DNA-based Analog Computing

Daniel Fu[1*], Shalin Shah[2*], Tianqi Song[1], John Reif[1,2]

[1]Department of Computer Science, Duke University, Durham, North Carolina 27708, United States

[2]Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27708, United States

***The authors contributed equally to this work.**

1. Introduction to Deoxyribonucleic acid (DNA)

Deoxyribonucleic acid (DNA) is a well-known biological molecule whose structure was discovered by Watson and Crick, in 1953 [8]. It consists of two negatively charged sugar-phosphate backbones coiled in helical fashion. These backbones are connected to nucleotide bases, and each nucleotide base has a highly specific affinity making DNA programmable. There are four types of bases: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). An A can bind with T, and G can bind with C, through hydrogen bonding. The programmable self-assembly behavior has since been exploited to construct an array of complex, 2D and 3D, DNA nanostructures and patterns [6]. Besides complex DNA structures, this bio-molecule can also be used as a substrate for nanoscale computing [2]. Not only simple logic gates but also extremely sophisticated circuits have been experimentally demonstrated using DNA [3].

2. Computing by DNA Circuits

Since the early demonstration of DNA computing by Adleman [1] several groups have independently shown logic operations [2]. However, none of them scaled up to a sizable circuit. Georg Seelig et al., in 2006, reported first DNA-based large-scale enzyme-free circuit application [5]. First, they demonstrated the simple AND and OR logic operations. Using these gates and dual-rail logic, they built a NOT gate. Finally, they built a large circuit which consisted of 11 gates. The circuit consisted of signal restoration and signal amplification sub-operations. To demonstrate robustness, they used mircoRNAs as input. Their method solely relied on toehold mediated strand displacement, secondary DNA structures, domain sequestering, and sequence specificity making it simple. However, since toehold mediated strand displacement is slow, it takes hours for circuit to complete and a sophisticated approach is required to prevent leak. A better approach to modular and scalable DNA computing architecture was proposed by Qian and Winfree [3]. In 2011, they proposed seesaw architecture which used toehold-exchange mechanism. In this approach, each logic gate is abstractly represented as a seesaw which is indicative of toehold-exchange. However, in presence of a fuel strand, forward reaction becomes more likely analogous to heavier weight on one side of seesaw. Since their goal is to achieve Boolean logic, a threshold strand is incorporated to absorb input up to desired concentration. Since this approach abstracts out sequence specific DNA

details and design is modular, it can be scaled up to design large circuits. Using seesaw architecture, Qian and Winfree constructed a 4-bit square root circuit composed of 14 gates. Because of the robustness of seesaw architecture,

Qian and Winfree took a step further by demonstrating a synthetic neuron. They emulated the behavior of neuron by creating a perceptron circuit (also known as linear threshold circuit) [4]. In a perceptron, *n* weighted inputs are summed to check if their sum crosses a predetermined threshold. Since seesaw circuit can, in principle, perform any logic operation using dual-rail AND and OR gates, they were able to emulate a neuron. Finally, to demonstrate the behavior of a brain and robustness of emulation using DNA as a substrate, they also implemented 4-bit Hopfield associative memory circuit [4]. In such a network, a few questions are answered by user which act as an input to neuron. Depending on the weighted sum of answers, neuron circuit can give an answer since one of the possible answers has threshold lower than weighted sum. The emulation of a synthetic neurons using DNA as a substrate is a major step towards implementing an artificial brain.

DNA nanotechnology and related biological fields have lacked a demonstration of dynamic (analog) chemical systems [7]. Most of the system demonstrated always relied on the end product partly because monitoring a desired specie overtime is difficult due to undesired leaks. However, Srinivas et al. recently developed a CRN-to-DNA compiler, called Piperine, based on sophisticated design principles to convert any chemical reaction system to DNA sequences. As a test case, they use their compiler to experimentally demonstrate an oscillator. The actual design principles can be found in

Srinivas et al. [7]. A major advantage of this system is that it is enzyme-free, however, that also comes at the cost of lower reaction yield and long observation times.

