Computationally Inspired Biotechnologies: Improved DNA Synthesis and Associative Search Using Error-Correcting Codes and Vector-Quantization ⋆

John H. Reif ⋆⋆ and Thomas H. LaBean

Department of Computer Science, Duke University

Abstract. The main theme of this paper is to take inspiration from methods used in computer science and related disciplines, and to apply these to develop improved biotechnology. In particular, our proposed improvements are made by adapting various information theoretic coding techniques which originate in computational and information processing disciplines, but which we re-tailor to work in the biotechnology context.
(a) We apply Error-Correcting Codes, developed to correct transmission errors in electronic media, to decrease (in certain contexts, optimally) error rates in optically-addressed DNA synthesis (e.g., of DNA chips).
(b) We apply Vector-Quantization (VQ) Coding techniques (which were previously used to cluster, quantize, and compress data such as speech and images) to improve I/O rates (in certain contexts, optimally) for transformation of electronic data to and from DNA with bounded error.
(c) We also apply VQ Coding techniques, some of which hierarchically cluster the data, to improve associative search in DNA databases by reducing the problem to that of exact affinity separation. These improvements in biotechnology appear to have some general applicability beyond biomolecular computing.
As a motivating example, this paper improves biotechnology methods to do associative search in DNA databases. Baum [B95] previously proposed the use of biotechnology affinity methods (DNA annealing) to do massively parallel associative search in large databases encoded as DNA strands, but many remaining issues were not developed. Using in part our improved biotechnology techniques based on Error-Correction and VQ Coding, we develop detailed procedures for the following tasks:
(i) The database may initially be in conventional (electronic, magnetic, or optical) media, rather than the form of DNA strands. For input and output (I/O) to and from conventional media, we apply DNA chip technology improved by Error-Correction and VQ Coding methods for error-correction and compression.

⋆ A postscript version of this paper is at URL http://www.cs.duke.edu/~reif/paper/Error-Restore/Error-Restore.ps .

⋆⋆ Surface address: Department of Computer Science, Duke University, Box 90129, Durham, NC 27708-0129. E-mail: reif@cs.duke.edu. Supported by Grants NSF/DARPA CCR-9725021, CCR-96-33567, NSF IRI- 9619647 and EIA-0086015, ARO contract DAAH-04-96-1-0448, and ONR contract N00014-99-1-0406.

(ii) The query may not be an exact match or even partial match with any data in the database, but since DNA annealing affinity methods work best for these cases, we apply various VQ Coding methods for refining the associative search to exact matches.
(iii) We also briefly discuss how to extend associative search queries in DNA databases to more sophisticated hybrid queries that include also Boolean formula conditionals with a bounded number of Boolean variables, by combining our methods for DNA associative search with known BMC methods for solving small size SAT problems. For example, these extended queries could be executed on natural DNA strands (e.g., from blood or other body tissues) which are appended with DNA words encoding binary information about each strand, and the appended information could consist of the social security number of the person whose DNA was sampled, cell type, the date, further medical data, etc.

1 Introduction

1.1 Recombinant DNA Technology

DNA as a Storage Media. Recall that DNA is a linear molecule composed of 4 types of nucleotides, and thus provides a base 4 data encoding. DNA is an appealing media for data storage due to the very large amounts of data that can be stored in compact volume. It vastly exceeds the storage capacities of conventional electronic, magnetic, or even optical media. A gram of DNA contains about 10^{21} DNA bases, or about 10^8 terabytes. Hence, a few tens of grams of DNA may have the potential of storing all the human-made data currently stored in the world. DNA is about 10^8 times more compact than other storage media currently being used is. Most recombinant DNA techniques can be applied at concentrations of about 5 grams of DNA per liter of water.

Recombinant DNA Technology. Biotechnological methods, which are collectively known as recombinant DNA technology, have been developed for a wide class of operations on DNA strands. These operations include site-specific edits and splicing operations. In the DNA annealing operation, two single strands of DNA (with opposite $3'$ to $5'$ orientation) combine into a doubly stranded DNA if the DNA bases of these sequence are complementary (or nearly complementary) to each other. DNA separation techniques [KG97] make use of annealing [HG97] to separate out DNA strands that contain particular subsequences; typically one set of DNA strands is surface attached (e.g. to streptavidin-coated paramagnetic beads) and the affinity separation operates on another set of single stranded DNA which anneal to the affixed DNA. PCR [B94, R94] is a recombinant DNA operation that uses DNA annealing to amplifying the frequency of those DNA strands that have a particular chosen sequence.

1.2 Computationally Inspired Biotechnologies

Adleman has suggested the term *Computationally Inspired Molecular Technology* to describe molecular methods that are inspired by computational methods. While many of the more general applications to Molecular Technology are outside the scope of this paper, it is nevertheless a visionary concept which seems destined to have major impact

This paper has a narrower focus to biotechnology. The major theme of this paper is that we may provide improvements to biotechnology using methodologies similar to those developed by computer scientists. A field we term *Computationally Inspired Biotechnology* encompasses biotechnology methods that are inspired by computational methods. For example, the inventor of PCR has stated he was inspired by the technique of recursive programming in his discovery of PCR, which operates by a recursive doubling of concentrations of selected DNA strands.

As we shall see, our improvements in biotechnology will be made by adapting error-resilient and optimum rate techniques, which originate in computational disciplines.

A Motivating Example: Massively Parallel I/O using DNA Chips. For example, let us consider the conversion of a database from conventional electronic media to a "wet" database of DNA strands in solution or on solid support. To do these transformations of the database and queries, we investigate the use of a promising new biotechnology; namely that of DNA chips [FRP+91,DDSPL+ 93,PSS+94, BKH96, CYH+96], which provide the ability of highly parallel input/output over 2D surfaces. By use of photosensitive DNA-on-a-chip technology, 2D optical input is converted to DNA strands encoding the input data. If the database was initially in conventional electronic form, it can easily be displayed at a very high rate as a series of images in 2D optical form, and in principle parallel arrays of DNA chips can be used to synthesize strands of DNA each encoding the database elements. Furthermore, DNA chips can also be used for 2D optical output: using hybridization at the sites with fluorescent labeled DNA, the output can be read as a 2D image. (See Figure 1.)

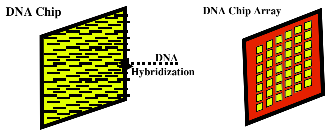

Fig. 1. DNA Chips and DNA Chip Arrays.

Each DNA chip can be optically addressed at up to 10^5 sites (and potentially many millions of sites in the immediate future), and each such chip is small enough so that arrays of up to a few thousand chips can be placed on a 2D array compact enough so all chips can be addressed by a single optical system. Hence there is the potential of parallel synthesis of DNA at 10^8 sites or more (and potentially many billions of sites in the immediate future). Thus, this massively parallel DNA synthesis for input (and affinity detection of DNA strands for output) has a potential for achieving a rate of input/output to convention optical/electronic media in the order of gigabit rates or more. However, this all is without consideration of error rates, which are considerable, and represent the bulk of the technical challenge in this case. The most common errors in optically addressed synthesis of DNA are premature truncations of the growing strand and base deletions, although insertions and substitutions are also possible. (There are also a variety of other sources for further errors, such as differential sequence-dependant binding and secondary structure of the DNA strands.) The error rate in optically addressed DNA synthesis methods used for DNA chips is roughly 4% to 8% per base [MBD+97]. This corresponds to an expected error in every 12 to 25 base pairs.

The commercial chips such as Affymetrix utilize only a fraction of the 10^5 or so sites that are in principle optically addressable; the current maximum is about $42,000$ sites, but the typical DNA chip uses only about $7,000$ sites (see Table 1 of [LFGL99]). Due to the proprietary nature of the Affymetrix technology, it is not entirely clear what their synthesis error rates are for each type for possible error. For the technology at its current scale of just a few tens of thousands of sites, and with most strands under 20 base pairs, the synthesis error rate is not likely to be the most dominant limiting factor, and it is only one of a number of other factors that can impact the technology at the current scale. Nevertheless, methods for decreasing error rates due to optically addressed base synthesis might well impact future scalability of DNA chip technology by facilitating increased numbers of addressable sites and increased strand length.

