A Report on the "Statistical Mechanics and Computation of DNA Self-assembly" Workshop

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 6, 2011

1 Introduction

This is a summary of the papers presented during the workshop "Statistical Mechanics and Computation of DNA self-assembly" which was held from 25 to 28 of May 2011 in Mariehamn, Finland. As the name suggests, the workshop consisted of a part on statistical mechanics of DNA self-assembly and another one on using DNA self-assembly to perform computational tasks. In what follows, I will briefly describe the presented papers. My goal is to just explain the main ideas and the results of each individual works and, hence, I will not go into much detail. Furthermore, I will not discuss principles of DNA computing as it has been explained in a separate report [19].

2 Statistical Mechanics of DNA Self Assembly

In this section, I will discuss the presentations on statistical mechanics of DNA self-assembly.

2.1 May 25

The conference started at 17 : 00 on May 26, 2011 by Prof. Karamer's talk on molecular beacons [1]. Molecular beacons are DNA strands that emit light when bound to a particular strand (their complement). As a result, once such reaction occurs, we will be able to detect it merely by having a light detector. This greatly simplifies the procedure for reading the output in DNA computing since one has to search for the solutions among all the other unwanted DNA sequences inside the test tube.

The main idea behind molecular beacons is to have a single stranded DNA molecule that falls back to itself and forms a *hairpin*, as shown in figure 1. Moreover, attached to one side of this strand is a fluorescent source that emits light with a particular wavelength. Attached to the other side is a light absorbing source that absorbs the same wavelength emitted by the fluorescent source. As a result, once the strand forms the *hairpin* structure, no light is emitted by the strand.

However, if the strand becomes close enough to its complement, they form a stronger bound and as a result, the hairpin will straighten up. Consequently, the fluorescent source and the absorbing ends are not near each

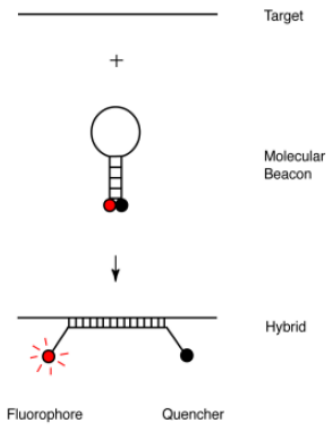

Figure 1: Molecular beacons and how they work [1].

other any more and there will be light in the test tube because of this reaction as illustrated in figure 2.

Molecular beacons have lots of applications. One example is to search for the genome of a particular virus in human blood by designing a molecular beacon whose sequence is complementary to the genome of the virus. In that case, if the virus is present in one's blood, we will observe some light in the test samples. Their team has solved the mutation issue in viruses by designing a sequence longer than what is required which makes the system tolerable to a number of mutations in the sequence. Another application comes from the fact that we can have different sequences with different fluorescent colors each of which react to different structures. Hence, we will be able to search for different patterns in a given sample at once.

The next talk by Prof. Dietz from technical university of Munich was on *DNA Origami* [2]. In brief, DNA origami is the art of constructing two and three dimensional nano-structures using DNA strands. The main technique, as explained during the talk, is to use a long available DNA strand and then fold it in the desired way by using smaller DNA strands, called scaffolds. We have to add these specifically designed scaffolds to particular sites in the long DNA strand in order to have the desired shape. Using this approach,

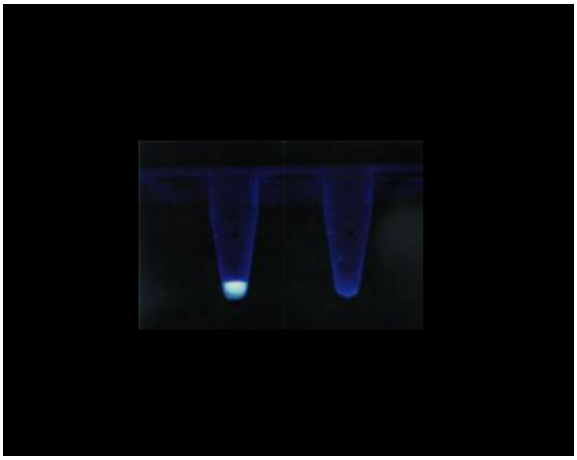

Figure 2: Existence of light in the left tube indicates desired biochemical reactions [1].

we literally can build whatever shape we want. Possible applications are in designing nano-devices for nano-machines.

