Molecular computation: RNA solutions to chess problems

Dirk Faulhammer*, Anthony R. Cukras*, Richard J. Lipton†, and Laura F. Landweber*‡

Departments of *Ecology and Evolutionary Biology and †Computer Science, Princeton University, Princeton, NJ 08544

Communicated by Andrew Chi-Chih Yao, Princeton University, Princeton, NJ, December 3, 1999 (received for review April 25, 1999)

We have expanded the field of ''DNA computers'' to RNA and present a general approach for the solution of satisfiability problems. As an example, we consider a variant of the ''Knight problem,'' which asks generally what configurations of knights can one place on an $n \times n$ chess board such that no knight is attacking any other knight on the board. Using specific ribonuclease digestion to manipulate strands of a 10-bit binary RNA library, we developed a molecular algorithm and applied it to a 3 × 3 chessboard as a 9-bit instance of this problem. Here, the nine spaces on the board correspond to nine ''bits'' or placeholders in a combinatorial RNA library. We recovered a set of ''winning'' molecules that describe solutions to this problem.

DNA computing | satisfiability | RNA evolution | *in vitro* selection | SELEX

Adleman (1) introduced DNA-based computing as an approach to solving mathematical problems, which uses DNA as a data carrier and techniques of molecular biology to operate on DNA. Since then, there have been relatively few experimental demonstrations of DNA-based computations (2, 3), although the theoretical foundation is strong (4, 5). Here we introduce a method for the construction of binary nucleic acid libraries, and we introduce RNA as a molecule for computation to present a general approach for the solution of the famous satisfiability (SAT) problems of propositional logic.

Using a combination of a binary RNA library and ribonuclease (RNase) H digestion, we developed a destructive algorithm (6) that would hydrolyze RNA strands that did not fit the constraints of a chosen problem, instead of an algorithm that required efficient hybridization extraction (1, 2). The use of restriction endonucleases (3) would have also permitted a logical binary-style operation, as each restriction enzyme cleaves only in the presence of its recognition site, and cleaves the double-stranded DNA sufficiently to completion. However, by using RNase H in the context of different sets of oligonucleotides, one can go beyond the set of available restriction enzymes. Here, RNase H acts as a ''universal restriction enzyme'' because it allows selective marking of virtually any RNA strands for digestion in parallel, and use of a thermostable RNase H (7) ensures fidelity of hybridization between DNA and RNA strands, minimizing incorrect marking of noncognate strands.

Materials and Methods

Chemical Synthesis of DNA ''Half Libraries.'' The DNA library was prepared as two halves by using mix and split phosphoramidite chemistry with the sequence codes shown in Table 1. In brief, bit *n* set to 0 and spacer *n* were synthesized on one column, and bit *n* set to 1 and spacer *n* were synthesized on the other column. The columns were decrimped and the two resins poured together and mixed, and then half of this mixture (containing approximately equal proportions of 0's and 1's at bit position *n*) was returned to each column for synthesis of the next variable position, and the entire process was repeated (Fig. 1). Each of the four combinatorial synthesis products was purified by polyacrylamide gel electrophoresis on a 6% denaturing gel (Sequagel, National Diagnostics) and was recovered by a ''crush and soak'' procedure (500 mM ammonium acetate/1 mM EDTA/0.1% SDS) and ethanol precipitation (8). Annealing of the four half-sized strands in a 50-μl PCR reaction containing 1 μM of each of four halves (9) and primer overlap extension during five thermocycles (94°C for 30 sec, 45°C for 30 sec, and 72°C for 30 sec per cycle) generated the DNA library of full sized strands with a total of 1,024 (2^{10}) unique strands. (Note that the 3 × 3 instance of this problem considers nine bits, or a total of 512 different chessboards. Bit 10 was unused in the actual selections.) The degeneracy of the DNA pool was verified by direct sequencing (10) using 5′ end-radiolabeled primers TXR (Prefix; $[CAT]_4$CTCGAGAATT) and 32.41 (Suffix complement; $[CTA]_4$CGGGATCCTAATGACCAAGG) in cycle sequencing reactions (SequiTherm Excel, Epicentre Tech).

