Brief Report on Applications of Coding Theory in DNA Computing

Amir Hesam Salavati
E-mail: hesam.salavati@epfl.ch

Supervisor: Prof. Amin Shokrollahi
E-mail: amin.shokrollahi@epfl.ch

Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)

June 3, 2011

1 Introduction

In this short report, I will briefly discuss a number of papers that I have read recently on DNA self-assembly and DNA computing. The papers were particularly interesting as they explain some applications of coding theory and neural networks in designing fault tolerant DNA computing schemes. After the brief summary, I will outline some ideas that I think might be relevant and interesting for the "Statistical Mechanics and Computation of DNA self-assembly" workshop in the end of this month.

2 DNA Computing: A Brief Introduction

In DNA computing, researchers employ *artificially* synthesized *single stranded* DNA molecules to perform computational tasks. More specifically, instead of having binary vectors to process, we use DNA strands as inputs and outputs of the system. Chemical reactions among these strands serve as the processor. As a result of such reactions, single stranded DNA molecules will form a double helix and using chemical procedures, these double stranded DNA molecules are extracted and read as the output of the system.

The main advantage of DNA computing is its highly parallel structure: at any given moment we have several strands being processed by means of chemical reactions. The goal is then to *program* the system in the way that given the input it produces the desired output. In that regard, we must engineer DNA strands such that the reactions among them proceed in the desired way.

2.1 Issue of Noise in DNA Computing

For any viable application, DNA computing must be highly reliable. This is in contrast to unpredictable nature of the environment DNA computation is performed, i.e. in a solution and by using chemical reactions. There are many sources of error in such environment but the two most important ones are: undesired reactions between two DNA strands, which results in unwanted helical DNA molecules in the output, and DNA strands reacting with themselves and folding in $3D$ forms, called secondary structures, which will stop the whole computation process.

Hence, we must develop techniques to overcome the unreliability issue

and sources of noise. In this regard, the situation is similar to transmitting data over noisy channels where we would like to eliminate noise. As a result, one might try applying coding methods to design DNA computing systems with smaller probability of error.

3 DNA Computing and Coding Theory

It is well known that two DNA strands that have a small distance from each other will most probably react. Therefore, in order to minimize the unwanted reactions among DNA strands, we must synthesize our DNA molecules in a way that they have a large enough minimum distance. Milenkovic et al. [1] have proposed three novel ways based on well-known coding techniques to design such DNA molecules:

- *Cyclic codes*: The authors employ cyclic codes over $GF(4)$ and map them to DNA sequences by assigning each of the 4 coding symbols to one of the four nucleotides. Cyclic codes are particularly interesting in the context of DNA computing because they make the testing procedure for secondary structure formation easier.

- *Generalized Hadamard Matrices*: Another method that the authors propose for picking codewords is to construct an $n \times n$ Hadamard matrix in a way that rows are cyclic shifts of each other. Furthermore, due to the special construction method, the minimum distance is the same between all rows. Now, each row is mapped to a DNA sequence and the result is a DNA code with equal distance properties among all the codewords.

- *Binary Mapping* The authors fist map the nucleotides to binary sequences in the following way: $A \rightarrow 00, T \rightarrow 01, C \rightarrow 10, G \rightarrow 11$. Then a binary code is developed in a way that the number of G and C nucleotides in the final DNA sequences is constant for all codewords. This property is specially desirable because it is related to the speed of chemical reactions.

In [2] the authors apply error correcting codes to decrease the error rate in DNA computing. Using this error correction technique with vector quantization methods, they also propose a new way to implement DNA databases

with associative search queries (similar to their neural networks counterparts). Their idea of implementing an error correcting method in the context of molecular biology is very interesting: First, they generate some codewords over an alphabet of 4 letters using any appropriate coding method. Then, they synthesize DNA strands that are composed of two parts with exactly the same size: the first part is the original codeword and the second one is a corrupted version of the codeword with some noise probability (the same as the one that occur during the computation process). If for each codeword we produce many such DNA strands, we get many versions of the channel output in the second part of the strand. In the process of error correction, we are given a probe and would like to find the codeword it represents. Since we have many corrupted versions of the codeword, chances are that one of them matches the probe. In that case, they react and form a double helix with the original codeword attached to it as a single strand. We can then extract the double helix part and retrieve the single strand.

4 DNA Computing and Neural Networks

It was quite surprising to see there has been attempts to design DNA computing modules based on the structure of neural networks. One talk in the upcoming workshop "Statistical Mechanics and Computation of DNA self-assembly" is also dedicated to this concept [3].

The idea here is that neural networks are good at doing fault tolerant computations. So why not try to implement a similar concepts using DNA molecules? Mills et al. have made an attempt to accomplish this goal by proposing a novel mechanism to develop a neural network in which the axons and neurons are replaced by the diffusion and molecular recognition of DNA [7].

