

In vitro Selection for a Max 1s DNA Genetic Algorithm

Junghuei Chen[†], Eugene Antipov[‡], Bertrand Lemieux, Walter Cedeño,
and David Harlan Wood[†]

ABSTRACT. Genetic algorithms, DNA computing, and *in vitro* evolution are briefly discussed. Elements of these are combined into laboratory procedures, and preliminary results are shown. The traditional test problem for genetic algorithms called the MAX 1s problem is addressed. Preliminary experimental results indicate successful laboratory “separation by fitness” of DNA encoded candidate solutions.

1. Introduction

Evolution is a concept of obtaining adaptation through the interplay of selection and diversity. Analogies from evolution have been used in both computing and molecular biology. These two areas are called respectively “evolutionary computation” and “*in vitro* evolution.” From the beginning of DNA based computing to the present there have been calls [11, 22, 28] to consider carrying out evolutionary computations using genetic materials *in vitro*.

In this paper we identify elements of evolutionary computations and *in vitro* evolution that we recommend combining to address three simple problems. Specifically, we choose “genetic algorithms” because they manipulate bitstrings using operations of pointwise mutation and crossover. These operations can be performed by modifications of techniques from *in vitro* evolution. In particular, single point crossover extends results due to Stemmer [30, 26].

We propose a laboratory implementation of one of these computations, and present our design. The crucial operation of physically separating DNA strands by their “fitness” is demonstrated using 2-d denaturing gradient gel electrophoresis.

This paper is an expansion of a paper [37] for the *Genetic and Evolutionary Computation Conference*. In the process, the size has roughly doubled by incorporating (1) details of experimental design, (2) laboratory demonstration of selection and readout, (3) computations confirming alignment specificity, and (4) computer simulation of sample candidate DNA strands in 2d denaturing gradient gel electrophoresis.

1991 *Mathematics Subject Classification*. Primary 68Q40, 68U99, 68Q15.

[†]Partially supported by DARPA/NSF Grant No. 9725021.

[‡]Partially supported an REU Supplement to DARPA/NSF Grant No. 9725021.

2. Genetic Algorithms and Other Evolutionary Computations

Generic algorithms and other types of “evolutionary computation” come in many, many different varieties, but most are variations and/or elaborations on the following very loose outline. Genetic algorithms typically cycle electronic computers through the following steps, maintaining a population of bitstrings representing candidate solutions of a problem.

Genetic Algorithm

Begin with a diverse initial population, perhaps chosen randomly.

1. Evaluate fitness of candidates.
2. Select more fit candidates to breed and less candidates to be replaced.
3. Induce variation by breeding.

Repeat.

It should be mentioned that this loose outline fits several evolutionary computation paradigms having varying techniques for fitness evaluation, selection, and breeding.

2.1. Evolutionary Computation Has Controversial Aspects. We are careful to make the following point. We do not take any stance on the virtues of any particular method of evolutionary computation. Instead, we aspire to *provide the means for assessing* some evolutionary computations using population sizes larger than is practical with conventional computers.

Evolutionary computation makes few assumptions and is ostensibly applicable to very broad classes of problems. Naturally, this makes it difficult to establish any provable guidelines. Just to list a few debatable issues, we have

- How does one model fitness?
- Is crossover disruptive?
- What are the roles of transposition, inversions, and introns?
- Would pointwise mutation alone suffice?
- Which evolutionary computations predictably converge?
- How does one recognize convergence?

3. DNA Is Suitable for Implementing Genetic Algorithms

Several means of DNA computation have been addressed. The first was, of course, by Adleman [2, 3]. Recent overviews can be found in [15] and [21]. See also the DNA computing bibliography of J. H. M. Dassen [10].

We expect computing time using DNA to be proportional to the number of generations required. This motivates incorporating both pointwise mutation and crossover, and for that matter any evolutionary analogies that might reduce the number of required generations.

Modifications to current technology suffice to implement crossover and pointwise mutation. However, selecting DNA strands for “breeding” in genetic algorithms is moderately challenging because one must physically separate DNA strands according to their “fitness.”

3.1. DNA Attributes Suit Genetic Algorithms. Of all evolution inspired approaches, genetic algorithms seem particularly suited to implementation using DNA. This is because genetic algorithms are generally based on manipulating populations of bitstrings using both crossover and pointwise mutation.

DNA computing techniques are desirable for genetic algorithm computations.

- The first main advantage is these techniques might process, in parallel, populations billions of times larger than is usual for conventional computers. The usual expectation is larger populations can sustain larger ranges of genetic variation and thus can generate high-fitness individuals in fewer generations.
- The second main advantage is the massive information storage available using DNA. For example, a gram of DNA primers contains about 10^{21} bases. The information content is approximately 2×10^{21} bits, greatly exceeding the 200 petabyte storage of all the digital magnetic tape produced in one year (1995) [36].
- A third advantage is that methods for implementing crossover using DNA are possible as variations on “Sexual PCR” pioneered by Stemmer [30].
- An additional issue is that biolaboratory operations on DNA inherently involve errors. These are more tolerable in executing genetic algorithms than in executing deterministic algorithms. To some extent, errors may be regarded as contributing to desirable genetic diversity.
- Oversimplified estimates indicate [37] DNA computing techniques can compare favorably to supercomputers in some cases. These favorable cases include executing genetic algorithms having simple fitness evaluations and very large populations of candidate solutions.

3.2. DNA Genetic Algorithms Compared to *In Vitro* Evolution. Genetic algorithms are reminiscent of methods in molecular biology referred to as “*in vitro* evolution.” Naturally, these methods use fitness criteria constrained to properties of biomolecular interest. Indeed, finding means to physically separate biological materials by “fitness” has determined which problems are addressed by *in vitro* evolution. *In vitro* methods suit some very important but relatively small classes of problems seeking ribozymes [34, 23, 16, 32], binding sites [33, 32], enzymes [27], etc.

