# Molecular computation via polymerase strand displacement reactions

Shalin Shah,[1,2] Ming Yang,[2] Tianqi Song[2] and John Reif[1,2]

[1]Department of Electrical and Computer Engineering
[2]Department of Computer Science
Duke University, Durham, NC, US - 27701
reif@cs.duke.edu

**Abstract.** The field of DNA computing has had many major breakthroughs since the early results of Adleman. Recent DNA computing works have focused primarily on enzyme-free computing architectures *i.e.* DNA-only circuits and devices. The rationale behind engineering such systems is to have biologically simpler machines. This offers several benefits such as no dependency on temperature control systems. However, recently [36, 38] the use of the enzyme-based system is gaining momentum yet again. These systems also compute using strand displacement, similar to enzyme-free architectures. However, the strand displacement is facilitated with a polymerase enzyme. Such enzymatic use provides an alternative method for the design and development of such systems. In this work, we discuss two computing avenues, namely, DNA circuits and chemical reaction networks along with a section on methods and protocols describing how polymerase-based systems operate. We end this chapter with a brief discussion and future possibilities of such emerging avenues of DNA computing.

**Keywords:** DNA computing · DNA polymerase · self-assembly · chemical reaction networks · logic circuits

## 1 Introduction

The field of DNA computing, now more than 25 years old, has come a long way since its first attempt at solving computationally hard problems [1]. Since then numerous computational architectures have been proposed [4, 7, 8, 12, 14, 16, 23, 28, 31, 36–38, 40], including molecular motors and robots [3, 5, 29, 44, 45], software packages to design DNA strands [18, 22, 30, 32, 34, 46, 50], protocols for liquid handling to design mesoscale structures [21, 26, 47, 49], DNA-based imaging architectures [19, 20, 24, 32, 33, 35], archival storage drives [10, 17, 34, 41] and several other related applications.

### 1.1 Logic circuits

The fundamental units of DNA computing include DNA strand and some supporting enzymes such as ligase, polymerase, nickase. For example, Adleman used

DNA hybridization and ligation to solve the traveling salesman problem for a directed graph [1]. Several recent architectures use enzyme-free DNA only strand displacement logic circuits [28,31,43]. Some robotic architectures use DNAzyme and an RNA target to implement a walker [11,29]. The popular DNA PEN toolbox uses three enzymes, namely, polymerase, exonuclease and nicking enzyme in addition to DNA strands for their computational ability [14]. Numerous other architectures have also been proposed which demonstrates this power and programmability of DNA computing systems. This work, in particular, will focus on the recent polymerase-based single-stranded DNA architecture [38] that uses strand displacing polymerase to compute logic circuits.

### 1.2 Chemical reaction networks

Chemical reaction networks (CRNs), traditionally, are used to model the specie dynamics of a well-mixed solution occurring in nature (for example, cell regulation). However, in DNA nanotechnology, mainly due to the phenomenal success of large-scale DNA logic systems [5, 7, 8, 15, 28, 31, 38, 43], it was realized that CRNs can form a programming language for synthetic DNA-based molecular systems [4, 36, 37]. This exciting realization has also been supported by rigorous explorations on the computational power of CRNs. It is understood that stochastic CRNs are Turing universal, under certain error tolerance assumptions, while what exactly are the limits of deterministic CRNs [6, 9]. More formally, a CRN is defined as a finite set of reactions $\mathbf{S}$ over a finite set of non-negative elements $\mathbf{E}$ such that reactants get consumed to produce products at some rate k *i.e.* $\mathbf{R} \xrightarrow{\text{k}} \mathbf{P}$ where $\mathbf{R}$, $\mathbf{P} \in \mathbf{E}$. A CRN can thus be represented using tuple ($\mathbf{S}$, $\mathbf{E}$, $\mathbf{K}$). Several DNA-based systems have been proposed [4, 27, 36, 37] to implement arbitrary CRNs and some of them have also been experimentally verified [7, 43]. In this work, we will mainly discuss the most recent scheme that uses a strand displacing polymerase to implement arbitrary CRNs.

### 1.3 Chapter organization

The rest of the book chapter is divided as follows: In section 2, we introduce the fundamental unit of computing *i.e.* DNA strand displacement. This includes both toehold-mediated strand displacement and polymerase-based strand displacement. In section 3, we discuss how a strand displacing polymerase can be used to perform DNA computing by building Boolean logic circuits. Section 4 shows how chemical reaction networks can be implemented using DNA substrate and strand-displacing polymerase enzyme. Unlike DNA-only reactions, polymerase reactions don't operate at room temperature and therefore the careful design of DNA strand and fluorescence setup is required. Section 5 describes the methods and protocol to be used. We conclude the chapter with a summary of the work and discussion about future possibilities.

