

Challenging Traditional Approaches to Computation: A Biomolecular Transducer Employing Ternary Language and Rendering a Biological Output

Mark Chaskes
44 Dauphin Drive
Williamsville, NY 14221
USA

Paul Lazarescu
20 Indiana
Irvine, CA 92606
USA

Mentor: Tamar Ratner
The Schulich Faculty of Chemistry
Technion, Haifa, Israel, 32000

Abstract

Biomolecular computing is a new field of research merging several sciences. Previous works were based on finite state automata and have had limited computing capabilities. In this project, a transducer model was used in order to design a biomolecular machine with greater computing potential. In this research two automata were designed to be used as software for the transducer: one able to divide a ternary input by three and another by two. Using a plasmid this transducer could perform multiple computations consecutively, using either of the two automata.

Introduction

Biomolecular computing is a relatively new science; the first article based on finite state automata was published only about a decade ago. In 1994, this topic of research ensued following Adleman's theoretical model demonstrating that laboratory molecular biology could be used to create a DNA based computation machine, *in vitro*¹. The first model was a proof of concept using a simple automata and it laid the foundation for higher computation². The potential provided by DNA based computing machines can have applications ranging from medicine to agriculture, due to the direct interface between the computation and the biological process. Subsequently, research groups have been working on broadening the capabilities and increasing the practicality of these biomolecular devices. Yaakov Benenson's experimental work created workable 2-symbol 2-state automaton which was the first biomolecular automaton built (Fig. 1)². Michal Soreni's work expanded into the production of more complex automata: a 3-symbol 3-state automaton greatly increased the number of possible functions³. Benenson also designed an automaton which regulated gene expression as the output, *in vitro*⁴. Elizaveta Kossoy's work further advanced the understanding of biomolecular computing, as she created an automaton that had a bacterial phenotype output, *in vivo*⁵.

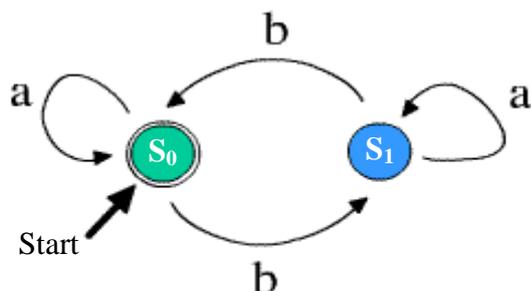


Figure 1: Schematic presentation of a 2-symbol (a and b) 2-state (S_0 and S_1) automaton.

Finite automata are theoretical models that can read sequences of symbols using a reading head, and can be in any one of a given number of states. These biomolecular devices are composed of different individual components which are placed in solution together. The hardware (Fig. 2A) consists of restriction enzymes (Endonuclease Type II), DNA ligase enzyme, and ATP (which provides energy). The input (Fig. 2B) is a double stranded DNA (dsDNA) sequence including a restriction enzyme recognition site, spacers (to

determine the initial cleavage site, which is the initial state of the automaton), a predetermined sequence, and a terminator. The software (Fig. 2C) is a set of dsDNA transition molecules which represent the transition rules. Each transition molecule includes a restriction enzyme recognition site, a spacer which determines the device's next state (Fig. 3), and a unique four nucleotide sticky end complimentary to the one resulting from cleaving the 'read' symbol at the specific state. Software determines the changes of the states according to the read information and the transition rules. The output is the final state in which the computation process ended. A detection molecule is used to detect that state (Fig. 2D). Each of the detection molecules has a unique four nucleotide sticky end complimentary to the sticky end of the terminator in the specific final state and a unique identifier to distinguish between the different molecules.