3. Analog Systems by DNA

Sarpeshkar in 1998 discussed and compared the benefits and pitfalls of analog and digital computing [9]. Classically then, computation was evaluated primarily in time and space with energy unbounded, but Sarpeshkar proposed that energy was an equally important parameter for greatly complex systems such as the brain. The comparison of computational forms on expanded criterion served to motivate insights into the efficiency of neurobiological systems that combine analog and digital mechanisms to reach synthetically unachievable efficiency. Upon evaluating the resource requirements and optimal operational ranges of pure analog and digital systems in areas such as precision costs and signal-to-noise ratio, Sarpeshkar concluded that it was not unthinkable that the resource efficiency of analog systems could be augmented by the precise computation of digital systems. In fact, the implications of the proposed hybrid system strongly supported experimental evidence found in neurobiological studies, suggesting that nature's computational masterpiece, the brain, is a refined hybrid system.

This general evaluation of computational modes has incited persisting interests in synthetically building such systems utilizing molecular species, of which DNA has increasingly shown to be a leading candidate. Soloveichik *et al.* established that DNA was a suitable medium for compiling arbitrary chemical reaction networks (CRNs) using

strand displacement as its reaction primitive [10]. Specific base sequence programming and domain design of DNA strands can control the reaction kinetics and concentrations of reactants and products of unimolecular and bimolecular reactions within the molecular environment. This design schema translates the computationally rich behavior of CRNs to nucleic-acid-base chemistries and suggests the availability of complex circuits for experimental implementation in a molecular environment.

Cardelli continues to refine the operating mechanisms of DNA computing, of which leak, describing unintentional hybridization that initializes incorrect circuit pathways, is a prevalent issue [11]. A new construction of two-domain nicked doubled-stranded DNA (ndsDNA) implements *join* and *fork* actions as its primitives for circuit composition. The construction specifically uses top-nicked double strands, where discontinuities only exist on one strand of a double-stranded structure. Domains all share and are separated by same short toeholds, such that in initial and final species, the entire strand is fully hybridized and thereby protected. The system prevents backflow of waste products that could otherwise interfere with active components. The implementation reinforces the correctness of gates by committing to irreversible reactions only when all correct inputs are present, and otherwise reversibly returning to its reactive forms through random walk if only single inputs are present. Used gates and signals are then hybridized to garbage collecting species such that logically inactive components of the circuit are also unreactive and do not interfere with the rest of the system.

Utilizing ndsDNA and the established translation of CRNs to DNA, Chen *et al.* implement system controllers in DNA, which will be necessary to run composite systems

of modular molecular components [12]. These systems would be able to interface with natural analog signals in cellular environments while retaining the mathematical richness and digital compatibility of CRNs. We begin to see the empirical culmination of hybrid computational systems from discussions started over a decade ago in DNA nanotechnology. Here, a simple consensus network is implemented through proof of non-catalytic, catalytic, and autocatalytic reactions in verified bimolecular reactions using ndsDNA. The system can work upon pico-scale concentrations to convert minority species to the majority by first combining majority and minority signals to produce a buffer signal, consuming the minority signal, then converting the buffer signal to the majority signal. An additional benefit of ndsDNA is also shown here is that its domains and reactions can be defined from existing plasmid DNA which retains greater purity than synthesized DNA and inhibits leaks.

Oishi and Klavins generalize the necessary reactions for enzyme-free DNA implementations of basic functional blocks in control systems and establish that any linear I/O system can be composed by the three reactions of catalysis, degradation, and analysis [13]. Implementations of integral, summation, and gain blocks as well as their associated input and output signal behavior are converted to their chemical concentration representations. Furthermore, it is also shown that ideal chemical reactions can accurately model the temporal dynamics of a linear I/O controller with time varying signals. Using these established reaction primitives, a Proportional Integral controller is constructed. Then with Soloveichik *et al.*'s DNA strand displacement schema, proving that DNA strand displacement systems could represent arbitrary

ordinary differential equations, an example DNA implementation of a PI controller was reported.

