Programming Complex Behavior in DNA-based Molecular Circuits and Robots

Thesis by
Anupama J. Thubagere

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy, Bioengineering

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2017
Defended on May 23, 2017

ACKNOWLEDGEMENTS

My journey through graduate school has seen some unconventional obstacles. My post graduate education began at Boston University with Franco Cerrina. He not only nurtured my growth in scientific research but also a love for the arts. He was a great scientist with an entrepreneurial spirit. His untimely death is an unfortunate loss for the scientific community and a greater personal loss for people who knew him. I will forever be grateful to have had the privilege to experience his intellect, leadership and warmth.

I moved to Caltech under the shadow of his loss, and found a welcoming community. I started working with Richard Murray and Paul Rothemund who were both excellent mentors. Richard has always helped me see the big picture, and not lose sight of the real purpose of my research. Paul helped me understand the precise details of my work, nothing was ever too small or trivial to discuss.

I am also thankful to everyone in the Murray lab for all the support I have received over the years. I feel lucky to have built special and lasting friendships at Caltech with Marcella Gomez, Victoria Hsiao, Fiona Chandra, Ophelia Venturelli, Eliza Franco, Melissa Tanner, Yong Wu, Zach Sun, Anandh Swaminathan and Vipul Singhal. Each in their own way have helped me be the person I am today.

I started working with Lulu Qian in my third year at Caltech. During this time, I also got more involved with the larger DNA nanotechnology community in Erik Winfree's lab. The DNA community at Caltech including Chris Thachuck, Robert Johnson, Damien Woods, Wei Li and Jongmin Kim have helped me with everything ranging from scientific discussions to feedback on experiments and data analysis to constructive criticism during rehearsal conference presentations. Lulu in particular has been an amazing mentor, challenging me to be my best self.

At about the same time that I started working with Lulu, I was diagnosed with a form of myeloid leukemia. My mentors, friends and family have all been an amazing source of comfort through the challenges I faced with my health. My husband, Ashwin Gopinath, has been encouraging and loving through the whole endeavor, helping me stay hopeful. My medical team, headed by Dr. David Snyder at City of Hope has been incredible in my treatment strategy, indulging my curiosity in experimental procedures in a safe and fruitful manner. I thank him for everything he has done for me.

Finally, after a few hiccups but with Lulu Qian's excellent guidance, I have come to the end of my journey through graduate school having reached milestones I can truly appreciate.

ABSTRACT

Integrated electronic circuits, like those found in cellphones and computers, are ubiquitous in our information-driven society. The success of electronics has, in part, been due its modular architecture that enables individual components to be independently improved while the overall device functionality remains unchanged. Over the last two decades the emerging field of dynamic DNA nanotechnology has been trying to apply the underlying philosophy of electronics to biochemical circuits. DNA nanotechnology employs rationally designed DNA molecules as building blocks of biochemical circuits that can, in principle, enable powerful applications like diagnostics and therapeutics.

Researchers in the field of DNA nanotechnology have developed simple elements to construct biomolecular systems with desired functions. They have also developed molecular compilers for defining design principles. The cost of DNA synthesis has decreased by over three orders of magnitude in the past decade. This has lead to a non-trivial number of small scale circuits, like DNA-based logic gates and chemical oscillators, being implemented. However, the scalability of this approach has yet to be clearly demonstrated.

In this thesis, we will discuss our main contributions to facilitating the advancement of DNA nanotechnology by developing systematic approaches for constructing modular DNA building blocks. These modules can be used to construct biochemical circuits and molecular robotic systems. The performance of the modules can be individually tuned and integrated into large-scale systems.

Using automated circuit-design software and cheap unpurified DNA, we demonstrated the design and construction of a complex synthetic biochemical circuit consisting of 78 distinct DNA species. The circuit is capable of computing the transition rules of a cell updating its state based on its neighboring cells, defined in a classic computational model called cellular automata. Using a bottom-up approach, we first characterized the component necessary for basic Boolean logic computation. We then systematically integrated more circuit elements and eventually constructed the full circuit. By developing a systematic procedure for building DNA-based circuits using unpurified components, we significantly simplified the experimental procedure. By using unpurified DNA components, we reduced the cost and technical barrier for circuit construction, thus making the design and synthesis of complex

DNA circuits accessible to even novice researchers.

Next we demonstrated a cargo sorting DNA nano-robot, using a simple algorithm and modular building blocks. The DNA robot has a leg and two foot domains for exploring a two-dimensional DNA origami surface, and an arm and hand domain for picking up randomly located cargos and dropping them off at their designated locations. It is completely autonomous and is programmed to perform a random walk without requiring an external energy source. Further, we demonstrated sorting multiple copies of two distinct cargo species on the same origami. Additionally, by compartmentalizing each sorting task on a single origami, we showed that two distinct sorting tasks can be implemented on different origami simultaneously in the same test tube. The recognition of a cargo is embedded in its destination, therefore it is possible to scale up the system simply by having multiple types of cargos. The same robot design can be used for performing multiple instances of distinct tasks in parallel. The different modules can be integrated to perform diverse functions, including applications in time-release targeted therapeutics.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Anupama J Thubagere et al. “Compiler-aided systematic construction of large-scale DNA strand displacement circuits using unpurified components”. In: *Nature Communications* 8 (2017).
A.J.T. designed and performed the experiments and analysed the data of the full circuit and led the project to completion. A.J.T and L.Q wrote the manuscript.

[2] Anupama J Thubagere et al. “A cargo-sorting DNA robot”. In: *Submitted* (2017).
A.J.T. designed and performed the experiments and analysed the data and led the project to completion. A.J.T and L.Q wrote the manuscript.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The origins and propagation of life has fascinated scientists for generations. Primordial life is believed to have originated with chemical molecules (nucleic acids, amino acids) reacting with each other in compartments. Over time the compartments developed into more sophisticated biological cells representing life. Scientists over time have worked to understand the complexities surrounding a single cell, its metabolism, its ability to respond to external stimuli and its replication properties using various scientific tools including different engineering approaches. Gregor Mendel in 1865 proposed the law of biological inheritance, suggesting the idea that biological life follow certain rules and copies of these rules get passed on to future generations. This inspired a new field of thought encouraging researchers to find evidence to aid in deciphering these rules.

Erwin Schrödinger's "What is life?" lecture [1] explains a theoretical physicists view on the physical aspect of the living cell. In his lectures, Schrödinger introduced the idea of an "aperiodic crystal" that contained genetic information in its configuration of covalent chemical bonds. Jon von Neumann inspired a new generation of scientists by introducing growth and reproduction as information processing tasks [2]. Alan Turing, with his work on chemical morphogenesis [3] provided insights into computations involved in biological growth.

Before Schrödinger, for over 50 years since its discovery, deoxyribonucleic acid (DNA) had been thought of primarily as a biological molecule. First isolated in 1869 by a Swiss physician Friedrich Miescher when he noticed a microscopic substance in discarded surgical bandages, DNA was mostly known to be a cell component that had important function as it was located in the nucleus. By early 1900s, it was also known that DNA consisted of two types of molecules: purines and pyrimidines. By 1952, Chargaff had discovered that all organisms have 1:1 ratio of purine and pyrimidine bases in their DNA and that different organisms had different amounts of these molecules [4]. Chargaff's rules helped James Watson and Francis Crick to solve the crystal structure of DNA in 1953 [5]. The double helical structure of DNA itself was crystallized by Rosalind Franklin and Maurice Wilkins. Resolving the structure of DNA led to understanding the physical properties of the

molecule at the atomic level(fig. 1.1). What came next were the constituent alphabets composing the DNA molecule. DNA is made of four chemical bases: adenosine (A), thymidine (T), guanadine (G) and cytosine (C).

There exist atleast three different conformation of DNA: A-DNA, B-DNA and Z-DNA. The B form described by James Watson and Francis Crick is believed to predominate in biological cells [6]. Each of the bases in the B-form string along a sugar-phosphate backbone separated by a distance of 0.43 nm. A binds to T via double hydrogen bonds and G binds to C via triple hydrogen bonds. A single strand of DNA binds to its complementary strand following this rule. The double helical structure is about 2 nm wide and the distance between base pairs reduces to 0.34 nm. The double helix is stabilized by the stacking interaction between the bases. Further, the double helix exhibits a left-handed turn approximately every 10.5 base pairs (bp), i.e, ~3.6 nm.

Figure 1.1: **Diagram of DNA structure.** The two ribbons symbolize the two phosphate-sugar chains and the horizontal rods symbolize the bases holding the chain together. The vertical lines mark the DNA fiber axis. Reproduced from [5].

In the last two decades, the emergent field of DNA nanotechnology has featured DNA molecules for designing a large library of molecular systems capable of performing complex information processing functions using networks of bimolecular components [7, 8]. The information of the physical structure of DNA gathered by Watson and Crick has been crucial to the molecular programming community in redefining DNA as an engineering material and a computing substrate implementing various structural properties and dynamic behaviors in DNA-based devices. In general, the understanding of the structural, thermodynamic and kinetic properties of DNA has aided the development of DNA nanotechnology. Furthermore, productivity of DNA synthesis has increased by nearly five orders of magnitude over the last 25 years, while the cost of gene synthesis per base has dropped to by nearly three orders of magnitude [9].

Along with DNA synthesis, technological advancements towards visualization and manipulation of matter at the nanoscale level have been of significance. Super resolution microscopy techniques such as DNA PAINT [10] and high-speed atomic force microscopy has enabled characterization of objects at the nanoscale and in some cases visualization of dynamic molecular events [11].

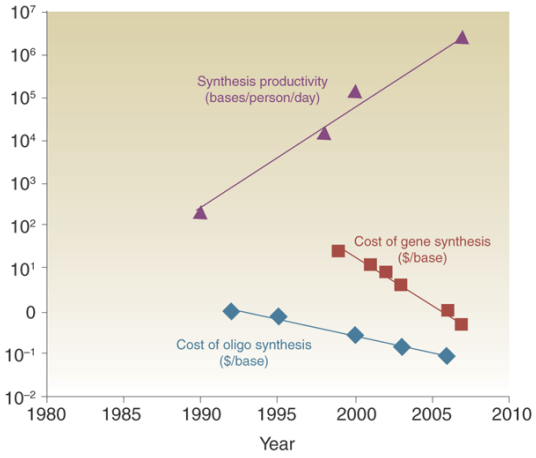

Figure 1.2: **Changing economics of DNA synthesis.** Since 1990, cost of oligo synthesis per base has dropped about two orders of magnitude while synthesis productivity has increased by five orders of magnitude. Reproduced from [9].

Advancements in oligo synthesis and visualization techniques at the nanoscale have contributed to the growth of DNA nanotechnology by helping researchers design various experimental and computational tools. Amongst these, a number of compilers that help design DNA for building sophisticated DNA-based devices have been developed [12–14]. Furthermore, the development of structural components at the nanoscale using DNA has contributed towards the development of localized computation schemes [15]. In this thesis, we will discuss our main contributions toward facilitating the advancement of DNA nanotechnology by developing systematic approaches for constructing modular DNA building blocks. These modules can be used to construct biochemical circuits and molecular robotic systems. The performance of the modules can be individually tuned and integrated into large scale systems.

1.1 Overview of DNA nanotechnology

DNA has well understood mechanical [16] and thermodynamic [17, 18] properties. The stability of a DNA double helix can be predicted from the contributions of each of its nearest-neighbor base pair stacks [17]. This ensures that efficient software and

fast algorithms can be designed to predict folding from DNA sequence and also permit sequence design given a prescribed structure.