## 2 Strand displacement

Just like a transistor forms a fundamental unit of silicon-based computing, strand displacement in DNA is the fundamental unit of DNA computing. If several thousand strand displacement units are combined, one can, in principle, design a fully-working bio-computer. *DNA strand displacement* [5,12–14,16,23,28,35,38] is defined as the replacement of an incumbent strand in a dsDNA by an input strand. There are two ways to do strand displacement: (a) Toehold-mediated strand displacement, and (b) Polymerase-based strand displacement. Most prior works relied on the principle of toehold-mediated strand displacement (TMSD) [3] as they followed the rationale of biologically simple systems. TMSD is an enzyme-free unit *i.e.* it only requires DNA strands. Polymerase-based strand displacement (PSD) [36], on the other hand, requires polymerase enzyme in addition to DNA strands. A pictorial representation of both techniques is shown in Fig. 1. In TMSD, a partial dsDNA molecule acts as the gate and the input strand comes in to bind with the dsDNA complex through a short single-stranded opening called toehold. Upon binding with the toehold region, the tug-of-war between input ssDNA and incumbent DNA strand starts. Since the toehold region makes full binding of input strand thermodynamically [25] more favorable, the input strand eventually completely displaces the incumbent strand which gets released as the output. This entire process can be abstracted as a pass gate where input strand I releases output strand O using gate complex G. TMSD has been used to construct interesting logic circuits, reaction networks and even targetted drug-delivery applications. More details about the applications of TMSD can be found in these papers [15, 23, 28, 42].

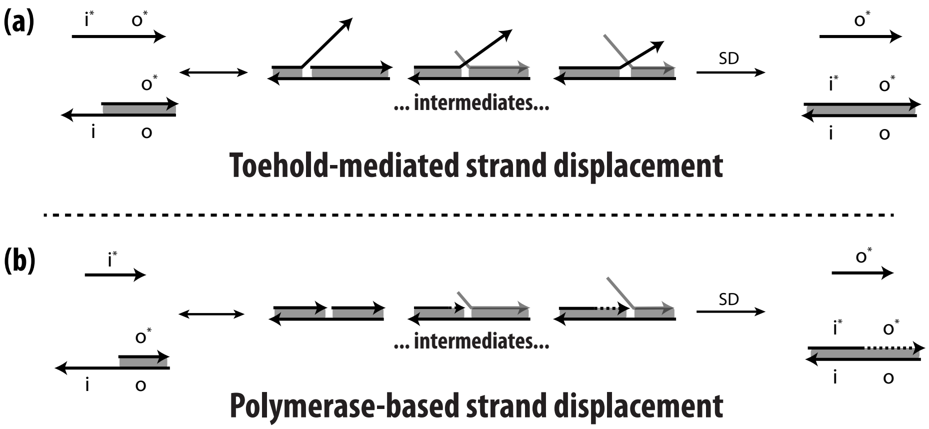

**Fig. 1.** Toehold-mediated strand displacement vs polymerase-based strand displacement along with the intermediate stages.

In PSD, a partial dsDNA molecule with partial single-strand open from the 5-

prime direction is used. An input strand can come in and bind to the toehold region and then polymerase enzyme comes in to simultaneously prime the full strand and displace the incumbent strand. The energy source in this scheme is polymerase enzyme as opposed to the branch portion of the input strand. PSD is a relatively old-yet-new idea in the field of DNA nanoscience. While polymerase enzyme has been used in standard techniques such as polymerase chain reaction (PCR) and early DNA computing works, the idea of using polymerase enzyme for DNA computing has faded away in the last few years. Most of the focus has been on enzyme-free DNA circuits as they are biologically simpler. However, there have been few works also exploring the idea of PSD for computing and reaction networks applications [14, 36, 38].

## 3 Using strand displacing polymerase for computation

DNA circuits have been increasingly explored over the last decade and the ones that have been implemented are primarily based on TMSD, which is an enzyme-free unit [28,31,43]. For example, Qian *et al.* experimentally demonstrate a four-bit square-root circuit that comprises 130 DNA strands and is based on seesaw gates and it takes hours to compute the result [28]. Although the principle of TMSD is biologically simple, it can introduce reaction leaks and is sometimes time-consuming especially when the length of the toehold increases. While the slow computing speed as a challenge has been approached from two different directions, namely, high concentration [48] and localization [3], our focus in this work is to explore an alternative avenue of PSD that can reduce circuit size.