The 271-bp DNA pool was amplified by PCR with primers T7TXR (TTCTAATACGACTCACTATAGG$[CAT]_4$CTCGAGAATT) and 32.41 (1 μM each; 15 cycles of 94°C for 30 sec, 60°C for 30 sec, and 72°C for 30 sec). The 251-bp RNA library was then synthesized by *in vitro* T7-transcription from the synthetic DNA pool and was purified on a 4% denaturing polyacrylamide gel.

DNA Bit Oligonucleotides. The set of DNA bit oligonucleotides used both to operate on the RNA library and as the downstream primers in readout PCR were from Operon Technologies (Alameda, CA) and are shown in Table 2.

Implementing the RNA Algorithm. RNase H digestions destroyed the RNA strand of RNA·DNA hybrids marked by hybridization of the pool RNA to the complementary bit oligonucleotides shown in Table 2. One hundred-microliter RNase H reactions contained 20 μg of gel-purified RNA (250 pmol), 1 nmol of each bit oligonucleotide, 5 units of Hybridase Thermostable RNase H (Epicentre Tech), 50 mM Tris·HCl (pH 7.5), 100 mM NaCl, and 10 mM $MgCl_2$. After preincubation at 72°C for 3 min to denature strands, enzyme and $MgCl_2$ were added at 45°C. The reaction was held at 45°C for 60 min and then was stopped by addition of 10 mM EDTA. Spin column chromatography (CLONTECH Chroma Spin-200, DEPC-H_2O) was used to remove DNA bit oligonucleotides and short RNA digestion products. The eluted RNA was ethanol-precipitated, and remaining full-length (undigested) RNA was purified on a 5% denaturing polyacrylamide gel. Recovered RNA was reverse transcribed in 80 μl with 1.25 μM suffix complement (Superscript II, BRL). One-tenth of this reaction was amplified by PCR (200 μl, 1 μM primers T7TXR and 32.41, 2–5 cycles of 94°C for 30 sec, 60°C for 30 sec, and 72°C for 30 sec). After ethanol precipitation, the two halves of each pool were combined, and half of the DNA was transcribed *in vitro* for the next step in the algorithm.

Multiplex Colony PCR Readout. PCR products were directly cloned by using the TOPO-TA cloning kit (Invitrogen). Cells were

Abbreviation: SAT, satisfiability.

See commentary on page 1328.

‡To whom reprint requests should be addressed. E-mail: LFL@princeton.edu.

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

Table 1. Nucleotide sequences for each bit and spacer of the combinatorial library

Bit	Knight absent code, bit set to 0	Knight present code, bit set to 1	Spacer code
a	CTCTTACTCAATTCT	TCCTCACATTACTTA	TCTAC
b	CATATCAACATCTTA	ACTTCCTTTATATCC	ATAAC
c	ATCCTCCACTTCACA	TTATAACAAACATCC	CTTAA
d	TTAAAATCTTCCCTC	ACATAACCCTCTTCA	TTTAC
e	CTATTTATCCACACC	ACCTTACTTTCCATA	TACAA
f	GCTTCAAACAATTCC	GTACATTCTCCCTAC	TCCTT
g	AACTCTCAAATTCAA	CATAATCTTATATTC	TCAAT
h	CTAACCTTTACTTCA	ATAATCACATACTTC	TCCAA
i	CATTCCTTATCCCAC	TCCACCAACTACCTA	ACACA
j	CACCCTTTCTCCTCT	TTTTAAATTTCACAA	SUFFIX

Table 2. Nucleotide sequences for each DNA bit oligonucleotide used in the computation