1.3 Applications to Biomolecular Computing. Biomolecular Computing (BMC) makes use of biotechnology to do computation. Given the extreme compactness of DNA and the ability via recombinant DNA technology to execute operations on vast numbers of DNA strands in massively parallel fashion, BMC has impressive potential for molecular-scale computation. We consider two of the greatest challenges for BMC:
(a) Error-Resilient, High Rate I/O. Although Error-resilient techniques for BMC have been developed in [BL95a, CW97, DMGFS98, DMRGF+97, and DHS97], a key issue we consider is the transformation of inputs, originally in conventional electronic media, into sequences of DNA. Assuming the inputs are static, the conversion needs only to be done one time, but this still may be a nontrivial task. The application of DNA chips for I/O in Biomolecular Computing may be limited by their relatively high error rates. In this use of DNA chips, each of the DNA strands synthesized may be quite long (likely well over 25 bases per strand), so as to transmit significant amounts of information, and then the majority of such strands can be expected to have at least one synthesis error. To address the synthesis error rate, we will apply Error-Correcting Coding methods.
(b) Scalable BMC Applications. Another key near term challenge [R96] in the field of BMC is to find applications of this technology that have the possibility of commercial utility in the near term, say in the next five years, and furthermore their resource requirements (number of recombinant DNA steps, volume of test tubes, etc.) should scale well so that future large scale demonstrations will be feasible. To date, there have been a number of BMC demonstrations and proposals for the solution of small size combinatorial search problems using separation techniques [A94, L95, ARRW96, BDL95, RWBCG+96] and surfaced based chemistry [CRFCC+96, LGCCL+96, BCGT96, CCCFF+97, LTCSC97, LFW+98, WQF+98]. But these applications are not scalable, since the volume grows exponentially with the problem size. Recently proposed applications of BMC that appear to be scalable include methods for hiding DNA [CRB99] and for encrypting DNA [GLR99], doing neural network learning [MYP98], and possibly certain other massively parallel computations [R95, GR98a], but their commercial utility may not be in the immediate future.
DNA Associative Search: A Scalable BMC Application. This paper takes associative search as our motivating example of a scalable BMC application, and provides solutions to the challenges listed above.

1.5 Organization of this Paper

In Section 1 we introduced recombinant DNA technology and the concept of computationally inspired biotechnologies In Sections 2 and 3 we present some coding methods used in computer science, namely Error-Correction coding and also Vector Quantization (VQ), that can be applied to improve biotechnology methods for error-resilient I/O and also optimum rate I/O btween DNA databases and conventional media. In Section 4 we introduce our motivating example application: associative search and describe how VQ can be used to improve DNA associative search by to refing the associative search to exact matches. In Section 5 we briefly present methods for extending associative search queries to sophisticated hybrid queries that include also Boolean formula conditionals. In Section 6 we conclude the paper.

2 Error-Correction Methods From CS Adapted to Biotechnology

This section proposes experiment methods for repairing faulty oligonucleotides contained within surface-bound probe arrays. We propose the use of Error-Correction DNA strands specifically designed to bind both error-containing and error-free probes.

(Error-Correction strands can also act as templates for primer extension reactions, which will append error-free code words onto the 5' ends of all probes on the chip.) Our Error-Correction strands are composed of two segments: an error-containing portion (the suffix) designed to be complementary to the immobilized probe sequences and biased in its synthesis to contain similar types and quantities of errors as the faulty-probe sequence; and an error-free portion (the prefix) which will provide a error-free probe (or can be used as a template for primer extension to create a error-free probe). After hybridization with the Error-correction strands, the resulting duplex probes contain single-stranded overhangs with the error-free portion. (There is also evidence [BSS+94] that duplex probes containing single-stranded overhangs are substantially less error-prone than simple single-stranded probes, due to stacking interactions that provide increased stringency.) The design of these Error-Correction DNA strands is provided by information theoretic error-correction methods.

2.1 Known Error-Correction Methods We will take inspiration from information theoretic error-correction methods (see Shannon [S48,S49], Hamming [H50], Berlekamp [B68], Pless [P82], Lint [L71]) used in computer science, with the goal of developing similar methods for various biotechnology applications.

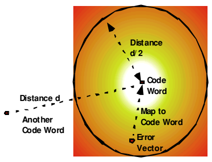

Fig. 2. Error-Correction by Mapping to Nearest Code Word.

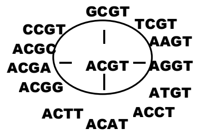

Fig. 3. DNA sequences of generalized Hamming distance 1 from ACGT.

These error-correction methods make use of a set C of code words. When code words are altered by errors (during transmission or storage), they can be mapped back into the original set of error-free code words by an *Error-Correction* procedure. (See Figure 2.) In one of the simplest of these coding schemes, to encode each $N-$vector X whose elements range over $0, 1, ..., b-1$, we use a code word $C(X)$ consisting of a $N'-$vector whose elements range over the same domain. In general, N' is larger than N; that is we are *increasing the length of the encoding to gain error-resiliency*. The distance metric is defined so that the *generalized Hamming distance* between two vectors is the number of elements where they differ. The code words are chosen so that there is at least distance $d = N' - N$ between each pair of code words. (See Figure 3.)

The *Maximum Likelihood Error-Correction* procedure simply maps each N'-vector to a code word that is closest in distance. Note that a restoration error does not occur as long as a vector is perturbed by at most d' errors, where d' is the floor of (the largest integer less than or equal to) d/2. The restoration error probability r is the likelihood that this Error-Correction procedure results in an error.

In the Boolean case (where the elements of vectors are 0 or 1), the error model (known as a *binary symmetric channel*) makes the assumption that each bit has a uniform, independent probability p of being flipped. In this case, the Hamming code [H50] provides an asymptotically optimum restoration error probability for any choice of the parameters N, $d = N' - N$, and p. In the more general case considered here, where the vector elements range over $0, 1, ..., b-1$, the error model provides an independent probability p of any element x of a vector being replaced by an element of $0, 1, ...b-1-x$ (note that the choice of the replacement element due to an error does not concern us, since that does not change the generalized Hamming distance).

There are known designs for sets of code words (e.g., see the Reed-Muller and BCH codes found in the texts [P82, L71]) which provide an asymptotically optimum restoration error probability $r = r(N, d, p, b)$ for any choice of the parameters $N, d, p,$ and b. These codes have the property that, for a fixed p, there is a constant $c > 0$ such that the restoration error probability r can be made an inverse exponential function 2^{-cN} of N, and simultaneously, $d = N' - N$ can be made to asymptotically approach 0 as N grows[1]. Our goal now is to take inspiration from this error-correction techniques and apply the concepts to improved biotechnology.

2.2 Applying Error-Correction Methods To Biotechnology

Recall that a *multiset* is a collection of items with possible repeats. Let the redundancy of a multiset S be the minimum number of repeats of any element of S. In the following, we assume multisets of DNA strands with very large redundancy, say at least 10^3. Let us consider the case where we attempt chemical synthesis of a multiset S of single stranded DNA strands (the strands might be on a DNA chip or on other solid support), where each of the strands has the same total number of bases. We assume that the chemical synthesis has errors. Let us suppose each base is synthesized within each of these strands with a uniform, independent probability p of a deletion error. We will approximately model these base deletions by replacements with other distinct bases. (To justify this, note that in the annealing process between two near-complementary strands s, s', a base deletion in strand s generally would result in a local DNA base bulge in strand s'. (See again Figure 3. Short stretches of double-stranded DNA are depicted showing: a) exact Watson-Crick(WC) complementary matching; b) a mismatch (T-T) imbedded within a WC match region; and c) a WC match region surrounding a bulged base (T). The bulged base can be described as a deletion from the left-hand strand or an insertion into the right-hand strand.) The effect of this base bulge requires a very complicated energetic model; for simplicity, we approximate the effect of this base bulge to first order by a same length sequence of base mismatches, although these are not strictly energetically equivalent.)

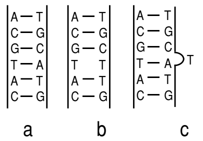

Fig. 4. Exact and Inexact Hybridization.