Song *et al.* proposed a fast and compact DNA logic circuits architecture that uses PSD [38]. Their architecture is based on single-stranded DNA logic gates, which largely reduces leakage reactions and signal restoration steps. While PSD is generally more effective at strand displacement as compared to TMSD since the enzyme is an energy source, it should also lead to more leaky reactions since the overall system has higher energy. However, the ssDNA gates are designed cleverly so they mitigate the leak issues without compromising the reaction speed. Therefore, the circuits achieve faster computation speed with fewer DNA strands, which enables easy construction of large-scale logic circuits. In their experiments, they use Bst 2.0 DNA polymerase which has strong strand-displacement activity and a large range of salt tolerance. The active temperature for Bst 2.0 DNA polymerase is around 65 $^o$C and they conduct the experiments at 55$^o$C. All the strand domains they use in the experiments is around 30 nt.

The functionality of the OR gate and AND gate is shown in Fig. 2. Each gate is of the form two-input (A, B) and one-output (O). Logic-OR means that if either input strand is present, it should be able to release the output strand. To implement this function, two ssDNA gates are used (refer to Fig. 2b and Fig. 2c). If the input follows right order then fuel comes in to produce a partially double stranded complex followed by an input which can release the output strand. The output of the gate is reported using standard fluorescence techniques [7, 28, 31, 36, 43]. The functionality of the OR-gate is observed in Fig. 2a, where all four inputs were tested. As expected, when either inputs are present, we see output fluorescence, however, when no input is present, we do not observe any output. Similar to the logic-OR gate, the logic-AND gate is also a two-input (A, B), one-output (O) gate. However, the output is triggered only when both the inputs coexist in the solution. The functionality of this gate is shown in Fig. 2e. The output fluorescence is only observed in one case where both the inputs are present.

Using these simple units of logic computing, the work demonstrated a large-scale circuit that comprises of 37 DNA strands and can compute a 4-bit square-root function with a half-completion time of around 25 minutes. It is beyond the scope of this chapter to discuss implementation details of the circuits, however, the reader is encouraged to refer to the full paper. These logic circuits demonstrated by Song *et al.* required less computation time and fewer DNA strands as compared to prior state-of-the-art in the field demonstrating the potential of PSD logic circuits.

## 4 CRNs using strand displacing polymerase

The first step towards designing arbitrary CRNs using a strand displacing polymerase includes unimolecular and bimolecular CRNs. This is because any complex set of CRNs can be broken down into these simplistic CRNs. To implement a unimolecular CRN, Shah *et al.* proposed a two-step process (refer to Fig. 3). Let's consider the unimolecular CRN $\mathrm{A} \xrightarrow{\mathrm{k}} \mathrm{B}$ where input A produces output B at some rate k. This can be done using two auxiliary complexes which converts the input A to intermediate species I, and intermediate species I to output B. Both the steps use strand displacing polymerase to undergo PSD. It might seem hard to imagine this two-step process approximating the unimolecular reaction, under certain assumptions, it can be proved that these reactions indeed do so. The reader is encouraged to refer to the full paper for details [36].

In unimolecular reactions, only one input is required. However, most systems use more than one input species and therefore bimolecular reaction form the simplest non-trivial reaction set. To implement such a system $\mathrm{A} + \mathrm{B} \xrightarrow{\mathrm{k}} \mathrm{C}$, we simply need a dsDNA complex and one of the inputs. If so, then the input strand can combine with the gate complex to produce intermediate species I, which then, in turn, produces output C with the help of auxiliary complex. Such implementation ensures the stoichiometry of both the inputs is maintained and intermediate being produced only in the presence of both the inputs (refer to Fig. 3). Similar to unimolecular reactions, it is possible to prove that this two-step process along with a supporting linker reaction, can approximate the bimolecular reaction. However, it is out-of-scope of this chapter and the reader is encouraged to read the original paper [36].

Using these simple systems that implement unimolecular and bimolecular reactions, very complex and interesting phenomenon can be implemented. For example, these systems can be used for population protocols such as molecular