	Hybridizes to bit set to 0		Hybridizes to bit set to 1
a_0	AGAATTGAGTAAGAG	a_1	TAAGTAATGTGAGGA
b_0	gttatTAAGAGGTTGATATG	b_1	gttatGGATATAAAGGAAGT
c_0	TGTGAAGTGGAGGAT	c_1	GGATGTTTGTTATAA
d_0	gtaaaGAGGGAAGATTTTAA	d_1	gtaaaTGAAGAGGGTTATGT
e_0	GGTGTGGATAAATAG	e_1	TATGGAAAGTAAGGT
f_0	GGAATTGTTTGAAGC	f_1	GTAGGGAGAATGTAC
g_0	attgaTTGAATTTGAGAGTT	g_1	attgaGAATATAAGATTATG
h_0	TGAAGTAAAGGTTAG	h_1	GAAGTATGTGATTAT
i_0	GTGGGATAAGGAATG	i_1	TAGGTAGTTGGTGGA

Twenty-base oligonucleotides are complementary to both the 15-nt bit sequence (uppercase) and adjacent 5-nt spacer (lowercase) to increase the melting temperatures of weaker hybridizing oligonucleotides.

grown on LB-agar plates containing 50 μg/ml ampicillin (8). Positive (white) colonies were randomly picked, were transferred to 50 μl of LB media containing 50 μg/ml ampicillin, and were grown for 4–6 hours at 37°C. Colony PCR (11) of 2 μl of media (95°C for 5 min, followed by 25 cycles of 95°C for 30 sec, 60°C for 30 sec, and 72°C for 30 sec with 0.2 μM primers TXR and 32.41) allowed rapid screening and recovery of individual DNA library strands. To readout the bit settings of cloned strands, 1 μl of a 1:100 dilution of the colony PCR product was first amplified by linear PCR (9) (10 μl, 0.2 μM primer T7TXR; 10 cycles as above) in the presence of prefix only to generate an excess of single-stranded DNA; then equimolar mixtures of 5′ end-radiolabeled "0"- or "1"-complementary DNA oligonucleotides ($\leq$1 nM each) were added to the same tube, and the reaction was continued for an additional five cycles. Reaction products were separated on a 6% denaturing polyacrylamide gel (Long Ranger gel solution, FMC Bioproducts). The dried gel was analyzed by autoradiography on a PhosphorImager (Molecular Dynamics).

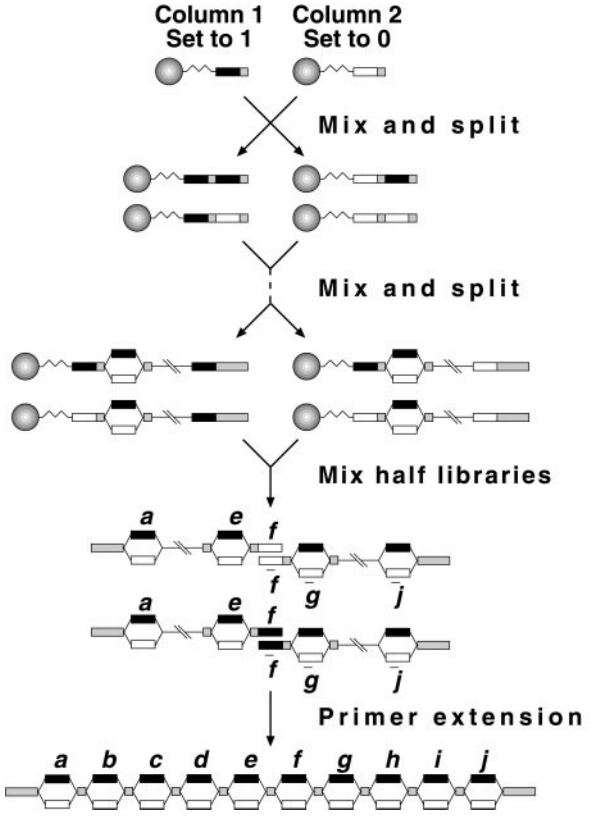

Fig. 1. Modular construction of the combinatorial DNA library from two half-libraries. Linear mixing and splitting steps yield exponential increases in pool complexity. Each half is synthesized on two columns beginning with bit *f* or its reverse complement. Synthesis of one half continues through bit *a* and the prefix; the other half contains the reverse complement of bits *f* through *j* and the suffix. Because the halves are complementary at position *f*, primer extension of the two halves creates the full-length 10-bit library. Black and white boxes represent 1 and 0, respectively; shaded boxes represent prefix, suffix, and spacer sequences.