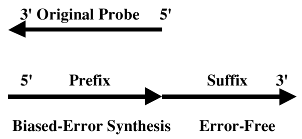

Fig. 5. EC Strand Prefix and Suffix

Our Model for Synthesis Errors. So these synthesis errors with independent base deletions will be *approximated by an error model with a uniform, independent probability p of base replacement (without deletion error).* (See Figure 4, which gives a 2D Projection of a local region in sequence space. Neighboring sequences are shown for a central tetramer (ACGT) with substitutions in the first position to the north, second to the east, third position south, and fourth west. Truncations, deletions and insertions are not shown.) This coincides with the uniform, independent replacement error model described in the above Subsection 2.1. Let $ERROR_p(S)$ be the multiset resulting from the attempted synthesis (with very high redundancy) of each of the single

[1] Similar results also hold for much more general channel error models, including discrete memoryless stationary processes. There are also more sophisticated coding methods known, for more complex models with non-uniform errors and also non-independent errors. For example, "burst" errors correspond to correlations of errors among consecutive elements, which necessitate modified methods for codings and the Error-Correction process.

stranded DNA in multiset S with a uniform, independent probability p of base error. Now our challenge is to perform a Error-Correction procedure on the strands by purely biotechnology (i.e., recombinant DNA operations) means. In context of our problem, the coding theory parameters N, d, p, and b are set so:

- N is the number of bases in each of the strands of S,
- $b = 4$ is the element domain size, since there are 4 DNA bases,
- p is given by the error model, and
- d is the minimum distance between each of the code words, and is a parameter that can be set to adjust the restoration error probability.

We choose a known design [P82, L71] for a set C of code words which provides an asymptotically optimum restoration error probability $r = r(N, d, p, 4)$ for choice of the parameters N, d, p. Our goal is to synthesize a multiset S of single stranded DNA, where each strand has N bases. For each strand s in S, we will define a *code word strand* $C(s)$ consisting of a single stranded DNA which gives the code word for the single stranded DNA s. Let $C(S)$ be the multiset of strands which are code word strands of the strands in S. We assume the encoding provides a restoration error probability $r = r(N, d, p, 4)$ for choice of the parameters N, d, p. There are two possible cases we consider:
(a) If S has the property that for each pair of strands in S differ by at least d bases (for example, we might expect this to be the case when S consists of a set of DNA words used in a DNA computation, and these DNA words were chosen so that all distinct pairs of these DNA words have low hybridization affinities), then we choose our code words $C(S)$ to be a set of DNA words with a 1 to 1 mapping C from S to $C(S)$, such that the elements of $C(S)$ all have length $N' = N$, and each s in S has low annealing affinity with the complement of $C(s)$ (note that this latter requirement implies that we can not set $C(s) = s$).
(b) Otherwise, we choose a known asymptotically optimum design [P82, L71] for the set C of code words, where for each s in S, code word strand $C(s)$ has $N' = N + d$ bases and moreover, s has low annealing affinity with the complement of $C(s)$. Since N' is larger than N, we are in this case *increasing the length of the encoding to gain error-resiliency.*
In either case, we now execute the following steps:
[0] Initialization. We first construct separately (by methods described in Subsection 2.3) a multiset of single stranded DNA strands, which we will call *Error-Correction (EC) strands.* (See Figure 5.) Each EC strand consists (in the $5'$ to $3'$ direction) of the prefix portion of the strand followed by the suffix portion of the strand. The prefix portion of the EC strand will be the result of synthesis of the complement of code word $C(s)$, with a uniform, independent probability p of DNA base synthesis errors. (Note that the synthesis of the prefix portion of the EC strand has the same error model as the synthesis of the code word $C(s)$.) The suffix portion of each EC strand will be the result NEW(s) of synthesis of s with a much lower error rate: with a probability q (where $q << p$) of even a single DNA base synthesis error on that suffix.
[**1**] Rather than directly attempt chemical synthesis of multiset S, where each strand has N bases, we instead do chemical synthesis with this error model of the multiset $C(S)$ code word strands, resulting in a synthesized multiset $ERROR_p(C(S))$. Without loss of generality, we assume that these synthesized strands of C(S) run from their attached $3'$ end to their unattached $5'$ end (however, note that we use the standard convention in our figures with arrows directed on the strands from the $5'$ end to the $3'$ end).
[**2**] Then we combine the EC strands with the DNA strands synthesized with this error model, and allow for hybridization. The hybridization products include doubly

stranded DNA complexes with elements of $ERROR_p(S)$ hybridized with the corresponding prefix of the EC strands, with single stranded overhangs consisting of the suffix portion of the EC strands. (See Figure 6.)

In summary, we begin with the error-prone synthesis of a code word $C(s)$, and this results in a ssDNA overhang NEW(s). Let S^* be the multiset of these single stranded overhangs NEW(s), for each s in S. We say S^* approximates S with *fidelity* f if f lower bounds the probability that NEW$(s) = s$, for each s in S.

A precise statement of our result. We now show:

Theorem 1 If we employ a error-correction code with restoration error probability $r = r(N, d, p, 4)$, then S^* approximates S with fidelity $(1-q)(1-r)^2 \geq 1 - 2r - q$.

Proof: Suppose we attempt synthesis of a strand s in S in this error model. With likelihood $1-r$, this yields a strand s' with at most d errors, where d is the floor of $(N' - N)/2$. Further suppose that the prefix portion of some EC strand is complementary to s', and anneals to a strand s'. The likelihood that that EC strand was perturbed by less than d errors, also has likelihood $1 - r$, and conditional on this event, the further likelihood that the suffix NEW(s) of that EC strand is s, is $1 - q$. Hence, the resulting single stranded overhang, consisting of the suffix portion NEW(s) of that EC strand, is the strand s, with likelihood given by $(1-q)(1-r)^2$, which is at least $1 - 2r - q$. **QED**

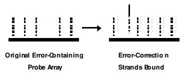

Fig. 6. Steps used to Error-Correct Synthesized DNA Strands, Resulting in Overhangs

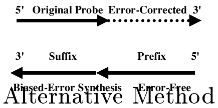

Fig. 7. An Alternative Method for Extension to Error-Free Probe Using Primer Extension on the EC' Strand

Alternatively, we may instead wish to extend the original strands (say in the $5'$ to $3'$ direction) of S, so that the strand extending each of the original strands corresponds to its error-free codeword. In this case, we need to assume that these synthesized strands instead run from an attached $5'$ end to an unattached $3'$ end. We also need to slightly redefine the EC strands (see step [1']), and apply a well-known recombinant DNA operation known as primer-extension (e.g., see its previous use by [OGB97] for DNA computation). In this case, we define the code words so that the complement of s (rather than s, as in the case above) has low annealing affinity with the complement of $C(s)$.

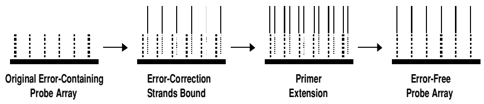

Fig. 8. Steps used to Error-Correct Synthesized DNA Strands, Resulting in Strand Extension

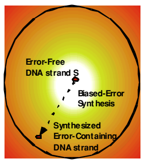

Fig. 9. Biased-Error DNA Synthesis

We now execute the following modified steps (see Figure 8):

[0'] We first construct separately (again, by methods described in Subsection 2.3) a multiset EC' of single stranded DNA strands, similar to the previously defined EC strands, except the suffix portion at the $5'$ end of the EC' strand is the complement of a possible result of synthesis of code word $C(s)$ (with the same synthesis error model), but the prefix portion at the $3'$ end of each EC strand is the *complement* to a strand s in S.

[1'] This step is the same as step [1] above.

[2'] This step is the same as step [2] above, except we anneal using the multiset EC' instead of multiset EC. This results in the hybridization products that include doubly

stranded DNA complexes with elements of $ERROR_p(S)$ hybridized with the corresponding suffix of the EC strands, with single stranded overhangs consisting of the prefix portion of the EC' strands.
[3'] We then apply primer-extension. These single stranded overhangs provide templates used in the primer-extension procedure that provides extension of the strands of $ERROR_p(S)$ with the restored codewords, as intended.
Since this modified method is identical to the original one, except that complements are synthesized, the fidelity is the same as stated in Theorem 1, and in particular, the fidelity that these single stranded overhangs approximate multiset S is again lower bounded by $1 - 2r - q$.