In contrast to *in vitro* evolution, genetic algorithms use *unconstrained fitness criteria* on bitstrings. Consider computations using bitstrings of length 100. Such computations can, in principal, evolve a population of fixed size in such a way as to create any one of $2^{100} \approx 10^{30}$ possible outcomes. Thus, genetic algorithm methods using DNA can address larger, but less structured, classes of problems than does *in vitro* evolution.

Genetic algorithms using DNA would be similar to conventional computers in that (virtually) all 10^{30} possible inputs and outputs would be equally suitable. In contrast, *in vitro* evolution is suited to the very rare DNA sequences encoding biological functionality. For example, bitstrings might be realized as DNA strands having 100 As and Gs. But as far as we know very few, if any, such DNA bitstrings code for biologically active functions. Or to put it another way, virtually all sequences are equally “meaningful” for the purposes of genetic algorithms. In contrast, *in vitro* evolution focuses on variations of the very rare DNA sequences of biological or biochemical interest.

4. The MAX 1s Problem Implemented in DNA

Here we present a problem that is relatively simple to address using DNA implementations of genetic algorithms. The essence of our implementation is that more

fit candidates strands of DNA can be physically separated from less fit candidates according to how well they match (hybridize with) “target” DNA strands.

The Max 1s problem is a traditional test problem for genetic algorithms. It involves binary bitstrings of fixed length. An initial population (usually randomly generated) is given. The objective is to evolve some bitstrings to match a prespecified “target” (generally taken to be all 1s).

We now outline a DNA implementation of the Max 1s problem. The corresponding information is shown in Figure 1. In the remainder of this section, details

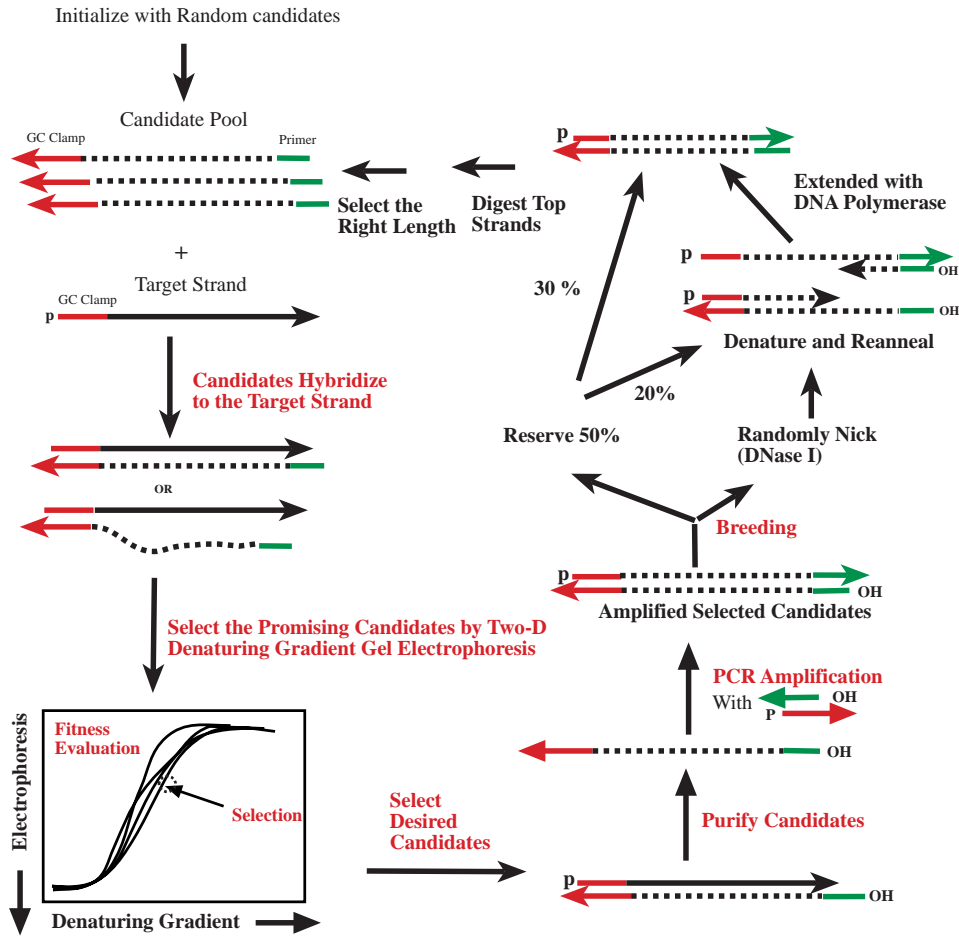


FIGURE 1. Outline of DNA implementation of genetic algorithms for MAX 1s problem. The candidate pool appears in the upper left. Selection using 2-d DGGE appears at the lower left. Purification and amplification of the more fit candidate strands appears at the lower right. Breeding using crossover appears at the upper right.

are given. Throughout this section, information is grouped in the following categories: (1) candidate pool, (2) fitness evaluation, (3) selection and (4) breeding.

Preliminary laboratory results are given later, in Section 5.

4.1. Outline of DNA Implementation. The implementation is given by the following outline. The same information, with a few added details, is shown in Figure 1.

DNA Genetic Algorithm for MAX 1s Problem

Begin with a diverse initial population of candidates.

1. Evaluate fitness by hybridizing to target strands and physically separate on a 2-d gel.
2. Select and purify more fit candidates to breed.
3. Amplify fit candidates with pointwise mutation and reserve a portion.
4. Breed candidates, using crossover.
5. Combine reserved and bred candidates, obtaining a new generation.