Inspired by the replication junction and the Holliday structure, a critical intermediate in genetic recombination, Ned Seeman in 1982 theorized the ability to study such biologically relevant reactions using strand displacement mechanisms [19] that could easily be mimicked using synthetic DNA molecules. He also envisioned creating three dimensional lattices from such three and four-armed junctions. The goal was to use these DNA lattices as scaffolds for facilitating the crystallographic study of protein structure. While this turned out to be more challenging to implement, it gave rise to the idea of self-assembled structures using synthetic DNA and led to the development of *structural* DNA nanotechnology. The initial idea for forming small tiles and simple structures along with the development of efficient software tools has given way to designing versatile DNA *origami* structures in two and three-dimensions.

Len Adleman had the pioneering thought of using DNA molecules to perform computations [20]. The parallelism provided by DNA molecules mixed in a test tube was used to solve a seven-node Hamiltonian path problem. While scaling up such reactions proved impractical, the work itself inspired several others to build DNA-based systems that could be programmed to perform interesting and complex computation tasks, thus paving way to building dynamic systems. Of particular consequence was Bernie Yurke's work on molecular tweezers using DNA strand displacement mechanism [21]. Strand displacement mechanism was first observed in a viral DNA replication where an invader strand uses the single-stranded overhang on a double-stranded DNA plasmid. Using the overhang, the invader strand manages to displace the bound strand and taking its place. *Dynamic* DNA nanotechnology has been used widely to implement all-DNA systems which can execute enzyme-free logic [22], implement cellular automaton rules [23], act as a catalyst [24], compute square root of a number [25], exhibit oscillations [26], and also used as a programmable module for dynamic self-assembly [27]. Specifically, the mechanism of toehold mediated strand displacement [21, 28–30] has been used extensively for implementing the above mentioned systems and continues to be extremely relevant in the field.

1.2 DNA as a computing substrate

Much like the way in which complex electrical circuits are built with simple primitives, molecular programming uses DNA strand displacement components to build computational systems which can implement catalytic reactions and feed-forward Boolean logic circuits. An active and growing community of molecular programmers have contributed to the development and widespread use of designed automation. For example, Visual DSD [31, 32] is a custom programming language which is commonly used for verifying, simulating and analyzing strand displacement circuits by enumerating and simulating the corresponding strand displacement reactions. Another great example is Nupack [12] which has gained popularity given its ease of use for analyzing and designing nucleic acid systems. The Nupack compiler provides an environment for testing nucleic acid sequences for structure verification at thermodynamic equilibrium. Such software tools provide great insight into implementing dynamical behaviors using DNA. Furthermore, the molecular architecture of DNA provides an elegant way of varying rate constants by orders of magnitude by simply tuning the length of the sub-sequence of the reactant DNA molecule called the "toehold". The following figure illustrates the strand displacement mechanism at the sub-sequence level, which will henceforth be referred to as the "domain" level.

A domain consists of a continuous string of nucleotides that are either fully bound or unbound in their stable configuration (fig. 1.3a). Figure 1.3 is reproduced from [7]. The system used in this study describes the interaction between a two-stranded DNA complex (X) and a single-stranded invader (A). The single-stranded DNA strand A reacts with double-stranded DNA complex X to release strand B and complex Y. Strands that initiate the reaction are called inputs (A) and strands released from the complexes are referred to as outputs (B). The strand-displacement reaction is facilitated by the toehold domains 3 and $3*$: the hybridization of these single-stranded toeholds co-localizes A and X, and allows domain 2 to branch migrate. Branch migration is an unbiased random walk process [29, 33] in which one domain displaces another of identical sequence through a series of reversible single nucleotide dissociation and hybridization steps. At the completion of branch migration, complex Y is formed and strand B is released. Overall, displacement is thermodynamically driven forward by the net gain in base pairs due to the toehold. The progress of strand-displacement reactions is typically assayed using fluorescence, mostly by means of fluorophore quencher reporter complexes that produce readout stoichiometrically proportional to the output.

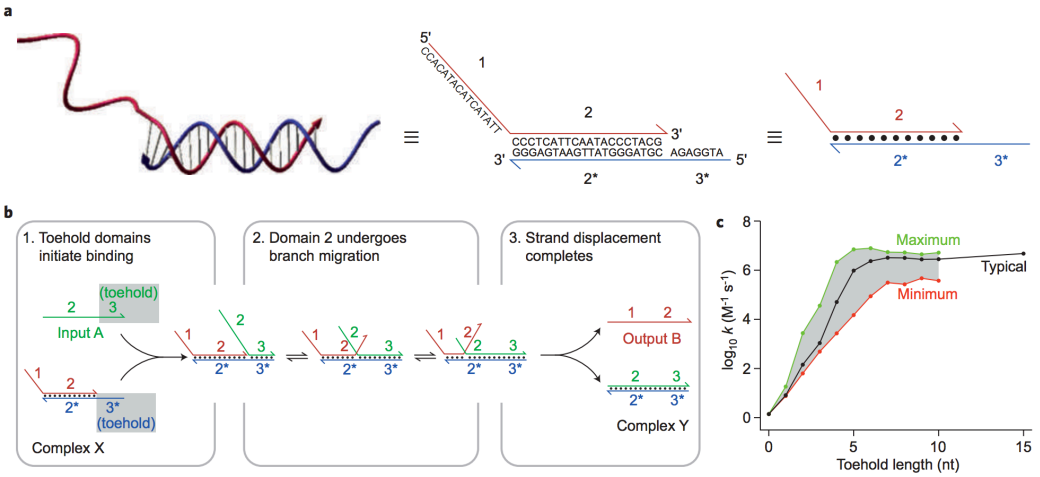

Figure 1.3: **Overview of DNA strand displacement. a,** Abstraction of contiguous DNA bases into functional DNA domains that act as a unit in hybridization, branch migration or dissociation represented by numbers, starred domains denote domains complimentary in sequence to domains without a star. **b,** An example of DNA strand displacement mechanism indicating initiation by toehold binding, continuing to branch migration and ending in disassociation of output strand completing the reaction. **c,** Kinetics of strand displacement modeled and predicted from length and sequence of toehold domain. This figure is reproduced from [7].

The overall effective rate of the strand displacement reaction k_{eff} was first studied by Yurke and Mills [28]. They observed an exponential acceleration in k_{eff} with toehold length. Zhang and Winfree [29] further explored the exponential acceleration of reaction rate with toehold strength and confirmed its saturation in the long toehold limit. They summarized the rate constant of the strand displacement reaction varies over a factor of 10^6, from 1 $M^{-1}s^{-1}$ to 6×10^6 $M^{-1}s^{-1}$ (fig. 1.3c). They verified the kinetics for different toehold strengths confirming that DNA devices based on strand displacement will work for most choices of domain sequences. They tested kinetics with multiple toeholds consisting only G/C nucleotides (green trace), A/T nucleotides (red trace) and finally toeholds that had roughly equal number of all four nucleotides (black trace). The gray region spanned by the green and red traces roughly shows the range of potential kinetics based on toehold length.

The data from both studies suggest that the k_{eff} is an effect of sequence design and variation of toehold sequence and experimental conditions may affect reaction rate. A special case of strand displacement initiation can occur without free nucleotides. Zhang and Winfree [29] investigated the kinetics of this "zero-toehold"

strand displacement and found it to be ~$1 M^{-1} s^{-1}$ at 25 °C in agreement with previous studies [34]. The zero-toehold reactions may cause undesired interactions and need to be considered while designing a strand displacement system. Such detailed biophysical studies of strand displacement systems along with the development of supporting design and analysis software has been the cornerstone of the achievement by the dynamic DNA nanotechnology community.

While DNA strand displacement has been leveraged to implement feedforward Boolean logic circuits [22], scaling up DNA-based digital circuit computation is a daunting challenge and a remarkable step was taken by Qian et al. [25] to build a DNA strand displacement circuit composed of 82 single- and partially double-stranded DNA circuit components in a single test tube representing a four-bit square-root calculating digital logic circuit. The circuit composed of multiple layers of cascading AND-OR seesaw logic gates [35]. Each logic gate had an inbuilt digital signal restoration mechanism that included thresholding and catalysis to assist in tuning circuit behavior. Thus far, another circuit of this size and complexity has not been attempted. The study also introduced a molecular compiler which can in theory, be used to design any arbitrary feedforward DNA-based digital logic circuit using seesaw logic gates. The compiler also provides necessary simulation tools to predict circuit behavior.

Designing and building large-scale DNA circuits is both laborious and time consuming. The square-root circuit has 82 DNA circuit components, in total 130 DNA strands, forming bimolecular complexes. To maintain the desired stoichiometry of molecules and reduce spurious/ undesired reactions, the DNA complexes have to be annealed and purified in-house which requires both time and skilled labor. The compiler that was developed in parallel with the circuit construction has not been independently validated to date.

Chapter 2 of this thesis seeks to independently validate the seesaw compiler by using it to design and implement a large 78-component logic circuit. We define a framework under which a circuit of this size can be tested in a matter of 3-6 days and cuts out the need for intensive purification techniques, making it more user friendly. We believe such a simplistic methodology which has already proved extremely successful in a classroom setting will redefine building complex molecular circuits and will open the testing field to scientists across various disciplines.

1.3 DNA as an engineering material

One restriction with building arbitrarily large DNA systems using toehold mediated strand displacement is the toehold sequence. While using orthogonal toeholds minimize cross talk between various reactant species, there are only a finite number of orthogonal toeholds available at any relevant toehold length. As the system gets larger, a way around is to use universal toeholds as designed in the seesaw system. Universal toeholds are common sequence toeholds wherein strand displacement reactions for all species in the reaction mixture can be initiated in using a single toehold. Since the strand displacement system is implemented in a well-mixed solution, the number of pair-wise interactions possible in the test tube scales quadratically with the size of the DNA implementation. This has the potential to exacerbate cross-talk between species in the test tube and perhaps puts a ceiling on the size of the circuit. Such a problem could be mitigated by lowering the concentration of the species, however, this would also scale down the concentration range of the intended dynamical behavior. In addition, the intended dynamical behavior would occur at much lower rates and would increase the observation time leading to other experimental challenges.

Implementing DNA circuits tethered to a surface could be a possible solution [15, 36, 37] since limiting interactions to species that are co-localized can reduce spurious interactions and allow domains to be re-used multiple times. DNA origami [38] provides one such surface where biomolecules can be tethered using DNA hybridization to localize molecules at distinct locations at the nanoscale.

Paul Rothemund in 2006 introduced the DNA origami technique for the self-assembly of DNA nanostructures with high fidelity. He employed a viral (M13) DNA strand with an established sequence as a scaffold, folding it into the target shape using smaller synthetic DNA strands (staples) with Watson-Crick complementarity to the scaffold strand (fig. 1.4). The staple DNA strand can be extended on either the 5' or the 3' end providing a binding sequence for hybridizing a molecule of interest to the origami. Each of these extensions can be easily made orthogonal by using the available DNA sequence design space, providing unique binding sites for localizing molecules of interest. An origami design software, caDNAno [39] provides an intuitive platform for designing target origami structures and respective staples. The software can be used to design and modify two and three-dimensional nanostructures [39–41]. Such structures have found various applications including drug delivery vesicles [42], mechanical DNA objects [43] and DNA structures that can

be inserted into an artificial cell membrane to act as active membrane channels [44].

