# On Designing DNA Databases for the Storage and Retrieval of Digital Signals

Sotirios A. Tsaftaris and Aggelos K. Katsaggelos

Department of Electrical and Computer Engineering,
Northwestern University, 2145 Sheridan Rd., Evanston, IL 60208, USA
{stsaft, aggk}@ece.northwestern.edu

**Abstract.** In this paper we propose a procedure for the storage and retrieval of digital signals utilizing DNA. Digital signals are encoded in DNA sequences that satisfy among other constraints the Noise Tolerance Constraint (NTC) that we have previously introduced. NTC takes into account the presence of noise in digital signals by exploiting the annealing between non-perfect complementary sequences. We discuss various issues arising from the development of DNA-based database solutions (i) *in vitro* (in test tubes, or other materials) for short-term storage and (ii) *in vivo* (inside organisms) for long-term storage. We discuss the benefits and drawbacks of each scheme and its effects on the codeword design problem and performance. We also propose a new way of constructing the database elements such that a short-term database can be converted into a long term one and vice versa without the need for a re-synthesis. The latter improves efficiency and reduces the cost of a long-term database.

## 1 Introduction

The history of DNA as an information storage and processing medium begins with Adleman's work on DNA-based solutions of the Hamiltonian path problem [1]. Subsequently, Baum proposed building a DNA database capable of storing and retrieving digital information [2].

Using DNA to solve a computationally hard digital signal processing problem was presented in [9, 11], where a DNA based solution to the disparity estimation problem was given.

DNA offers significant advantages when compared to other media for storing digital signals or data, in general. The DNA molecule, especially in its double stranded form, is very stable, compact, and inexpensive. Polymerase Chain Reaction (PCR) is an economical and efficient way to replicate databases. Querying the database can be implemented with a plethora of techniques. In digital databases the query time increases proportionally to the size of the database. However, in DNA databases when annealing is used as a search mechanism, the querying time is independent of the database size when the target molecules have equal concentrations.

As with any DNA computing application, the first step is to find reliable mechanisms for encoding digital signals into DNA sequences, also known as the codeword design problem. In our problem, the encoding has to be such that it enables

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3611, pp. 1192 – 1201, 2005.

content-based searches and at the same time limits the possibility of errors during retrieval. Furthermore, in the case of digital signals where perfect matches are almost impossible due to noise or the nature of the signals, the encoding scheme needs to allow for and account for imperfect matches. To accomplish this we introduced a new constraint, the Noise Tolerance Constraint (NTC) [9, 11].

The second step is to decide on the structure of the storage elements each of which has unique properties and characteristics. DNA databases consist of a collection of elements. Usually each element has a unique address (or index) block, which uniquely identifies (and therefore enables the retrieval of) a usually larger information-carrying block (data block).

The third step is to decide on the host environment of the database. It can be a test tube, a polymer, or even a living organism. Each host offers unique capabilities but at the same time imposes constraints on the information capacity, database longevity, and ease of use. Finally, input and output methods need to be chosen. Some input (information storage) and output (information retrieval) methods are faster than others.

The steps mentioned above are not independent. For example, the choice of host and database element has a large impact on the codeword design problem and the capabilities of the database.

In this paper we briefly describe the codeword design problem (section 2) and propose and analyze possible schemes for database construction, storage, and retrieval. In section 3 we present the state of the art in database element design and suggest a modification that can relax the constraints on codeword design. In section 4 we discuss short-term databases and their performance and present an input/output protocol. In section 5 we repeat the task for a long-term database but also emphasize the construction of a long-term database from a short-term version. Finally, in section 6 we conclude this paper.