Repeat.

4.2. Design of Candidate Solutions and Target DNA Strands. Figure 2 shows our design. A target strand and a perfect candidate strand are shown

```

/-----TARGET-----\
/----Comp-CLAMP----\ /----- 40 A's-----\
3' <- GGATAGGCACCGCTCCGTGGAAAAAAAAA.....AAAAAA <- 5'
5' -> CCTATCCGTGGCGAGGCACCTTTTTT.....TTTTTTTCACCGTGGCTCGCACCGAG -> 3'
\-----CLAMP-----/\----- 40 T's-----/\----- TAIL -----/
\-----PERFECT CANDIDATE-----/

3' <- AGTGGCACCGAGCGTGGCTC <- 5'
\----Comp-TAIL-----/

```

FIGURE 2. Design of target and a perfect candidate. Imperfect candidates would have a mixture of Ts and Cs in place of the 40 Ts in the perfect candidate.

in the figure. Imperfect candidate strands have a mixture of 40 Ts and Cs instead of 40 consecutive Ts. At the 5' end of all candidate strands there is a universal section known as the CLAMP section of the target strands. All 3' ends of the candidate strands are extended by a universal TAIL sequence. The candidate strands are longer to facilitate eventual separation of target and candidate strands using denaturing gel electrophoresis.

Since candidate strands have known primer sites at both ends, they can be amplified by PCR. One primer we will call the TAIL-primer, which is the reverse complement of the TAIL. The other primer is simply the CLAMP sequence.

The CLAMP and TAIL sequences have been designed to encourage correct alignment of the PCR primers and to avoid sticking where they are not supposed to stick. Figure 3 shows melting temperatures T_m (a measure of the strength of hybridization) at all possible alignments of the CLAMP and the TAIL-primer and their reversals. These short strands are tested against a full-length candidate strand.

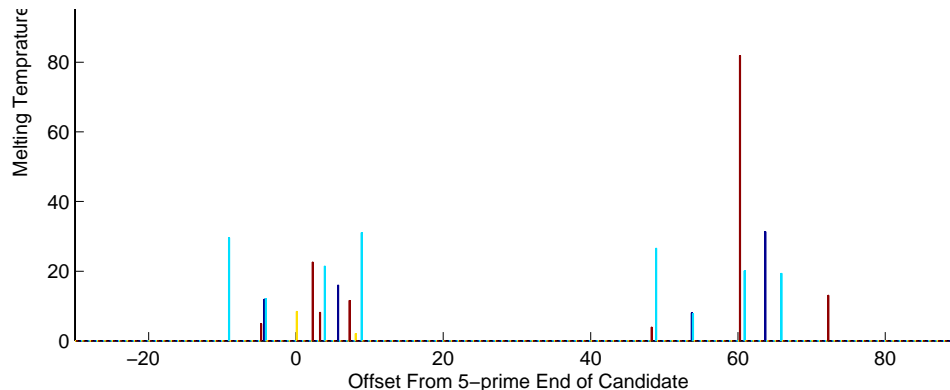


FIGURE 3. At an offset of 60, the correctly aligned TAIL-primer sticks tightly to the candidate strand, providing a melting temperature of 82 degrees. Misalignments of the CLAMP and the TAIL-primer, and their reversals, always stick weakly with melting temperatures of 31 degrees or less.

This figure and the next was generated using the BIND software [17] from MIT, for which we have a license for limited use at UDel.

Figure 4 shows melting temperatures T_M at all possible alignments of the

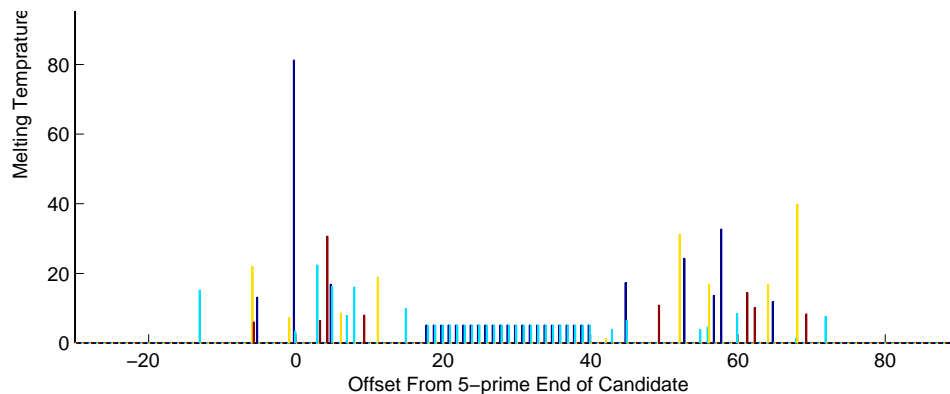


FIGURE 4. At an offset of 0, the correctly aligned CLAMP sticks tightly to the reverse complement candidate strand, having a melting temperature of 81 degrees. Misalignments of the CLAMP and the TAIL-primer, and their reversals, always stick weakly with melting temperatures of less than 40 degrees.

CLAMP and the TAIL-primer and their reversals. These short strands are tested against the full-length reverse complement of a candidate strand.

