

DNA Computing

IT3708

Author: Eirik Baug



NTNU

Norwegian University of
Science and Technology

April 18, 2018

1 Literature Search Protocol

1.1 Research Questions

I have identified the the following research questions.

- How can problems be solved by using biological operations on DNA strands, and why could this be the preferred way to solve a problem?
- How can DNA computing technology possibly be used in the future?

The research questions are inspired from my lack of understanding of how DNA computing works and how it is applied in practice. I wish to expand my knowledge on the topic by understanding how DNA strands can be computed and used to express solutions to hard computational problems and what problems can be solved by such technology.

1.2 Research Strategy

To find relevant articles to answer the preceding research questions a few criteria has to be met. First of all only relevant search engines like Ieeexplore, Google Scholar or similar sites, as well as the course textbook will be used as credible literature. Additionally, the relevancy of the articles will have to be assessed, firstly by looking at the publishing date. When looking for future applications of the technology I wish to only select relatively recent articles (2016 to 2018) as this is the most interesting for a field that is constantly changing. However, when it comes to finding relevant theory on the subjects older articles might be just as relevant to explain the underlying concepts of the techniques. I will also only select papers that give me a broader understanding of the field, as learning really specific techniques is not my intention for this research.

Search queries used includes words like **DNA computing, theory, practice, application** and **future**. Included articles will have to contain DNA computing as its main topic as well as at least one of the other relevant search words (**IC1**). I also wish to include articles that explain how the methods discussed can used in a relevant context (**IC2**) and articles that discuss how the technology can be used and further developed in the future **IC3**.

Additionally, a set of **quality criteria** has been added to assess the quality of the articles in relation to my search. First of all, the article needs to be understandable in that it provides enough theory and background information for me to actually understand what it is presenting (**QC1**). Also, sources must be properly cited in the article and the article must come from a credible source (e.g. a university)(**QC2**). The research must also clearly state what it aims to accomplish and it must be clear that the research is actually useful to the field (**QC3**).

2 DNA Computing, Theory and Application

2.1 Introduction

DNA computing is an approach to parallel problem solving which differs quite a bit from how problems are normally solved in computers today. The concept was developed by Leonard Adleman in 1994 when he attempted to show how DNA molecules could be used to find the shortest path in the traveling salesman problem. The method involves encoding every part of the solution space as unique sequences of nucleotides. While one city could be encoded as TGAAT and another ATAGAT the route between them could be encoded as the concatenation of the complementary strands, where the second half of the first strand is the start city and first part of the second strand corresponds to the ending city. For this example this would be AATATA.

These simple DNA strands can reproduce extremely fast with several strands replicating in parallel, and after a few iterations exceed the data-processing capacity of the fastest supercomputers available today. While the DNA strand reproduction rate is very large the amount of strands required for more complex problems are so large that Adleman was uncertain as to what problems could actually be efficiently solved by this type of computing [1].

However, what Adleman did effectively do was open up a new field of computing where problems could be solved in a new and biologically inspired way, allowing for solving problems that previously seemed too complex for conventional algorithms.

2.2 Background Theory

Adelmans experiment was a so called proof of principle where the possibility of such a system was shown. Since then the area of DNA computing has become a multidisciplinary area, and DNA computing has been applied a lot of different problems.

As one might expect DNA computing is based on molecular biology and attempt to solve complex problem by taking inspiration from how DNA molecules are combined in nature. To elaborate further on the DNA strands presented in the introduction; a DNA molecule is made of nucleotides. The nucleotides all includes one of four distinct bases; adenine, guanine, cytosine and thymine which are simplified as A, G, C and T respectively. As it is the bases that differentiate the nucleotides it is with these they are identified. In the preceding section the concept of DNA strands was introduced which is how potential solutions are presented. Single-stranded DNA molecules are simply chains where nucleotides are bound together by a strong covalent bond [2].

Another aspect of working with DNA is the concept of complementary bases. Strands attract and combine based on the bases. A bonds with T and G bonds with C [2]. It seems that combination of these attracting bases is what is known as complementary strands which is what most of the DNA computing field is based on by making nucleotides naturally attract and combine with each other through a process called annealing [3][2]. I believe that the concept of complementary bases is probably the most important aspect of DNA computing as it defines how nucleotides attract each other making it possible for new solutions to form.

2.3 Bio-chemical Operations

DNA computing involves doing biological operations on a set of DNA strands. Exactly what operations are used may differ from what DNA computing model is used, but what they have in common is that they are all inspired from already existing biological operations. Some of the most important operations are mentioned below. These techniques are usually ran in iterations until a satisfiable strand has been found [1], but what these operations have in common is that they all exploit nucleotides natural tendency to merge and split under certain conditions.

One of the most important operations on the DNA strand is the synthesis, which is the process of designing and restructuring the nucleotides in the strand to form a new strand [3][2]. When solving computational computer problems this will be important for finding new solutions to the given problem since when the nucleotides are restructured a new solution to the problem is also formed.

