Encoding Methods for DNA Computers*

J. Blustein

29 October 1996

*A handout to accompany a tallk given in partial fulfillment of the requirements for *CS 881a – Topics in Computing and Biology: DNA Computing* in the department of Computer Science at the University of Western Ontario.

Abstract

A brief survey of some methods, as described in the literature, for encoding computational data using DNA. Methods are classified into three broad groups: those that try to use DNA to mimic conventional electronic computers, those that use it in novel ways and those that seek to improve on Aldeman's original encoding. The advantages and disadvantages of each model are briefly discussed.

Contents

1 Introduction

The papers I will discuss [Bau95, Bau96, RWB+96, DMG+96, Mir96, Rot96, GFB+, Adl94] are about methods of encoding information in DNA for computing. The approaches in these papers differ from each other in two ways. Firstly some of the papers are about encodings used (or to be used) in solving specific problems, e.g., Adleman's original article [Adl94], and some are about general methods of encoding. Secondly some of the solutions described are attempts to replicate methods used in electronic computers but some present novel approaches.

2 Novel Approaches

All of the approaches in this section are based on two conditions: 1) that fixed-length sequences are used as units of encoding, and 2) that sequences are extracted by binding to complementary sequences.

2.1 Adleman's TSP program

Adleman was the first to use DNA to compute the solution to a problem [Adl94]. He encoded a 7-vertex directed graph by representing each vertex by a different 20-mer oligonucleotide. Edges were represented by other 20-mer oligonucleotides each of which was composed of two 10-mer oligonucleotide subsequences. The $3'$ 10-mer subsequence of the edge encoding was complementary to the $5'$ end of the originating vertex and similarly with the other subsequence. He used complementary sequences to selectively retrieve DNA sequences encoding the results.

Advantages

- The approach demonstrated the possibility of DNA computers.
- The approach is non-deterministic and therefore unlike programs that are usually run on electronic computers.

- The approach used may be applicable to other problems, but see Baum's comments [Bau96] for provisos.

Disadvantages

- The solution given is specific to a single instance of a single problem.

2.2 Turing Machines in Bacteria

Rothemund proposes encoding a Turing Machine (TM) within bacterial DNA [Rot96]. He shows, in detail, how two TMs could be encoded in the circular DNA of a bacterium using commercially available restriction enzymes. The encoding and functioning of a three state Busy-Beaver TM is shown as well as a plan for the implementation of a four symbol, seven state universal TM.

His proposed encoding has the read/write head and the transition table encoded in a single circular piece of DNA. The head is inside the tape and has its position changed with each operation.

Advantages

- The system can be shown to work in theory.
- The system uses only standard, well understood components.

Disadvantages

- The model has not been tested.
- It is unclear how to generalize from the result.

2.3 Associative Memory

Computers store and retrieve data by address. To retrieve a datum from memory we must know at what address it is stored. In electronic computers computers the concept of an address refers to an exact location in memory. Baum proposes a new and more powerful form of memory for DNA computers [Bau95].

He proposes that data not be stored at specific locations, but rather be stored with an attached address tag. Under the proposed scheme DNA sequence encoding data will be retrieved by filtering the chemical solution comprising the memory through a selective sieve that will retain molecules with certain addresses but not others. The filtering could be accomplished using sequences complementary to the address encodings attached to solid supports, as Adleman did [Adl94].

To understand the power of the proposed encoding we must consider some of its details. Address tags would be composed of single stranded DNA while the data would be composed of double stranded DNA. The filtering operations could thus use single stranded DNA so as to match only the address portion. By varying conditions within the memory solution, the exactness of the match can be varied.

The smallest addressable component of electronic computers is the binary bit. Bits encode either 0 or 1 so a sequence of three bits can store up to eight values. By contrast, Baum's proposed method uses sequences of the four genetic nucleotides. Three bases can be combined to form seven distinct sequences. Address tags can be composed of subsequences, each of which is composed of a number of bases. Baum proposes that the order of the subsequences be unimportant for addressing. Thus while conventional computers use sequences of bits to encode data, Baum proposes using sets of DNA sequences to encode the address tags.