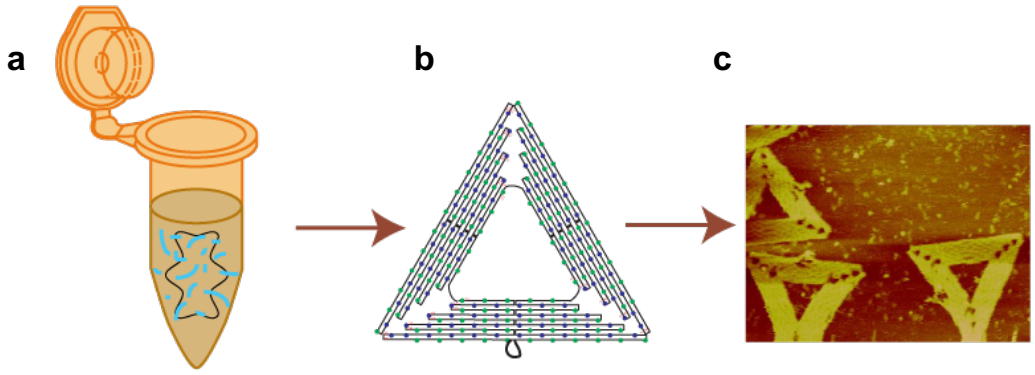

Figure 1.4: **Making DNA origami. a,** M13 viral scaffold is mixed with designed staple strands in 1 × TAE/12.5 mM Mg^{++} and annealed to form target triangle structure designed in caDNAo. **b,** Schematic of designed triangle DNA origami. The blue and green dots represent uniquely addressable 5' staple extension locations. **c,** The triangular nanostructures are visualized using atomic force microscopy.

Harnessing the computational power of the DNA molecule as well as its ability to form complex structures which can act as scaffolds upon which computation can take place has led to the development of nanorobots. Information is programmed into the design of the sequences of the DNA strands making up the robot and the tracks. Programming robots that are capable of tasks ranging from taking a few steps in a single direction [45], to making decisions based on environmental cues [46], to picking up and carrying cargo in the form of gold nanoparticles [47] are a few examples which showcase the extent of control possible at the nanoscale. While there are molecular transporters that can operate without needing external control [48, 49], designing more complex robotic systems which can process information autonomously can be beneficial for implementing complex behaviors. Developing molecular robotic systems capable of responding to external stimuli and pass control signals would be highly applicable in the field of targeted therapeutics. Having molecular robots perform complex cooperative tasks could help implement 'social' robotics [50] and perhaps even be useful in fabrication of molecular devices [51]. Furthermore, these robots provide a unique opportunity to study the biophysics of localized molecules while using minimally invasive techniques for localization [52–54].

Most of the previously designed robots were used to implement a specific task like

undirectional walking or choosing a particular path or picking up cargo. The functionality of these robots have mostly been their ability to move. The cargo-picking robot [47] is capable of walking and picking up 3-cargo molecules using its triangular geometry, and is non-autonomous. The robotic system gets quickly complicated if the number of cargo molecules increase. Also, the robot can only perform a single iteration of cargo pick-up and scaling up the functions of this robot is a non-trivial task. Our work (covered in chapters 4, 5) combines the modularity of an autonomous robot capable of walking and the performance of a programmed task using simple algorithm.

Chapter 3 of this thesis addresses a systematic approach to identifying and characterizing an important module for a nanorobot capable of performing 1-dimensional random walk on a DNA nanostructure using reversible strand displacement reactions. The robot walks exclusively by using base-pair hybridization and disassociation, allowing it to be used for continuous exploration of the nanoscale surface using uniquely positioned tracks, without exhausting any kind of fuel.

Chapter 4 expands the functionality of the designed random-walking robot by programing it to explore the 2-dimensional surface of the nanostructure. In concert with the exploration, the robot is programmed to perform the complex task of sorting randomly distributed cargo to their designated goal locations on the nanoscale surface of DNA origami. Our experimental demonstration shows the sorting of two types of cargo, but in principle, the system can be scaled up to sort a larger number of cargo types to their designated goal locations, using the same robot design.

Chapter 2

SYSTEMATIC CONSTRUCTION OF DNA CIRCUITS

A hallmark of electrical engineering has been the development of simple primitives aiding the construction of complex electrical circuits. One of the primary aims of molecular programming is to develop a library of primitives which can be used to construct complex molecular circuits. To that end, biomolecular DNA components have been successfully implemented using toehold mediated strand displacement reactions to device small circuits. The simple bimolecular components are cascaded to engineer molecular devices with interesting dynamic properties which include nano-scale machines like molecular tweezers [21], catalytic amplifiers [22] and molecular oscillators [26, 55].

To further advance the engineering of such complex devices, software tools such as molecular compilers capable of translating high-level instructions to low-level molecular interpretation have been developed. A molecular reaction system compiled to DNA molecules has three main steps. The first step is where the system is run through a compiler to break it down into domain specific DNA molecules. Next, the domains are converted into DNA sequences and nucleotides. And finally, the sequences are sent out to a commercial vendor for synthesis before the reaction can be implemented in a laboratory setting.

Multiple compilers that convert reaction mechanisms to DNA domains have been developed. Soloveichik et al., [56] developed a compiler to translate arbitrary chemical reactions to real and experimentally implementable reactions. Visual DSD [32] was developed as a design and analysis tool for DNA strand-displacement systems. The software provides support for enumerating the designed reactions and simulating them.

The next step to implementing molecular systems is to translate the DNA domains produced by the compiler into DNA sequences. Some compilers have an inbuilt function to generate DNA sequences for the corresponding circuit design [25], special software tools like NUPACK [12] have been developed specifically to design nucleic acid sequences and ascertain the thermodynamic properties of designed molecules. Several other tools assisting with calculating kinetic properties of designed molecules [57] and verifying molecular system behavior agree with the de-

sired higher-level functions [32, 58, 59] have also been developed.

While some of the compilers mentioned above have been developed by incorporating experimental data [25, 56], none have been independently validated, i.e, these compilers have not been used to design and test additional DNA circuits.

The final stage of circuit construction involves the synthesis of designed DNA sequences. Most researchers use commercial vendors for this purpose. Decades of research has resulted in the synthesis of synthetic oligonucleotides being cheaper, efficient and easily obtainable [9]. The oligos can be ordered purified or unpurifed. Purification of oligos is performed either by poly acrylamide gel electrophoresis (PAGE) or high pressure liquid chromatography (HPLC) columns, where molecules with truncated bases are separated from designed molecules. Purification increases the cost of oligos by 10-fold. Several researchers prefer to perform in-house purification using PAGE to separate malformed DNA structures from desired structures. In-house purifications are time and labor intensive. However, purification of circuit components reduce undesired products that are caused by stoichiometric errors [25, 60]. While purification has certain advantages, it significantly increases the cost and time required for building large-scale DNA circuits.

This chapter focuses on independent validation of a molecular compiler [14] that was used to generate DNA sequences for implementing a large-scale DNA circuit. Using a systematic hierarchical method, we show that unpurified strands can be successfully used to implement DNA circuits.

Review of the DNA Seesaw motif

This section is a brief review of the seesaw DNA motif developed by Qian et al to scale up complexity of DNA circuits [35] and demonstrating their ability to implement digital logic computation [25] and neural network computation [61]. In the seesaw abstraction, each DNA gate is represented by a two-sided node (fig. 2.1a). Each DNA signal is represented by a wire. Each side of the node can be connected to any number of wires. Each wire connects two different sides of two nodes. Each red number indicates one DNA species with its initial relative concentration. Each number on a wire corresponds to a free signal strand; each number within a node at the end of a wire corresponds to a bound signal strand (positive number) or a threshold that absorbs a signal when it arrives at the gate (negative number). The fuel molecule contributes to the catalytic nature of the reaction. A reporter that transforms a DNA signal into a fluorescence signal is represented by half a node

with a zigzag arrow, with its initial relative concentration indicated by the number inside the hemispherical node.

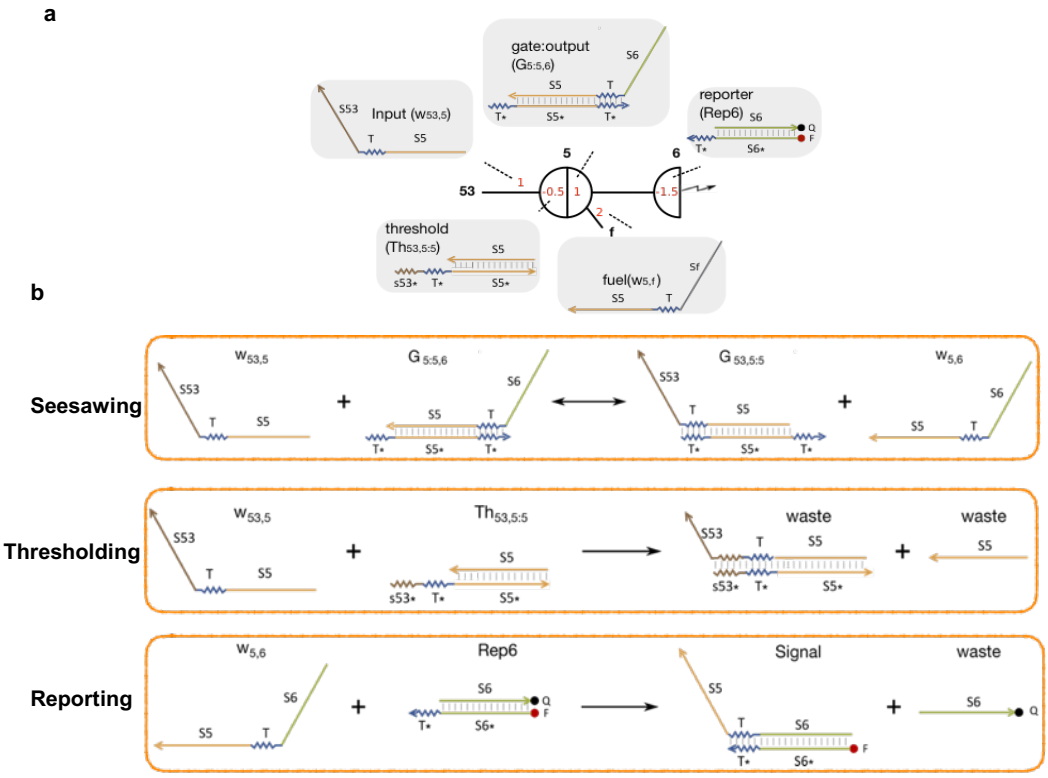

Figure 2.1: **T**he Seesaw DNA motif. **a**, Each DNA gate is represented by a two-sided node, connected through wires. **b**, There are 3 types of reactions: the first is seesawing where the input strand can reversibly displace the top-strand of the gate:output complex. The second is the thresholding reaction that irreversibly binds the input strand. The final is the reporting reaction where the output strand from the seesawing reaction displaces the quencher strand from the fluorophore-quencher reporter complex producing a fluorescent signal.

There are three basic reactions involved in a seesaw network (fig. 2.1b). The first is seesawing: A free signal on one side of a gate can release a signal bound on the other side of the gate by toehold-mediated strand displacement. The process starts with the free signal strand (e.g., w53,5) hybridizing to the gate:signal complex (e.g., G5:5,6) at the uncovered toehold domain (e.g., T*) and then undergoing branch migration through the recognition domain (e.g., S5). The previously bound signal will fall off when it is attached to the gate base strand only by the short toehold. The resulting gate:signal complex (e.g., G53,5:5) will have an uncovered toehold on the other side, and therefore the now-free signal (e.g., w5,6) can reverse the process symmetrically. The second reaction is thresholding: a threshold species associated

with a gate and an impinging signal can react with the signal by means of a longer toehold (e.g., s53*T*), producing only inert waste species that have no exposed toehold. Thresholding is much faster than seesawing because the toehold-mediated strand displacement rate grows exponentially with toehold length for short toeholds [29]. As a result, seesawing effectively only happens when the input signal exceeds the threshold. The third reaction is reporting: A reporter species similar to a threshold, but modified with a fluorophore and quencher pair, can absorb an impinging signal while generating a fluorescence signal. Unlike thresholding, reporting does not compete with seesawing, and it therefore does not require a longer toehold.