With annealing being the process of combining DNA strands it is also important to include the process of splitting the DNA strand to again obtain its single stranded form. This process is what is referred to as denaturation which in nature occurs when the temperature of a solution raises to the point where the hydrogen bonds between the strands are split [3][2].

I believe these operations could be observed by putting a lot of DNA strands together in a test tube in a lab and expose the tube to different temperatures to make the strands interact split and with each other. This would be sort of like a primitive DNA computer. The different bio-chemical operations would all work in parallel because of the DNA strands natural tendencies to operate independently and attract. The difficult part of using this in a digitalized system will probably be to integrate this functionally in computer hardware, manipulate the DNA in a satisfactory way, as well as implementing and making the system understand logic. We can control the DNA to a certain degree by exposing it to different conditions, but in the end DNA strands combine and split in natural ways that we cannot completely predict. The trick will be to manipulate enough DNA strands at the same time until one of them gives us the desired solution. If we could combine this sort of naturally "brute-force" approach where nucleotides form all different solutions with some intelligent decisions that manipulates the DNA to form in a certain way I believe we could have something truly powerful. Maybe by manipulate the temperature of the DNA strand population we could also manipulate the way the strands form as strands tend to split at certain temperatures.

2.4 Why choose DNA computing?

After looking at some of the background that is what makes the field of DNA computing possible it is important to consider why one would choose such a method which for most people is a lot less intuitive than traditional programming.

The main advantage of solving problems with the help of DNA is the enormous parallelism as strands are formed in parallel. Billions of DNA molecules can in theory function in parallel in a DNA computing system [4]. This in turn ensures good scalability for problems like for instance a cumulative XOR computation system developed by Nadrian C. Seeman where the time complexity increases linearly with the problem size [4]. This is one of the most important aspects since in traditional algorithms these complex problems scale exponentially with the problem size. This is also shown by Adleman's traveling salesman experiment and due to the enormous parallelism available in the DNA molecules[1]. It seems to me that each DNA molecule is basically its own mini-computer. This does, however, not ensure that the problems are always solved faster with DNA computing than with traditional computing algorithms, so it is important, at least with current limitations, to select the correct problems to apply the techniques to and preferably select problems that are too complex for traditional systems as the parallelism and non-deterministic nature of DNA computing might solve such problems faster.

At the moment the main area where DNA computing computing is applied in an efficient

manner is when solving certain NP problems; problems that are so complex that traditional optimization algorithms are currently not able to solve them in polynomial time [5].

DNA computing techniques are typically non-deterministic which makes them similar to most other evolutionary algorithms. Solutions are found by running a number of iterations letting DNA strands naturally split and combine until a satisfactory DNA solution is found. This means that while it is not always able to solve problems faster than traditional algorithms on non-NP problems DNA computing can solve all types of computational problems with a varying degree of efficiency by using its immense capacity for parallelism and non-deterministic properties.

DNA computing is something that is generally associated with future technology where we may be able to create computers able to exploit the enormous parallelism of the nucleotides. However, today a lot of DNA computing algorithms suffer from being extremely time consuming which Adleman also noted in his experiment [1] that the amount of strands needed to be checked for a solution is immense. While there are a lot of challenges the DNA computing field has made some breakthroughs. For instance, DNA computing has been successfully applied to logic functions and boolean circuits where Ogiwara and Ray first proposed a boolean circuit system with a time complexity of $O(1)$ [3]. DNA computing techniques have also been successfully applied to cryptographic algorithms, but while these algorithms have a linear time complexity they seem to often still be slower than their traditional algorithm counterparts. It looks like the largest limitation of DNA computing is in the current hardware. It will be exciting in the future to see if we can build proper hardware suited for these types of algorithms. The speed of current computer hardware will probably reach a limit to how fast it can become at some point, but a system based on DNA computing could theoretically be limitless in its capacity for parallelism, and therefore potentially greatly outperform traditional computers, which makes it a field worth researching for the future. I believe avoiding the bottleneck of evaluating every strand in sequence will be important for this goal as a system is only as strong as its weakest link.

3 Potential Application Areas

In my opinion the most exciting aspect of DNA computing is not what it can do right now, but rather what it can do in the future. As mentioned earlier DNA computing today suffers from not having proper hardware able to exploit the enormous theoretical parallelism of these types of algorithms. A theoretical DNA computer would allow for billions of DNA strands to operate in parallel at the same time and solve problems in polynomial time that would otherwise be impossible or too resource exhaustive for conventional computers to solve. Systems like this have already been proposed to some degree with some really successful attempts. For instance in 2002 a research group at Weizmann Institute of Science created a molecular computing machine composed of enzymes and DNA molecules [6].

In 2004 an autonomous bio-molecular computer that logically analyses the levels of messenger RNA species and based on this creates a molecule that is capable of affecting levels of gene expression. Such a computer could potentially be capable of diagnosing cancer and producing an anti-cancer drug in the future [7].