2.3 Synthesizing the EC Strands

The remaining problem is to synthesize the EC strands as defined above in Section 2.2 (synthesis of the EC' strands is similar). This turns out to be the most interesting and technically demanding of the steps. (See Figure 9.) We consider a variety of methods:
(a) Direct synthesis and purification of Error-Correction Strands. Given detailed knowledge of the types and rates of sequence errors observed for oligonucleotides synthesized by light-directed chemistry on 2D chips, one could imagine constructing, by high fidelity solid-phase synthesis, a unique repair strand for each of the possible error sequences which then maps the error back to the intended probe sequence, as described above. This could be prohibitively expensive due to the large number of strands required. Nevertheless, this could turn out to be the method of choice in the case of small-scale applications.
(b) Biased-Error Chemical Synthesis of Error-Correction Strands. Here combinatorial synthesis is used for reducing the total number of separate syntheses required by simultaneously producing EC strands for a large number of possible flawed probes. Here the prefix portions of the EC strands are synthesized with errors in a manner mimicking the error model for the original probe strands, but where the suffix portion of the EC strands are synthesized with the lowest possible error rate. We propose the use of high-sequence-fidelity oligonucleotides produced by automated synthesis for the repair of errors incorporated during the production of probes by light-directed synthesis of addressable probe arrays. We will go from an expected error rate of $4 - 8\%$ to something less than 1%. The proposed method would require a total number of syntheses and purifications equal to the intended diversity of the original probe array; while this may seem a daunting task, the payoff in error reduction and the corresponding increase in chip efficiency may be required for accurate I/O on critical computational or associative search applications.

Current protocols for chemical synthesis of oligonucleotides have been optimized to minimize sequence errors. The coupling efficiency for light-directed synthesis of DNA on planar solids are various estimated to be between 92% and 96% [MBD97, MH98] per cycle, compared to $98 - 99\%$ for conventional synthesis using acid cleavage of methoxytrityl protecting groups on the same planar supports [MH98]. Higher coupling yields ($> 99\%$) are routinely observed for automated, solid-phase oligonucleotide synthesis. The most commonly used method is the phosphite triester (phosphoramidite) procedure as modified by Beaucage and Caruthers [BC81]. The $3'$-hydroxyl group of the first nucleotide is immobilized on a solid support (either controlled pore glass or a polystyrene-copolymer). The chain grows by the nucleophilic attack of the $5'$-hydroxyl of the immobilized oligo on an activated $3'$-phosphoramidite moiety of a $5'$-protected building block. Reactive chemical groups on the bases are protected with various organic groups which are removed by treatment with ammonium hydroxide following the final coupling step. The reactive phosphates of the DNA phosphodiester backbone are protected throughout synthesis with -cyanoethyl which is also removed in the final ammonia deprotection step.

A quick review of the steps involved in chemical DNA synthesis will assist our discussion. Each cycle of chain elongation requires four steps: 1) deprotection, 2) coupling, 3) capping, 4) oxidation. *Deprotection* involves the removal of the dimethoxytrityl chemical-protecting group from the $5'$-hydroxyl of the previous nucleotide. Trichloroacetic acid ($1 - 3\%$ w/v) is used in dichloromethane; the reaction requires less than one minute. *Coupling* provides for addition of the next nucleotide to the growing polymer. Excess soluble protected nucleosides and coupling reagent drive the reaction nearly to completion. Phosphoramidites can not react directly with a free $5'$-hydroxyl and must first be activated by a weak base like tetrazole. Tetrazole protonates the dialkyamino group of the phosphoramidite moiety and become a nucleophile, generating a very reactive tetrazolophosphane intermediate. Coupling reaction with these compounds is very fast (less than 2 minutes) and nearly quantitative. *Capping* is used to eliminate free $5'$-hydroxyls which failed to react in the coupling step, thus capping decreases the frequency of sequence deletion products in the final oligo, converting them instead into truncations. Products of the capping reaction are terminated waste products and are no longer active during the remainder of synthesis. Acetic anhydride/N-methylimidazole is generally used as the capping reagent. *Oxidation* of the newly formed phosphite internucleotide linkage is unstable and must be oxidized to the more stable phosphate before chain extension can proceed. Iodine in tetrahydrofuran is a mild oxidizer with water as the oxygen donor. Reaction occurs within ≈ 30 seconds in very high yield.

Synthesis errors can occur at each of the steps in each cycle of the synthesis. The rates of different varieties of errors can be tuned in order to prepare oligos complementary to the probe libraries and containing, as an ensemble average, errors which match those spontaneously created in the light-directed chip synthesis. In order to achieve this tuning, very good data will be needed as to the types and frequencies of errors within the immobilized probes.

- **Tuning the rate of truncation errors:** Truncations occur when the deprotection step succeeds, the coupling step fails, and the capping step succeeds. Truncations on the chip can be converted into deletions if the capping step is eliminated. Truncations in the EC strands can be increased to match the chip rate by decreasing the efficiency of the coupling reaction (decreasing reaction time or by decreasing the concentrations of either the phosphoramidite or the activator (tetrazole/water). No changes are needed in the other steps.
- **Tuning the rate of deletion errors:** Deletion errors can occur if the deprotection step fails, or when it succeeds but the coupling and capping steps simultaneously fail. The easiest way to increase the rate of deletions on the EC strands is to decrease the efficiency of the deprotection step, either by decreasing the acid concentration or the reaction time or possibly both together.
- **Tuning the rate of substitution errors:** Substitution errors can be most easily incorporated into EC strands by spiking the nucleoside phosphoramidite stock solutions with an appropriate amount of contaminating phosphoramidite from the other nucleosides. For example, if a substitution rate of 1observed for sites meant to be base G, then the G-monomer can be deliberately contaminated with 1% A-monomer. Reactivities of the phosphoramidites vary slightly with base identity, but at this stage, these differences will be ignored. We must also note that we will be synthesizing the complement of the error-containing strands and so must spike with T-monomer when we expect to basepair with erroneous A, etc.

(c) Other Methods for Generating Diverse Prefixes for EC Strands. The relative simplicity of the biased-error chemical synthesis approach makes it the most appealing of methods for generating diverse prefixes for EC strands. However, for purposes of completeness, it should also be mentioned that there are a number of other possible methods.

Mutagenisis via Polymerase Enzymes. Sequence diversity can be generated during enzymatic polymerization of DNA by the use of polymerase enzymes and conditions which favor the introduction of mutations to the newly synthesized strands. Error rates of DNA polymerase enzymes vary from about one per million bases for high fidelity, proof-reading enzyme, to nearly one per hundred bases for highly mutagenic polymerization. In general the nature of errors may differ between chemical and enzymatic DNA synthesis, so biased-error chemical synthesis would most likely provide EC strands which most closely match the error profiles for probe DNA synthesized on a chip. Other mutagenisis Methods used for more general computations are given in [KG98].

DNA Self Assembly. It is also, in principle, possible to construct EC strands using a self-assembly method [R97, WLW+98, WYS96, LYK+00, LWR99, RLS00, MLR+00] involving a universal base pairing nucleotide like inosine to generate diverse populations of prefixes.

3 Adapting to Biotechnology VQ Methods Used in Computer Science

3.1 VQ Coding Methods Used in Computer Science

We next consider information theoretic Vector Quantization (VQ) Coding methods (see Gray [G90], Gersho, Gallager, and Gray [GGG91]) used in computer science for compressing data (such as speech and images) within bounded error. Again let $V = B^n$ be the set of all possible n-vectors over domain B of consecutive integers, and consider a database of vectors in V. VQ methods (which are also known as source coding and clustering methods) partition the vectors of the database into clusters of vectors, where each cluster is a subset of the database vectors. For each cluster G, the *center vector*, which may not be originally in the database of vectors, is the average of all the vectors of the cluster. The *radius* of a cluster is the maximum distance between any vector of the cluster to the center vector. There are well-known algorithms (Jain, Dubes [JD88]) which cluster the vectors of the database so the cluster radius is minimized and the average number of vectors in each cluster is bounded by a *cluster size* parameter m.

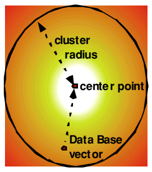

Fig. 10. A Vector Quantization Cluster

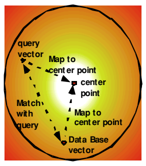

Fig. 11. Mapping the Query Vector to the Center of a Nearby Cluster

3.2 Applying VQ Coding Methods To Increase DNA Chip I/O

The clusters are enumerated and each assigned a *cluster index*, which is an integer that uniquely identifies the cluster. The number of clusters is a multiple $1/m$ of the original number of vectors of the database. Within a given cluster, each vector is approximated by the center point of the cluster and the code for a vector is an index to the cluster the vector is in. (See Figures 10 and 11.) Software for this process of VQ clustering has been developed by Eve Riskin's group at the University of Washington and can be obtained from their FTP site (http://isdl.ee.washington.edu/compression/code/).