Error Analysis. To determine potential sources of error, we measured the efficiency and recovery of each operation.

Digestion of the RNA library by thermostable RNase H. 5′ end radiolabeled RNA library (5 pmol) was subjected to RNase H digestion in the presence of combinations of DNA bit oligonucleotides (a_0, f_1, h_1; b_0, g_1, i_1; c_0, d_1, h_1; d_0, c_1, i_1; f_0, a_1, g_1; g_0, b_1, f_1; h_0, a_1, c_1; i_0, b_1, d_1; 75 pmol each; Table 2) under conditions as described above in 15 μl and 0.5 units RNase H. The digestion products were quantified on a 10% denaturing polyacrylamide gel by using a PhosphorImager. The amount of digestion products were in the expected range [12.5% undigested pool RNA, 12.5% first (longest), 25% second, and 50% third (shortest) digestion product]. Digestion of single RNA clones under the same conditions was almost complete ($\geq$98% in all cases but 96% for i_0, b_1, d_1).

Spin-column chromatography and gel purification. Column chromatography removed 5′ end radiolabeled DNA bit oligonucleotides (100 pmol) with >99% efficiency and recovery of 40–50% of 50 pmol 5′ end radiolabeled RNA library. Recovery of 50 and 30 pmol of cloned RNA molecules (the expected range after a digestion step) from a 6% denaturing polyacrylamide gel was 52% and 75%, respectively.

Reverse transcription and PCR. Thirty and fifteen picomoles of three cloned RNA molecules were reverse transcribed in the presence of 5′ end radiolabeled primer 32.41. Efficiencies were 50% in one of the clones and 25% in two other clones. One-tenth of these reactions was subjected to PCR (200 μl, 1 μM primers T7TXR and 5′ end radiolabeled 32.41, 1–5 cycles of 94°C for 30 sec, 60°C for 30 sec, and 72°C for 30 sec). Samples were removed after each cycle and were resolved on a 12% denaturing polyacrylamide gel. Amplification increased exponentially after cycles 1–4, but plateaued after the fifth cycle of PCR.

Results and Discussion

The "Knight Problem." Chess problems create a class of SAT problems that are flexible in both scale and complexity. A "true"

or "1" variable represents the presence of a knight at positions *a–i* whereas a "false" or "0" variable represents the absence of a knight at that position. This is a particular instance of a class of NP-complete SAT problems. For a 3 × 3 board

a	*b*	*c*
d	*e*	*f*
g	*h*	*i*

we represent the problem as follows (∧, "and"; ∨, "or"):

$((\neg h \wedge \neg f) \vee \neg a) \wedge ((\neg g \wedge \neg i) \vee \neg b) \wedge ((\neg d \wedge \neg h) \vee \neg c) \wedge ((\neg c \wedge \neg i) \vee \neg d) \wedge ((\neg a \wedge \neg g) \vee \neg f) \wedge ((\neg b \wedge \neg f) \vee \neg g) \wedge ((\neg a \wedge \neg c) \vee \neg h) \wedge ((\neg d \wedge \neg b) \vee \neg i)$.

In this particular example, this simplifies to

$((\neg h \wedge \neg f) \vee \neg a) \wedge ((\neg g \wedge \neg i) \vee \neg b) \wedge ((\neg d \wedge \neg h) \vee \neg c) \wedge ((\neg c \wedge \neg i) \vee \neg d) \wedge ((\neg a \wedge \neg g) \vee \neg f)$.

This reduces the number of operations that one has to perform.

Design and Synthesis of a 10-Bit RNA Library. A 10-bit pool size was chosen because this represents a pool of significant complexity, containing 1,024 different strands. Each strand of RNA follows the template below (4):

5′ Prefix	Bit *a* (0/1)	Spacer 1	Bit *b* (0/1)	Spacer 2	. . .
24 nt	15 nt	5 nt	15 nt	5 nt	. . .