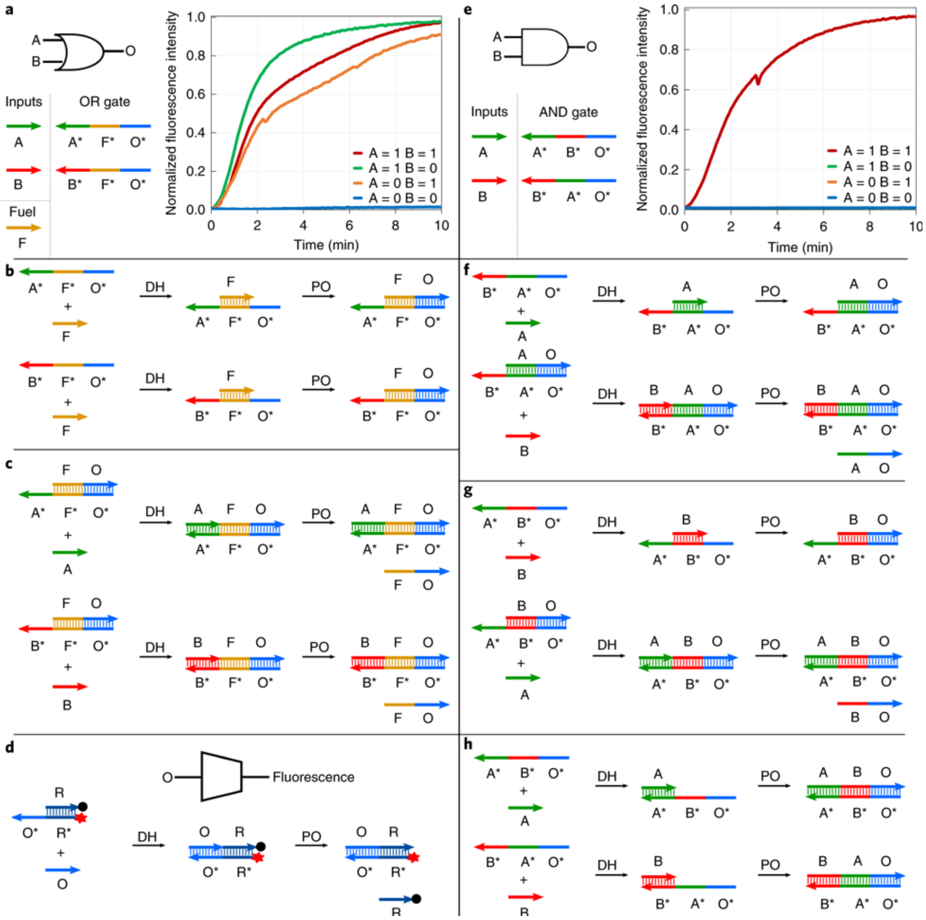

**Fig. 2.** DNA implementation for logic gates OR and AND. a) OR gate. b) Fuel strand (F) is bound to the OR gate by DNA hybridization (DH) and then extended by polymerization (PO) reaction. c) Input hybridizes with the gate complex and releases the output by PO reaction. d) Reactions with a reporter complex. Input (O) hybridizes with reporter complex, then separates fluorophores and quenchers by PO reaction and release fluorescence signal e) AND gate. f, g) Two possible reaction pathways to produce output (AO) or (BO). h) Side-reactions without any output generation. [38]. The figure is adopted from Song *et al.* (2019) with permission [38].

consensus. In such a network, as shown in Fig. 4a, two molecular species interact with each other to reach a democracy if a majority is found. The *in silico* demonstration of a molecular consensus network is shown in Fig 4b. Depending on different relative amounts of species A and B, the network, which is a set of CRNs, can compute the majority. Upon computation, all the minority species are consumed and converted to the majority. A more complex system that has also been demonstrated includes a rock-paper-scissor oscillator. Such a

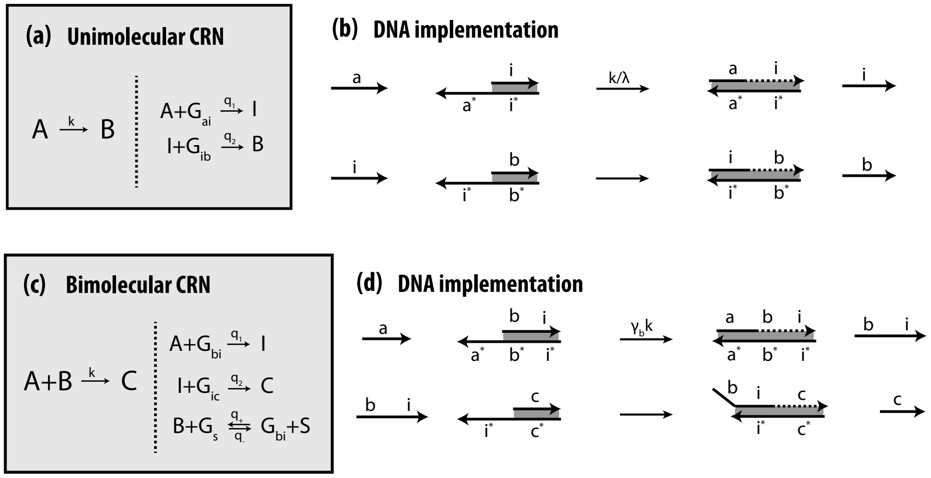

**Fig. 3.** Unimolecular and bimolecular reaction along with their low-level DNA implementation. Both reactions are a two-step process where input releases an intermediate strand, and the intermediate strand releases an output upon combining with an auxiliary dsDNA complex. The figure is adopted from Shah *et al.* (2019) with permission [36].