Note that in contrast to Error-Correcting codes, the VQ coding induces errors, which are bounded by the choice of the clusters and can be tuned by setting the parameter m. For certain statistical source models for the data, for example memoryless or finite-state stationary processes (see Gray [G90], p 44), the resulting data-rate/distortion of VQ coding has been shown to be asymptotically optimal. However, natural data sources such as speech, images, and natural DNA can not be well modeled as memoryless or finite-state stationary processes (this seems to be related to the fact that speech and image data and also natural DNA [GT94, LY97, NW99] are known to be relatively uncompressible if it is compressed without errors e.g., via LZ compression [CT 91]), and so the performance of VQ coding on these data sources must be judged by empirical testing rather than by precise formulas. Extensive empirical testing of VQ coding (see Gray [G90], Gersho, Gallager, and Gray [GGG91]), has shown it to provide high factors of compression for many types of natural data, for example approximately 20 for speech and 30 for images, without much noticeable degradation.

An immediate application of VQ data clustering techniques is to improve I/O rates for transformation of electronic data to and from DNA with bounded error. Recall each vector of the database assumed to have a unique identification tag. After determining the clusters, their center points need to be transmitted (at $1/m$ the cost of transmitting the entire set of the database), and each vector v of the database is simply represented by a strand consisting of a series of DNA words encoding the unique identification tag for v and also an identification tag for the center point of the cluster that contains v. While for arbitrary databases, the performance (improved I/O rate) of VQ coding can not be precisely predicted by analytic techniques, we have noted that empirical evidence indicates it has excellent performance if the data base consists of speech or image data. The next section discusses how to apply VQ coding methods to refine DNA associative search to exact matches.

4 Application to Associative Search

4.1 Definition of Associative Search. Let $V = B^n$ be the set of all possible $n-$vectors, whose elements range over a set B of consecutive numbers. Given two vectors u, v in V, let distance(u, v)= $|u_1 - v_1| + |u_2 - v_2| + ...|u_n - v_n|$, that is, sum of absolute values of the differences between corresponding elements of the vectors. We will use this distance metric in the context of associative search. The associative search problem assumes a database which is an ordered list of elements of V. In general, the input to an associative search query consists of a query vector in V and a distance bound d. The task is to search the entire database for those vectors (called the distance d near-matches) of the database that are of distance at most d from the query vector. If the distance bound is not specified, then the task is to find a vector (called the closest match) of the database that is of smallest distance from the query vector. (See Figure 12.) Each of the vectors of the database is assumed to have a unique identifying index in the list comprising the database, so the output vectors can be specified by their indices.

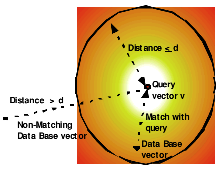

Fig. 12. Associative Search

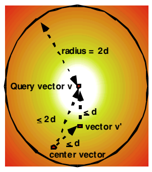

Fig. 13. Possibility Vectors of Distance $2d$ from the Query Vector.

Motivation. For example, the associative search problem arises as a basic task in the image-processing context. The image database is assumed to be preprocessed by a fixed procedure A to form an preprocessed database, which has a list of low level image attributes for each image or sub-image (e.g., $A(I)$ may apply a series of linear filters or perhaps a wavelet transform to the Image I). Given an input image I, we use A to determine the vector $A(I)$. Then an associative search in the preprocessed database provides the closest match to $A(I)$. This provides an index to that image in the image database whose attributes best match that of the input image I. This motivates the development of an ultra-compact storage media (e.g., DNA) that supports highly parallel associative searches within the entire media.

Associative Search in Conventional Storage Media. In conventional highly compact storage (e.g., RAM, magnetic or optical), the time for an associative search though the entire database grows linearly with the size of the database. For example, the time for a sequential associative search through an entire database of 1 terabyte stored on optical disk may take at least a few hours. These searches can be sped up by a factor of P if P independent storage systems are accessed in parallel, but in conventional systems this degree of parallelism P is at most a few hundred. Image databases of size nearly 1000 terabytes (a terabyte = 10^{12} bytes) are being constructed by NASA and other federal agencies for space science and reconnaissance. The time for an associative search through a database of such size might, even with this degree of parallelism, could take at least a few days or even weeks.

4.2 DNA Annealing as an Associative Search Engine. An application previously proposed by Baum was massively parallel associative search in large databases, who sketched some approaches using known recombinant DNA methods for DNA ligation affinity separation such as such the use of streptavidin-coated paramagnetic beads. PCR also provides a way of doing associative search, since it uses DNA annealing to amplifying the frequency of those DNA strands that have a particular chosen sequence. (PCR is currently used for searching within large biological databases; for example, to fingerprint human DNA.) In the context of the DNA databases we consider, the database might have size $n = 1000$ terabytes.

Scalability. This application appears to be scalable, since (i) before the maximum concentration is reached, the number of recombinant DNA operations required (using PCR) for the search is independent of the size of the database and the database can be stored without a volume expansion, and (ii) after the maximum concentration is reached, the number of recombinant DNA operations required for the search is at most linear in the size of the database and the volume scales linearly.

Use of DNA Word Design in Associative Search. *DNA Word Design* is the problem of designing a library of short n-mer oligonucleotide sequences (DNA words) for information storage. These DNA words encode finite alphabets of symbols by appropriately chosen sets of DNA oligonucleotide sequences. Ideally, a good word design will maximize binding specificity, and minimize cross-binding affinity (mismatching) and also minimize secondary structure. DNA word design is crucial to error control in BMC, so is well studied. Some of the basic techniques required by this application, such as DNA word design, have already been developed in prior work[2] Each element of a vector of the database is encoded by a DNA word. Since the number of possible values of each element is $|B|$, we require a library of $|B|$ distinct DNA words for encoding the possible elements at a given position in the vector. To decrease associative match

[2] Researchers have provided DNA word designs using random strings [A94, L94], evolutionary search [DMRGF+97], error-correcting codes [W98], automated constraint-based procedures [HGL98], and other methods [A96, B96, DMGFS96, M96, GDNMF97, JK97a].

misalignments, database vectors can use a distinct DNA word library of $|B|$ distinct DNA words (with minimal cross-binding affinity), for each of the n positions in the vector. Each vector of the database is thus encoded by a length n sequence of these DNA words, followed by DNA word encoding an identifying index to that vector.

4.3 Major Challenges Remaining. Nevertheless, key aspects of the associative search application needed development, including development of methods for the following key tasks (not considered by Baum):

(a) Input and Output (I/O) to Conventional Media: The database may initially be in conventional electronic media, rather than the form of DNA strands. The conversion of a static database needs only to be done one time, but then the queries also need to be so converted. Hence, the methods of Sections 2 and 3 can be used to improve I/O rates to and from conventional media, with error-resiliency and optimal I/O rate for a given error rate.

(b) Refining the Associative Search to Exact Affinity Separation: A key difficulty in the use of DNA annealing to do associative search is due to the high stringency of DNA annealing. Suppose the query vector v has a partial match with two data base vectors: (i) the vector v_1 matches the query nearly exactly except for a small number k of base mismatches scattered in the interior, while (ii) another vector v_2 matches the query exactly except for a much larger number $k' >> k$ consecutive base mismatches at one end. Then v_1 is a much closer match with the query vector than v_2. But if the vectors v_1, v_2 are encoded as DNA strands s_1, s_2, respectively, and the query vector v is encoded in complementary fashion as a strand s, then it is quite possible that s might anneal to strand s_1 much better that to strand s_1. The reason for this discrepancy is that in an annealing of two nearly complementary DNA strands, base mismatches that occur in scattered fashion in the interior of the strands can be less stable than mismatches at the end of strands. In general, in the annealing process between two single stranded DNA, the energetic properties of a set of mismatched bases in one strand (generally results in a local DNA bulge along the mismatch subsequence) varies dramatically depending on the position of the mismatches. Therefore, DNA annealing does not in general provide a very uniform metric for associative matching in the case of partial matches. Hence methods were needed for refining the associative search method to require only annealing on complementary sequences for which DNA annealing affinity methods work best, even if the query in not an exact match or even partial match with any data in the database.

4.4 Applying VQ Coding Methods To Associative Search: Refining the Associative Search to Exact Matches

DNA annealing affinity methods work best on complementary sequences. Yet, we need to process an associative match query, even if the query in not an exact match or even partial match with any data in the database. We now develop two methods for refining the associative search method to require only annealing on exactly complementary sequences. Both methods apply VQ-Coding clustering techniques.