4.3. Fitness Evaluation by DGGE Physical Separation of DNA. Our fitness evaluation is carried out in the laboratory using so-called 2-d denaturing gradient gel electrophoresis (DGGE) [18]. Let us first review the nature of DGGE. Figure 5 shows DGGE from our laboratory having perfect candidates combined

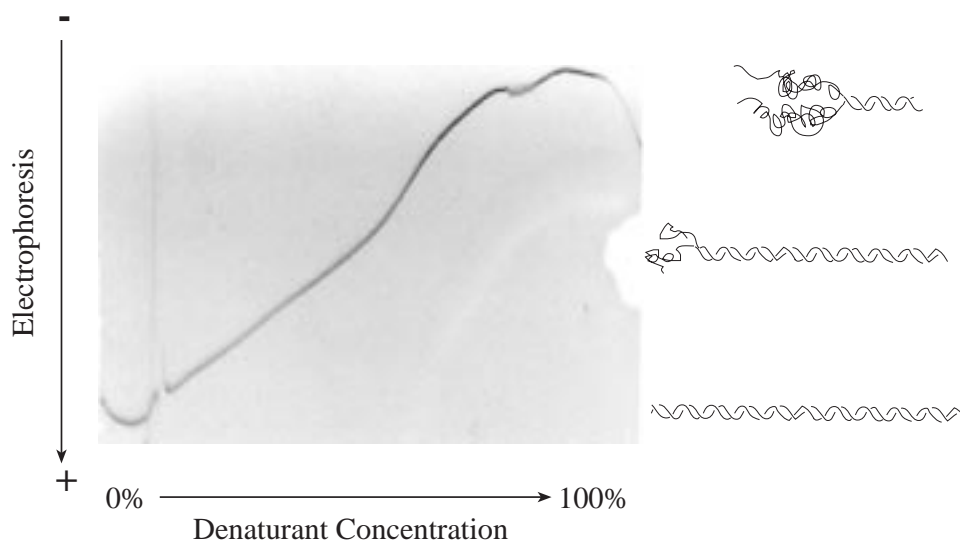


FIGURE 5. DGGE using perfect candidates. DNA strands move downward from a reservoir at the top of the figure. The speed of vertical strand migration is retarded as strands come apart (denature) as shown schematically on the right edge of the figure.

with target strands. The target strands hybridize (stick) to the perfect candidate strands. Each candidate is extended by a universal TAIL at its 3' end. During hybridization a short strand complementary to the TAIL is added to stabilize this region, preventing it from hybridizing (sticking) in unforeseen places. The mixture of hybridized strands is placed uniformly along the top of the gel. The hybridized strands travel vertically downward in the gel as a result of an applied electric field. However, their speed of migration is determined by their initial placement from left to right; that is, by how strongly they are denatured (pulled apart). On the left, where no denaturant is encountered, the strands move relatively quickly downward. In the center, they move more slowly because they encounter intermediate denaturing. At the extreme right, the strands are able to move only very slowly because the strands are almost completely pulled apart except in the more resistant CLAMP region.

An important fact for DNA computing is that *DGGE can detect a single base mismatch*. Indeed, this was an early application of DGGE in molecular biology [18].

4.4. Selection of More Fit Candidate Solutions. Selection is done by literally cutting out a portion of the 2-d gel and extracting the DNA strands from it. The most fit candidates are presumably lowest on any vertical line.

This allows a wide latitude for selection criteria. Experience in genetic algorithm computing demonstrates the desirability of maintaining genetic diversity to prevent the loss of genetic materials which may be needed in later stages of evolution.

4.5. Removing Target Strands From More Fit Candidates. The selected DNA can be purified by digesting target strands. Since these strands are fabricated with phosphorylated 5' ends, they can be selectively digested by λ -exonuclease.

4.6. Amplification (Breeding) With Mutation via PCR. The purified candidate strands are amplified by PCR, which also induces pointwise mutation at a rate of about 10^{-4} nucleotide per cycle when *Taq* DNA polymerase is used [12]. Special PCR protocols can be used to induce pointwise mutation at a rate of 10^{-2} to 10^{-4} [5, 6, 8, 19, 20, 35, 38].

One of the PCR primers (the complement of the TAIL), that is used to make the strands which are complementary to the candidates, is phosphorylated on its 5' end so that these strands can later be digested with λ -exonuclease.

Notice that candidates strands and their complements amplify at an exponential rate, doubling in number with each PCR cycle. They would increase a million-fold in 20 PCR cycles. In contrast, target strands can hybridize with *only one* of the two primers, namely the CLAMP primer. Thus, any target strands present can only amplify at a linear rate. They would increase twenty-fold in 20 PCR cycles. It is important to note that these complements of target strands are both rare and *shorter* than candidate strands.

4.7. Purification of Mutated Candidates. The PCR product can be purified by length using conventional denaturing gel electrophoresis. This gets rid of any target strand residue because these will be shorter than the candidates and their complements. Any excess primers are also eliminated. The purified candidates and their complements can be further amplified with mutation using additional primers and PCR. A relatively large quantity is generated to allow for the very low yield expected from the subsequent crossover step.

A portion of the double stranded product is temporarily reserved; the remainder is used for crossover.

4.8. Breeding Using Single Point Crossover. The double stranded product to be used for breeding is partially digested with *DNase I* to mostly nick (cut only one strand) at random locations about once per strand. The nicked strands are combined with a similar amount of reserved unnicked strands. The mixture is denatured (strands are melted apart) and allowed to reanneal forming new combinations. Many, many possible configurations could be formed. But among these, some will be portions of complements of candidate strands annealed at most positions to a 5' end of a candidate strand. The TAIL of the candidate strands is designed to enforce alignment. A typical desirable hybridization is illustrated in the upper right corner of Figure 1. By adding *DNA Polymerase I*, the partial candidate strand is extended to a full length candidate. In this process it combines its genetic information with that encoded in the intact strand. *DNA Polymerase I* is used because it will remove any annealed fragments in its way as it proceeds in the 5' to 3' direction. This ensures the maximum possible extension.

The net result is single point crossover. The offspring candidate strand produced by the polymerase has a block of genetic information from one parent followed by another block from a different parent. Our present approach limits us to using single point crossover (which is the most usual approach used in genetic algorithms).