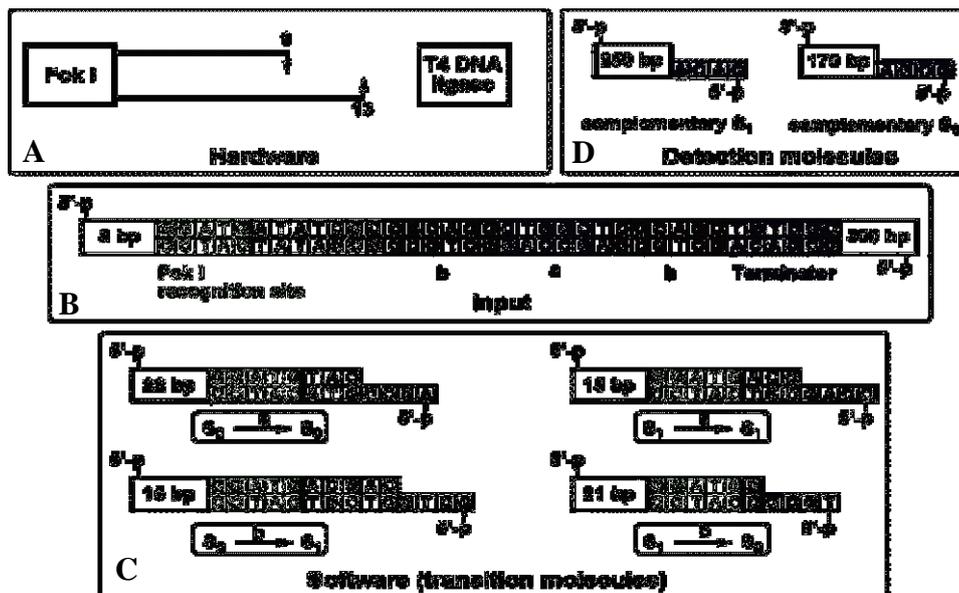


Figure 2: Components of 2-symbol 2-state Biomolecular Computing Device³. **A.** The hardware – restriction enzyme (FokI) and T4 DNA ligase. **B.** The input – dsDNA sequence including the recognition site (for FokI), sequence of symbols and the terminator. **C.** The software – dsDNA molecules including the recognition site (for FokI) and unique sticky end. **D.** The detection molecules – dsDNA with unique sticky ends and different lengths.

The first step of the computation occurs when restriction enzymes cleave the dsDNA based input and leaves four nucleotide sticky ends (an overhanging strand; Fig. 3). Only the transition molecule with a complementary sticky end to one of the cleaved inputs can hybridize. After this step the DNA ligase covalently bonds the transition molecule to the remainder of the input. The transition molecule contains a new recognition site for the restriction enzyme which enables this process to repeat itself. Next, the restriction enzymes again cleave the input, 'read' the next symbol at the given state, and again leave four nucleotide sticky ends. The new transition molecule then bonds to the sticky ends left by the enzyme. This cycle continues until all of the input has been processed and the terminator molecule has been reached. Once this region has been cleaved, only detection molecule which contains a complementary four nucleotide sticky end will hybridize to the terminator's sticky end. DNA ligase then covalently bonds the dsDNA.



Figure 3: Three ways to cleave a six nucleotide dsDNA symbol resulting in a four nucleotide sticky end. Cleaving outside the symbol represents state 0; one nucleotide inside the symbol represents state 1; two nucleotide bases inside the symbol represent state 2.

There are several ways to determine the final state of the automata. For example, gel electrophoresis, as used in Benenson's first paper², surface plasmon resonance (SPR), as used

in Soreni's paper³, etc. Both of these methods use the length of the output to determine in which state the computation ended. The length in each state varies due to different lengths of the specific detection molecules. Different methods can be used (other than length), to determine the output such as measurement of fluorescent colors, and most desirable technique - biological function indicating the final state.

Although biomolecular computing devices are significantly slower than their silica based counterpart, the full extent of their possibilities have not yet been completely realized. Due to the fact that a biomolecular computing device is easily assimilated into a biological system, these devices have the unique ability to communicate through a direct interface between the computing process and the system. This direct interaction can result in a biological response, such as the release of a suppressant⁴, a new phenotype⁵, etc. Biomolecular devices also have the ability to compute in parallel which allows an automaton to perform several computations on different inputs simultaneously. In addition, a biomolecular devices are microscopic in size however, can still store large amounts of data.