Spacer 9	Bit *j* (0/1)	Suffix 3′
5 nt	15 nt	32 nt

Each bit can be set to an "on" (= 1) or "off" (= 0) sequence. Using computer simulations, we incorporated three important criteria into the design of the combinatorial pool: (*i*) Each of the bit encodings must be fundamentally different. Hence, sequences were chosen to maximize the Hamming Distances between different library strands and between the different parts of individual strands (no more than five matches over a 20-nt window, both within and between all 2^{10} possible strands). Bit sequences were also selected to have an average melting temperature of 45°C. (*ii*) The strands were biased to avoid secondary structure so that each bit position would be equally accessible to the enzymes and oligonucleotides that operate on the pool molecules. This was accomplished by using a three-letter alphabet for the bits and spacers in the library, A, C, and U, eliminating the potential for both G-C pairs and G:U pairs (which form in RNA) as well as G stacking. (*iii*) The strands were controlled to avoid hybridization to themselves or to any other library strands by more than seven consecutive base pairs (12) because such interactions would interfere with our ability to operate on the RNA strands by making these regions inaccessible to reagents.

To satisfy all of these criteria, a simple computer program (PERMUTE, published as supplemental material on the PNAS web site, www.pnas.org) generated random nucleotides for bit and spacer sequences to satisfy the second criterion and then permuted the sequences until they fulfilled all three criteria. The prefix and suffix provide PCR primer-binding sites, with the prefix containing the *T7* RNA polymerase promoter for transcription. The resulting sequences are shown in Table 1.

We synthesized a 10-bit DNA library as two halves by a recursive mix and split strategy (Fig. 1). The first half contained bits *a* through *f*, and the second half contained the complement of bits *f* through *j*. Bit *n* set to 0 and spacer *n* were synthesized on one column, and bit *n* set to 1 and spacer *n* were synthesized on the other column. The contents of both columns were mixed, and half of the mixture was returned to each column. The partially synthesized molecules were further extended by synthesis of the next bit set to 0 and corresponding spacer on one column, and the next bit set to 1 and corresponding spacer on the other column, and the entire process was repeated. The first half (130 nt) contained the prefix and bits *a* through *f*, and the second half (110 nt) contained the complements of bits *f* through *j* and the suffix. Primer extension from overlapping bit *f* by *Taq* DNA polymerase created full-length library strands, and *in vitro* transcription produced the RNA library. Equal two-fold degeneracy of each bit position was verified by directly sequencing the library (data not shown). Note that use of a 10-bit library to solve an instance of a 9-bit SAT problem allows one to ignore any position that computes less reliably, analogous to the problem of avoiding "bad blocks" on computer disks.

Algorithm for Solving the Knight Problem Using the RNA Library. We executed the algorithm for solving a variant of the knight problem with our RNA combinatorial library as follows (Fig. 2): Initially we have a test-tube that encodes all 1,024 possible 10-bit strings. Each string has the general form $x_1 \ldots x_n$, where a variable x_i is either 1 or 0, representing a bit set to on or off. (In our experiment, $x_1 = a, x_2 = b, \ldots, x_9 = i$.) Then we systematically perform the first operation to destroy strings that fail to satisfy the first clause $((\neg h \wedge \neg f) \vee \neg a)$ as follows: *(i)* One executes an OR clause by dividing the library into two "equal" collections. In one test-tube, we select those strands that contain a 1 at position *a* by annealing DNA bit oligonucleotide a_0 to the library, hence digesting those strands with position *a* set to 0 (thereby setting bit *a* to 1). Simultaneously, we destroy any 1's at those bit positions that must be set to 0 to fulfill the clause (bits *f* and *h* in this case). In the other test-tube, we anneal DNA bit oligonucleotide a_1 to perform the mirror operation setting bit position *a* to 0 ($\neg a$). (*ii*) Undigested molecules are recovered and reverse transcribed. After not more than five PCR cycles, the contents of both tubes are mixed and amplified to high copy number during *T7* RNA polymerization. The library is split again to execute the next OR clause, and this process is repeated until all clauses have been processed. The combinations of bit oligonucleotides in each step were as follows: $a \rightarrow 1$: a_0, f_1, h_1; $\rightarrow 0$: a_1; $b \rightarrow 1$: b_0, g_1, i_1; $\rightarrow 0$: b_1; $c \rightarrow 1$: c_0, d_1, h_1; $\rightarrow 0$: c_1; $d \rightarrow 1$: d_0, c_1, i_1; $\rightarrow 0$: d_1; $f \rightarrow 1$: f_0, a_1, g_1; $\rightarrow 0$: f_1.