## 2 The Codeword Design Problem

In most DNA computing applications only perfect hybridizations are acceptable. In our case, we want to design the DNA codewords such that the melting temperature between DNA words is *inversely proportional to the absolute difference* between the encoded signal values. To accomplish this, we have introduced a new constraint, the *Noise (or inexact match) Tolerance Constraint (NTC)* [9, 10, 11]. According to the NTC, duplexes formed by codewords assigned to neighboring signal values must

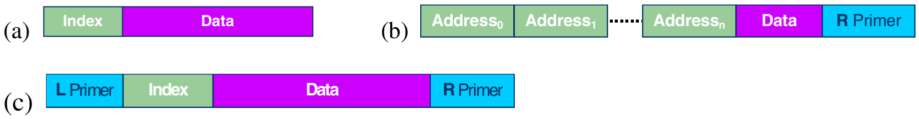

**Fig. 1.** Commonly used database elements (DE): (a) An index-data DE; (b) An address based DNA DE approach; (c) A DNA DE with left and right primers added for easy replication. All single border rectangles represent single stranded sequences.

have high (similar) melting temperatures, while duplexes outside this neighborhood must have melting temperatures lower than a predefined temperature threshold. In our case, we combine the NTC with other commonly used constraints, such as the self-complementarity, consecutive bases, GC content, frame-shift, and the reverse complement constraints. Another useful constraint is the illegal code constraint, that is, all codewords must not contain any word belonging to a set of illegal codes. In many cases recognition sites of enzymes are such codes. To solve this constrained optimization problem we have developed a random generate and test algorithm [10] and an iterative algorithm with stochastic non-improving steps [12].

## 3 Database Elements

### 3.1 State of the Art

The capabilities of a DNA database are highly dependent on the selection of the DNA database elements (DE). The sequences are created in such a way that it is clear which part is the entry identifier and which is the data. Database elements are then synthesized and mixed together to form the database.

Various designs of database element have been proposed in the literature. In Fig. 1 we illustrate some of the most commonly used ones. Baum [2] was the first to suggest the index-data approach (Fig. 1(a)) in building DNA memories. The left part is the index that identifies the data, which appear on the right. The index part of different DEs should be very dissimilar. Using even this simple DE with the appropriate encoding of information and annealing, database searches are possible.

As suggested in the introduction, using DNA to store information and annealing to perform the search is highly beneficial. A search of a database with a given query will return all the DEs that contain the query information. Although such feasibility is available with today's computers, the search time in a computer database is dependent on the size of the database (number of DEs). In a DNA database, the search time is not dependent on the number of the DEs but on the concentration of the DEs instead. Furthermore, a query search not only will return the DEs that match the query, but will provide information on the number of matching instances and their quality (strength), a feature very useful for signal processing applications [11].

Another possible DE structure is the address-data one shown in Fig. 1(b). This design was used in [6] for the implementation of a hierarchical DNA memory based on nested PCR, termed Nested Primer Molecular Memory (NPMM). The advantages of this approach are the theoretically unlimited scalability and capacity of the DNA database. An obvious disadvantage is that the lengths of DEs may differ, thus complicating the extraction of information.

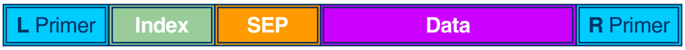

**Fig. 2.** A DNA database element with a separation element (SEP) between the index and data parts. Double border rectangles represent double stranded sequences.

Ordinary DEs can be retrofitted (before synthesis) with unique left and right primers at each end shown in Fig. 1(c). A PCR procedure applied to the database with primers, defined as the complements of the L and R primer sites, can easily replicate the whole database, thus reducing replication and distribution costs.

### 3.2 Proposed Design

When single stranded DNA DEs are used, the index and data have to be designed in such a way that an index query will not hybridize within the data part and vice versa. Such specificity falls in the category of the illegal code constraint, but unfortunately has its limitations (section 2).

Let us assume K different DEs are needed to carry digital signals (N integer values). In this case we have to (i) define the primers L and R, (ii) find K unique and highly dissimilar indices, and (iii) find a mapping between digital values and DNA sequences (N different sequences). A reasonable path for designing the system will be to first decide on the primers and then find the K unique indices using a codeword design algorithm with high Hamming distance as one of the constraints. Consecutively, data codewords will be designed to satisfy the illegal code constraint with illegal codes the K indices and the primers.