In this particular research project, a more complex computing device was designed. In place of finite state automata, a transducer was used. This device has many advantages over automata. Firstly, a transducer has a larger computing capacity than finite automata, due to the fact that a transducer can encode new information. Secondly, a transducer can create a biological output. This feature is attributable to the dsDNA's insertion into a plasmid (a circular vector found in bacteria). This ability allows the computer to 'read', encode and save information within the DNA sequences and to have a biological function as the output. An automaton can only process information and indicate its final state, which limits the number of possible programs directly influencing its complexity. The use of a plasmid, as suggested in this project, may revolutionize biomolecular computing. In the future it may have applications in the fields of medicine, agriculture, environment, etc.

Results

In this project a theoretical finite state transducer was designed. The advantage of

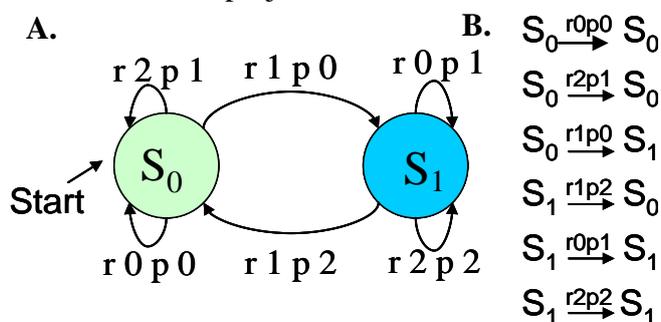


Figure 4: The divide-by-two transducer used in this project. The automaton always begins in state 0 (S_0). **A.** A schematic diagram of the automaton. 'r' represents the symbol 'read' and 'p' represents the symbol 'printed'. **B.** Transition rules of this automaton. For each transition rule there is one transition molecule.

transducers over automata is that they can 'print' data enabling them to perform more complex computations. Two different automata were created in order to demonstrate the abilities of this biomolecular machine: one can divide a number by three and the other can divide a number by two. The device is based on a ternary language (that includes only the numbers 0, 1, and 2). The initial step was designing the automata upon which the transducer would be based. A set of rules were established for each automaton. The transition rules encode for the written data and the state change when a symbol is 'read' in a certain state. For the division

by 2, a 3-symbol 2-state automaton was fashioned (Fig. 4). The final state reveals the decimal besides the whole number answer (which was 'printed').

The divide-by-three automaton (Fig. 5) is a three-symbol three-state program. In this case, the end state also represents the fraction left over. S_0 stands for 0/3 (or no remainder); S_1 stands for 1/3; and S_2 stands for 2/3. The written symbols of both automata will reveal the whole number numerical answer. The reading of the printed section is possible, but will not be discussed in this project

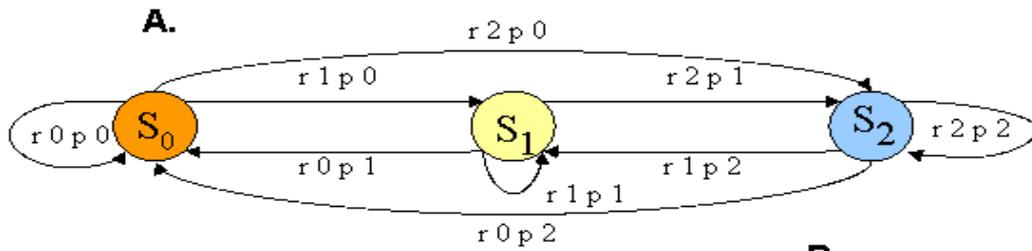
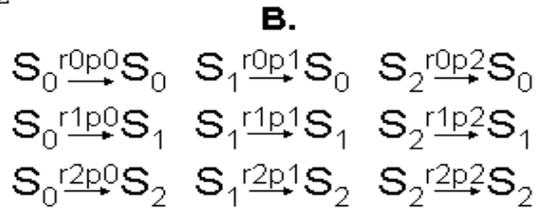


Figure 5: The divide-by-three transducer used in this project. The divide-by-three automaton is more complicated than the aforementioned divide-by-two automaton. **A.** A schematic diagram of the automaton. 'r' represents the symbol 'read' and 'p' represents the symbol 'printed'. **B.** Transition rules of this automaton.



To create a transducer that both 'reads' and 'writes', a procedure to insert the input into a plasmid can be used. The plasmid was cleaved by an EcoRI restriction enzyme; in essence, it is now a linear strand of dsDNA. The input hybridizes to the plasmid because of its sticky ends, which were designed to be complementary to the sticky ends left by EcoRI. Then, DNA ligase strengthens the connection by covalently re-bonding complimentary nucleotides.