molecular protocol is considered complex since they require careful calibration of reaction rates and stoichiometry of species. In such a cyclic system, a rock can beat scissors, a scissor can beat paper and a paper can beat rock. Therefore, such a network, under certain initial conditions, can keep oscillation indefinitely. The DNA implementation of such a system closely approximates the actual oscillatory phenomenon, however, eventually starts diverging after auxiliary gates are consumed (refer to Fig. 4d). These example networks demonstrate the power of CRNs and their DNA implementations. Of course, the experimental demonstration of these systems using DNA and PSD will add a cherry to the cake, however, the problem still remains as work-in-progress [36].

## 5 Methods and protocol

### 5.1 Oligonucleotide design, synthesis, and purification

It has been long known [2] that DNA strand design plays a crucial role in the successful operation of large-scale circuits and therefore utmost care should be taken while designing the strands. Based on our prior works, here are some general guidelines which can be used to design DNA strands:

1. The GC content for DNA strands should be roughly in the range of 40 and 60% along with no more than 3 consecutive G, C, A, T nucleotide, also called homopolymers (eg. CCCC).

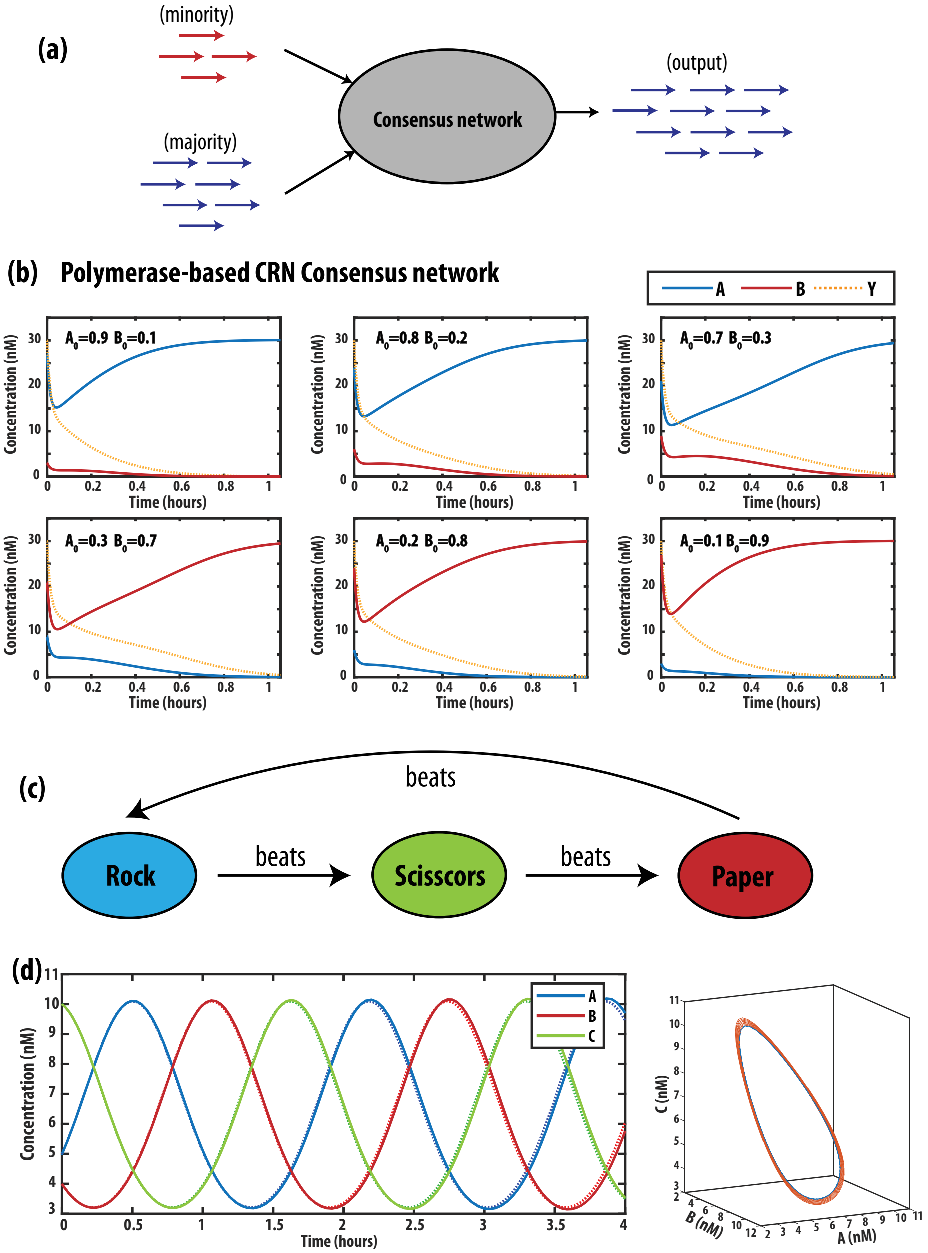

**Fig. 4.** (a-b) A consensus network. (c-d) A rock-paper-scissor oscillator. Both the systems were designed and implemented using the strand displacing polymerase scheme [36]. The figure is adopted from Shah *et al.* (2019) with permission [36].