"Bit Shuffling." Recombination events are likely to occur during PCR amplification of heterogeneous target sequences (13), especially because the strands in our 10-bit library share several stretches of sequence identity constrained by the bit structure of the library. To measure this effect, the PCR products derived from two clones (*wt*-I 011 111 001 and *wt*-II 010 110 010, *a*, . . . ,*i*) were combined and amplified for 25 cycles of PCR (0.2 μM primers TXR and 32.41; conditions as above). The resulting PCR products were cloned, and 20 clones were randomly chosen and analyzed by multiplex colony PCR readout at positions *c*, *f*, and *h*. Eight of twenty resulting clones (40%) were the product of bit shuffling. This indicates that bit shuffling poses a serious threat to successful execution of the algorithm; however, when only 15 instead of 25 PCR cycles were used, no bit shuffling was detected in 20 randomly chosen clones. This suggests that bit shuffling in our system is mostly due to a high proportion of incompletely extended strands annealing to heterologous target sequences, and it suggests that we can sufficiently reduce this problem by limiting the number of PCR cycles.

Consequently, no more than five cycles of PCR were performed between any two digestion steps and before cloning. Instead, we exploited *T7* RNA polymerase as an amplifier, because this enzyme is more processive than *Taq* DNA polymerase.

Methods of "Readout." Readout of a randomly sampled set of clones was accomplished by colony PCR (10) followed by

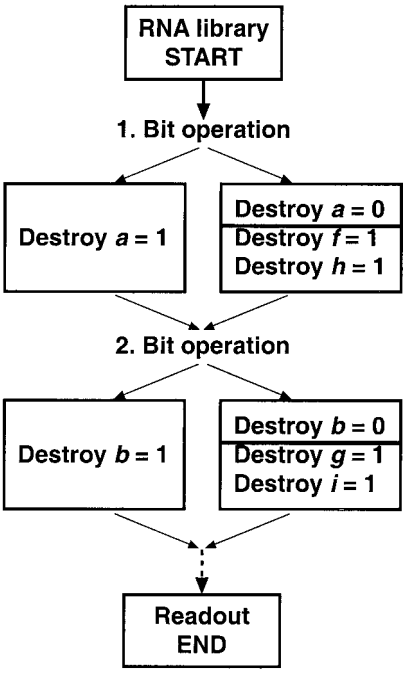

Fig. 2. Outline of the RNA algorithm. A value of 1 or 0 is assigned to a specific bit position by destroying all strands in the RNA library which do not have the value at this position (the step shown sets bits *a*, *f*, and *h*). The library is divided into two equal ensembles. In one test-tube, a specific bit position is set to 1 along with its accompanying constraints; in the other set, this bit is set to 0, using targeted digestion by RNase H. Continuing with this series of operations on the combined ensemble of strands leads to execution of the algorithm, which ends with readout of a randomly chosen subset of the surviving strands.

multiplex linear PCR. This creates a "bar code" for each strand (Fig. 3). If one wanted to solve a stricter version of the knight problem, one could select the board containing the maximal

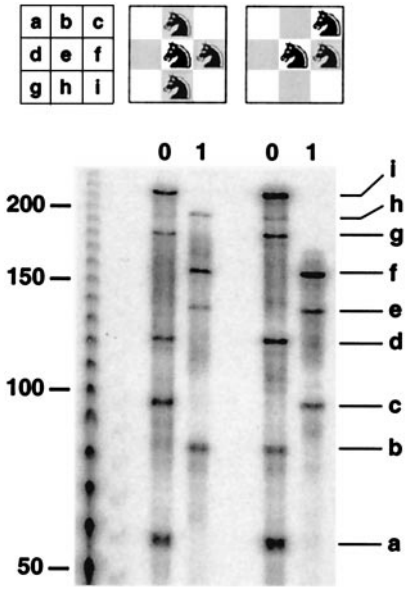

Fig. 3. Multiplex colony PCR readout. Colony PCR (10) allowed rapid screening and recovery of individual library strands. The boards represented by two clones are shown above their respective readouts on a 6% polyacrylamide gel. These clones represent the strings 010 011 010 and 001 011 000. Left lane, 10-bp ladder.