However, to get a fully operational DNA computer that is able to solve a larger range of problems than what is possible today a new type of arithmetic and logical unit (ALU) would need to be proposed. This would make the DNA computer more of a general purpose machine at the same time as being able to solve NP problems that traditional systems cannot efficiently solve today. The computer would then also be able to compute logical operations like NOT, OR, AND, XOR etc. that a normal computer today would find easy, but DNA computers have a harder time computing because of its' non-deterministic nature. With this functionality a programmer would be able to enter logical language directly into the machine without any additional alterations, and thus the computer would be able to implement conventional computing architectures. Techniques for such an ALU unit has been proposed in [6], but has yet to be implemented in practice, but it seems relatively likely that at some point we will have a general purpose DNA computer.

While I do not doubt that such a system is feasible it is beyond my ability and understanding of the subjects to declare whether such a system will be able to outperform conventional computers. While it seems that traditional computers are able to do most things faster because of their deterministic nature, DNA computers with parallelism and non-determinism makes DNA computers solve a lot of problems which are too hard for conventional computers. If the parallelism can make the DNA computer compute normal arithmetic operations faster than a traditional computer then the DNA computer would likely be superior. However, because of how efficient traditional computers are at this task today I doubt this will be the case. Even with enormous parallelism these operations are still not implemented in a deterministic manner in a DNA computer, as opposed to a conventional computer where these simple operations are very easy, meaning that simple decisions might take a long time to compute in DNA computers as a large amount of solutions may need to be tested. I think the success of DNA computers will come rely a lot on how effective we can become at parallelising and processing the operations. The potential is huge, but after all data or strands have been computed some system would have to verify that we have the correct solution, and this could possibly cause a bottleneck in the system as every solution needs to be checked.

Lastly, it would be really interesting in the future to see DNA hardware applied to the field of robotics. Maybe by taking inspiration from biochemical operations robots and other artificial intelligent programs could create intelligent systems that before has been too complex for current AI and therefore take a step in the direction of a true general intelligence.

4 Conclusion

DNA computing is a field of computing that attempts to solve problems by applying biochemical operations to strands of nucleotides. These operations allow for splitting and combining DNA strands such that strands containing a valid solution will naturally form after a number of iterations and combinations of complementary strands. It attempts to exploit the enormous parallelism and non-deterministic nature of these types of operations to solve NP problems in polynomial time [2]. The time complexity of such algorithms generally grows linearly with the size of the problem. However, that does not always mean that these algorithms always outperform conventional algorithms for the same problems as the amount of DNA strands required to solve a relatively simple optimization problem become extremely large really fast [1]. A lot of success have still been achieved in recent years by applying several different DNA computing models different NP problems [6].

DNA computing is a field that is still very much evolving. Ever since it was first proposed in 1994 several different models have been proposed, able to solve different problems of varying degree of difficulty, but we still have not achieved a general purpose DNA computer. However, recently a DNA computing ALU have been theorized for this goal [6] so that a general purpose DNA computer will probably be achievable in the future. Moreover, how well such a computer will be able to perform in comparison to traditional computers is still uncertain, but my guess would be that while DNA computers will be superior when solving complex NP problems traditional computers will still be the stronger choice for a lot of simpler conventional computing, at least for a while yet until the extreme theoretical parallelism of a DNA computer is able to outperform our conventional computers in every way.

References

- [1] Dario Floreano and Claudio Mattiussi. *Bip-Inspired Artificial Intelligence*. Massachusetts Institute of Technology, 2008.
- [2] Martyn Amos et al. “Topics in the theory of DNA computing”. In: *Theoretical Computer Science* 287.1 (2002). Natural Computing, pp. 3–38. ISSN: 0304-3975. DOI: [https://doi.org/10.1016/S0304-3975\(02\)00134-2](https://doi.org/10.1016/S0304-3975(02)00134-2). URL: <http://www.sciencedirect.com/science/article/pii/S0304397502001342>.
- [3] K. Boruah and J. Ch. Dutta. “Development of a DNA computing model for Boolean Circuit”. In: *2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. Feb. 2016, pp. 301–304. DOI: 10.1109/AEEICB.2016.7538295.
- [4] D. I. Lewin. “DNA computing”. In: *Computing in Science Engineering* 4.3 (May 2002), pp. 5–8. ISSN: 1521-9615. DOI: 10.1109/5992.998634.
- [5] Eric Bach et al. “DNA Models and Algorithms for NP-Complete Problems”. In: *Journal of Computer and System Sciences* 57.2 (1998), pp. 172–186. ISSN: 0022-0000. DOI: <https://doi.org/10.1006/jcss.1998.1586>. URL: <http://www.sciencedirect.com/science/article/pii/S0022000098915860>.
- [6] M. Sarkar, P. Ghosal, and S. P. Mohanty. “Exploring the Feasibility of a DNA Computer: Design of an ALU Using Sticker-Based DNA Model”. In: *IEEE Transactions on NanoBio-science* 16.6 (Sept. 2017), pp. 383–399. ISSN: 1536-1241. DOI: 10.1109/TNB.2017.2726682.
- [7] “An autonomous molecular computer for logical control of gene expression”. In: (2004). URL: <https://www.nature.com/articles/nature02551>.