4.9. Purification to Select Candidate Strands. The reaction products from crossover are combined with the reserved material. Complementary strands have been created by extended phosphorylated primes so they can be digested with λ -exonuclease. This eliminates, or at least greatly shortens, complements of candidate strands.

Purification by length (using denaturing gel electrophoresis) completes the breeding operation.

4.10. The New Generation. The new generation is now complete and ready to be processed.

4.11. Readout of Candidate Pool. We want to monitor the generation-to-generation progress of our *in vitro* evolution. Two protocols follow.

1. A readout of the average “fitness” of the candidates in a pool. That is, we estimate the average number of Ts between the CLAMP and TAIL of candidates strands in the pool.
2. A sensitive detection of “perfect” candidates in a candidate pool. That is, we detect if any candidate has all Ts between its CLAMP and its TAIL.

We assume we are given a pool of candidate strands. (Any contaminating complementary strands, or even some contaminating target strands, will not affect the readout results.)

4.11.1. *Estimation of Average Fitness.* A large number of primers complementary to the TAIL section of candidates are added to the pool of candidates. By first raising, then lowering, the temperature each candidate preferentially anneals to a primer. The primer is extended in the 3' direction by including *Taq* polymerase and including radioactive nucleotides. *We use only the nucleotide dATP.* Thus, the primers are extended as long as the candidate contains only Ts, and extension stops upon encountering any other base.

The result is: for each candidate strand, we obtain a primer extended by k radioactive As is generated, where k is the number of consecutive Ts in the candidate strand (counting from its TAIL). Each temperature cycle extends a primer on each candidate strand. The resulting radioactive extended primers of various lengths can be separated and read out using denaturing gel electrophoresis. Thus, an estimate of the average fitness of a candidate pool can be obtained.

4.11.2. *Sensitive Detection of Perfect Candidates.* Consider the situation at the finish of a primer extension step in the previous protocol. If any perfect candidates are present, the primers annealed to them will have been extended all the way up to the CLAMP region. They can not be extended any further, because the CLAMP has a C at its 5' end, and only *dATP* nucleotides are available. Because of this, the complement of the CLAMP can not be constructed by the polymerase. At this point, we add (lengthened) copies of the complement of the CLAMP and allow them to anneal onto the CLAMP portions of each candidate. In the case of perfect candidates, and only in that case, the extended primer and the complement of the CLAMP will be in physical contact. Adding ligase bonds these two together. Since we use copies of the complement of the CLAMP that have been lengthened by twenty bases at their 3' end, the ligated DNA strands are twenty bases longer than the candidate strands. In fact, they are longer than any other strands present on the mixture: candidates, primers, lengthened CLAMP complements, contaminating complements of candidates, target strands, etc. In summary, a strand twenty bases

longer than the candidate strands is constructed if and only if a perfect candidate was present in the candidate pool.

We then selectively amplify and detect any such longest strands by using PCR and gel electrophoresis separation by length. PCR primers for this task are determined by the twenty bases at each of the ends of the long strand. One of these primers (determined by the lengthening) can only anneal to the long strand. Thus, the only long strands that can be produced by PCR are copies of the long strands produced by ligation (or their complements). Notice the other PCR primer (the reverse complement of the TAIL) can be extended on other templates as well, but this results in short strands because all the other templates are short.

Gel electrophoresis separation by length will show a band at the longest length when and only when a perfect candidate was present in the candidate pool.

5. Experiment Demonstrating Selection and Readout

5.1. Candidate Solution Complements Used for Readout. In our experiments the 40 variable positions between the CLAMP and the TAIL of each imperfect candidate are independently chosen to be Ts with probability .8 and Cs with probability .2. Figure 6 shows how the number of Ts in a candidate varies according to the random mixture used to fabricate the candidates. We see that

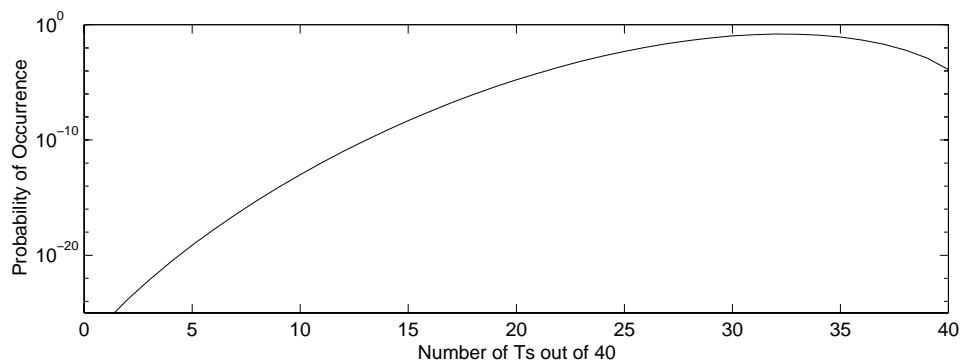


FIGURE 6. Relative frequency as a function of the number of Ts, for a mixture of 80% Ts and 20% Cs.

about .01% of the candidate strands should already be a perfect solution, so in our experiments we prevent this by always placing a C at the 11th position between the CLAMP and the TAIL. This universal C is located at the 31st base from the 5' end of each candidate DNA strand.

Figure 7 shows the method of generating radiolabeled primer extensions for readout. Notice that we are reading the *complements* of candidates, rather than the direct readout of the candidates themselves (as was described in Section 4.11.1). The process is entirely analogous.

The candidates in the initial pool all contain a C 31 bases from the 5' end. In Figure 7, this corresponds to the G shown on a strand complementary to a candidate, which is to be read. All primer extensions would, in principle, halt at or before this G. However, *Taq* polymerase will in fact complement a G with a T, with a small probability. (This probability is enhanced when no Cs are available.)