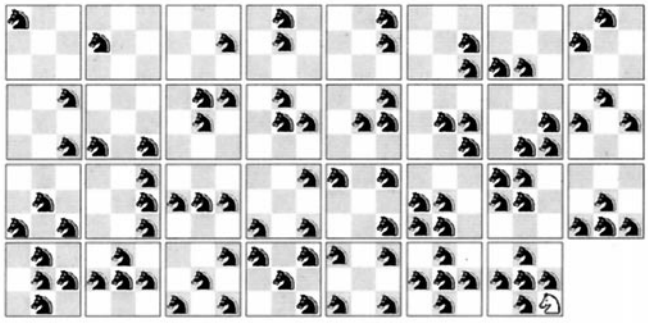

Fig. 4. Representations of the 31 unique boards analyzed by PCR readout. The last board contains one "illegal" white knight.

number of knights by choosing sequences of unequal length for 1's and 0's and purifying either the longest or shortest strands (3), but the use of bit sequences of unequal length would preclude the quick and streamlined readout in Fig. 3 because it relies on equal length for 1's and 0's (Table 1) to generate a bar code pattern. Furthermore, unlike that of Ouyang *et al.* (3), our approach requires neither the isolation of plasmid DNA nor actual sequencing of any clones for readout.

RNA Solutions. After subjecting the RNA library to the algorithm outlined above, 43 clones were randomly chosen and interpreted by readout PCR. Forty-two of these represented solutions to our version of the Knight problem (Fig. 4). Ten solutions occurred more than once, even though they were selected at random. However, the overall distribution of these 42 clones was not significantly different from a random sampling of all 94 possible solutions (Table 3), with a very modest preference for higher numbers of knights. We did not attempt to recover the other 64 solutions, as they are all variants of the classes of solutions shown in Fig. 4, and it would require a much larger sampling to ensure representation of each unique solution at least once. One clone (represented as 010 111 011) contained two illegal interactions, resulting from incorrect placement of one knight. Hence, of 127 knights on 43 boards, only one knight had an unacceptable placement. This was effectively a 97.7% success rate in finding correct solution strands, which is indeed satisfactory for choosing a clone that is likely to encode a solution.

DNA sequence analysis of four clones confirmed the results obtained by readout PCR and the robustness of our style of encoding that uses hybridization discrimination between "words" in DNA of 15 letters or more (14–16); each clone contained one to three point mutations or deletions, compared with the sequences in Table 1, but in general this did not affect either the algorithm or readout. However, the board with one incorrectly placed knight did contain an adjacent deletion and point mutation in bit 9 (TCCAC*T*ACTACCTA instead of TCCAC*CA*ACTACCTA), resulting in ineffective RNase H cleav-

Table 3. Expected and observed frequencies of boards with specific numbers of knights

No. of knights per board	Expected frequency (percent of 94)	Observed frequency (percent of 42)
5	2 (2.1)	1 (2.3)
4	18 (19.1)	11 (26.2)
3	36 (38.3)	16 (38.1)
2	28 (29.8)	10 (23.8)
1	9 (9.6)	4 (9.5)
0	1 (1.1)	0 (0)

age but correct readout with DNA bit oligonucleotide i_1. Four additional clones that could not be interpreted by readout PCR contained 3–5 deletions in single bit positions, one of which would have led to incorrect placement of another knight. This suggests that the main source of error is clusters of mutations in the same bit positions, but one could reduce the prevalence of such deletions by very accurate size purification, essential for the scalability of this system. Algorithms that employ recursive cycles of selection allow enrichment of the number of solution strands in each round as a form of progressive error correction (17–20). Repeated rounds of digestion might therefore enhance recovery of solution strands over ''non-solutions.'' Thus, because $2^{50} \approx 10^{15}$ is approximately the number of RNA molecules that *in vitro* selection protocols can currently search, this projects an upper bound for the size of DNA or RNA computing experiments that could use exhaustive search algorithms. Fortunately, this is on the same order as many interesting problems in computer science, such as the Data Encryption Standard (DES) (20).