2.2 May 26

The session on statistical mechanics continued on May 26 by several talks closely related to that of [2] on the day earlier. Prof. William Shih from Harvard medical school gave a talk on how to construct $3D$ nano-structures using DNA strands. The technique employed by their team is construct $2D$ DNA structures, which is relatively easy nowadays, and stack them to end up with a $3d$ structure [3].

Since $3D$ DNA origami structures are becoming more and more important in the realm of nano-devices, there have been attempts to design special computer packages and *CADs* to help researchers visualize the $3D$ structure in a high-end software which automatically give them the DNA sequence once the desired shape is finalized. One example of such packages was written by Prof. HGBERG, Bjrn as an add on to autodesk Maya [5], which is a specifically designed software to design $3D$ animations.

The next two talks in this session was about DNA and RNA thermody-

namics. The first one summarized the attempts of a team from the university of Barcelona to estimate the estimate the free energy of an RNA molecule [6]. The idea was quite simple and elegant: apply force to both ends of a single stranded RNA molecule and keep pulling until it tears apart. The amount of required force for the strand to tear apart is then proportional to its free energy.

The next talk was on modeling DNA as a collection of solid balls and springs to model nucleotides and bounds between them [7]. Despite its simplicity, experimental results show that the model is rather accurate in estimating the free energy of DNA molecules.

3 Computation of DNA Self-assembly

The second part of the workshop was on computational aspects of DNA self-assembly, i.e. using DNA strands to perform computational tasks. It was held on May 27 and May 28, 2011.

3.1 May 27

The session started with a fascinating talk by Prof. David Soloveichik from Caltech about computing with DNA strands [8]. They have developed a chemical framework to perform computational tasks which is based three simple rules:

1. If two complementary single stranded DNA molecules become close enough to each other, they bind and form a double stranded DNA molecule.

2. If two strands are attached together in only a small section break the bound and separate them.

3. If two strands are bound together, replace one of the strands with a third strand only if the new bound results is a longer DNA strand with stinger bounds.

Using the above simple rules, one can construct logic gates and virtually any other computer circuit to perform computation. Together with Microsoft research, they have even written a computer package to simplify the circuit design process [9]. Using this package, one can design circuits as

usual and then the software return the sequences for the DNA strands used in the circuit. Overall, since we have simple operations such as AND, OR, $AMPLIFY$, etc., we can construct any computational circuit using these simple biochemical modules. The team is now working on implementing the proposed model in the test tube where still there are some works required to do in order for the results to match theory.

On a rather different topic, Prof. Jarkko Kari from the University of Turku discussed the problem of termination in DNA self-assembly [10]. He addressed the question of unbounded growth in DNA self-assembly, i.e. given a DNA strand and a set of rules for its growth, can we determine if the growing procedure terminates or it continues unboundedly. This process is widely modeled as a tiling problem. Using such models, the speaker showed that this problem is NP-hard and undecidable even in the simplest cases. He proved this by showing that given a set of Wang tiles, no algorithm is able to determine if the tiles can form an infinite (planar) path where consecutive tiles match with each other.

DNA codeword design problem was another topic addressed in this session. Prof. Max Garzon from the University of Memphis explained his and his team's research on finding suitable DNA sequences for DNA computing [11]. Briefly speaking, we are given a set of constraints and required to find DNA strands that comply with these constraint. For instance, they must be designed in such a way that unwanted strands do not bind together or strands fold back on themselves. For a more comprehensive list of constraints see [12].

One possible constraint is to have large Hamming distance between all DNA codewords. However, this is not enough since due to spatial structures of DNA strands, two strands might bind even if they have large Hamming distances. Hence, they have developed an *in vitro* genetic algorithm to find the appropriate sequences. The idea is to produce all possible strands and filter out the unsuitable ones. Filtering is done by putting all generated sequences in a tube and extracting all those that bind together [12]. For DNA sequences with 20 nucleotides, this approach yields between 50000 to 100000 appropriate DNA codewords.

