 when it remains steady after approximately 500 generations.
Figure S4: Sample secondary structure of GFPool2A sequence using primer sites “ATACCAGCTTATTCAATT” and “AGATTGCACCTTACTATCT”, designed to generate a SELEX starting pool with uniform structure distribution. This pool is 100nt in length and contains 25 random nucleotide positions.
Figure S5: Sample data from dissociation constant experiments. Top: ATPF8 ($K_d = 3.7 \mu M$); Bottom: ATPF4 ($K_d = 22.4 \mu M$)
**Figure S6:** Sample sequences and representative secondary structures for RFPoolA, shown in Figure 2.  A) a 5J structure B) a 4J structure C) a 3J structure and D) a 2J structure.
Figure S7: Resulting structural distribution (number of sequences displaying a given junction order) after addition of two 15 nt primer sites to 20 million five-way junctions sequences of length 70nt generated by Inverse Folding. Less than 1% of sequences retain five-way junctions and instead result in simpler structures.
Figure S8: Comparison of average structural distribution for the pools generated by Random Filtering (light grey) and Inverse Folding (dark grey). Random Filtering generates more complex structures than Inverse Folding.
Figure S9: Tree structure distribution for the 12 tree-structures, of the pool, \textit{GFPool3}, generated by the \textit{Genetic Filtering} method. A 35.23\% error from a uniform distribution is observed. The pool design obtained by this experiment is

“\texttt{NNNNCGANNGAAANCCAUUGACGUGNCCACGUAUGUGNACUNNCUCNNNNGAGNNUU}
\texttt{UCGACGUNNNGCGUCGAGANNAGUUNNNNNUUUCNNUUG}”.

For the definition of tree structure IDs (i.e. \texttt{2}_1 to \texttt{6}_5), see (Gevertz et al., 2005).
Figure S10: Comparison of predicted structural distributions of RNAfold (dark grey), GTfold (light grey) and MC-Fold (medium grey) for the same 1000 random sequences of 100 nucleotides each (containing no fixed primer sites). The structure distributions generated by RNAfold and GTfold are almost identical. However, MC-Fold predicted considerably more complex structures.
Figure S11: Comparison of junction distribution with different stem lengths. Selecting a smaller minimum stem length results in a larger number of predicted high complexity structures.
Figure S12: A sample five-way junction structure used to demonstrate our method for counting junctions.
Table S1: DNA sequences (predicted secondary structures in brackets) obtained experimentally from the **RFPool A.** Sequences were aligned by ClustalW2.
Table S2: Structural distribution, distance from the optimal, and number of single stranded positions present for DNA pools with primer sites. *GF Pool2*: Pool shown in Figure 5 using primer sites “ATACCAGCTTATTCAATT” and 3’ primer “AGATTGCACCTTACTATCT” *GF Pool2A*: Pool 2 modified by keeping the 20th single stranded position fixed.
ATPS11 (1J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS12 (2J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS14 (2J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS15 (2J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS16 (1J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS17 (2J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS18 (5J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS19 (3J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS20 (2J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS21 (1J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS22 (4J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS23 (5J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS24 (4J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS25 (3J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS26 (5J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS27 (5J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS28 (5J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS29 (4J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC
ATPS30 (4J)  --ATACCAGCTTTACCTTATTGCCATCTGGTTCAGAGCTGACTGTC

**  ** **** *   * *         **    *    * **    *   **

ATPS11 (1J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS12 (2J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS14 (2J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS15 (2J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS16 (1J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS17 (2J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS18 (5J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS19 (3J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS20 (2J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS21 (1J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS22 (4J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS23 (5J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS24 (4J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS25 (3J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS26 (5J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS27 (5J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS28 (5J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS29 (4J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-
ATPS30 (4J)  GGCGGGCCGCCGATCTATCTTAAGAAGATAGTAAG-TGCAATCT-

**  ** **** *   * *         **    *    * **    *   **

Table S3: DNA sequences (predicted secondary structures in brackets) obtained experimentally from the starting GFPool1 used in SELEX experiments. Sequences were aligned by ClustalW2.
** Table S4: DNA sequences (predicted secondary structures in brackets) obtained after 8 selection rounds against ATP. Sequences were aligned by ClustalW2.**
Table S5: Structural distribution determined with different numbers of mutations
(averages and variances in brackets).