Using the seesaw motif, we can build two distinct type of seesaw gates. The first kind is the amplifying gate which has the threshold and fuel. If input is greater than the threshold, the output is released catalytically. It can support multiple outputs. The second kind is the integrating gate which has no threshold or fuel and the output is released as sum of inputs. It can support multiple inputs. An integrating gate followed by an amplifying gate can compute either OR or AND logic. A two-input OR gate will have an integrating gate that outputs the sum of the two inputs. The downstream amplifying gate will output 1 when the sum is greater than 0.6 and will output 0 otherwise. In practice, the outputs will not be exactly 0 or 1 because of spurious or incomplete reactions, so we must ensure that logic gates will function correctly even with imperfect inputs. Assuming a digital abstraction where OFF signals may be in the range 0 to 0.2 and ON signals in the range 0.8 to 1, we see that only when both inputs are OFF can the output remain OFF. Similarly, changing the threshold from 0.6 to 1.2 computes AND. In this case, only when both inputs are ON can the sum exceed the threshold and catalyze the output to be ON.

2.1 Circuit design

The Seesaw Compiler [14, 25] automatically translates an arbitrary feedforward digital logic circuit into its equivalent seesaw circuit (fig. 2.2). Input to the compiler is a logic circuit consisting of AND, OR,NAND,NOR and NOT gates with their connectivity specified. The compiler then generates an equivalent dual-rail circuit file consisting of AND-OR gates only [62]. NOT gates are difficult to implement directly using representations where the ON or OFF state of an input is determined by the presence or absence of a single DNA species: a circuit might compute a false output before all input strands are added, because NOT gates already produce ON signals in the absence of their inputs, and for use-once circuits (such as seesaw circuits), computations cannot be undone. Therefore, we use dual-rail logic. Each

input is replaced by a pair of inputs, representing logic ON and OFF separately. Each logic gate in the designed circuit is replaced by a pair of AND or OR gates. (Taking the NOR gate as an example, output being OFF is the OR of both inputs being ON; output being ON is the AND of both inputs being OFF.)

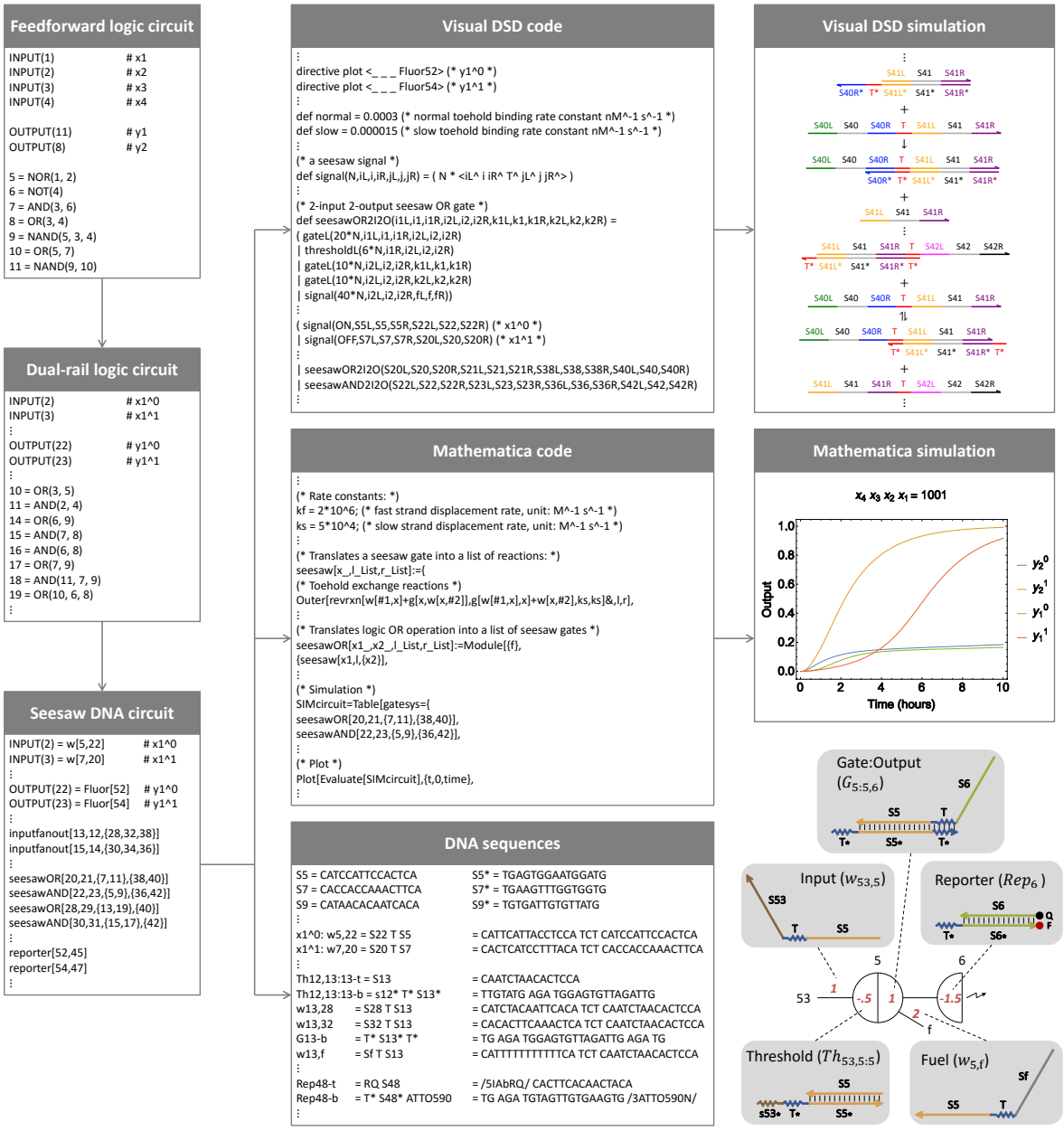

Figure 2.2: **Automated circuit design steps using the Seesaw Compiler.** A feedforward digital logic circuit is first translated into an equivalent duel-rail logic circuit, and then translated into an equivalent seesaw DNA circuit. Visual DSD code and Mathematica code are generated for analyzing and simulating the seesaw DNA circuit, and DNA sequences are generated for constructing the circuit. Bottom right diagram introduces the notations of seesaw circuits: black numbers indicate identities of nodes. The locations and values of red numbers indicate the identities of distinct DNA species and their relative initial concentrations, respectively.

The compiler then translates the dual-rail circuit file into seesaw circuit files speci-

fying gates and their connections, and relative concentrations of all initial species. As discussed in chapter 2, each signal molecule in the DNA circuit is defined as a wire $w_{j,i}$ connecting seesaw nodes j and i and is implemented using a single stranded DNA molecule. The seesaw nodes consist of double stranded threshold, gate: output and single stranded fuel molecules. Input fan-out gates are introduced to accept an input signal used or multiple logic gates and produce the corresponding number of output signals. Reporter molecules are fluorophore:quencher double stranded components which generate fluorescence in the presence of an output molecule, providing a distinct readout.

The compiler also generates the simulation code for the circuit that can be used as input in the Visual DSD tool (https://www.microsoft.com/en-us/research/project/programming-dna-circuits/) [13, 32] in order to generate chemical reaction diagrams with DNA domain level representation and simulate circuit behavior based on the chemical reactions. More importantly, the compiler provides a Mathematica code to simulate circuit behavior using mass action kinetics. This code uses the CRNSimulator package [63] and allows the user to alter reaction rates, concentrations and spurious interactions to get a detailed understanding of circuit dynamics. The compiler output also includes a file of DNA sequences for signal strands and for each top and bottom strand of threshold, gate, and reporter complexes to implement the seesaw circuit intended to perform the original digital logic computation.

To experimentally validate the seesaw compiler, we designed a single feedforward Boolean circuit to implement two one-dimensional distinct elementary cellular automata transition functions. Elementary cellular automata (CA) are one of the simplest models of computation [64] known to be Turing-complete. An elementary CA consists of one-dimensional grid consisting of individual cells with a binary state of 1 or 0. The fate of a single central cell (C) in each subsequent generation is based on its current state and the state of its neighboring Left cell (L) and Right cell (R). A state transition rule maps each of the 8 (2^3) possible states for L, C and R to either 1 or 0. Therefore, a binary string of length 8 can identify each of the 2^8 transition functions to specify the evolution of the elementary CA in subsequent generations. While the rule 110 CA (binary representation : 01101110) is famously known to be Turing-universal [65], other related but equally powerful CAs are not given their due. This circuit investigates the suitability and differences of the rule 124 CA (binary representation : 01111100) with respect to its famous counterpart. Applying

a mirror transformation to rule 110 CA generates rule 124 CA. The combined rule 110-rule 124 circuit consisting of five logic gates in two layers has been carefully designed so that it can be composed from smaller, yet interesting circuits to implement interesting logic function but does not implement the actual CA model. The efficacy of individual gates is evaluated to incrementally build the circuit in multiple phases in a well-mixed solution and does not involve spatial dynamics.

The rule 110-rule 124 DNA circuit generated by the seesaw compiler implements dual rail logic with six layers consisting a total of 78 initial DNA species (fig. 2.3c). We then use the Mathematica code generated by the compiler to simulate the designed circuit to predict the expected behavior for 8 different inputs under ideal experimental conditions (fig. 2.3c). Finally, the compiler designed DNA sequences for the individual circuit components were purchased unpurified. The circuit was constructed using the unpurified DNA strands with no further in-house purification. Overall, the seesaw compiler provided an extremely easy and efficient way to automate the design process. On the experiment side, using unpurified DNA components meant additional noise introduced by synthesis and stoichiometric errors which can pose a challenge to build large-scale circuits. Hence, we took a bottom-up approach in building this system, beginning by testing its simplest functional component - digital signal restoration.