Advantages

- Recall by tag means you do not need to know exact location, only the tag encoding.
- The encoding allows for complex descriptions of the data, the tags could function like a relational database.

Disadvantages

- Addresses cannot be zero because at least one base must be present in address tag.
- Because the retrieval process is essentially non-deterministic there is no certainty that the correct data will always be retrieved.

3 Restrictions on Encoding Sequences

Molecules can form chemical bonds that will interfere with their encoding function. Bonds can form between different parts of the molecule (intramolecular) or between different molecules (intermolecular). Several authors [Bau96, GFB$^+$, Mir96] suggest constraints on the components of DNA sequences to avoid some practical problems with encodings like Adleman's (see Section 2.1).

3.1 Restricted Alphabet

Mir suggests encoding sequences using one of the following combinations: A/G, A/C, T/G or T/C to eliminate the possibility of intramolecular bonds [Mir96]. According to the author, chemical methods of controlling the formation of intermolecular bonds are well known.

Advantages

- Successfully tested with sequences composed of Ts and Cs.
- Devised by a biochemist with a knowledge of the chemical interactions involved.
- Sequencing by Hybridization described in the article can be easily incorporated into DNA computer [Mir96, p. 128].

Disadvantages

- Avoids only intramolecular folding, not intermolecular bonds.

3.2 Spacer Sequences

Baum's solution uses 'spacer' sequences which will separate the encoding portions. For example, if E is a sequence encoding part of the data for a computation and S is a spacer sequence then SES would be used in place of E.

Without spacer sequences there is a strong possibility that the wrong sequences will be retrieved or that correct sequences will not be retrieved. The constraints are intended to stop you from using sequences that may bind to themselves or to a sequence other than their complement.

Baum shows the length of the shortest possible spacer sequences for encodings of various lengths. Sequences of five bases can safely provide 208 distinct values for example.

Gray et al. provide a C++ program to identify exactly which sequences are best.

Advantages

- The method can make other methods more robust.

Disadvantages

- Many sequences are eliminated from use by the constraints.
- There is no guarantee that more sequences will need to be eliminated in future.
- Little consideration given to how DNA encoding functions in natural information preserving systems.

3.3 Genetic Algorithms to Select Sequences

Deaton et al. [DMG$^+$96] used a genetic algorithm to test possible encodings for Adleman's experiment (see Section 2.1). They determined that, by including various constraints on the sequences, that the Hamming distance is a good guide to how similar two sequences can be without risk of confusion between them. Some of their results were confirmed in laboratory experiments.

Advantages

- Apparent confirmation that the Hamming distance is a useful guide to selecting encoding sequences.

Disadvantages

- No general proof given.

4 Making DNA Act Like Circuitry

4.1 Sticker Manipulation

Roweis et al. propose mixing single- and double-stranded DNA to simulate bits within [RWB+96]. A strand of N bases long can be considered as divided into k non-overlapping regions each M bases long. Each of those regions is used to represent one bit of information. Bits are *on* if the corresponding region has a second (complementary) strand annealed to it (see Figure 1.) Information is encoded in a complex of DNA, not just a single or double stranded molecule.

The authors define chemical methods for (1) multiset union of DNA complexes, (2) selection of complexes with selected bits set, (3) setting and (4) clearing individual bits. They provide detailed suggestions for machinery to manipulate the molecules to carry out these operations.

They also suggest that the study of 'comma free code' and 'de Bruijn sequences' might help to discover useful sequences for encoding [RWB+96, p. 10].

Advantages

- Model does not require strand extension or use of enzymes.

- Error rates and implementation details are well considered.

$$\begin{array}{cccccccccc}
 & & \overline{\mathsf{NNNNN}} & & \overline{\mathsf{NNNNN}} & \overline{\mathsf{NNNNN}} & & & \overline{\mathsf{NNNNN}} & \\
5' & \mathsf{NNNNN} & \mathsf{NNNNN} & \mathsf{NNNNN} & \mathsf{NNNNN} & \mathsf{NNNNN} & \mathsf{NNNNN} & \mathsf{NNNNN} & \mathsf{NNNNN} & 3'
\end{array}$$

Figure 1: s.s. DNA with four 'bits' set by 'stickers'

- Model allows for many practical computations.