(A) Associative Search with Given Match Distance. Our first method makes the assumption that we are given a bound d on the allowed match distance (recall that this is the distance between the query vector and the selected database vectors) for an associative match query. Using a conventional computer, we apply known VQ-Coding clustering techniques [G90, JD88], which provide a clustering of the database vectors so that the radius of each cluster is at most d. Recall that the elements of each vector in V range over a finite domain B. For each VQ cluster G of the database vectors, let $v(G)$ be the center point of G, and let the *possibility vectors* $P_{2d}(G)$ be the set of those vectors in V that are within distance $2d$ of the center point $v(G)$ of the cluster G (see Figure 13). We now describe our method for doing associative search:

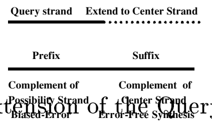

Fig. 14. Extension of the Query Strand to the Center Strand

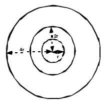

Fig. 15. Hierarchical Associative Search.

[0] Initialization: For each cluster G of the database vectors, we synthesize a DNA strand $s(G)$, to be called the *center stand* of that cluster G whose words encode the center point of the cluster, along with a unique identification tag. Then, using a biased-error chemical synthesis similar to the procedure given in Section 2.3 for synthesis of the EC strands, we construct from the center stand $s(G)$ of each of the cluster G, a multiset PS of single stranded DNA, each consisting of a prefix portion that encodes the complement of a possibility vector in $P_{2d}(G)$, followed by a suffix portion consisting of the complement of the center stand $s(G)$. This step is done only once, for a static data base.

Associative Query Processing

Input: A query vector v.

[1] We synthesize a multiplicity of DNA strands, called *query strands* encoding the query vector v.

[2] We then combine these query strands with the multiset PS. The hybridization products include doubly stranded DNA complexes. Each of these consist of a query strand hybridized with a corresponding prefix of a PS strand, with single stranded overhang consisting of the suffix portion of the PS strand consisting of the complement of the center strand (see Figure 14).

[3] Primer extension is applied to these hybridization products with overhangs, so that each query strand is extended to include the center strand as its new suffix, forming a *result strand.*

[4] Then we denature to single stranded DNA, and separate out the result strands using affinity separation with strands complementary to the center strands.

[5] Output: Finally, we output to conventional media (e.g., by the use of DNA chips) the center vectors corresponding to the suffix portions of the result strands. We use a conventional computer to enumerate the vectors of the clusters of each of these center vectors and to determine which of each cluster's vectors are of distance at most d from the query vector v. (Recall we have already precomputed, using a conventional computer, the clustering and the center of each cluster. Hence these result strands suffice for us to determine the cluster index and list of vectors of the clusters.)

The main point is that this construction reduces the associative search problem to that of finding just exact matches (via complementary hybridization), and this can be done very effectively by known DNA annealing methods (e.g., PCR). Note that the set of possibility vectors $P_d(G_i)$ defines an n-dimensional ball, with respect to the defined distance metric, with radius $2d$ and centered at the center point of cluster G_i. Since elements of the vectors range over a set of size $b = |B|$, the size of the set $P_{2d}(G)$ of possibility vectors of any cluster G is at most $b^{2d} - 1$. For bounded, modest size b and d (where say $b = 28 = 64$ and $d = 5$), we are able to do this construction in the DNA domain, since DNA is quite compact.

Proof of the Algorithm and Analysis. We now show:

Theorem 2 The final output determined in step [5] consists of all database vectors that are at most distance d to the query vector. Furthermore, the number of distinct database vectors that are enumerated by a conventional computer in step [5] is at most $b^{2d} - 1$.

Proof: Let $G_1, \ldots, G_j$ be the selected clusters whose centers are of distance at most d from the query vector v, and let $v_1, \ldots, v_j$ be the centers of these clusters selected in step [4]. Note that the query vector, represented by a DNA strand, will be included among the possibility vectors of each of those clusters. We need to prove that these center vectors are of distance at most $2d$ from the query vector, and their clusters include all database vectors that are at most distance d to the query vector. The elements of each cluster G_i are of distance at most d from their center vector v_i, and each center vector v_i is of distance at most d from the query vector v. Hence, the vectors in these clusters are at most twice this distance, that is of most distance $2d$, to the query vector v. So the number of distinct database vectors that are enumerated in step [5] is at most $b^{2d} - 1$. Furthermore, consider any database vector v' that is at most distance d to the query vector v. Then v' will be in a cluster G whose center $v(G)$ is at most distance d from v', and so the center $v(G)$ will be of distance at most $2d$ from the query vector v. Hence, v' will be included as an element of one of these selected clusters $G_1, \ldots, G_j$. **QED**

(B) Hierarchical Associative Search. Our next method, which we just sketch, is a more complex procedure which makes only the assumption that the match distance be upper bounded by d. For example, we will assume that the match distance be at most the radius of the entire set of database vectors. We apply known hierarchical VQ-Coding clustering techniques [G90, JD88]. These make a series of distinct clusterings of the data at exponentially increasing cluster radius. These hierarchical sequence of clusterings can be represented by a tree, whose root is the original set of vectors, and where each level of the tree is a clustering, refining the clustering at the previous level, and with cluster radius half that of the previous level (see Figure 15). Given an associative match query, let the *near query vectors* within radius r be the set $P_r(G)$ of possibility vectors of a hypothetical cluster of radius r with center vector being the input query vector. For each level of the tree with a radius bound of say r, we synthesize a multiset of DNA strands, which will be called *near query strands* which encode the near query vectors within radius r. To execute an associative search, we do not search just using the input query vector, and instead augment the query vector with these near query vectors of increasing radius. The search proceeds on decreasing levels $= 0, 1, \ldots$ starting at the lowest level of the tree. At each of these levels, of radius bound of say r, we execute an affinity separation to determine if there is any exact matches between the near query vector within that radius r that exactly matches a vector of the database. We terminate where either r exceeds d, or at the first level where at least one near query vector exactly matches a vector of the database. Again, the main point of this construction is to reduce the associative search problem to that of finding just exact matches (via complementary hybridization), and again this can be done very effectively by known DNA annealing methods such as PCR. Note that the number possibility vectors of any cluster of radius r is at most $b^r - 1$, and this is upper bounded by $b^{2d} - 1$, as in the previous method. Hence for bounded, modest size b and d, we are able to do this construction in the ultra-compact DNA domain.

5 Extension of Associative Search to Include Boolean Conditionals

Finally, we briefly describe how to extend associative search queries to sophisticated hybrid queries that include also Boolean formula conditionals (with a small bounded number n' of Boolean variables), by combining our methods for DNA associative search with known BMC methods for solving the SAT problem (e.g., using surface chemistry techniques [CRFCC+96, LGCCL+96, BCGT96, CCCFF+97, LTCSC97, LFW+98, WQF+98]). We assume that each of the vectors of the database are augmented with "digital tag vectors" consisting of a list of n' Boolean values, encoding binary information about the vector. An extended query consist of (i) a query vector to be matched with and (ii) a Boolean formula to be satisfied. The extended query requires finding those database vectors that closely match the query vector and also whose Boolean variables satisfy the queries Boolean formula. The DNA strands encoding these database vectors also are augmented with prefix *digital tag strands* consisting of a sequence of DNA words encoding these Boolean values.

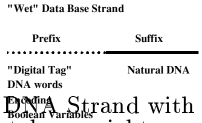

Fig. 16. A Natural DNA Strand with a Digital Tag Prefix

For example, the extended database might consist of natural DNA strands (e.g., from blood or other body tissues) onto which are appended at their 5' ends prefix digital tag strands (see Figure 16) consisting of DNA words encoding identifying information about each strand (such as social security number of the person whose DNA was sampled, cell type, the date, further medical data, etc.). The digital tag strands may have been constructed by previous BMC processing. There is also proposed methods [LL97] for recoding natural DNA, by the use of nonstandard DNA bases, into a form more amenable to computation.

Our technique is to execute the extended query in two stages:

(a) We first execute the Boolean formula portion of the query as a SAT problem, using biomolecular computing techniques previously developed (e.g., using surface chemistry techniques referenced above). These method can be implemented so thatthose strands not encoding SAT solutions are deleted, and all the remaining DNA strands satisfy the Boolean formula.

(b) Then we execute the associative search, as previously described in Section 3, on the remaining strands, to find the closest match to the query vector that satisfies the query's Boolean formula.

6 Conclusion

We have provided some examples of how ideas from information processing disciplines can be applied to biotechnology. (As we have already pointed out, this is not the first time this was done, if we consider PCR as essentially a recursive algorithm for selective strand amplification.) We believe that this approach of "Computationally Inspired Biotechnologies" will be a profitable approach for overcoming key biotechnology challenges remaining, for example:

- increased affinity selectivity for a wider range of molecules, and
- decreased errors in chemical synthesis.