2. The toehold region should always be on the 5' end (or 3' exposed region) since the polymerase primes from 5' to 3' direction. The toehold can range from 10 nt to 30 nt depending on the desired reaction rate.
3. Polymerase enzyme has varying activity with temperature and therefore thermal chamber is essential. In the case of Bst polymerase, a temperature of 50 - 65 $^o$C is suitable.
4. Use 5 - 8 nt clamping strategy [7, 28, 38] on the ends of dsDNA *i.e.* use GC-rich ends to avoid DNA breathing from opening down-stream gates.

It is very well-known that once the DNA is synthesized, two rounds of purification is required [7, 43]: (a) 12% denaturing PAGE purification, and (b) non-denaturing PAGE. This will greatly mitigate the unwanted reactions, also called leaky [39] reactions, and get rid of synthesis errors. The voltage and time for running gels depend on the size of gel and length of DNA, however, 200 V for 30 minutes can be a good starting point for denaturing gel and 150 V for 4 hours for non-denaturing gel. For gate preparations, several protocols are available suggesting different anneal times, however, a simple 2 hour anneal cycle with a linear temperature gradient from 95 $^o$C to 25$^o$C works well.

### 5.2 Fluorescence sample preparation and measurement

DNA circuits are generally tested using fluorescence spectrometer [7, 12, 16, 28, 31, 38, 45]. Some care should be taken here while observing reaction output since polymerase activity is highly correlated with temperature. Prepare the reaction master mix with all the required buffers and DNA strands. This should include dNTPs, polymerase, isothermal amplification buffer, dsDNA gates as well as fluorescent DNA reporters. Incubate this solution at the desired temperature in the PCR machine or cuvette for 20 minutes. Add the input strand quickly to the sample and start measuring the reaction. It should be ensured that the volume of input is minimal as compared to the rest of the sample volume since the input is added at room temperature. For example, 1 uL (1% of total volume) of input can be added to 90 uL of the master mix. This ensures minimal temperature fluctuations upon adding input.

## 6 Discussion and outlook

In this chapter, we introduced an alternative architectural unit for DNA computing *i.e.* polymerase strand displacement which uses a polymerase enzyme as an energy source for strand displacement. It was compared with the traditional toehold-mediated strand displacement briefly followed by a short review of the state-of-the-art techniques for DNA computing and DNA-based CRNs using a strand displacing polymerase. As with other systems, the biggest challenge to build large-scale polymerase systems would be reaction leak and therefore leakless principles need to be adopted to the polymerase design. However, it largely remains an open problem. However, these new studies [36,38] promise an exciting future of DNA computing since they demonstrate that to engineer large-scale DNA computing systems, there are new research avenues yet to be explored.