In the second part of his talk, Prof. Garzon talked about using coding theoretical approaches to find proper DNA codewords. To this end, we must utilize a different distance measure than the usual Hamming distance to better capture the spatial structure of the DNA molecules. They have developed a new metric which combines the Hamming distance with the $3D$ structure

of the DNA sequence. Using this metric, they show that the problem of finding suitable DNA codewords is the same as the sphere packing problem in coding theory. Therefore, whatever method used to find dense sphere packs can be applied to determine DNA codewords as well [11]. The speaker also mentioned applying this technique to find phylogenetic trees.

Although it is possible to perform computational tasks with DNA molecules, an important issue in DNA computers is the fact that DNA molecules are consumed. Hence, we must be careful with how much material we consume in DNA computations and recycle the "wastes" resulted in computational operations as unwanted DNA strands. Prof. Anne Condon from the University of British Columbia address the recycling issue [13] and showed that we can design a $3-bit$ counter with DNA strands that consumes less amount of material than its counterparts.

The idea of a $3-bit$ DNA counter is quite interesting: we must have three *pairs* of DNA strands, one for each bit. Then, in each pair one of the strands represent the digit 1 and the other one the digit 0. At any given moment, only one strand in each pair is present in the test tube. Hence, the three molecules present in the test tube at any given moment represent the state of the counter. Now the brilliant point is the transition rule between these strands and formulating them in terms of chemical reactions. In principle, this procedure is closely related to Chemical Reaction Networks (CRN) of [8].

Prof. Condon gave an example of such reactions and how the counter work indeed. She also showed the recycling which is based on a simple trick: to count from 0 to 7, we must have 7 reactions. Out of these 7, 3 of them are the opposite direction of the other 3. Hence, all the wastes produced in one cycle will again be used in another cycle later (except for one remaining reaction which is a result of the odd number of reactions). The main drawback of this approach is that it is slow and we must have only one single copy of each strand. The latter is the main reason this technique is not yet implemented in practice.

Biochemical neural networks was another presented technique to employ DNA sequences to perform computation. Dr. Lulu Qian from Caltech explained their team's attempts to implement a biochemical neural networks using DNA strands [14]. The key component in their model is a *DNA switch*, which is a DNA strand that can have two conformations, which we call UP and $DOWN$. Each DNA switch is represented by a *unique double stranded* DNA molecule. The conformation of this DNA strand indicates the state of the neuron. Furthermore, for each neuron we have a single stranded DNA

molecule with the opposite conformation which we replaces the current conformation in the double stranded DNA molecule, changes the state of the neuron.

Having these DNA switches, we will be able to implement logic gates and amplifiers. For instance, an AND gate could be implemented by the chemical reaction $A+B \rightarrow C$ when C is only produced when both A and B are present. Similarly, amplification can be realized by the reaction $A+F \rightarrow 2A$, where F is what is called a *fuel molecule*. We can also have threshold gates, i.e. gates that once the amount of fuel molecule exceeds a threshold some reaction is fulfilled. Hence, we can design biochemical circuits in a high level manner and implement it using DNA sequences. Interestingly, there will be a computer package available soon which will simplify the whole process as the user can design logical circuits in a high level fashion and get the sequence for the strands that implement the circuit.

In order to construct a biochemical neuron using DNA switches, they have used the fuel molecules as amplifiers to act as connection weights. Then, in a neural network with size n, we need n such amplifiers for each neuron and a threshold gate, as shown in figure 3. Using these neurons, the speaker presented an example of the Hopfield network with four neurons that could memorize four patterns.

Note that this is a different scheme from the other biochemical neural network designed by this lab where they use transcriptional circuits [16]. While this approach is more scalable in terms of the size of the circuit one can build, the method proposed in [16] is more suitable for dynamic and interconnected neural networks as the above approach only works for feed-forward neural networks since the proposed "neurons" will freeze in their states once it is decided. There are other limitation to the proposed approach at this point too. One of the more important ones is the lack of learning ability, which seems solvable in near future.

In the last talk of this session, Dr. Eugen Czeizler presented an interesting method to design nano-circuits based on nanotubes mounted on DNA strands as platforms [17]. The proposed approach is to use DNA strands as *building blocks* of circuits and design some simple blocks which we can then use to produce any type of electrical circuit. The suggested method was also closely related to DNA origami.