FIGURE 7. Readout of strand complementary to a candidate strand by means of primer extension. A radiolabeled primer CLAMP is annealed to the strand and extended by *Taq* polymerase in a solution containing *only* dTTP. Thus, extension halts upon encountering the first base which is not an A.

5.2. Experimental Fitness Evaluation by DGGE Physical Separation of DNA. Recalling the nature of 2d DGGE discussed in section 4.3, we heuristically reason about a mixture of targets and imperfect candidates. We expect that everywhere across the gel the candidate strands that best match (hybridize to) the targets will migrate downward relatively faster. In fact, imperfect matches exhibit vertical spreading in our experiments. See Figure 8. We assume that on any vertical line the most fit candidates are lowest on the 2d DGGE gel. However, the nature of variation from left to right is not clear. In our preliminary experiments, we have selected from several places at the lower edge of the 2d DGGE in Figure 8 and combined the extracted DNA. Further experiments will be needed to optimize a selection strategy.

Using the on-line POLAND software, we obtain estimates of the mobility of our particular candidate design, having (selected) mismatches. Figure 8 shows DGGE predictions from the POLAND software [25].

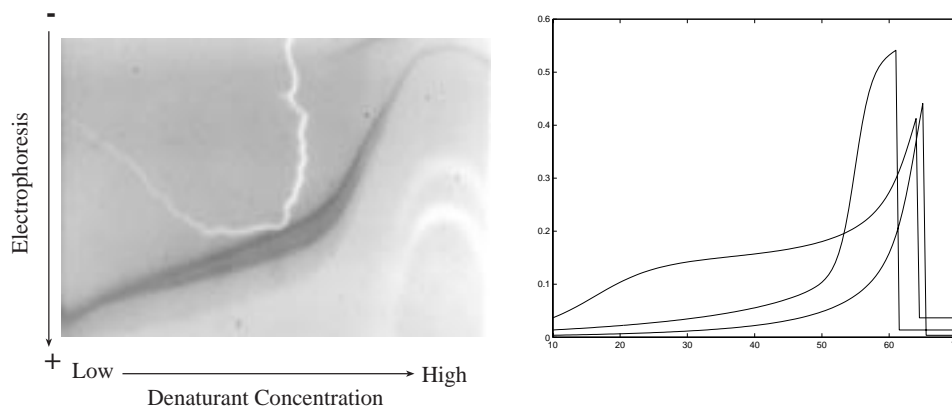


FIGURE 8. DGGE vertically separates imperfect candidates. The speed of vertical strand migration is retarded as strands come apart (denature). This is due to two factors: increasing denaturant concentration and decreasing quality of target-candidate matching (hybridization). On the right, DGGE predictions for selected imperfect candidates using the POLAND software. At 50° , the curves correspond, from the bottom up, to (1) perfect candidate, (2) single mismatch at position 50, and (3) single mismatch at position 31.

5.3. Experimental Readout of the Result of First Round of Selection.

Figure 9 contains data from a preliminary experiment demonstrating readout before and after selection.

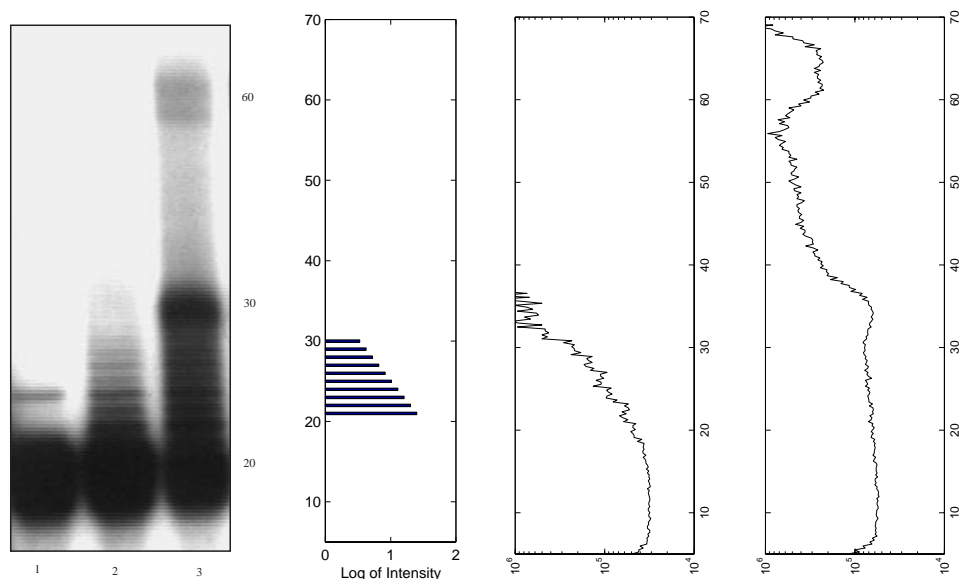


FIGURE 9. The gel on the left: Lane 1 contains only the radiolabeled primer. Lane 2 is a readout generated by complements of the pool of initially random candidates. Lane 3 is a readout generated after the first round of selection. (All strands to be read contain a G in position 31.) On the near left: A theoretical computation of expected intensities based on the probability (.8) used to generate the initial pool. On the near right: A logarithmic scan of Lane 2. On the far right: A logarithmic scan of Lane 3, reading the results of the first round of selection.

Lane 2 of the gel and its scan show a nearly exponential decay. Exponential decay would be appropriate for this outcome, corresponding to the question of “What is the frequency of observing runs of all tails using a biased coin?”