For large databases, such a procedure can be rather daunting. In Fig. 2 a different approach on DE design is shown, where the DEs are double stranded sequences with a separator sequence in the middle (labeled in the figure as SEP). With this approach, the design of the indices and the mapping can be separated and index sequences can actually be similar to data codewords. The idea is that the part needed for a search is exposed (single stranded, correct direction) upon request. For example, for a search in the data part, only that part is exposed. Furthermore, the parts can be exposed with different DNA direction. For example, the index part can be exposed in the 3' to 5' direction, thus requiring a query in the 5' to 3' direction, while the data part can be exposed in the 5' to 3' direction requiring a query in the 3' to 5' direction.

A similar idea was presented in [5] using hairpin formations of DNA as a conformational addressing mechanism. In this case, data access could be given after the addressing was complete, which is not exactly sought after in our research.

Our current design suggests the recognition site of a nicking enzyme as a separator sequence. Nicking enzymes are restriction enzymes that introduce an incision in one of the single strands instead of cutting the double stranded sequence in two parts. Our scope is to use a combination of nicking enzymes and two types of exonucleases. An appropriate exonuclease can be applied to expose the sequence in the desired direction for processing.

In this case the constraints for the codeword design problem are augmented with the addition of the recognition site as another illegal code.

## 4 Short Term Storage

We have previously discussed various DEs and their properties. Here we propose a possible platform for short-term storage and retrieval of digital signals. DNA sequences can be either kept in liquid form placed in test tubes, freeze-dried to save volume, or even placed within an exotic substance.

We have focused our attention on DEs of the form shown in Fig. 1(c) and Fig. 2. For added security, the database can be mixed with random genetic material as in [3]. To guarantee correct separation and amplification of the database, the primers need to be highly dissimilar compared to the genetic material.

## 4.1 Input-Output

There are many ways to implement a query search. Overall it requires the synthesis of the *query strand,* introduction of the query in the database solution and detection of the hybridization event using one of several spectroscopic techniques, i.e., fluorescent labels attached to the query strand.

In the case where a simple yes or no answer (whether the query can be found in the database or not) is needed, a change in the fluorescent response will indicate success.

If, in addition, we want to know the elements in which a match was found, affinity purification or Fluorescence Activated Cell Sorting (FACS, a technique that allows the separation of fluorescence activated molecules) can separate the strands that had a match. For affinity purification, query strands should be attached to beads. For FACS, query strands should be fluorescently labeled. Assuming the existence of a DNA chip (microarray) with all the indices immobilized in spots, placing a sample of the washed solution onto the chip will return the index. If the position of a match is sough after, a laboratory procedure would include a PCR step using as primer the query strand followed by a gel electrophoresis and a DNA sequencing step.

## 4.2 Performance

To evaluate the performance of a DNA database, three factors have to be evaluated: (a) speed, (b) information capacity and scalability of the database, and (c) accuracy.

### Speed

The biggest advantage of a DNA approach is that search speed is constant for any database size and that multiple searches can take place in parallel. Speed is estimated by the amount of time a query strand needs to react such that a high detectable concentration of outcomes can be guaranteed. The time spent for extracting the outcomes is not accounted for. In order for the constant speed assumption to hold, DEs must have equal concentrations and the query strands must be in excess.

The key to constant speed is that a query search using DNA annealing is, in fact, a competitive hybridization. This means that for a given high concentration of DEs, the best match will find the query in a given amount of time. Following the proposed DE design, this is relatively easy since only a PCR amplification step is needed. It is clear that there is a trade-off between speed and molar concentrations. If speed is a concern and a constant search time is sought after, an increase in the database concentration is needed. On the other hand, if volume is a concern then a higher reaction time is needed resulting in a reduction in speed.