	0	1	2	Terminator
Sequences decided upon for each symbol	AGTCTT TCAGAA	GGTATT CCATAA	CTCGTT GAGCAA	TGCTGA ACGACT
Sticky ends left by sequences cleaved in S ₀	AGTCTT AA	GGTATT AA	CTCGTT AA	TGCTGA CT
Sticky ends left by sequences cleaved in S ₁	GTCTT A	GTATT A	TCGTT A	GCTGA T
Sticky ends left by sequences cleaved in S ₂	TCTT	TATT	CGTT	CTGA

Figure 6: The symbols sequences cut in the different states, leaving unique sticky ends.

Plasmids have two advantages: to save the printed information so it will not be lost once the input is cleaved, and to have the ability to be transformed into bacteria. By doing so, the bacteria express's the output as a biological function. The plasmid that would be used can not contain a second recognition site for EcoRI or any of the other three restriction enzymes used for the computation process (BbvI, BseRI, EagI).

The computational process of a transducer requires the creation of many individual molecules for each of the automata, which each have a unique function. Because this project makes use of ternary, sequences that represent the numbers 0, 1, and 2 and sequence for the terminator was created (Fig. 7).

These six base pair sequences were created in such a way that they alone, or when put next to each other in the input, do not accidentally encode the recognition site for any of the three restriction enzymes used. These sequences were also created in such a way that when cleaved in any state, no sequence had the same sticky end.

To build the transducer, fifteen transition molecules were engineered (six for the 3-symbol 2-state transducer and nine for the 3-symbol 3-state transducer). These molecules are designed in such a way that they only bond to the dsDNA that contains its complementary sticky end. Each molecule is composed of a two nucleotide sticky end, AA, a written symbol, an eight nucleotide spacer, recognition sites for BseRI and BbvI, a varying length spacer and unique four nucleotide sticky end. Varying length spacers are required since the states of the transducer are defined by the cutting position of the restriction enzyme in the symbol, so the transition between different states is directly correlated to the length of the spacer. The four nucleotide sticky end on the right is complimentary to the sticky end obtained from cleaving the 'read' symbol at a specific state characteristic to the transition molecule. BseRI leaves a TT

S_0 to S_0 (0,0)
 AGTCTT...8 base...CTCCTCGCAGC...2 base
 AATCAGAA...pairs ...GAGGAGCGTCG...pairs...TCAG

S_0 to S_1 (1,0)
 AGTCTT...8 base...CTCCTCGCAGC...1 base
 AATCAGAA...pairs ...GAGGAGCGTCG...pairs...CCAT

S_0 to S_2 (2,0)
 AGTCTT...8 base...CTCCTCGCAGC
 AATCAGAA...pairs ...GAGGAGCGTCGGAGC

S_1 to S_0 (0,1)
 GGTATT...8 base...CTCCTCGCAGC...3 base
 AACCATAA...pairs ...GAGGAGCGTCG...pairs...CAGA

S_1 to S_1 (1,1)
 GGTATT...8 base...CTCCTCGCAGC...2 base
 AACCATAA...pairs ...GAGGAGCGTCG...pairs...CATA

S_1 to S_2 (2,1)
 GGTATT...8 base...CTCCTCGCAGC...1 base
 AACCATAA...pairs ...GAGGAGCGTCG...pair ...AGCA

S_2 to S_0 (0,2)
 CTCGTT...8 base...CTCCTCGCAGC...4 base
 AAGAGCAA...pairs ...GAGGAGCGTCG...pairs...AGAA

S_2 to S_1 (1,2)
 CTCGTT...8 base...CTCCTCGCAGC...3 base
 AAGAGCAA...pairs ...GAGGAGCGTCG...pairs...ATAA

S_2 to S_2 (2,2)
 CTCGTT...8 base...CTCCTCGCAGC...2 base
 AAGAGCAA...pairs ...GAGGAGCGTCG...pairs...GCAA

Figure 7: The transition molecules for the divide-by-three transducer. For every transition rule of the transducer, one transition molecule had to be designed. Another six transition molecules were created for the divide-by-two transducer.

encode a biological phenotype output, as Kossoy's experiments proved⁵. Therefore, these biological functions could give the bacteria antibiotic resistance to a certain drug or even cause the bacteria to change colors.