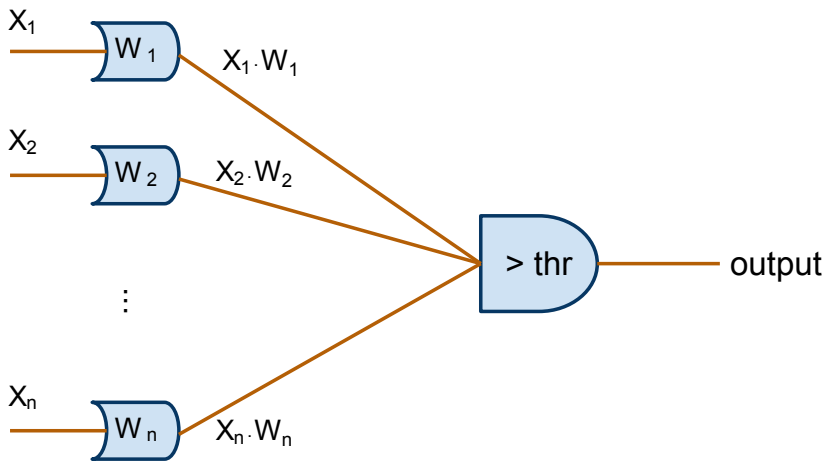

Figure 3: Schematic of a biochemical neuron based on DNA strands [14].

3.2 May 28

On the last day of the workshop, Prof. Erik Winfree from Caltech gave an interesting talk on the algorithm self-assembly of DNA structure [18]. As one of the first pioneers of the field, he explained the theory behind DNA self-assembly, summarized past achievements and explained future challenges. The idea is that we can model the process of DNA self-assembly as a tiling problem. Now the goal would be to construct different tile patterns using a given set of tiles.

The type of pattern we can build depends on the the initial set of tiles. Nevertheless, given a proper set of tiles and a number of simple rules, it can be shown that one can build any sort of structure using these tiles as well as *performing computational tasks, e.g. evaluate a function of some inputs*. More specifically, if we put n tiles together and define some simple assembly rules, the structure will grow to a pattern of tiles with 2^n tiles and an algorithm shape. The initial pattern determine the final shape.

Another interesting topic in his talk was the use of fault-tolerant tiling systems, i.e. the automatic self-assembling structures that will correct tiling errors occurred during the process.

References

[1] F. Kramer, "Finicky and sloppy molecular beacons", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[2] H. Dietz, "Sequence, shape, function: a primer to DNA origami", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[3] W. Shih, "Self-assembly of DNA into nanoscale three-dimensional shapes", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[4] B. Hgberg, "Free-form design of 3D DNA nanostructures using vHelix for Autodesk Maya", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[5] http://usa.autodesk.com/maya/

[6] J. M. Huguet, "Thermodynamics of RNA hybridization inferred from out of equilibrium unzipping experiments", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[7] T. Ouldridge, "Coarse-grained modelling of DNA and DNA self-assembly", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[8] D. Soloveichik, "Toward a programmable chemistry with strand displacement cascades", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[9] http://lepton.research.microsoft.com/webdna/

[10] J. Kari, "The termination problem in self-assembly", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[11] M. Garzon, "DNA codeword design: theory and application", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[12] A. J. Neel, M. H. Garzon, "DNA-based memories: a survey", Studies in Computational Intelligence, Vol. 113, 2008, pp. 259-275.

[13] A. Condon, "On recycling and its limits in strand displacement systems", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[14] L. Qian, "Building a DNA brain", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[15] L. Qian, D. Soloveichik, E. winfree, "Efficient Turing-universal computation with DNA polymers", Lecture Notes in Computer Science, Vol. 6518, 2011, pp. 123-140.

[16] 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.

[17] E. Czeizler, "A Design Framework for Carbon Nanotube Circuits Affixed on DNA Origami Tiles", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[18] E. Winfree, "Algorithmic self-assembly of DNA: theory and experiment", Workshop on *Statistical mechanics and computation of DNA self-assembly*, Finland, 2011.

[19] R. K. Kumar, A. H. Salavati, "DNA computing and why it might be interesting for a coding theorist", Technical report, EPFL, May 2011.