A limiting factor of experimental discovery has been the high costs of empirically testing unique DNA systems. Robust simulation suites are a necessary tool for any field to drive the throughput and efficiency of design and test phases. Often noted by previous literature, to test the correctness of gate compositions, resultant systems, and their permutations necessitates automated case analysis. Yordanov *et al.* report on a substantial addition to Visual DSD that improves upon simulated testing capabilities for DNA strand displacement, DNA enzyme, and RNA enzyme systems [14]. The device under test is a Proportional Integral controller. It had been previously established by Oishi & Klavins that catalysis, degradation, and annihilation were a sufficient class of reactions to represent any linear I/O system. Their simulation scheme also showed that for certain DNA enzyme implementations, each degradation reaction could be replaced by a catalytic reaction coupled with an annihilation reaction, reducing the required elementary reactions to only catalysis and annihilation, and the simplification was termed *catalytic degradation*. It was a necessary circumvention as in some DNA enzyme systems, signal degradation was uniform and specific signals could not be chosen. However, pitfalls to the Visual DSD additions still include dealing with waste, as it would still be very computationally expensive to keep track of waste signals in large systems that would continue compounding. Nonetheless, the work still adds extensive capability for designing biochemical control circuits.

Song et al. reported an architecture to make DNA circuits for analog arithmetic computing [15]. The architecture is based on three operations: addition, subtraction and multiplication. Each operation is conducted by a corresponding gate. Polynomials can be computed by circuits made from the gates. Using approximation strategies (i.e. Taylor series), computing beyond polynomials can also be doable.

References

[1] Leonard M Adleman. Computing with dna. Scienti c american, 279(8):34-41, 1998.

[2] Gheorghe Paun, Grzegorz Rozenberg, and Arto Salomaa. DNA computing: new computing paradigms. Springer Science & Business Media, 2005.

[3] Lulu Qian and Erik Winfree. Scaling up digital circuit computation with dna strand displace- ment cascades. Science, 332(6034):1196-1201, 2011.

[4] Lulu Qian, Erik Winfree, and Jehoshua Bruck. Neural network computation with dna strand displacement cascades. Nature, 475(7356):368, 2011.

[5] Georg Seelig, David Soloveichik, David Yu Zhang, and Erik Winfree. Enzyme-free nucleic acid logic circuits. science, 314(5805):1585-1588, 2006.

[6] Nadrian C Seeman. Structural DNA nanotechnology. Cambridge University Press, 2016. [7] Niranjan Srinivas. Programming chemical kinetics: engineering dynamic reaction networks with DNA strand displacement. California Institute of Technology, 2015.

[8] James D Watson, Francis HC Crick, et al. Molecular structure of nucleic acids. Nature, 171(4356):737-738, 1953. 2

[9] Sarpeshkar, R. (1998) Analog versus digital: extrapolating from electronics to neurobiology. Neural Comput. 10, 1601−1638.

[10] Soloveichik, D., Seelig, G., and Winfree, E. (2010) DNA as a universal substrate for chemical kinetics. Proc. Natl. Acad. Sci. U. S. A. 107, 5393−5398.

[11] Cardelli, L. (2013) Two-domain DNA strand displacement. Mathematical Structures in Computer Science 23, 247−271.

[12] Chen, Y.-J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., and Seelig, G. (2013) Programmable chemical controllers made from DNA. Nat. Nanotechnol. 8, 755−762.

[13] Oishi, K., and Klavins, E. (2011) Biomolecular implementation of linear I/O systems. IET Syst. Biol. 5, 252−260.

[14] Yordanov, B., Kim, J., Petersen, R. L., Shudy, A., Kulkarni, V. V., and Phillips, A. (2014) Computational design of nucleic acid feedback control circuits. ACS Synth. Biol. 3, 600−616.

[15] Song, Tianqi, et al. "Analog computation by DNA strand displacement circuits." ACS synthetic biology 5.8 (2016): 898-912.