Moreover, in the immediate future, the separation between computational and biological technologies should narrow. As the miniaturization of biotechnology continues, we can expect DNA chip technology to have also MEMS microflow devices and computational processing capability as well (see Gehani and Reif [GR98] and Suyama [S98]).

References

[A94] Adleman, L.M., "Molecular Computation of Solution to Combinatorial Problems", Science, 266, 1021, (1994).

[ARRW96] Adleman, L.M., P.W.K. Rothemund, S. Roweis, E. Winfree, "On Applying Molecular Computation To The Data Encryption Standard", 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton, June, 1996

[BCGT96] Bach, E., A. Condon, E. Glaser, and C. Tanguay, "Improved Models and Algorithms for DNA Computation", Proc. 11th Annual IEEE Conference on Computational Complexity, J. Computer and System Sciences, to appear.

[B94] Barnes, W.M., "PCR amplification of up to 35-kb DNA with high fidelity and high yield from bacteriophage templates", Proc. Natl. Acad. Sci., 91, 2216–2220, (1994).

[B95] Baum, E. B., "How to build an associative memory vastly larger than the brain", Science, pp 583-585, April 28, 1995.

[B96] Baum, E. B. "DNA Sequences Useful for Computation, 2nd Annual DIMACS Meeting on DNA Based Computers", Princeton University, June 1996.

[BC81] Beaucage,S.L., and Caruthers,M.H. (1981). "Deoxynucleoside phosphoramidites-A new class of key intermediates for deoxypolynucleotide synthesis", Tetrahedron Lett. 22,1859-1862.

[B68] E. R. Berlekamp, "Algebraic Coding theory", McGraw-Hill Book Company, NY (1968).

[BKH96] Blanchard, A. P., R. J. Kaiser and L. E. Hood, "High-density oligonucleotide arrays", Biosens. Bioelec., Vol. 11, 687-690, (1996).

[BDL95] Boneh, D., C. Dunworth, R. Lipton, "Breaking DES Using a Molecular Computer", Princeton CS Tech-Report number CS-TR-489-95, (1995).

[BL95a] Boneh, D., and R. Lipton, "Making DNA Computers Error Resistant", Princeton CS Tech-Report CS-TR-491-95, Also in 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, June 1996.

[BL95b] Boneh, D., and R. Lipton, "A Divide and conquer approach to DNA sequencing", Princeton University, 1996.

[BSS+94] N. E. Broude, T. Sano, C. L. Smith, and C. R. Cantor, "Enhanced DNA Sequencing by hybridization", Proc. Natl. Acad. Sci., Vol. 91, pp. 3071-3076, (April, 1994).

[CCCFF+97] Cai, W., A. Condon, R.M. Corn, Z. Fei, T. Frutos, E. Glaser, Z. Guo, M.G. Lagally, Q. Liu, L.M. Smith, and A. Thiel, "The Power of Surface-Based Computation", Proc. First International Conference on Computational Molecular Biology (RECOMB97), January, 1997.

[CRFCC+96] Cai, W., E. Rudkevich, Z. Fei, A. Condon, R. Corn, L.M. Smith, M.G. Lagally, "Influence of Surface Morphology in Surface-Based DNA Computing", Submitted to the 43rd AVS National Symposium, Abstract No. BI+MM-MoM10, (1996).

[CYH+96] Chee, M. ,R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhart, M. S. Morris and S. P. A. Fodor, "Accessing genetic information with high-density DNA arrays", Science, Vol. 274, 610-614, (1996).

[CW97] Chen, J., and D. Wood, "A New DNA Separation Technique with Low Error Rate", Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).

[CRB99] Clelland, C.T., Risca, V., and C. Bancroft. "Genomic Steganography: Amplifiable Microdots". To appear in Nature, 1999.

[CT 91] Cover, T. M. and J. A. Thomas, "Elements of Information Theory", John Wiley, New York, NY, (1991).

[DMGFS96] Deaton, R., R.C. Murphy, M. Garzon, D.R. Franceschetti, and S.E. Stevens, Jr., "Good encodings for DNA-based solutions to combinatorial problems", Proceedings of the 2nd Annual DIMACS Meeting on DNA Based Computers, June 1996.

[DMGFS98] Deaton, R., R.C. Murphy, M. Garzon, D.R. Franceschetti, and S.E. Stevens, Jr., "Reliability and efficiency of a DNA-based computation", Phys. Rev. Lett. 80, 417-420 (1998).
[DMRGF+97] Deaton, R., R.C. Murphy, J.A. Rose, M. Garzon, D.R. Franceschetti, and S.E. Stevens, Jr., "A DNA Based Implementation of an Evolutionary Search for Good Encodings for DNA Computation", ICEC'97 Special Session on DNA Based Computation, Indiana, April, 1997.
[DHS97] Deputat, M., G. Hajduczok, E. Schmitt, "On Error-Correcting Structures Derived from DNA", Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[DDSPL+ 93] Drmanac, R, S. Drmanac, Z. Strezoska, T. Paunesku, I. Labat, M. Zeremski, J. Snoddy, W. K. Funkhouser, B. Koop, L. Hood, and R. Crkenjakov "DNA Sequence Determination by Hybridize: A Strategy for Efficient Large-Scale Sequencing", Science, 260, 1649–1652, (1993).
[FRP+91] Fodor, S. P. A., J. L. Read, C. Pirrung, L. Stryer, A. T. Lu and D. Solas, "Light-directed spatially addressable parallel chemical synthesis", Science, Vol. 251, 767-773, (1991).
[FTCSC97] Frutos, A.G., A.J. Thiel, A.E. Condon, L.M. Smith, R.M. Corn, "DNA Computing at Surfaces: 4 Base Mismatch Word Design", Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[GDNMF97] Garzon, M., R. Deaton, P. Neathery, R.C. Murphy, D.R. Franceschetti, S.E. Stevens Jr., "On the Encoding Problem for DNA Computing", Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[GLR99] Gehani, A., T. H. LaBean, and J.H. Reif, "DNA-based Cryptography", 5th DIMACS Workshop on DNA Based Computers, MIT, June, 1999. DNA Based Computers, V, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. E. Winfree), American Mathematical Society, 2000.
http://www.cs.duke.edu/~reif/paper/DNAcypt/crypt.ps
[GR98] Gehani, A. and J. Reif, "Micro flow bio-molecular computation", 4th DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June, 1998. DNA Based Computers, IV, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. H. Rubin), American Mathematical Society, (1999). Also, special issue of Biosystems, Vol. 52, Nos. 1-3, (ed. By L. Kari, H. Rubin, and D. H. Wood), pp 197-216, (1999).
http://www.cs.duke.edu/~reif/paper/geha/microflow.ps .
[GGG91] Gersho, A., R. Gallager, and R. M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, (1991).
[GFBCL+96] Gray, J. M. T. G. Frutos, A.M. Berman, A.E. Condon, M.G. Lagally, L.M. Smith, R.M. Corn, "Reducing Errors in DNA Computing by Appropriate Word Design", University of Wisconsin, Department of Chemistry, October 9, 1996.
[G90] Gray, R. M., "Source Coding Theory", Klewer Academic Publishers, Boston, (1990).