### Capacity and Scalability

Information capacity is a critical component of a database, although in many cases, capacity is a function of speed and accuracy. The study of information capacity in

DNA databases has to be coupled with the accuracy of the database. The scalability of the database depends on the available indices.

**Accuracy**

Although the scope of the codeword and index design algorithms is to find sequences that will not cause hybridization errors, the models used to simulate hybridization procedures have inherent statistical errors. It is therefore expected that outcomes of *in vitro* database searches will have some errors. The most important sources of errors are: hybridization errors, polymerase errors, enzymatic errors, extraction errors, and human errors.

## 5 Long Term Storage

For long-term storage, we propose the insertion of the database in living organisms following the insertion technique used in [13]. The choice of the hosting organism depends on various parameters, such as mutation rate, capacity, maintenance cost, and others.

### 5.1 Proposed Database Element Construction

To embed a sequence in the genome of an organism the message sequence (in our case the database) is encapsulated between two extraction primers, the L and R keys. The primers are used to extract the database from the organism's genome using a PCR procedure. The L and R keys need to be (i) highly dissimilar compared to the genome of the organism for extraction and (ii) include multiple repetitions of STOP codons to protect the organism [7].

In [13] the message was pre-synthesized and was part of the overall embedding sequence. In our case, the message is either the whole or a part of the database consisting of concatenations of DEs shown in the previous section. Following the same approach, the construction of a long-term version of our database would require the resynthesis of the whole database as a big linear molecule, which is rather inefficient and uneconomical.

Here we propose a slight modification of the DEs used for short-term storage to accommodate further long-term storage applications with minimal laboratory work. Furthermore, using only two laboratory operations, the database extracted from an organism is converted to the short-term equivalent and the same information retrieval procedures can be applied.

The proposed modification affects only the design of the L and R primers used to encapsulate the DEs. The primers are designed such that when two DEs are concatenated, they create the recognition site, RS, of a restriction enzyme as shown in Fig. 3.

As shown in Fig. 3, with the addition of the restriction enzyme, the concatenated database constructed from DEs is transformed into individual DEs with sticky ends. To transform the DEs into their original single stranded form, only the addition of a 3'→5' exonuclease is needed.

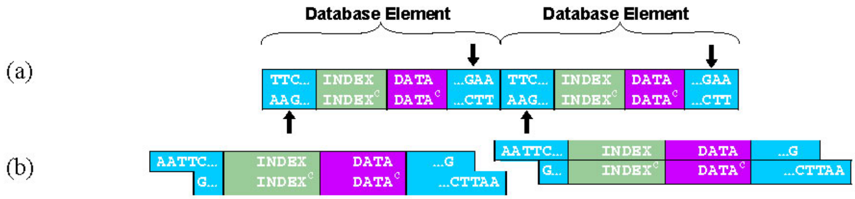

**Fig. 3.** Transformation from a long database strand (a) to individual database elements with sticky ends by adding a restriction enzyme (b).

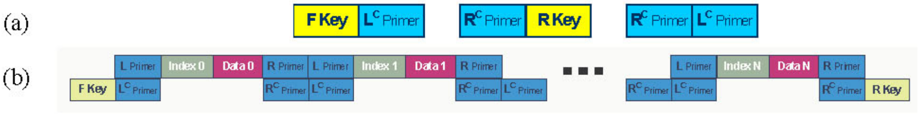

**Fig. 4.** (a) The “helper strands.” (b) Molecular self-assembly of database elements for long term storage with the addition of complementary helper strands.

If the DEs have a separator sequence in the middle, the double stranded form is adequate. Consequently, only an exonuclease that digests single stranded parts is needed to eliminate the sticky ends.

To construct a long concatenation of DEs from a given short-term database, “helper strands” are employed. The “helper strands” are illustrated in Fig. 4(a), where the F and R keys are the extraction primers, and $R^C$ and $L^C$ are the complements of the L and R primers.