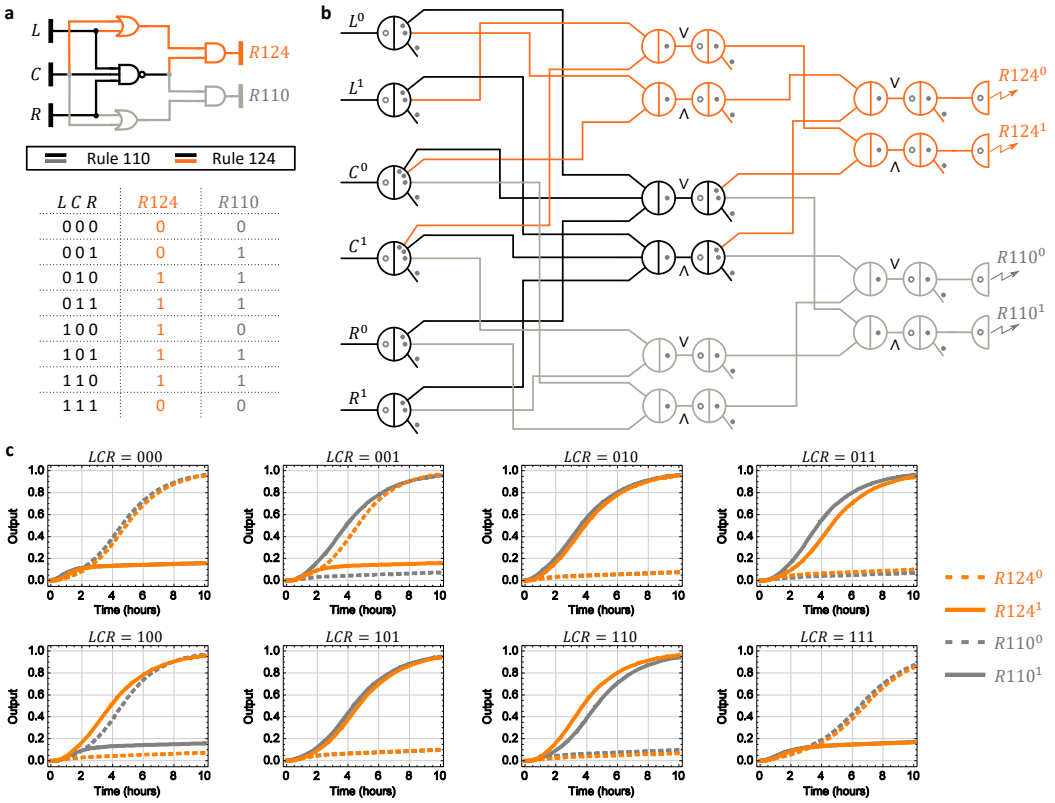

Figure 2.3: **Design of a rule 110-124 circuit using the Seesaw Compiler.** **a**, Gate diagram and truth table of a digital logic circuit that computes the transition rules 110 and 124 of elementary cellular automata. **b**, Seesaw gate diagram of the equivalent DNA strand displacement circuit. Each seesaw node connected to a dual-rail input implements input fan-out. Each pair of seesaw nodes labeled $\wedge$ and $\vee$ implements a dual-rail AND and OR gate, respectively. Each pair of dual-rail AND and OR gates implements an AND, OR or NAND gate in the original logic circuit. Each dual-rail output is converted to a fluorescence signal through a reporter, indicated as a half node with a zigzag arrow. Each circle and dot inside a seesaw node indicates a double-stranded threshold and gate molecule, respectively. Each dot on a wire indicates a single-stranded fuel molecule. **c**, Simulations of the DNA strand displacement circuit using the previously developed model for purified seesaw circuits. Trajectories and their corresponding outputs have matching colors. Overlapping trajectories were shifted to be visible. Dotted and solid lines indicate dual-rail outputs that represent logic OFF and ON, respectively. For example, when input $LCR = 001$, meaning L^0, C^0 and R^1 were introduced at a high concentration and L^1, C^1 and R^0 at a low concentration, two output trajectories $R124^0$ and $R110^1$ reached an ON state and the other two output trajectories $R124^1$ and $R110^0$ remained in an OFF state, indicating that the output was computed to be 0 and 1 for rule 124 and 110, respectively. Simulations were performed at $1\times = 50$ nM — the compiler recommended standard concentration for large-scale purified seesaw circuits.

2.2 Calibrating effective concentrations.

To begin building the circuit bottom-up, we first tested the digital signal restoration, which involves signal amplification and thresholding. The expected circuit behavior using the seesaw compiler predicts that input signal less than the threshold should be cleaned up to an ideal OFF state, and input greater than the threshold should be amplified to an ideal ON state eq. (2.1). The seesaw circuit has the digital signal restoration as a component in every logic gate.

$$y = \begin{cases} 1 & x > th \\ 0 & x \leq th \end{cases} \tag{2.1}$$

At the molecular level, the digital signal restoration process consists of two basic reactions: catalysis and thresholding. Catalysis is implemented with two toehold exchange pathways that release free output strands $w_{i,k}$ from double-stranded gate molecules $G_{i:i,k}$, using the input strands $w_{j,i}$ as a catalyst (Supplementary fig. 2.13a):

$$w_{j,i} + G_{i:i,k} \xrightarrow{k_s} w_{j,i} + w_{i,k} \tag{2.2}$$

Catalysis can be used for signal amplification, since a small amount of input can trigger the release of a much larger amount of output.

Thresholding is implemented with double-stranded threshold molecules $Th_{j,i:i}$ consuming the input at a much faster rate ($k_f \gg k_s$) than the input acting as a catalyst (Supplementary fig. 2.13b):

$$w_{j,i} + Th_{j,i:i} \xrightarrow{k_f} \emptyset \tag{2.3}$$

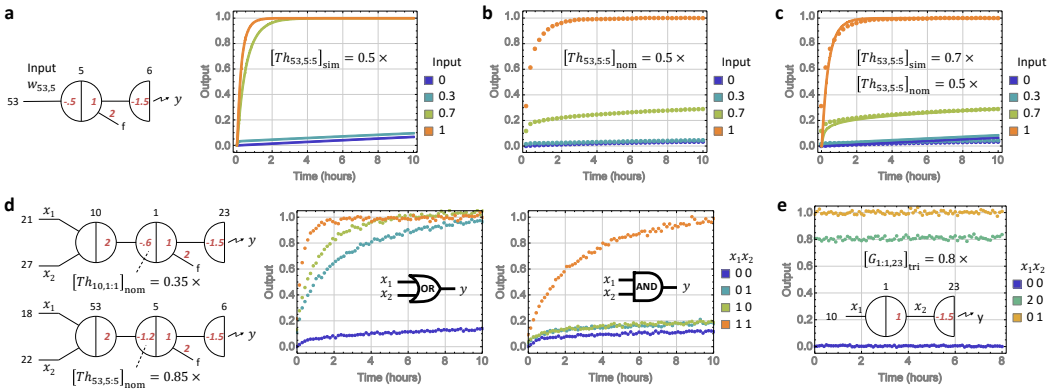

Figure 2.4: **Calibrating effective concentrations.** **a**, Simulations and **b**, experimental data of digital signal restoration. **c**, Estimating effective threshold concentration by fitting simulations to the data obtained. **d**, OR and AND logic gates constructed using adjusted nominal threshold concentrations. **e**, Estimating effective gate concentration. Data show steady state fluorescence level. $1\times = 100$ nM. Here and in later figures, all output signals in the data were normalized using the minimum fluorescence signal (the first data point) of an OFF trajectory as 0, and the maximum fluorescence signal (the average of the last five data points) of an ON trajectory as 1.

The Seesaw Compiler was to used simulate digital signal restoration with threshold molecule concentration of $0.5\times$ ($1\times = 100\ nM$), with multiple input signal concentrations ranging from $0\times$ to $1\times$. The expected behavior (fig. 2.4a) resulted in an OFF state with an input less than the threshold molecule concentration (i.e, $0\times$ and $0.3\times$), cleaned up by eq. (2.3). An input greater than the threshold molecule concentration (i.e, $0.7\times$ and $1\times$), resulted in an ON state, amplified via eq. (2.2).

However, the observed circuit behavior (fig. 2.4b) input $0.7\times$ that was more than the threshold concentration of $0.5\times$ did not reach the ideal ON state. The difference between expected (simulation) and observed (data) circuit behaviors suggests that the effective concentration of an unpurified threshold species, compared to that of an unpurified signal species, was higher than expected.

In order to resolve the above conflict, we needed to investigate the source leading to this concentration difference. Typically, spectrometric absorbance at 260 nm is used to quantify the nominal concentration of DNA species. This measured concentration can be greater than the effective concentration of the species that can actively take part in the desired reaction. Assuming that the DNA sequences are correctly designed, synthesis errors including nucleotide insertion, deletion and mutation can cause the effective concentration of the DNA species to be less than the

nominal concentration. To estimate the effective concentration of unpurfied DNA molecules, we defined the ratio of effective (eff) to nominal (nom) concentrations of an arbitrary signal, threshold and gate molecules,

$$\alpha_{j,i} = \frac{[w_{j,i}]_{\text{eff}}}{[w_{j,i}]_{\text{nom}}} \tag{2.4}$$

$$\beta_{j,i} = \frac{[Th_{j,i:i}]_{\text{eff}}}{[Th_{j,i:i}]_{\text{nom}}} \tag{2.5}$$

$$\gamma_{i,k} = \frac{[G_{i:i,k}]_{\text{eff}}}{[G_{i:i,k}]_{\text{nom}}} \tag{2.6}$$

The digital signal restoration is composed of an input signal, threshold and gate:output molecule which triggers a reporter molecule for fluorescence readout. Hence the absolute concentrations of each of the species cannot be measured individually. Further, the absolute concentrations will only affect the speed of the reactions, but does not guarantee the correctness of computation. We therefore proceeded to compare the data (fig. 2.4b) with simulation (fig. 2.4a) to estimate the effective concentration of an unpurified threshold species, compared to that of an unpurified signal species. By changing the threshold value in the simulation to 0.7× the simulation results agreed with the experiment results. From this, we inferred that the effective concentration of the signal was similar to that of the signal for nominal concentration of threshold $[Th_{53,5:5}]_{\text{nom}} = 0.5\times$ and nominal concentration of the input signal $[w_{53,5}]_{\text{nom}} = 0.7\times$.

Thus the threshold to signal ratio can be calculated as:

$$\begin{aligned}\frac{\beta_{53,5}}{\alpha_{53,5}} &= \frac{[Th_{53,5:5}]_{\text{eff}}}{[Th_{53,5:5}]_{\text{nom}}} \times \frac{[w_{53,5}]_{\text{nom}}}{[w_{53,5}]_{\text{eff}}} \\ &= \frac{[w_{53,5}]_{\text{nom}}}{[Th_{53,5:5}]_{\text{nom}}}\Big|_{[Th_{53,5:5}]_{\text{eff}}=[w_{53,5}]_{\text{eff}}} \\ &= \frac{0.7}{0.5} = 1.4\end{aligned} \tag{2.7}$$

The effective concentration of unpurified threshold molecules may be higher than the signal molecules when the nominal concentrations are the same because error introduced during synthesis. The process of chemical synthesis involves the coupling of nucleotides sequentially, and the coupling efficiency in each step is less than 100% [66]. Signal molecules (33 nucleotides) are longer than threshold molecules (top strand: 15 nucleotides, bottom strand: 25 nucleotides), hence the probability

of synthesis error is higher in the signal molecule than in the components of the threshold molecule.

Signal restoration experiments were performed on additional components of the rule110-rule124 circuit and the threshold to signal ratio $\beta/\alpha = 1.4$ was consistent for different threshold and signal molecules (Supplementary fig. 2.14). Hence this ratio was used to adjust the nominal thresholds for computing 2 and 3-input AND and OR logic.

An upstream integrating node coupled with a downstream amplifying node is used to compose a seesaw logic gate. Ideally, an integrating node outputs the sum of all inputs:

$$y = \sum_{i=1}^{n} x_i \tag{2.8}$$

A two-input logic function can be computed as:

$$y = \begin{cases} 1 & x_1 + x_2 > th \\ 0 & x_1 + x_2 \leq th \end{cases} \tag{2.9}$$

Assuming that an ideal OFF state is $[0, 0.2]$ and an ideal ON state is $[0.8, 1]$, $th = 0.6$ will compute logic OR and $th = 1.2$ will compute logic AND, if the effective concentrations of the threshold and input signals are comparable to each other (i.e. $\beta/\alpha = 1$).

Since $\beta/\alpha \neq 1$ for unpurified threshold and signal molecules, we can take this ratio into consideration while calculating the lower and upper bounds of the nominal threshold for an n-input logic gate:

$$0.2n \times \frac{\alpha}{\beta} \leq [Th_{\text{OR}}]_{\text{nom}} < 0.8 \times \frac{\alpha}{\beta} \tag{2.10}$$

$$[(n-1) + 0.2] \times \frac{\alpha}{\beta} \leq [Th_{\text{AND}}]_{\text{nom}} < 0.8n \times \frac{\alpha}{\beta} \tag{2.11}$$

Using $\beta/\alpha = 1.4$, we chose a nominal threshold of 0.35× and 0.85× for two-input OR and AND gate, respectively, and 0.4× and 1.6× for three-input OR and AND gate. Experiments of the logic gates showed desired behaviors (fig. 2.4d and Supplementary fig. 2.15).