## References

1. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science pp. 1021–1024 (1994)
2. Arita, M., Nishikawa, A., Hagiya, M., Komiya, K., Gouzu, H., Sakamoto, K.: Improving sequence design for DNA computing. In: Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation. pp. 875–882. Morgan Kaufmann Publishers Inc. (2000)
3. Bui, H., Shah, S., Mokhtar, R., Song, T., Garg, S., Reif, J.: Localized DNA Hybridization Chain Reactions on DNA Origami. ACS nano **12**(2), 1146–1155 (2018)
4. Cardelli, L.: Two-domain DNA strand displacement. Mathematical Structures in Computer Science **23**(2), 247–271 (2013)
5. Chao, J., Wang, J., Wang, F., Ouyang, X., Kopperger, E., Liu, H., Li, Q., Shi, J., Wang, L., Hu, J., et al.: Solving mazes with single-molecule DNA navigators. Nature materials **18**(3), 273 (2019)
6. Chen, H.L., Doty, D., Soloveichik, D.: Deterministic function computation with chemical reaction networks. Natural computing **13**(4), 517–534 (2014)
7. Chen, Y.J., Dalchau, N., Srinivas, N., Phillips, A., Cardelli, L., Soloveichik, D., Seelig, G.: Programmable chemical controllers made from DNA. Nature nanotechnology **8**(10), 755–762 (2013)
8. Cherry, K.M., Qian, L.: Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature **559**(7714), 370 (2018)
9. Cook, M., Soloveichik, D., Winfree, E., Bruck, J.: Programmability of chemical reaction networks. In: Algorithmic bioprocesses, pp. 543–584. Springer (2009)
10. Erlich, Y., Zielinski, D.: DNA fountain enables a robust and efficient storage architecture. Science **355**(6328), 950–954 (2017)
11. Eshra, A., El-Sayed, A.: An odd parity checker prototype using DNAzyme finite state machine. IEEE/ACM transactions on computational biology and bioinformatics **11**(2), 316–324 (2013)
12. Eshra, A., Shah, S., Song, T., Reif, J.: Renewable DNA hairpin-based logic circuits. IEEE Transactions on Nanotechnology (2019)
13. Fu, D., Shah, S., Song, T., Reif, J.: DNA-Based Analog Computing. In: Synthetic Biology, pp. 411–417. Springer (2018)
14. Fujii, T., Rondelez, Y.: Predator–prey molecular ecosystems. ACS nano **7**(1), 27–34 (2012)
15. Garg, S., Bui, H., Eshra, A., Shah, S., Reif, J.: Nucleic Acid Hairpins: A Robust and Powerful Motif for Molecular Devices, chap. 7, pp. 175–199. World Scientific (2019)
16. Garg, S., Shah, S., Bui, H., Song, T., Mokhtar, R., Reif, J.: Renewable Time-Responsive DNA Circuits. Small **14**(33), 1801470 (2018)
17. Goldman, N., Bertone, P., Chen, S., Dessimoz, C., LeProust, E.M., Sipos, B., Birney, E.: Towards practical, high-capacity, low-maintenance information storage in synthesized dna. Nature **494**(7435), 77 (2013)
18. Goyal, A., Limbachiya, D., Gupta, S.K., Joshi, F., Pritmani, S., Sahai, A., Gupta, M.K.: DNA pen: A tool for drawing on a molecular canvas. arXiv preprint arXiv:1306.0369 (2013)
19. Jungmann, R., Avendaño, M.S., Dai, M., Woehrstein, J.B., Agasti, S.S., Feiger, Z., Rodal, A., Yin, P.: Quantitative super-resolution imaging with qPAINT. Nature methods **13**(5), 439 (2016)

20. Jungmann, R., Avendaño, M.S., Woehrstein, J.B., Dai, M., Shih, W.M., Yin, P.: Multiplexed 3D cellular super-resolution imaging with DNA-PAINT and Exchange-PAINT. Nature methods **11**(3), 313 (2014)
21. Ke, Y., Ong, L.L., Shih, W.M., Yin, P.: Three-dimensional structures self-assembled from DNA bricks. science **338**(6111), 1177–1183 (2012)
22. Lakin, M.R., Youssef, S., Polo, F., Emmott, S., Phillips, A.: Visual DSD: a design and analysis tool for DNA strand displacement systems. Bioinformatics **27**(22), 3211–3213 (2011)
23. Li, W., Yang, Y., Yan, H., Liu, Y.: Three-input majority logic gate and multiple input logic circuit based on dna strand displacement. Nano letters **13**(6), 2980–2988 (2013)
24. Lin, C., Jungmann, R., Leifer, A.M., Li, C., Levner, D., Church, G.M., Shih, W.M., Yin, P.: Submicrometre geometrically encoded fluorescent barcodes self-assembled from DNA. Nature chemistry **4**(10), 832–839 (2012)
25. Machinek, R.R., Ouldridge, T.E., Haley, N.E., Bath, J., Turberfield, A.J.: Programmable energy landscapes for kinetic control of DNA strand displacement. Nature communications **5**, 5324 (2014)
26. Ong, L.L., Hanikel, N., Yaghi, O.K., Grun, C., Strauss, M.T., Bron, P., Lai-Kee-Him, J., Schueder, F., Wang, B., Wang, P., et al.: Programmable self-assembly of three-dimensional nanostructures from 10,000 unique components. Nature **552**(7683), 72 (2017)
27. Qian, L., Soloveichik, D., Winfree, E.: Efficient Turing-universal computation with DNA polymers. In: International Workshop on DNA-Based Computers. pp. 123–140. Springer (2010)
28. Qian, L., Winfree, E.: Scaling up digital circuit computation with DNA strand displacement cascades. Science **332**(6034), 1196–1201 (2011)
29. Reif, J.H., Sahu, S.: Autonomous programmable DNA nanorobotic devices using DNAzymes. Theoretical Computer Science **410**(15), 1428–1439 (2009)
30. van Roekel, H.W., Meijer, L.H., Masroor, S., Felix Garza, Z.C., Estevez-Torres, A., Rondelez, Y., Zagaris, A., Peletier, M.A., Hilbers, P.A., de Greef, T.F.: Automated design of programmable enzyme-driven DNA circuits. ACS synthetic biology **4**(6), 735–745 (2014)
31. Seelig, G., Soloveichik, D., Zhang, D.Y., Winfree, E.: Enzyme-free nucleic acid logic circuits. science **314**(5805), 1585–1588 (2006)
32. Shah, S., Dubey, A., Reif, J.: Programming temporal DNA barcodes for single-molecule fingerprinting. Nano letters (2019)
33. Shah, S., Dubey, A.K., Reif, J.: Improved optical multiplexing with temporal DNA barcodes. ACS Synthetic Biology **8**(5), 1100–1111 (2019)
34. Shah, S., Limbachiya, D., Gupta, M.K.: DNACloud: A Potential Tool for storing Big Data on DNA. arXiv preprint arXiv:1310.6992 (2013)
35. Shah, S., Reif, J.: Temporal DNA Barcodes: A Time-Based Approach for Single-Molecule Imaging. In: International Conference on DNA Computing and Molecular Programming. pp. 71–86. Springer (2018)
36. Shah, S., Song, T., Song, X., Yang, M., Reif, J.: Implementing arbitrary CRNs using strand displacing polymerase. In: International Conference on DNA Computing and Molecular Programming. pp. 21–36. Springer (2019)
37. Soloveichik, D., Seelig, G., Winfree, E.: DNA as a universal substrate for chemical kinetics. Proceedings of the National Academy of Sciences **107**(12), 5393–5398 (2010)