When the “helper strands” are added to a solution containing DEs in the form of Fig. 1(c), molecular self-assembly will follow, which will create formations similar to the one illustrated in Fig. 4(b). With the addition of ligase, each formation will be joined at the hybridized parts, and long, partially single stranded molecules composed of DEs are obtained.

Clearly, the self-assembly will happen at random, and it is not guaranteed that each formation will contain at least one copy of each DE in the database. With gel electrophoresis, the molecules can be sorted according to length, and a subset D of those with the longer length can be finally selected, with the expectation that in D, at least one copy of each DE is present. In the performance section, we will further discuss this issue.

Overall the problem of codeword design does not change significantly. In this case the only difference is the design of the L and R keys and the L and R primers and the addition of the restriction site RS, the keys and the primers as illegal codes in the design problem.

### 5.2 Input Output

Here we assume that a short-term storage solution is already available and the DEs conform to the constraints listed in the previous paragraph. This conformation assumes prior knowledge of the hosting organism. The procedure for acquiring a long-term storage from a short-term storage solution that follows the appropriate DE structure involves: mixing the helper strands with a sample of the solution, extracting the self-assembled strands, converting them into cloning vectors, and inserting them in the organism.

To extract information from the organism, the procedure requires extracting the genetic material, amplifying the database using PCR and the L/R keys, and cutting the long strands into DEs with the restriction enzyme.

### 5.3 Performance

Since most of the discussion of section 4.2 is applicable here, we will only mention some further considerations.

**Speed**
Transformation between short-term and long-term is a tedious one-time event; hence, it is not considered as overhead. Search performance will be the same as in the short-term case.

**Capacity**
The capacity in this case is also affected by the design of the database due to the self-assembly process, and the organism itself. The construction of the database is based on stochastic molecular self-assembly. To increase the probability of inclusion of at least one DE in the cloning vector, high redundancy is needed. Multiple vectors of high dissimilarity need to be formed and inserted into different populations of organisms. With high probability, this mixture of organisms will contain all DEs. A second issue is the capacity of the organism itself. Certain limits and workarounds exist and are available throughout the literature of recombinant libraries [8].

**Accuracy**
The long-term storage is comprised of the following major components: (1) database creation (DEs to self-assembled strands), (2) database insertion (cloning vectors), (3) database maintenance (maintaining the organism), (4) database extraction, and (5) information retrieval. Each component introduces possible errors.

In (1) the errors are introduced from the random creation of the self-assembled strands, which leads to ambiguity in the final molar concentration of the DEs in (4).

In (2) the error arises from the event of unsuccessful insertion. To minimize its probability, multiple clones are inserted into multiple copies of the organism.

Component (3) introduces a very important source of errors, namely mutations. Organisms, especially bacteria, evolve very fast. To minimize the probability of such an error, an organism with low mutation rates should be chosen and redundancy should be exploited by inserting the same information into multiple bacteria.

In (4) enzymatic reactions are used that have certain success rates.

Finally in (5) the same sources of errors as in the short-term case apply. They are amplified by the fact that molar concentrations are random variables.

## 6 Concluding Remarks

In this paper we proposed various approaches towards the DNA-based storage and retrieval of digital signals. We considered two different approaches: one for short-term storage and one for long-term storage. For short-term storage, the database is kept in test tubes or other materials, whereas for long-term storage, the database is inserted inside an organism.

As a search mechanism we use the annealing property of DNA. To perform searches based on signal similarity, we stressed the importance of the Noise Tolerance Constraint, which was previously established.

We argued that a DNA based approach offers significant advantages when compared to traditional computer based techniques.

For database construction, we analyzed and compared various forms of its elements and identified the necessary changes in the codeword design problem. For each form, we provided an analysis of materials and equipment needed and also laid out procedures for database creation, replication, maintenance, and information extraction. We also proposed a procedure that can be used to transform one database form to another, thus avoiding re-synthesis costs. Finally, for each storage solution, we discussed performance issues, identified problems, and proposed a framework for simulation.