An alternative approach for adjusting the nominal threshold is to use the following equations:

$$[Th_{\text{OR}}]_{\text{nom}} = 0.6 \times \frac{\alpha}{\beta} \tag{2.12}$$

$$[Th_{\mathrm{AND}}]_{\mathrm{nom}} = [(n-1) + 0.2] \times \frac{\alpha}{\beta} \tag{2.13}$$

Compared to choosing a nominal threshold based on the lower and upper bounds, this approach is less flexible but simpler.

Similarly the ratio between γ and α for a gate releasing a signal can be estimated using an experiment that compares the fully triggered (tri) concentration of the gate with the signal when their nominal concentrations are the same. For example, the data in fig. 2.4e showed that $[G_{1:1,23}]_{\mathrm{tri}} = 0.8\times$ when $[G_{1:1,23}]_{\mathrm{nom}} = [w_{1,23}]_{\mathrm{nom}} = 1\times$. Thus the gate to signal ratio can be calculated as:

$$\begin{aligned} \frac{\gamma_{1,23}}{\alpha_{1,23}} &= \frac{[G_{1:1,23}]_{\mathrm{eff}}}{[G_{1:1,23}]_{\mathrm{nom}}} \times \frac{[w_{1,23}]_{\mathrm{nom}}}{[w_{1,23}]_{\mathrm{eff}}} \\ &= \frac{[G_{1:1,23}]_{\mathrm{eff}}}{[w_{1,23}]_{\mathrm{eff}}}\Big|_{[G_{1:1,23}]_{\mathrm{nom}}=[w_{1,23}]_{\mathrm{nom}}} \\ &= \frac{0.8}{1} = 0.8 \end{aligned} \tag{2.14}$$

Additional gate calibration experiments suggested that the ratio $\gamma/\alpha = 0.8$ was consistent for different gate and signal molecules (Supplementary fig. 2.16). We suspect that due to synthesis errors in gate molecules, not all gates can successfully release a signal, which is why an unpurified gate has a lower effective concentration compared to a signal.

As signal restoration was built in within every logic gate to accept an ON state of $[0.8, 1]$, we decided not to make any adjustment for nominal gate concentrations if $\gamma/\alpha \geq 0.8$. Otherwise, nominal concentration of an amplifying gate and an n-input integrating gate can be adjusted as:

$$[G_{\mathrm{AMP}}]_{\mathrm{nom}} = 1 \times \frac{\alpha}{\gamma} \tag{2.15}$$

$$[G_{\mathrm{INT}}]_{\mathrm{nom}} = n \times \frac{\alpha}{\gamma} \tag{2.16}$$

Importantly, the values of α, β and γ should depend on the strand quality, and thus could vary with different DNA synthesis providers, procedures, and batch orders. It is necessary to recalculate the ratios β/α and γ/α, if these conditions change.

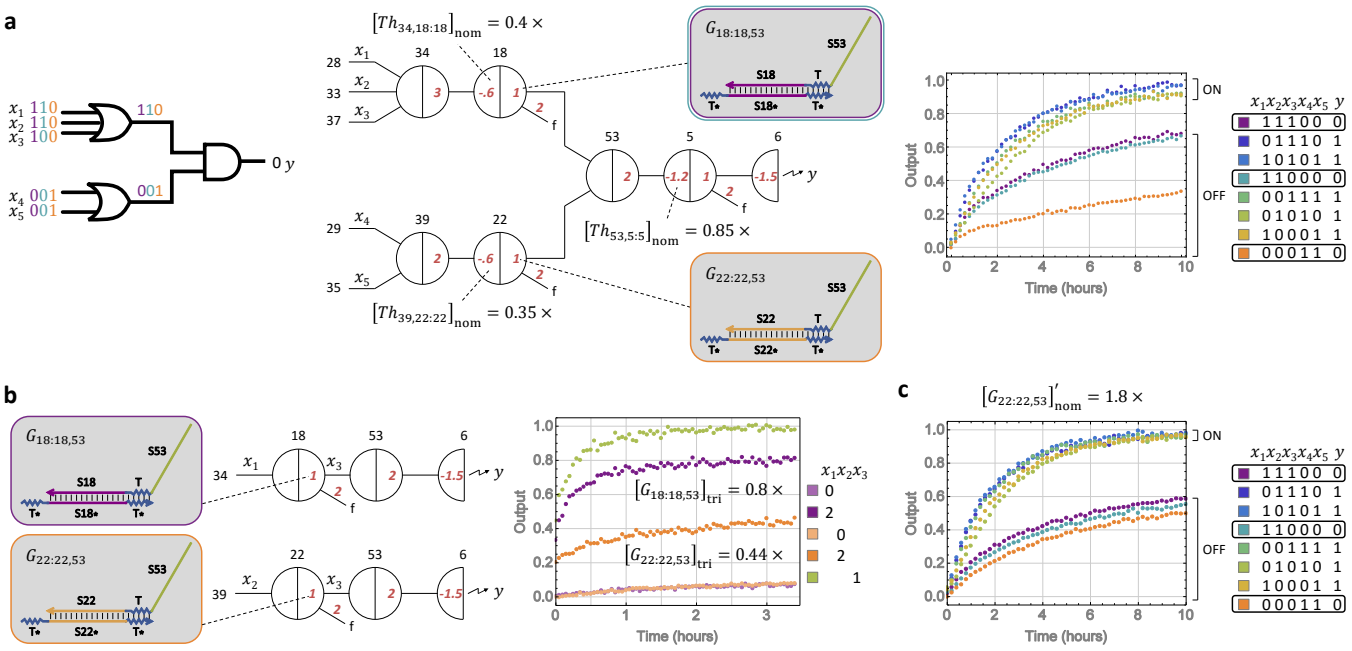

Figure 2.5: **Identifying an outlier gate. a**, Logic circuit diagram, seesaw circuit diagram, and experimental data of a two-layer logic circuit. **b**, Measuring the effective concentrations of the gate species. Three independent circuits were used to measure the effective concentrations of two gates fully triggered by x_1 and x_2, respectively, comparing to the effective concentration of x_3 (using signal strand $w_{18,53}$). **c**, Experimental data of the two-layer logic circuit using adjusted nominal gate concentration. $1\times = 100$ nM.

2.3 Identifying outliers.

Next we move to test a larger part of the designed circuit by cascading an upstream layer of OR gates into the single AND gate already tested (fig. 2.4d). The upstream OR gates were not tested individually, but the nominal threshold concentration of the 3 and 2-input OR gate was calculated using the measured ratio from the digital restoration gate eq. (2.10). The components of this two layer logic circuit is part of the rule 124 sub-circuit. The expected circuit behavior was plotted showing that the output should remain OFF when only one of the upstream OR gates is ON. The observed circuit behavior showed that the output was reasonably OFF when one upstream OR gate was ON ($x1 = x2 = x3 = 0; x4 = x5 = 1$), but was half ON when the other upstream OR gate was ON ($x1 = x2 = x3 = 1; x4 = x5 = 0$) (fig. 2.5a). This experimental result suggests that the two inputs to the AND gate from the upstream OR gates were not symmetrical, i.e , it there was a variation in the effective concentrations of the upstream unpurified gate species leads to unbalanced signal wires.

The two upstream OR gates were then tested to verify if $\gamma/\alpha \neq 0.8$ (eq. (2.16). The data in fig. 2.5b shows a gate calibration experiment to check for outlier gates. The ratios resulted in $\gamma_{18,53}/\alpha_{18,53} = 0.8\times$ for one gate and $\gamma_{22,53}/\alpha_{22,53} = 0.44\times$ for another, highlighting the outlier gate.

Lower concentration of one of the gates could be the result of synthesis error in the strands used to compose the gate. There is a higher probability that the synthesis of DNA strand for this particular gate were lower than average quality, than the DNA strands for the other gate being higher than average quality.

The threshold or the gate concentration of this outlier could be modified using its threshold to signal ratio (β/α) or gate to signal ratio (γ/α), the common nominal concentration described in the previous section, and the common ratio for other thresholds and gates:

$$[Th_{j,i:i}]'_{\text{nom}} = [Th_{j,i:i}]_{\text{nom}} \times \frac{\alpha_{j,i}}{\beta_{j,i}} \times \frac{\beta}{\alpha} \tag{2.17}$$

$$[G_{i:i,k}]'_{\text{nom}} = [G_{i:i,k}]_{\text{nom}} \times \frac{\alpha_{i,k}}{\gamma_{i,k}} \times \frac{\gamma}{\alpha} \tag{2.18}$$

In this case, the unbalanced wires do not completely reduce the ON/OFF gap and can therefore be fixed by increasing the downstream threshold ($[Th_{53,5:5}]_{\text{nom}}$). But in other cases, unbalanced wires could result in significantly reduced ON/OFF gaps that cannot be resolved by increasing the downstream thresholds. In order to find a universal solution that can be applied across the entire circuit, we measure the difference between the effective concentrations of unpurified gate species that caused the unbalanced wires. In order to do this, we first connect the downstream seesaw node directly to a reporter. We then remove the threshold and fully trigger the gate in question with excess input signals (2×) and compare the triggered levels of the two gates (fig. 2.5b). To balance the two wires, we could either increase the nominal concentration of one gate ($[G_{22:22,53}]_{\text{nom}}$) or decrease the nominal concentration of the other ($[G_{18:18,53}]_{\text{nom}}$). We chose the former, because the triggered output of $[G_{18:18,53}]_{\text{nom}}$ was found to be equal to the output triggered by signal $w_{18,53}$, while the triggered output of ($[G_{22:22,53}]_{\text{nom}}$) was much lower than then output triggered by $w_{22,53}$. We adjusted the nominal concentration of unpurified gate $[G_{22:22,53}]'_{\text{nom}} = 1/0.44 \times 0.8 = 1.8\times$. Hence our solution to the challenge of unbalanced wires is to measure the difference between the effective concentrations of unpurified gate species that caused the unbalanced wires, and adjust the nominal

concentrations of them accordingly. By implementing this change in gate concentration, we observed that the ON trajectory was at an ideal high fluorescence state (fig. 2.5b). The OFF trajectories were lower, at roughly the same level for different combinations of inputs, but were not at an ideal low fluorescence state. This required further tuning of the unpurified cascaded sub-circuit.

2.4 Tuning circuit output.

The downstream AND gate when tested in isolation showed expected computation (fig. 2.4d) with significant ON/ OFF separation, but cascading the AND gate with two upstream OR gates decreased the ON/OFF signal gap of the circuit. The expected circuit behavior is that the output should remain OFF when either upstream OR gate is OFF. However, erroneous ON cases were observed experimentally when one of the OR gates was expected to be OFF while the other OR gate had more than one input being ON. This suggests that unpurified circuits are noisier than expected, compared with purified circuits [25], and the behavior becomes less robust with increasing number of layers.

One possible reason for the increased noise in unpurified circuit is stoichiometry. All gate molecules were prepared by annealing equal proportion of top signal strand and bottom sink strand constituting the gate molecule. Both top and bottom strands have toeholds and migration domains that can interfere with downstream components if present in excess. The equal proportion was assumed by the concentration information provided by the oligo vendor. However, small discrepancies in pipetting volumes or concentration accuracy could easily off-set the stoichiometry of the two strands. In-house purification of such annealed complexes remove excess unbound strands. In the absence of this purification step, even a small excess of strands would result in undesired output release in down stream gates, introducing extra noise. Therefore, with every additional layer, this noise may increase.