In a very interesting paper, Kim, Hopfield and Winfree have developed another framework to implement any neural network using DNA molecules [4]. They key element is a *DNA switch* which is a normal single stranded DNA molecule that can be in either of the two states ON of OFF. Each such DNA switch has several input/output terminals by means of which it affect other DNA switches by releasing RNA molecules. In neuroscience parlance, DNA switches act as neurons and the concentration of RNA strands act as spikes. Similar to neural networks then, we will have a network of DNA switches that can collaborate with each other to perform computational tasks. The authors

have provided several examples of such computational tasks. However, they only talk about how we can have a general DNA neural network. The issue of fault tolerance remains an open issue.

In [2] the authors use error correcting algorithm to implement an association mechanism similar to neural associative memory. In a nutshell, the problem to solve is that we have a huge database formed as encoded DNA strands. We would like to retrieve the closest stored vector to the query pattern. For DNA association systems, we must design proper DNA words to be stored as our patterns, such that the probability of mismatch in reduced. Another criterion that has to be as low as possible for good DNA words is the probability of a single stranded DNA fold back to itself. The Authors have proposed a method for associative searches when all we are required is to return all stored vectors that are in distance d from the query vector. The idea is to cluster patterns that are in distance d from each other, assign them a center codeword to represent the cluster and then do a normal error correction problem (as explained above) using the query vector and the encoded version (according to the above method) of the center codewords. The authors have also extend the above approach to the case that we are required to return a vector that is in distance *at most* d from the query vector.

5 suggestions for Presentation subject

In this section, you will find a couple of suggestions on the topic of the presentation/poster for the workshop "Statistical Mechanics and Computation of DNA self-assembly".

5.1 Coding Theory to Design Proper DNA Codewords in Advance

The first idea, which is in line with [1] and [2] and many other similar works, is to use coding techniques to design DNA strands that first of all minimize probability of an error happening at all (by having large enough minimum distances) and secondly, in case of errors, can correct them to some extent. The main challenge then is to design proper codes based on the desired criteria in DNA computing, i.e. large minimum distance, constant number of G and C nucleotides in codewords, etc.

5.2 Neural Coding Theory to Perform Iterative Error Correction

The second idea, which I find more interesting and is in line with our current research on neural networks, is to employ the framework developed in [4] and [7] to design *iterative* error correcting algorithms for DNA computing. Because we have a molecular realization of neural associative memories and similar arguments to what we have developed in our recent papers also applies here, specially our work on constrained neural networks (the ITW paper).

Moreover, the idea about designing DNA neural networks is closely related to Gene Regulatory Networks (GRN) and how they evolve over time. We discussed possible applications of coding techniques in gene regulatory networks in detail as a part of my candidacy exam [5], [6]. There, the idea was to see if GRNs use coding techniques to overcome the issue of noise and if so, what type of coding algorithm is used. However, as correctly argued in the exam session by the committee members, it is not necessarily the case that GRNs use error correcting codes as we know them. Nevertheless, in the realm of DNA computing, since we are synthesizing DNA molecules ourselves, we can use any coding technique that we want.

6 Conclusion

This short report describes some possible applications of coding theory and neural networks in DNA computing. Based on current research in this field, a couple of suggestions was made for our presentation in the upcoming workshop "Statistical Mechanics and Computation of DNA self-assembly". The suggested subjects were selected according to their relevance to our current research projects as well as the topics of interest in the mentioned workshop.

References

[1] O. Milenkovic, N. Kashyap, "On the Design of Codes for DNA Computing" Lecture Notes in Computer Science, Vol. 3969, 2006, pp. 100-119.

[2] J. H. Reif, T. H. LaBean, "Computationally Inspired Biotechnologies: Improved DNA Synthesis and Associative Search Using Error-

Correcting Codes and Vector-Quantization", Lecture Notes in Computer Science, Vol. 2054, 2001, pp. 145-172.

[3] Q. Lulu, "Building a DNA brain", To be presented at Statistical Mechanics and Computation of DNA self-assembly, May 2011, Finland.

[4] J. Kim, J. J. Hopfield, E. Winfree, "Neural network computation by in vitro transcriptional circuits", Advances in Neural Information Processing Systems (NIPS), Vol. 17, 2004, pp. 681-688.

[5] Amir Hesam Salavati, Applications of Coding Theory in Biological Systems, PhD candidacy exam report, Ecole Polytechnique Federale de Lausanne (EPFL), June 2010

[6] Amir Hesam Salavati, Applications of Coding Theory in Molecular Biology: An Overview Technical Report, Ecole Polytechnique Federale de Lausanne (EPFL), March 2010

[7] A. P. Mills Jr., M. Turberfield, A. J. Turberfield, B. Yurke, P. M. Platzman, "Experimental Aspects of DNA Neural Networks", Soft Computing - A Fusion of foundations, Methodologies and Applications, Vol. 5, No. 1, 2001, pp. 10-18.