38. Song, T., Eshra, A., Shah, S., Bui, H., Fu, D., Yang, M., Mokhtar, R., Reif, J.: Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase. Nature nanotechnology **14**(11), 1075–1081 (2019)
39. Song, T., Gopalkrishnan, N., Eshra, A., Garg, S., Mokhtar, R., Bui, H., Chandran, H., Reif, J.: Improving the Performance of DNA Strand Displacement Circuits by Shadow Cancellation. ACS nano **12**(11), 11689–11697 (2018)
40. Song, X., Eshra, A., Dwyer, C., Reif, J.: Renewable DNA seesaw logic circuits enabled by photoregulation of toehold-mediated strand displacement. Rsc Advances **7**(45), 28130–28144 (2017)
41. Song, X., Reif, J.: Nucleic acid databases and molecular-scale computing. ACS nano
42. Srinivas, N., Ouldridge, T.E., Šulc, P., Schaeffer, J.M., Yurke, B., Louis, A.A., Doye, J.P., Winfree, E.: On the biophysics and kinetics of toehold-mediated DNA strand displacement. Nucleic acids research **41**(22), 10641–10658 (2013)
43. Srinivas, N., Parkin, J., Seelig, G., Winfree, E., Soloveichik, D.: Enzyme-free nucleic acid dynamical systems. Science **358**(6369), eaal2052 (2017)
44. Teichmann, M., Kopperger, E., Simmel, F.C.: Robustness of localized DNA strand displacement cascades. ACS nano **8**(8), 8487–8496 (2014)
45. Thubagere, A.J., Li, W., Johnson, R.F., Chen, Z., Doroudi, S., Lee, Y.L., Izatt, G., Wittman, S., Srinivas, N., Woods, D., Winfree, E., Qian, L.: A cargo-sorting DNA robot. Science **357**(6356) (2017)
46. Thubagere, A.J., Thachuk, C., Berleant, J., Johnson, R.F., Ardelean, D.A., Cherry, K.M., Qian, L.: Compiler-aided systematic construction of large-scale dna strand displacement circuits using unpurified components. Nature communications **8**, 14373 (2017)
47. Tikhomirov, G., Petersen, P., Qian, L.: Fractal assembly of micrometre-scale DNA origami arrays with arbitrary patterns. Nature **552**(7683), 67 (2017)
48. Wang, B., Thachuk, C., Ellington, A.D., Winfree, E., Soloveichik, D.: Effective design principles for leakless strand displacement systems. Proceedings of the National Academy of Sciences **115**(52), E12182–E12191 (2018)
49. Wei, B., Dai, M., Yin, P.: Complex shapes self-assembled from single-stranded DNA tiles. Nature **485**(7400), 623 (2012)
50. Zadeh, J.N., Steenberg, C.D., Bois, J.S., Wolfe, B.R., Pierce, M.B., Khan, A.R., Dirks, R.M., Pierce, N.A.: NUPACK: analysis and design of nucleic acid systems. Journal of computational chemistry **32**(1), 170–173 (2011)