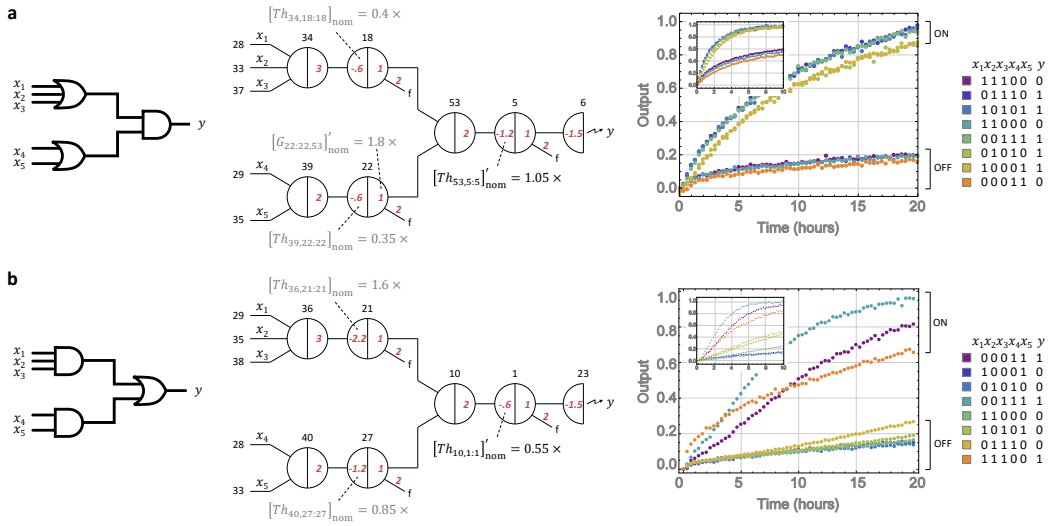

Figure 2.6: **Tuning circuit output.** Logic circuit diagram, seesaw circuit diagram, and experimental data of a two-layer logic circuit with **a**, two upstream OR gates connected to an downstream AND gate, and **b**, two upstream AND gates connected to an downstream OR gate. Nominal concentrations shown in gray and black indicate adjustments made in a previous step and in this step, respectively. Small insets of experimental data show the circuit behaviors before adjustments. 1× = 100 nM.

The threshold function in every seesaw logic gate can absorb some of the noise from upstream components. Hence, by simply tuning the threshold, the output of the circuit can be tuned. This was done by estimating the ON/OFF separation of the circuit output in fig. 2.5b. From the data, we can choose a trajectory that computes logic ON and a logic OFF. Next, we can calculate the difference in the observed OFF value and the ideal OFF value (δ) when the ON trajectory is at its ideal ON value. With 0.7 and 0.3 as the lower bound and 0.9 and 0.1 as the upper bound for an ideal ON/ OFF separation, the range of δ can be determined:

$$y^{\text{OFF}}\Big|_{y^{\text{ON}}=0.7} - 0.3 \leq \delta \leq y^{\text{OFF}}\Big|_{y^{\text{ON}}=0.9} - 0.1 \tag{2.19}$$

The nominal threshold in the logic gate that produces the circuit output can then be adjusted accordingly:

$$[Th_{j,i:i}]'_{\text{nom}} = [Th_{j,i:i}]_{\text{nom}} + \delta \times \frac{\alpha}{\beta} \tag{2.20}$$

Using the data of the two-layer logic circuit shown in fig. 2.5c, we chose the trajectory with input = 01010 and 11100 as the reference ON and OFF trajectory, respectively, and calculated $0.08 \leq \delta \leq 0.41$. We then increased the threshold in

the downstream AND gate to $[Th_{53,5:5}]'_{\text{nom}} = 0.85 + 0.28/1.4 = 1.05\times$ and repeated the experiment. The circuit behavior was improved with a much better ON/OFF separation (fig. 2.6a).

The same method was then used to test another two layer sub-circuit of rule-124 consisting of an OR gate cascaded with two upstream AND gates. (fig. 2.6b). In this case, using input = 00011 and 01110 as the reference ON and OFF trajectories, we obtained a similar range of δ, and decided to apply the same amount of increase to the threshold in the downstream OR gate (fig. 2.6b). We chose the slowest ON trajectory and the fastest OFF trajectory for threshold adjustment, but it is possible to make other choices based on knowledge of the data set. Choosing the slowest ON trajectory was a choice made on the theory that increasing the threshold concentration would slow down circuit dynamics in general (fig. 2.5b versus fig. 2.6a). Hence, threshold concentration can only be adjusted when the circuit dynamics show ON trajectories being significantly faster than the OFF trajectories.

After having individually tuned the two sub-circuits of rule-124, we combined the two and added the remaining fan-out gates for proving input signal fro the multiple gates, and were successful in implementing rule-124 successfully (fig. 2.7).

2.5 Systematic procedure

From our experience gained in implementing rule-124 sub-circuit, we pooled the different aspects of building unpurified circuits using a bottom-up approach. The testing of digital signal restoration, adjusting threshold concentrations, identifying and adjusting outliers and the final tuning of threshold could be formalized into a systematic procedure and can provide a general guideline to build unpurified seesaw circuits.

Figure 2.8 summarizes the steps discovered while constructing rule-124 sub-circuit into a flowchart. The flow chart provides a detailed guide to experimentally implement a DNA based seesaw logic circuit and includes troubleshooting procedures. We start with the construction of the simplest functional component, the digital signal restoration and estimate the effective threshold concentration with respect to the signal. If the threshold to signal ratio $\beta/\alpha > 1.2$, we adjust the nominal thresholds in all logic gates. We then use the adjusted threshold to compute AND, OR logic. If the logic is computed incorrectly, it may signify that the threshold species of this gate is an outlier, in which case case it needs to be recalibrated using the first step.

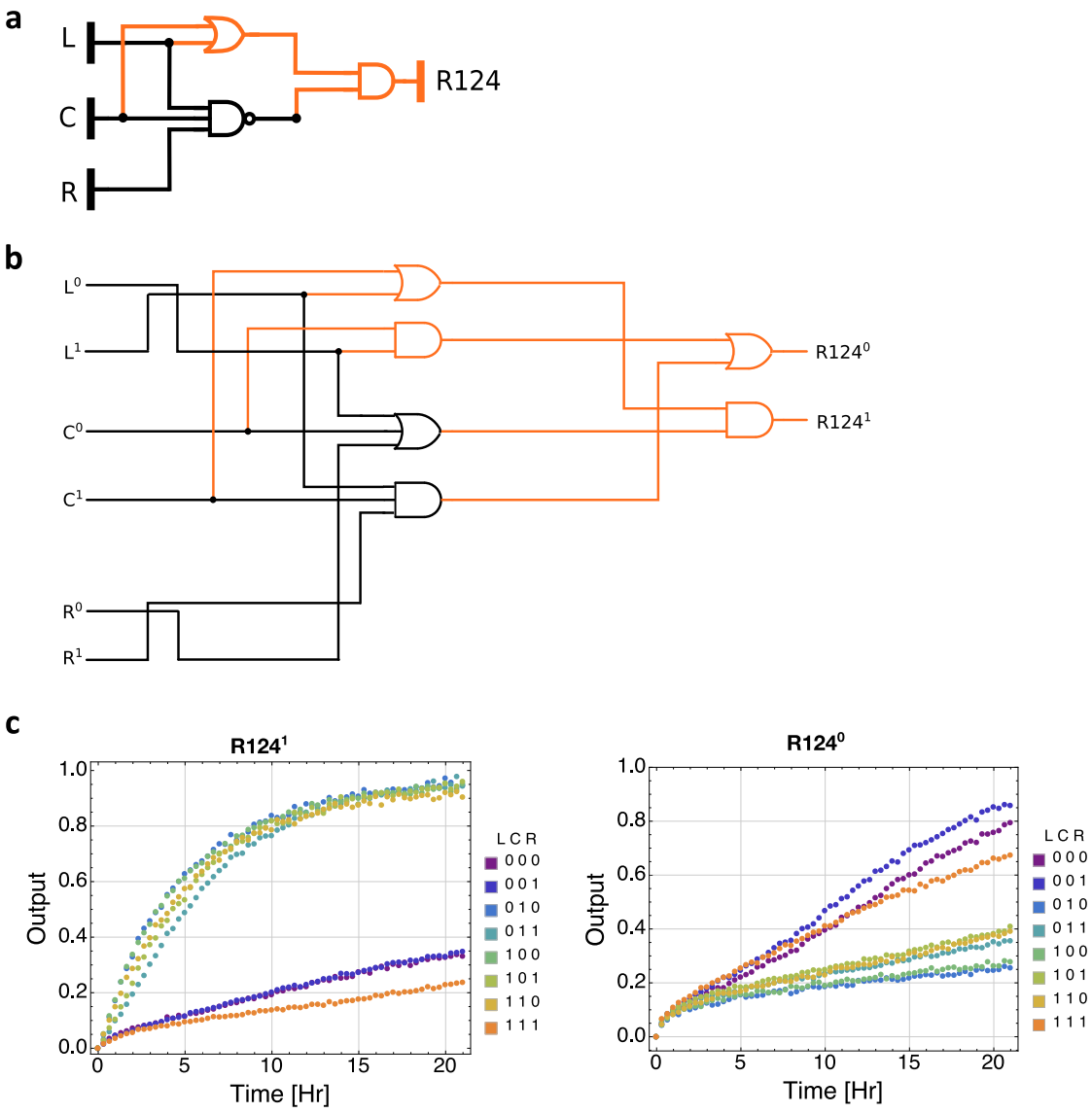

Figure 2.7: **The rule 124 sub-circuit.** **a**, Logic circuit diagram. **b**, Dual-rail circuit diagram. **c**, Experimental data. $1\times = 100$ nM.

If the logic was correctly computed, we test the gates for calibration. If the gate to signal ratio $\gamma/\alpha < 0.8$, we adjust all nominal gate concentrations.

The next step involves constructing a two-layer logic circuit. At this point, if the logic is incorrectly computed, check for outlier gates. If so, adjust the gate concentration and check the circuit computation again. It is possible that at this point the circuit may not not show desired ON/OFF separation. Making sure that ON trajectories are significantly faster than the OFF trajectories, increase the nominal concentration of the threshold in the logic gate that produces the circuit output.

Continue to construct the next layer of the circuit. If it fails, it is probably because of an outlier gate. The outlier gate needs to be identified and recalibrated. The process is continued until all layers of the circuit is built producing desired computation.

The flowchart was used to build rule-110 sub-circuit and successfully implement in 3-days. If purification of components was involved, an extra day would get added to testing each step. If multiple iterations of the flowchart need to be performed at each step, the time required for constructing such a circuit would increase multi-fold.