An example input was created encoding the string 200, equivalent to 18_{10} (Fig. 8). Notice the recognition sites for the two restriction enzymes BseRI and BbvI (colored brown and pink, respectively). BseRI cuts eight bases upstream from its recognition site on the sense strand, and ten bases away on the antisense strand. This leaves a two base sticky end. BbvI cleaves eight bases downstream on the sense strand and twelve on the antisense strand, leaving a four nucleotide sticky end. In

sticky end onto which every transition molecule can bond. BbvI leaves a four base sticky end, to which only one transition molecule can hybridize. Although every transition molecule has the AA sticky end, the one with the complementary four nucleotide sticky end will bond fastest.

After the terminator has been cleaved, at the end of the computation process, detection molecules are used. Detection molecules, like transition molecules, hybridize to the specific four nucleotide sticky end left by restriction enzymes. The significance about detection molecules is that once bonded to the terminator, with the use of DNA ligase, a detection molecule can indicate the final state. It does this because each detection molecule has a specific sticky end and a specific identifier. Identifiers used include length and fluorescence. Because each terminator has a specific sticky end and unique identifier, the final state could be determined by analyzing the specific identifier unique to the single detection molecule. Detection molecules can also

Final S_0
 TGCTGA...Biological...
 AAACGACT.....Gene 1.....ACGA

Final S_1
 TGCTGA...Biological...
 AAACGACT.....Gene 2.....CGAC

Final S_2
 TGCTGA...Biological...
 AAACGACT.....Gene 3.....GACT

Figure 8: The detection molecules. The molecules bond to the terminator when it is cleaved in a specific state. The molecules also "print" the terminator again so the divide-by-two process can begin.

order to compensate for the distance between the recognition site and the cleavage point, spacers have been added. The input also includes a string of symbols ending with the terminator sequence. The two sides of the input have sticky ends enabling cloning to a plasmid. The bright green section is another recognition site for a different restriction enzyme (EagI) that is necessary in order to perform consecutive computations.

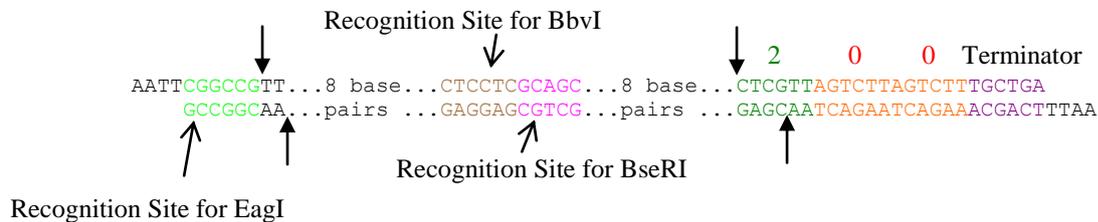


Figure 8: The complete input that was used in this research project. The input used in this research project includes three recognition sites (EagI, BseRI, BbvI). In order to better influence the cleavage sites, spacers were used. The symbol 200 (represented by the green and red sections) is also encoded in the input. The arrows represent cleavage sites by the restriction enzymes.

To initiate the computation process the restriction enzymes are added to the input solution. They cleave the input according to their recognition sites at the initial state. The correct transition molecules then hybridize to the sticky ends left by the restriction enzymes, and DNA ligase forms covalent bonds between the transition molecule and the input. This cycle of restriction, hybridization, and ligation repeats until the entire input has been processed (Fig. 9).

One of the DNA based transducer's advantage is the ability to perform more than one computation consecutively. In this case it divides by both three and two. This complicates the process because the recognition sites for the restriction enzymes BbvI and BseRI must be reinserted into the input of the second computation (the output of the first process) for the process to restart. For this method to work, a new restriction enzyme, EagI, was used. EagI's recognition site has been represented by bright green (Fig.10). EagI cleaves within its recognition site, so it does not destroy any other part of the input. Then, the recognition sites for BbvI and BseRI are inserted into the sequence and a second process can begin. The second process can be with either of the automata. The following process involves cycles of restriction, hybridization, and ligation.