<table>
<thead>
<tr>
<th>No. of mutations</th>
<th>Percent of 1J structures</th>
<th>Percent of 2J structures</th>
<th>Percent of 3J structures</th>
<th>Percent of 4J structures</th>
<th>Percent of 5J structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10.04 (9.72)</td>
<td>49.45 (30.23)</td>
<td>27.12 (20.19)</td>
<td>8.78 (9.71)</td>
<td>4.61 (4.54)</td>
</tr>
<tr>
<td>$10^3$</td>
<td>10.26 (0.70)</td>
<td>49.99 (3.05)</td>
<td>26.45 (2.04)</td>
<td>8.44 (0.65)</td>
<td>4.86 (0.41)</td>
</tr>
<tr>
<td>$10^4$</td>
<td>10.31 (0.09)</td>
<td>49.87 (0.29)</td>
<td>26.69 (0.19)</td>
<td>8.34 (0.07)</td>
<td>4.80 (0.04)</td>
</tr>
<tr>
<td>$10^5$</td>
<td>10.36 (0.01)</td>
<td>49.93 (0.02)</td>
<td>26.57 (0.02)</td>
<td>8.36 (0.01)</td>
<td>4.78 (0.00)</td>
</tr>
<tr>
<td>$10^6$</td>
<td>10.35 (0.00)</td>
<td>49.93 (0.00)</td>
<td>26.58 (0.00)</td>
<td>8.35 (0.00)</td>
<td>4.79 (0.00)</td>
</tr>
</tbody>
</table>

Table S6: Structural distribution of sequences created by mutating 50 different 3J sequences 1 million times each
**Genetic Filtering Algorithm (Pseudo-Code)**

initialize currentGeneration[popSize] //all five-way junctions

for i from 0 to popSize

    jct[i] is the junction distribution after mutating SS positions of
    currentGeneration[i] \(10^5\) times without changing primer sites

    distance[i] = \(\sum_{j=1}^{5} |jct[j] - djct[j]|\) //djct[i] is the desired junction distribution

    fitness[i] = fitness_function_with_penalty(distance[i]) //see below

Fitness = \(\sum fitness[i]\)

for i from 0 to popSize

    probF[i] = 1 - finess[i]/Fitness

    probC[i]=\(\sum_{j=0}^{i} probF[j]\) //cumulative probability

numGeneration=0; // create initial generation

for i from 0 to popSize

    oldGeneration[i]= currentGeneration[i]

    oldFitness[i]=fitness[i]

while (minFitness \(\neq\) 0) and numGeneration \(\leq\) NG

    and NoChangeCounter < NoChangeCutoff)

    /* create new generation */

    j=0

    while (j < popSize)

        /* every 10 generations put best so far into new generation*/

    Luo et al.
if numGeneration %10=0
    newGeneration[0]=bestSequence
    fitness[0]=best_fitness
    j++;

P_random = random number from [0, 1]

/*CROSSOVER method*/
if 0 ≤ P_random < P_crossover
    seq1=select() /*roulette wheel selection */
    seq2=select()
    midGeneration, midGeneration2= crossover(seq1, seq2)
    if midGeneration, midgeneration2 not in newGeneration
        newGeneration[j]= midGeneration
        newGeneration[j + 1]= midGeneration2
        j=j+2

/*MUTATION method*/
else if P_crossover ≤ P_random < P_crossover + P_mutation
    seq=select()
    midGeneration= mutate positions(seq)
    if midGeneration not in newGeneration
        newGeneration[j]=midGeneration
        j++

/*COPY method*/
else
midGeneration= select()

if midGeneration not in newGeneration
    newGeneration[j]=midGeneration
    j++

currentGeneration = newGeneration

for i from 0 to popSize
    if currentGeneration[i] in oldGeneration
        ix = index of currentGeneration[i] in oldGeneration
        fitness[i] = oldFitness[ix]
    else
        distance[i] = \[\sum_{j=1}^{5} | jct[j] - djc[t][j]| \]
        fitness[i]=distance[i]

    oldGeneration[size oldGeneration] = currentGeneration[i]
    oldDistance[size oldGeneration] = distance[i]
    size_oldGeneration++

update Fitness, probF[i], probC[i]

numGeneration ++

bestSequence = MAX(bestSequenceP_newGeneration, bestSequence_current)

if bestSequence has improved
    NoChangeCounter = 0
else
    NoChangeCounter++
Algorithm `Fitness_function_with_penalty(distance)`

```plaintext
definition:
num_ss = number of ss positions

if num_ss ≥ dnum_ss // dnum_ss is the desired number of ss positions
    fitness = distance + 0.2*(num_ss - dnum_ss)
else
    fitness = distance + (dnum_ss – num_ss)
```

Supplemental References