Disadvantages

- Current form of model does not allow for if/else branching.
- The space/time tradeoff may be large.

5 Conclusion

Appropriate encoding, like the choice of an abstract data type, depends on the problem being solved and the algorithm being used. As well, any DNA encoding will need to restrict the sequences it uses for encoding. Baum's approach [Bau96] to sequence selection is the most rigorous I have seen.

If the sticker model [RWB$^+$96] were used with modifications to allow for conditional branching then we could use DNA as sequences of bits. Such sequences can be used to model the encodings used in traditional computers or directly as bitvectors. Bit vectors do not readily suit most problems, and it would be a mistake to ignore the power of DNA computers by using them only to simulate traditional computers.

Baum's suggestion to use DNA as a form of associative memory [Bau95] has no direct parallel with the traditional bits and bytes approaches of traditional computers. It can, of course be simulated by higher order structures, but with DNA computers we can implement some interesting (and hopefully useful) operations directly.

There is no clear best encoding scheme yet. However, we can say with certainty that arbitrary sequences must not be used for encoding. Care must be excercised to ensure that no unexpected intramolecular bonds form and that the number of unexpected intermolecular bondings is kept low in whatever encoding is used.

References

[Adl94] Leonard M. Adleman.
Molecular computations of solutions to combinatorial problems.
Science, pages 1021 – 1024, 11 November 1994.
HTML version at `http://www.hks.net/%7ecactus/doc/science/molecule_comp.html`.

[Bau95] Eric Baum.
Building an associative memory vastly larger than the brain.
Science, pages 583 – 585, 28 April 1995.

[Bau96] Eric Baum.
DNA sequences useful for computation.
In Baum et al. [BBK+96], pages 122 – 127.
I have the version at `http://www.neci.nj.nec.com/homepages/eric/seq.ps`.

[BBK+96] Eric Baum, Dan Boneh, Peter Kaplan, Richard Lipton, John Reif, and Nadrian Seeman, editors.
Second Annual Meeting on DNA BASED COMPUTERS, DIMACS Workshop, Princeton University, Department of Computer Science, 10 – 12 June 1996.

[Das] Ray Dassen.
A bibliography of molecular computation and splicing systems.
HTML version at `http://liinwww.ira.uka.de/bibliography/Misc/dna.html`
Maintainer's address is ⟨`jdassen@wi.LeidenUniv.nl`⟩.

[DMG+96] R. Deaton, R. C. Murphy, M. Garzon, D. R. Franceschetti, and S. E. Stevens, Jr.
Good encodings for DNA-based solutions to combinatorial problems.
In Baum et al. [BBK+96], pages 131– 140.

[GFB+] Jesse M. Gray, Rony G. Frutos, A. Michael Berman, Anne E. Condon, Max G. Lagally, Lloyd M. Smith, and Robert M. Corn.

Reducing errors in DNA computing by appropriate word design.
Draft version date: 10/9/96.

[Mir96] Kalim U. Mir.
A restricted genetic alphabet for DNA computing.
In Baum et al. [BBK+96], pages 128– 130.

[Rot96] Paul Wilhelm Karl Rothemund.
A DNA and restriction enzyme implementation of Turing machines.
In Eric B. Baum and Richard J. Lipton, editors, *DNA Based Computers*, number 27 in DIMACS: Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1996.
Bibliography details from A Bibliography of Molecular Computation and Splicing Systems [Das]. I have used the version at `http://www.ugcs.caltech.edu/~pwkr/oett/dimacs/dimacs.ps`
See `http://www.ugcs.caltech.edu/~pwkr/oett.html` for a simulation of one of the TMs.

[RWB+96] Sam Roweis, Erik Winfree, Richard Burgoyne, Nickolas V. Chelyapov, Myron F. Goodman, Paul W. K. Rothemund, and Leonard M. Adleman.
A sticker based model for DNA computation.
In Baum et al. [BBK+96], pages 1 – 27.