When the final computation was finished, a printed section was left over. The computation worked correctly; therefore the printed section was three, (or 10 in ternary). Eighteen (ternary 200) divided by three and then divided by two is three. S_0 (as there is no remainder after computation) was the final state so detection molecule 0 was used and reporter gene 0 was expressed.

To check that all the molecules that were created fit only in the appropriate positions, a special program had to be used, Clone Manager. It was found that the transition molecules fit into the right positions and only the right positions, and that none of the sequences accidentally encoded the recognition sites for the restriction enzymes.

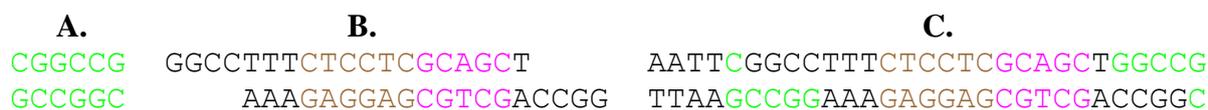


Figure 10: Restriction enzyme EagI (green) was needed in order to insert the recognition sites of BbvI (pink) and BseRI (brown). **A.** Restriction enzyme EagI's recognition site. This enzyme cleaves within the recognition site allowing the two other previously used recognition sites to be inserted without damaging neighboring sequences. **B.** The inserted two sticky ends which are complementary to the sticky ends which are created when EagI cleaves. In between the sticky ends on this sequence are the recognition sites of BbvI and BseRI. **C.** This is the entire sequence which is found in the modified plasmid. The green represents the recognition site EagI and the brown and pink represent the two recognition sites of BseRI and BbvI, respectively.

Discussion

This project is a proof of concept which focuses on the theoretical design of a finite state transducer capable of dividing a ternary input by three and two. In this project, two different sets of software were created. For each set of software, a theoretical model was built (which is the transducer). A variety of different types of molecules were also created as the components of the design. The design and molecules were checked using a computer program (Clone Manager). After all the molecules were processed in the computer program the molecules functioned as expected, and the design worked properly.

Creating a transducer using a plasmid has not yet been accomplished experimentally. Biomolecular transducers' importance does not lie in computing numerical equations, but rather in biological responses to stimuli. The application for this science holds a lot of potential. In the near future, scientists hope to integrate these biomolecular computers into biological systems. Therefore, this project is literally cutting-edge science; a new step into the future of finally integrating computers with bioorganic systems.

Conclusion

The aim of this project was to prove that, theoretically, it would be possible to engineer a DNA based transducer. The design for the transducer created proved correct by computer simulation. A biomolecular transducer that will be produced in the future is unlikely to replace the common computer. Instead, it will more likely integrate with biological systems, as biomolecular devices can scan DNA and produce a response. The immense power of a biomolecular computer with a direct interface with organisms will allow an incredible amount of possible functions, most of which today cannot even be imagined.

Acknowledgments

We would like to sincerely thank our mentor, Tamar Ratner, for her endless dedication to our project and her continued assistance. We would like to thank Professor Ehud Keinan. We would also like to thank Mr. Russell N. Stern and the Louis Herman Israel Experience Fund for their generosity and donation.

References

- ¹ J. McCaskill, Biomolecular Computing. *ERCIM News* 43, 30-32 (2000).
- ² Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, E. Shapiro, Programmable computing machine made of biomolecules. *Nature* 414, 430-434 (2001).
- ³ M. Soreni, S. Yogev, E. Kossoy, Y. Shoham, E. Keinan, Parallel Biomolecular Computation on Surfaces with Advanced Finite Automata. *J. AM. CHEM. SOC.* 127, 3935-3943 (2005).
- ⁴ Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, E. Shapiro, An autonomous molecular computer for logical control of gene expression. *Nature* 429, 423-429 (2004).
- ⁵ E. Kossoy, N. Lavid, M. Soreni-Harari, Y. Shoham, E. Keinan, A Programmable Biomolecular Computing Machine with Bacterial Phenotype Output. *ChemBioChem* 8, 1255-1260 (2007).