Lane 3 of the gel and its scan show the pool of candidates has been dramatically altered by the first round of selection. Now one observes many more long runs of Ts. Therefore, the strands being read out represent much more nearly homogeneous sequences. This represents significant progress toward our goal of evolving some completely homogeneous strands, which represent solutions to the Max 1s problem.

Also, in Lane 3 of the gel and its scan, we see that when *Taq* polymerase breaks through the barrier at the 31st position, we see that the selection has considerably homogenized the strands. The bump at 60 bases into the scan of the third lane (that is, the rightmost piece of Figure 9) shows that relatively many primer extensions run to the maximum possible length. That is to say, it appears that many solutions of the Max 1s problem have been generated. Regrettably, not all the full length extensions are due to perfect solutions because of unwanted bonding of Gs to Ts. However, unwanted bonding is relatively rare and equally present in Lanes 2 and 3.

It is the the comparison of these two lanes that shows the significant improvement resulting from the first round of selection.

However, it must be said that the experiment shown in Figure 9 is only preliminary and is in need of further refinement. For example, the leftmost lane of the gel shows that the primer does not form a single distinct band. This leads to ambiguities in the other two lanes. The G stopper placed in position 31 was neither effective nor important — it was simply irrelevant. (We used material we already had on hand.) Selection would be better illustrated without any stoppers. The unwanted bonds of G to T recommends reading candidate solutions “from the other end,” as we describe earlier in Section 4.11.1. In that case, the analogous situation involves unwanted bonds of C to A. These are much more uncommon.

6. Conclusions

Although much work remains to be done, it seems clear that even a single round of selection using 2d DGGE is capable of significantly enhancing the “fitness” within a population of candidate solutions of the Max 1s problem. The interest is that candidate solutions are encoded in strands of DNA. Thus, we are dealing with populations billions of times larger than is the usual practice with conventional computers. All candidate solutions in the entire population have their fitness evaluated at the same time. Selection by fitness is merely a single additional laboratory step. Of course, laboratory steps are slow compared to computer cycle times, but DNA can represent populations so large that its computational power can compare favorably with supercomputers [37].

References

- [1] J. Ackermann, B. Wlotzka, and J. S. McCaskill. *In vitro* DNA-based predator-prey system with oscillatory kinetics. *Bulletin of Mathematical Biology*, 60(2):329–354, March, 1999.
- [2] Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, November 1994.
- [3] Leonard M. Adleman. Computing with DNA. *Scientific American*, 279(2):54–61, August 1998.
- [4] J. J. Bull and C. M. Pease. Why is the polymerase chain reaction resistant to *in vitro* evolution? *Journal of Molecular Evolution*, 14(6):845–850, December, 1998.
- [5] R. Craig Cadwell and Gerald F. Joyce. Randomization of genes by PCR mutagenesis. *PCR Methods and Applications*, 2:28–33, 1992.
- [6] R. Craig Cadwell and Gerald F. Joyce. Mutagenic PCR. In Carl W. Dieffenbach and Gabriela S. Dveksler, editors, *RCP Primer: A Laboratory Manual*. Cold Spring Harbor Laboratory Press, Plainview, New York, 1995.
- [7] Junghuei Chen, Eugene Antipov, Bertrand Lemieux, Walter Cedeño, and David Harlan Wood. DNA computing implementing genetic algorithms. In Laura Landweber, Erik Winfree, Richard Lipton, and Stephan Freeland, editors, *Preliminary Proceedings DIMACS Workshop on Evolution as Computation*, pages 39–49, DIMACS, Piscataway NJ, January 1999. Available on request from DIMACS. Paper found at URL: <http://www.cis.udel.edu/~wood/papers/DIMACS99.ps>.
- [8] R. Cheynier, S. Gratton, J. P. Vartanian, A. Meyerhans, and S. WainHobson. G→A hypermutation does not result from polymerase chain reaction. *AIDS Research and Human Retroviruses*, 13(12):985–986, August 10, 1997.
- [9] J. C. Cox, P. Rudolph, and A. D. Ellington. Automated RNA selection. *Biotechnology Progress*, 14(6):845–850, December, 1998.
- [10] J. H. M. Dassen. A bibliography of molecular computation and splicing systems. HTML source: <http://www.wi.LeidenUniv.nl/jdassen/dna.html>, BibTeX source: <http://www.wi.LeidenUniv.nl/jdassen/dna.bib>. This bibliography is also hooked into <http://liinwww.ira.uka.de/bibliography/>, The Collection of Computer Science Bibliographies.