[GT94] Grumbach, S., and F. Tahi, “Compression of DNA Sequences”, Proceedings of the IEEE Data Compression Conference (DCC’94), Snowbird, UT, 72–82, March 1994.
[H50] Hamming, R. W., “Error Detection and error correcting codes”, Bell System Technical Journal, Vol. 29, 147-160, (1950).
[HGL98] Hartemink, A., David Gifford, J. Khodor, “Automated constraint-based nucleotide sequence selection for DNA computation”, 4th DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June, 1998. DNA Based Computers, IV, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. H. Rubin), American Mathematical Society, (1999).
[HG97] Hartemink, A.J., D.K. Gifford, “Thermodynamic Simulation of Deoxyoligonucleotide Hybridize for DNA Computation”, Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[JD88] Jain, A. K. and R. C. Dubes, “Algorithms for clustering data,” Prentice Hall, Englewood Cliffs, N.J., (1988).
[KG97] Khodor, J., and David K. Gifford, “The Efficiency of Sequence-Specific Separation of DNA Mixtures for Biological Computing”, Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[KG98] Khodor, J., D. Gifford, “Design and implementation of computational systems based on programmed mutagenesis”, 4th DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June, 1998. DNA Based Computers, IV, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. H. Rubin), American Mathematical Society, (1999).
[FLGL99] Lipschutz, R.J., Fodor, P.A., Gingeras, T.R., and Lockhart, D.J. Nature Genetics Supplement, vol 21, pp 20-24 (1999).
[LYK+00] LaBean, T. H., Yan, H., Kopatsch, J., Liu, F., Winfree, E., Reif, J.H. and Seeman, N.C., “The construction, analysis, ligation and self-assembly of DNA triple crossover complexes”, J. Am. Chem. Soc. 122, 1848-1860 (2000).
www.cs.duke.edu/∼reif/paper/DNAtiling/tilings/JACS.pdf
[LWR99] LaBean, T. H., E. Winfree, J. H. Reif, “Experimental Progress in Computation by Self-Assembly of DNA Tilings”, 5th International Meeting on DNA Based Computers(DNA5), MIT, Cambridge, MA, (June, 1999). To appear in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, ed. E. Winfree, to appear American Mathematical Society, 2000. http://www.cs.duke.edu/∼thl/tilings/labean.ps
[LL97] Landweber, L.F. and R. Lipton, “DNA 2 DNA Computations: A Potential ‘Killer App’?”, 3nd Annual DIMACS Meeting on DNA Based Computers, University of Pens., (June 1997).
[L71] van Lint,J. H., “Coding Theory”, Lecture Notes in Mathematics, Springer Verlag, NY, (1971).
[L95] Lipton, R.J. “DNA Solution of Hard Computational Problems”, Science, 268, 542–845, (1995).
[LFW+98] Liu, Q., A. Frutos, L. Wang, A. Thiel, S. Gillmor, T. Strother, A. Condon, R. Corn, M. Lagally, L. Smith, “Progress towards demonstration of a surface based DNA computation: A one word approach to solve a model satisfiability problem”, 4th DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June, 1998. DNA Based Computers, IV, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. H. Rubin), American Mathematical Society, (1999).

[LGCCL+96] Liu, Q., Z. Guo, A.E. Condon, R.M. Corn, M.G. Lagally, and L.M. Smith, "A Surface-Based Approach to DNA Computation", Proc. 2nd Annual Princeton Meeting on DNA-Based Computing, June 1996.
[LTCSC97] Liu, Q., A.J. Thiel, A.G. Frutos, R.M. Corn, L.M. Smith, "Surface-Based DNA Computation: Hybridize and Destruction",Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[LY97] Loewenstern, D. and Yainilos, P., "Significantly lower entropy estimates for natural DNA sequences", J.A Storer and M Cohn (Eds.), IEEE Data Compression Conference, Snowbird, UT, pp. 151-161, (March, 1997).
[MLR+00] Mao, C., T.H. LaBean, J. H. Reif, and N.C. Seeman, "An Algorithmic Self-Assembly", Nature, Sept 28, (2000).
www.cs.duke.edu/∼reif/paper /SELFASSEMBLE/AlgorithmicAssembly.pdf
[MH98] Marshall, A., Hodgson, J. 1998 Nature Biotechnology 16, pp 27-31.
[MBD+97] McGall, G.H., Barone, A.D., Diggelmann, M.,Ngo, N.,Gentalen, E., and Fodor, S.P.A. "The Efficiency of Light-Directed Synthesis of DNA Arrays on Glass Substrates". J. Am. Chem. Soc., 119(22): 5081-5090, (1997).
[MYP98] Mills, A., B. Yurke, P. Platzman, "Error-tolerant massive DNA neural- network computation", 4th DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June, 1998. DNA Based Computers, IV, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. H. Rubin), American Mathematical Society, (1999).
[M96] Mir, K.U., "A Restricted Genetic Alphabet for DNA Computing", 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, (June 1996).
[NW99] Nevill-Manning, C.G. and I.H. Witten, "Protein is Incompressible", J.A Storer and M Cohn (Eds.), IEEE Data Compression Conference, Snowbird, UT, pp. 257-266, (March, 1999).
[PSS+94] Pease, A. C. , D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes and S. P. Fodor, "Light-generated oligonucleotide arrays for rapid DNA sequence analysis", Proc. Natl Acad. Sci. USA, Vol. 91, 5022-5026, (1994).
[P82] V. Pless, "Introduction to the theory of error-correcting codes," John Wiley and Sons, , NY (1982).
[OGB97] Orlian, M., F. Guarnieri, C. Bancroft, "Parallel Primer Extension Horizontal Chain Reactions as a Paradigm of Parallel DNA-Based Computation", Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[R93] Reif, J. (ed.), Synthesis of Parallel Algorithms", Morgan Kaufmann, (1993).
[R95] Reif, J.H., "Parallel Molecular Computation: Models and Simulations", Seventh Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA95), ACM, Santa Barbara, 213–223, June 1995. Algorithmica, special issue on Computational Biology, 1999. (http://www.cs.duke.edu/∼reif/paper/ paper.html)
[R97] Reif, J.H., "Local Parallel Biomolecular Computation", 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997). DIMACS Series in Discrete Mathematics and Theoretical Computer Science, ed. H. Rubin, (1999).
(http://www.cs.duke.edu/∼reif/paper/Assembly.ps and /Assembly.fig.ps)
[R98] Reif, J.H., "Paradigms for Biomolecular Computation", First International Conference on Unconventional Models of Computation, Auckland, New Zealand, January

1998. Unconventional Models of Computation, edited by C.S. Calude, J. Casti, and M.J. Dinneen, Springer Pub., Jan. 1998, pp 72-93.
(http://www.cs.duke.edu/∼reif/paper/paradigm.ps)
[RLS00] J.H. Reif, T. H. LaBean, and Seeman, N.C.,Challenges and Applications for Self-Assembled DNA Nanostructures, Invited paper, Sixth International Meeting on DNA Based Computers (DNA6), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Leiden, The Netherlands, (June, 2000) ed. A. Condon. To be published by Springer-Verlag as a volume in Lecture Notes in Computer Science, (2000). http://www.cs.duke.edu/∼reif/paper/SELFASSEMBLE/selfassemble.ps
[R94] Roberts, S.S., "Turbocharged PCR", Jour. of N.I. H. Research, 6, 46–82, (1994).
[RDGS97] Rose, J.A., R. Deaton, M. Garzon, and S.E. Stevens Jr., "The Effect of Uniform Melting Temperatures on the Efficiency of DNA Computing", Third Annual DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June 23-26, 1997. Published in DNA Based Computers, III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol 48 (ed. H. Rubin), American Mathematical Society, (1999).
[RWBCG+96] Roweis, S., E. Winfree, R. Burgoyne, N.V. Chelyapov, M.F. Goodman, P.W.K. Rothemund, L. M. Adleman, "A Sticker Based Architecture for DNA Computation", 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, June1996, Also as Laboratory for Molecular Science, USC technical report A Sticker Based Model for DNA Computation, May 1996.
[R96]R96 Rubin, H. "Looking for the DNA killer app.", Nature, 3, 656–658, (1996).
[S48] Shannon, C. E., "A mathematical theory of communication", Bell System Technical Journal, Vol. 27, 379-423 and p 623-656, (1948).
[S49] Shannon, C. E., "Communication in the presence of noise", Proceedings of the I. R. E., Vol. 37, 10-21, (1949).
[S98] Suyama, A., "DNA chips - Integrated Chemical Circuits for DNA Diagnosis and DNA computers", To appear,(1998).
[WQF+98] Wang, L., Q. Liu, A. Frutos, S. Gillmor, A. Thiel, T. Strother, A. Condon, R. Corn, M. Lagally, L. Smith, "Surface-based DNA computing operations: DESTROY and READOUT", 4th DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June, 1998. DNA Based Computers, IV, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. H. Rubin), American Mathematical Society, (1999).
[WLW+98] Winfree, E., F. Liu, Lisa A. Wenzler, N. C. Seeman, "Design and Self-Assembly of Two Dimensional DNA Crystals", Nature 394: 539–544, 1998. (1998).
[WYS96] Winfree, E., X. Yang, N.C. Seeman, "Universal Computation via Self- assembly of DNA: Some Theory and Experiments", 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton, June, 1996.
[W98] Wood, D. H., "Applying error correcting codes to DNA computing", 4th DIMACS Workshop on DNA Based Computers, University of Pennsylvania, June, 1998. DNA Based Computers, IV, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (ed. H. Rubin), American Mathematical Society, (1999).