Because *in vitro* selection experiments (18) that search for binding or function within an enormous sequence space have captured molecules as unique as one in a quadrillion, and because these molecules may even encode the solution to a mathematical problem, like knights on a chessboard, nucleic acid-based computing may be viewed as a natural extension of simulated evolution experiments to searches of a defined mathematical space.

We thank Steve Lawrie, Vernadette Simon, and Saw Kyin for synthesis of the oligonucleotide portions of the combinatorial pool. We also thank Leonard Adleman, Paul Rothemund, and Erik Winfree for helpful discussions and two anonymous reviewers for comments. This research was supported in part by the National Science Foundation/Defense Advanced Research Planning Agency, the Air Force Office of Scientific Research, and a Deutsche Forschungsgemeinschaft fellowship to D.F.

1. Adleman, L. M. (1994) *Science* **266,** 1021–1023.
2. Guarnieri, F., Fliss, M. & Bancroft, C. (1996) *Science* **273,** 220–223.
3. Ouyang, Q., Kaplan, P. D., Shumao, L. & Libchaber, A. (1997) *Science.* **278,** 446–449.
4. Lipton, R. J. (1995) *Science.* **268,** 542–545.
5. Landweber, L. F. & Baum, E. B., eds. (1999) *DNA Based Computers II* (Am. Math. Soc., Providence, RI).
6. Amos, M., Gibbons, A. & Hodgson, D. (1999) in *DNA Based Computers II*, eds. Landweber, L. F. & Baum, E. B. (Am. Math. Soc., Providence, RI), pp. 151–161.
7. Porter, D. & Curthoy, N. P. (1997) *Anal. Biochem.* **247,** 279–286.
8. Sambrook, J., Maniatis, T. & Fritsch, E. F. (1989) *Molecular Cloning: A Laboratory Manual* (Cold Spring Harbor Lab. Press, Plainview, NY).
9. Landweber, L. F. & Kreitman, M. (1993) *Methods Enzymol.* **218,** 17–26.
10. Rao, V. B. (1994) *PCR Methods Appl.* **4,** S15–S23.
11. Trower, M. K. (1996) *Methods Mol. Biol.* **58,** 329–333.
12. Zuker, M. (1989) *Science* **244,** 48–52.
13. Meyerhans, A., Vartanian, J. & Wain-Hobson, S (1990) *Nucleic Acids Res.* **18,** 1687–1691.
14. Frutos, A. G., Liu, Q., Thiel, A. J., Sanner, A. M. V., Condon, A. E., Smith, L. M. & Corn, R. M. (1997) *Nucleic Acids Res.* **25,** 4748–4757.
15. Smith, L. M., Corn, R. M., Condon, A. E., Lagally, M. G., Frutos, A. G., Liu, Q. & Thiel, A. J. (1998) *J. Comp. Biol.* **5,** 255–267.
16. James, K. D., Boles, A. R., Henckel, D. & Ellington, A. D. (1998) *Nucleic Acids Res.* **26,** 5203–5211.
17. Stemmer, W. P. (1995) *Science* **270,** 1510.
18. Landweber, L. F. (1999) in *DNA Based Computers II*, eds. Landweber, L. F. & Baum, E. B. (Am. Math. Soc., Providence, RI), pp. 181–189.
19. Boneh, D., Dunworth, C., Lipton, R. J. & Sgall, J. (1999) in *DNA Based Computers II*, eds. Landweber, L. F. & Baum, E. B. (Am. Math. Soc., Providence, RI), pp. 163–170.
20. Boneh, D., Dunworth, C. & Lipton, R. J. (1999) in *DNA Based Computers*, eds. Lipton, R. J. & Baum, E. B. (Am. Math. Soc., Providence, RI), pp. 37–65.