Although our goal is to offer a demonstrational small scale *in vitro* DNA database, in order to reduce laboratory costs we are in the process of implementing an *in silico* DNA database equivalent. This equivalent simulates various procedures, such as target-to-primer hybridization (already implemented in [9]), PCR, and bead based searches, and will be used to measure and improve the performance of the database. Laboratory experiments are planned to commence after the completion of the simulations.

## Acknowledgements

The authors would like to thank Prof. Osamu Ono and Zuwairie Ibrahim for their invitation to participate in this session. We are also grateful to Prof. Papoutsakis and Dr. Paredes for their help with our laboratory work. Finally Mr. Tsaftaris would like to thank the *Alexander S. Onassis* public benefit foundation for their financial support.

## References

1. L. Adleman, Molecular computation of solutions to combinatorial problems, Science, vol. 266, no. 5187, pp.1021–1024, 1994.
2. E.B. Baum, Building an associative memory vastly larger than the brain, Science, vol. 268, no. 5210, pp.583-585, 1995.
3. C.T. Clelland, V. Risca, and C. Bancroft, Hiding messages in DNA microdots, Nature, vol. 399, no. 6736, pp. 533–534, 1999.
4. H. Kakavand, D. O'Brien, A. Hassibi, and T.H. Lee, Maximum a Posteriori (MAP) Estimator for Polymerase Chain Reaction (PCR) Processes, In Proc. of 26th Int. Conf. of IEEE Engineering in Medicine and Biology, 2004.

5. Kameda, M. Yamamoto, H. Uejima, M. Hagiya, K. Sakamoto, and A. Ohuchi, Conformational addressing using the hairpin structure of single-strand DNA, In Revised Papers from DNA Computing, 9th International Workshop on DNA-Based Computers, Lecture Notes in Computer Science, Springer-Verlag, vol. 2943, pp. 219–224, 2004.
6. S. Kashiwamura, M. Yamamoto, A. Kameda, T. Shiba, and A. Ohuchi, Hierarchical DNA Memory Based on Nested PCR, In Revised papers from DNA Computing: Eighth Int'l Meeting on DNA Based Computers, Lecture Notes in Computer Science, vol. 2568, pp. 112–123, Springer-Verlag, 2003
7. Y. Ozawa, S. Hanaoka, R. Saito, and M. Tomita, Differences of Translation Termination Sites Among the Three Stop Codons, in Asai, K. and Miyano, S. and Takagi, T. (eds.), Genome Informatics 1999, vol. 10, pp. 328-329, Universal Academy Press, Tokyo, 1999.
8. K. She, So you want to Work with Giants: The BAC Vector, BioTeach Journal, vol. 1, Fall 2003, available online at http://www.bioteach.ubc.ca.
9. S.A. Tsaftaris, DNA-based Digital Signal Processing, M.S. Thesis, Northwestern University, Dept. of Electrical and Computer Engineering, June 2003.
10. S.A. Tsaftaris, A.K. Katsaggelos, T.N. Pappas and E.T. Papoutsakis, DNA based matching of digital signals, in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 5, Montreal, Quebec, Canada, pp. 581-584, 2004.
11. S.A. Tsaftaris, A.K. Katsaggelos, T.N. Pappas and E.T. Papoutsakis, How can DNA-computing be applied in digital signal processing?, IEEE Signal Processing Mag., vol. 21, no. 6, pp. 57-61, 2004.
12. S.A. Tsaftaris and A.K. Katsaggelos, A New Codeword Design Algorithm for DNA-Based Storage and Retrieval of Digital Signals, Pre-Proceedings of the 11th International Meeting on DNA Computing, June 6-9, London, Canada, 2005.
13. P.C. Wong, K. Wong, and H. Foote, Organic Data Memory Using the DNA Approach, Communications of the ACM, vol. 46, no. 1, 2003.