- [11] Alan Dove. From bits to bases: Computing with DNA. *Nature Biotechnology*, 16(9):830–832, September 1998.
- [12] K. A. Eckert and T. A. Kunkel. High fidelity DNA synthesis by the *Thermus aquaticus* DNA polymerase chain reaction. *Nucleic Acids Research*, 18:3739–3744, 1990.
- [13] A. D. Ellington, M. P. Robertson, and J. Bull. In vitro evolution - Ribozymes in wonderland. *Science*, 276(5312):546–547, April 25, 1997.
- [14] C. V. Forst. Molecular evolution. *Journal of Biotechnology*, 276(5312):546–547, April 25, 1997.
- [15] Tino Gramß, Stephan Bornholdt, Michael Gramß, Melanie Mitchell, and Thomas Pellizzari. *Non-Standard Computation*. Wiley-VCH, Weinheim, 1998.
- [16] Rachel Green, Andrew D. Ellington, David P. Bartel, and Jack W. Szostak. In vitro genetic analysis: Selection and amplification of rare functional nucleic acids. *Methods*, 2:75–86, 1991.
- [17] Alexander J. Hartemink and David K. Gifford. Thermodynamic simulation of deoxyoligonucleotide hybridization for DNA computation. In Harvey Rubin and David Harlan Wood, editors, *Preliminary Proceedings of the Third Annual Workshop on DNA Based Computers, held at the University of Pennsylvania, June 23-25, 1997*, pages 15–25. University of Pennsylvania, 1997.
- [18] L. S. Lerman, K. Silverstein, and E. Grinfeld. Searching for gene defects by denaturing gradient gel electrophoresis. *Trends in Biochemical Sciences*, 172(3):89–93, 1992.
- [19] M. M. Ling and B. H. Robinson. Approaches to DNA mutagenesis: An overview. *Analytical Biochemistry*, 254(2):157–178, December 15, 1997.
- [20] J. L. LinGoerke, D. J. Robbins, and J. D. Burczak. PCR-based random mutagenesis using manganese and reduced dNTP concentration. *Biotechniques*, 23(3):407–, September 1997.
- [21] Carlo C. Maley. DNA computation: Theory, practice, and prospects. *Evolutionary Computation*, 6(3):201–229, 1998.
- [22] Robert Pool. Forget silicon, try DNA. *New Scientist*, 151(2038):26–31, July 13, 1996.
- [23] D. L. Robertson and F. G. Joyce. Selection in vitro of an RNA enzyme that specifically cleaves single-stranded DNA. *Nature*, 344(6265):467–468, March 29, 1990.
- [24] A. Skandalis, L. P. Encell, and L. A. Loeb. Creating novel enzymes by applied molecular evolution. *Chemistry and Biology*, 4(12):889–898, December 1997.
- [25] Gerhard Steger. Thermal denaturation of double-stranded nucleic acids: Prediction of temperatures critical for gradient gel electrophoresis and polymerase chain reaction. *Nucleic Acids Research*, 22(14):2760–2768, July 25, 1994. On line server is found at the URL: <http://www.biophys.uni-duesseldorf.de/POLAND/poland.html>.
- [26] Willem P. C. Stemmer. DNA shuffling by random fragmentation and reassembly: In vitro recombination for molecular evolution. *Proceedings of the National Academy of Science, U.S.A.*, 91:389–391, 1994.
- [27] Willem P. C. Stemmer. Rapid evolution of a protein by DNA shuffling. *Nature*, 370:389–391, 1994.
- [28] Willem P. C. Stemmer. The evolution of molecular computation. *Science*, 270:1510–1510, December 1, 1995.
- [29] Willem P. C. Stemmer. Searching sequence space. *Bio/Technology*, 13:549–553, 1995.
- [30] Willem P. C. Stemmer. Sexual PCR and assembly PCR. In Robert M. Meyers, editor, *The Encyclopedia of Molecular Biology and Molecular Medicine*, volume 5, pages 447–457. VCH, New York, 1996.
- [31] Willem P. C. Stemmer, Andreas Cramer, Kim D. Ha, Thomas M. Brennan, and Herbert L. Heyneker. Single-step assembly of a gene and entire plasmid from large numbers of oligodeoxyribonucleotides. *Gene*, 164(1):49–53, 1995.
- [32] Jack W. Szostak. In vitro genetics. *Trends in Biochemical Sciences*, 172(3):89–93, 1992.
- [33] H. J. Thiesen and C. Bach. Target detection assay (TDA) — A versatile procedure to determine DNA-binding sites as demonstrated on SP1 protein. *Nucleic Acids Research*, 18(11):3203–3209, 1990.
- [34] C. Tuerk and L. Gold. Systematic evolution of ligands by exponential enrichment — RNA ligands to bacteriophage-T4 DNA-polymerase. *Science*, 249(4968):505–510, August 3, 1990.
- [35] J. P. Vartanian, M. Henry, and S. WainHobson. Hypermutagenic PCR involving all four transitions and a sizeable proportion of transversions. *Nucleic Acids Research*, 24(14):2627–2631, July 15, 1996.
- [36] Roy Williams. Data powers of ten. Web page at <http://www.ccsf.caltech.edu/~roy/dataquan>.

- [37] David Harlan Wood, Junghuei Chen, Eugene Antipov, Walter Cedeño, and Bertrand Lemieux. A DNA implementation of the Max 1s problem. In Wolfgang Banzhaf, A. E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference, July 13-17, 1999, Orlando, Florida USA.*, San Francisco, 1999. Morgan Kaufman.
- [38] M. Zaccolo, D. M. Williams, D. M. Brown, and E. Gherardi. An approach to random mutagenesis of DNA using mixtures of triphosphate derivatives of nucleoside analogues. *Journal of Molecular Biology*, 255(4):589–603, February 2, 1996.

(J. CHEN) DEPT. OF CHEMISTRY & BIOCHEMISTRY, UNIVERSITY OF DELAWARE, NEWARK, DE
E-mail address: `junghuei@udel.edu`
URL: `http://www.udel.edu/chem/chen/chen.html`

(E. ANTIPOV) DEPT. OF CHEMICAL ENGINEERING, UNIVERSITY OF DELAWARE, NEWARK, DE
E-mail address: `ultra@udel.edu`

(B. LEMIEUX) DEPT. OF PLANT & SOIL SCIENCES, UNIVERSITY OF DELAWARE, NEWARK, DE
E-mail address: `blemieux@udel.edu`
URL: `http://bertrand.ags.udel.edu`

(W. CEDEÑO) HEWLETT-PACKARD COMPANY, 2850 CENTERVILLE ROAD, WILMINGTON, DE
E-mail address: `wcedeno@lf.hp.com`
URL: `http://www.personal.psu.edu/users/w/x/wxc28`

(D. WOOD) DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF DELAWARE, NEWARK, DE
E-mail address: `wood@cis.udel.edu`
URL: `http://www.cis.udel.edu/~wood`