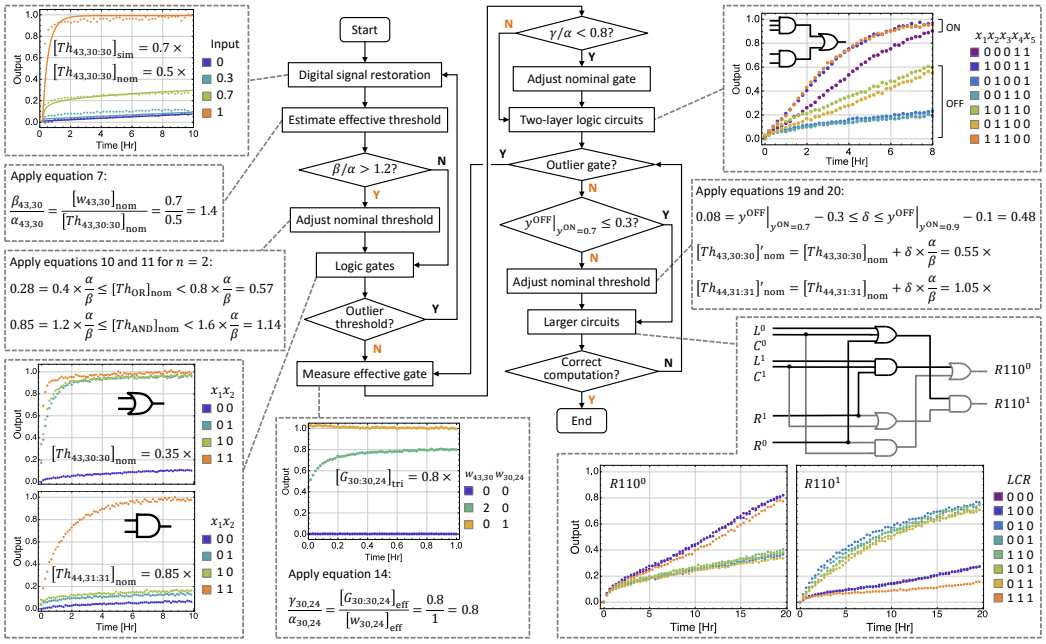

Figure 2.8: **F**lowchart for building seesaw DNA circuits using unpurified components. Insets show how the flowchart was used to construct the rule 110 sub-circuit. Y (yes) and N (no) highlighted in orange in the flowchart indicate the situations encountered and decisions made while building the rule 110 sub-circuit. 1× = 100 nM.

Finally, the components of rule-124 and rule-110 were combined using shared fan-out input gates and a 3-input NAND gate as originally designed (fig. 2.3b). All 78-distinct DNA species were combined in a single test tube and tested for all eight possible inputs. The kinetics fluorescence data showed correct ON/ OFF computation for the two pairs of dual rail output (fig. 2.9a). Data at the 24 hr time point was used to plot an array of 7x8 cells, representative of eight cellular automata generations on a torus with starting configuration (0,0,0,1,0,0,0), for a pair of dual-rail outputs of the rule 110-rule 124 circuit. Figure 2.9b pictorially represents ideal

circuit behavior which corresponds to an image of a black dog with a white background for R124^1 and an inverted image for R124^0. The ideal image for rule 110 is the mirror image of rule 124. The observed DNA circuit behavior showed less contrast with the dogs and the background than ideal behavior but still contained enough clarity to make out the pattern.

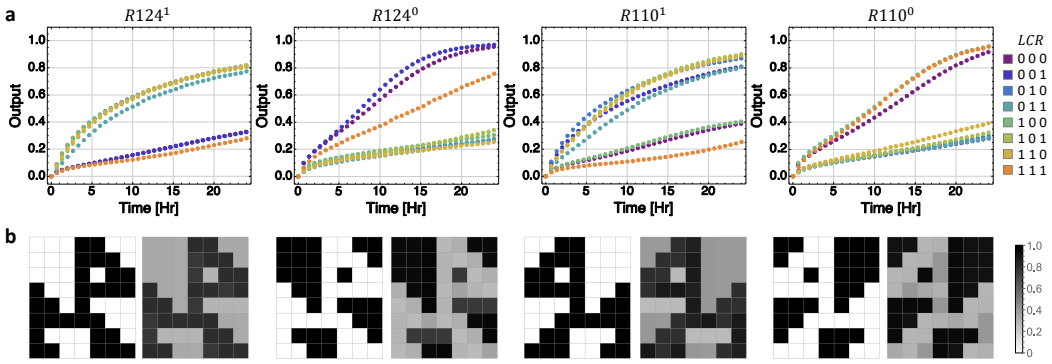

Figure 2.9: **I**mplementing the rule 110-124 full circuit. **a**, Fluorescence kinetics data of the two pairs of dual-rail outputs. 1× = 100 nM. All DNA sequences are listed in Supplementary Table 1. **b**, Comparing the ideal logic circuit behavior (left) with the DNA circuit behavior (right). Each of the circuit outputs is illustrated by an array of 7×8 cells, representative of eight cellular automata generations on a torus with starting configuration (0,0,0,1,0,0,0). The arrays for the DNA circuit were plotted using the output values at 24 hours from the data. The ideal logic circuit behavior corresponds to an image of a black dog with a white background for $R124^1$, an inverted image for $R124^0$, and their mirror images for $R110^1$ and $R110^0$, respectively.

2.6 Model

While the construction and implementation of rule110-124 was successful in producing the expected results, the experimental computations performed at a 100 nM concentration took longer than what the simulations predicted at 50 nM (fig. 2.3c). The simulations were performed assuming purified components. Hence, to better predict the behavior of unpurified components, we developed a model that incorporates synthesis errors in molecules.

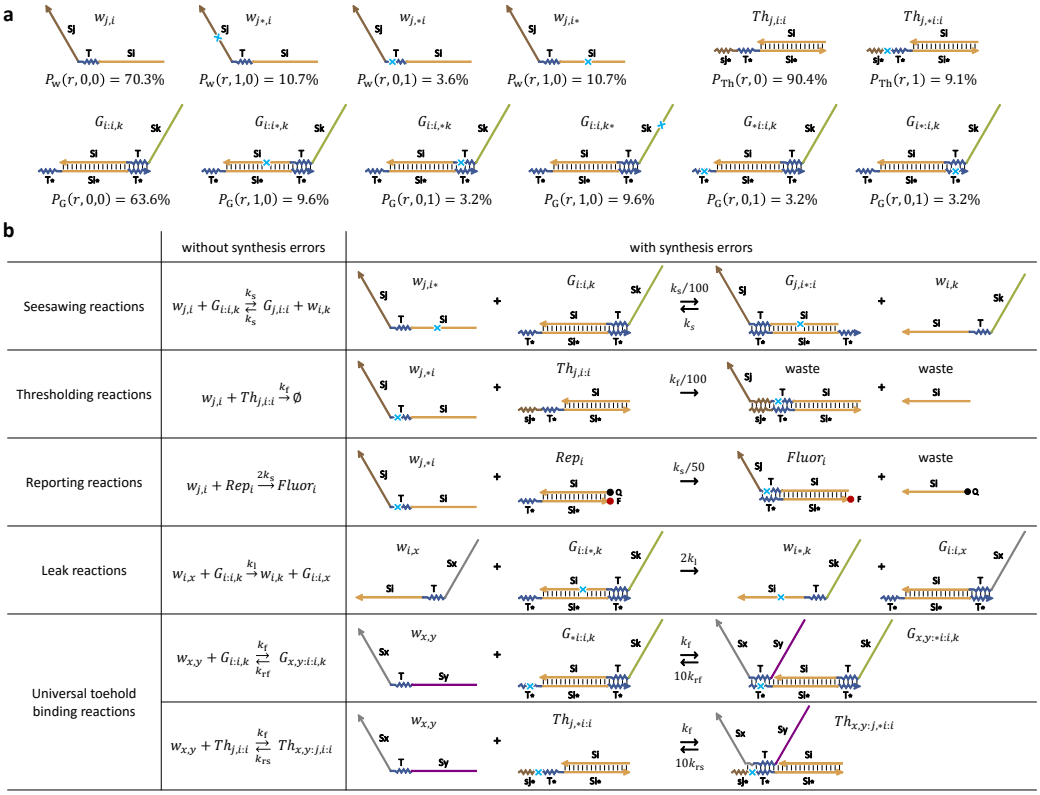

Figure 2.10: **A** model for unpurified seesaw circuits. **a**, Populations of signal, gate and threshold molecules without and with synthesis errors in the marked locations. $r = 0.01$. **b**, Example reactions that involve DNA strands without and with synthesis errors. $\forall i, j, k, x$ and y.

We first define the probability of having n errors in a chemically synthesized DNA strand of l bases, given that r is the probability of synthesis error per base:

$$P(r,l,n) = \binom{l}{n} \times (1-r)^{l-n} \times r^n \tag{2.21}$$

We then calculate the populations of signal, gate and threshold molecules with and without synthesis errors (fig. 2.10a). In order to build a simple but accurate model, we divide all the molecules involved into distinct populations. A very small population of molecules may have more than one synthesis error and are treated as "non-reactive". The remaining population with a single synthesis error is classified based on the domain where the error occurs.

For example, a signal strand is composed of two branch migration domains flanking a toehold domain. Given that a branch migration domain has 15 bases and a toehold domain has 5 bases, the probability of a signal strand having s errors in a specific branch migration domain (and thus not in the other) and t errors in the toehold

domain can be calculated as:

$$P_{\mathrm{w}}(r,s,t) = P(r,15,s) \times P(r,5,t) \times P(r,15,0) \tag{2.22}$$

Synthesis purity documents published by the vendor reports a failure rate of 1% or less with each nucleotide coupling event during DNA synthesis [66], so we chose $r = 0.01$ in all the calculations. The signal molecules with domains S_j, T and S_i can be divided into five populations,
$P_w(r,0,0) = 70.34\%$ with no synthesis errors (named $w_{j,i}$),
$P_w(r,1,0) = 10.7\%$ with an error in the Sj domain (named $w_{j*,i}$),
$P_w(r,0,1) = 3.6\%$ with an error in the T domain (named $w_{j,*i}$),
$P_w(r,1,0) = 10.7\%$ with an error in the Si domain (named $w_{j,i*}$), and
$1 - P_w(r,0,0) - 2 \times P_w(r,1,0) - P_w(r,0,1) = 4.8\%$ with two or more errors (considered non-reactive and not participating in any reactions). The star next to the variable represents the location of synthesis error. A universal toehold domain is used in all signal species in seesaw circuits and are represented by the star next to the comma in the given name ($w_{j,*i}$).

A gate molecule consists of two strands, a top strand that is the signal species and a bottom strand having two toehold domains flanking a branch migration domain (fig. 2.10a). The gate bottom strand participates in the *seesaw* reactions (fig. 2.1a) where two signal species compete with each other to bind to the gate bottom strand. Hence it is never free in solution. Any error in the branch migration domain of the bottom strand should not significantly affect the reaction rate, because it does not bias the competition in either direction, and after the initiation of strand displacement, random walk steps of adjacent base pair opening and closing should remain sufficiently fast [30]. Thus we only consider errors in the remaining two branch migration domains and three toehold domains. The probability of a gate molecule having s errors in a specific branch migration domain (and thus not the other) and t errors in a specific toehold domain (and thus not the other two) can be calculated as:

$$P_G(r,s,t) = P(r,15,s) \times P(r,5,t) \times P(r,25,0) \tag{2.23}$$

The gate molecule can hence have three toeholds and two long domains with synthesis errors. The branch migration domain in the bottom strand of the gate molecule is considered error-free as it does not affect the rate of the reaction. Therefore, there are seven populations of gate molecules considered (fig. 2.10a)

$P_G(r,0,0) = 63.6\%$, with no synthesis errors (named $G_{i:i,k}$),
$P_G(r,1,0) = 9.6\%$ with an error in the Si domain (named $G_{i:i*,k}$),
$P_G(r,1,0) = 9.6\%$ with an error in the Sk domain (named $G_{i:i,k*}$),
$P_G(r,0,1) = 3.2\%$ with an error in the T domain (named $G_{i:i,*k}$),
$P_G(r,0,1) = 3.2\%$ with an error in the T domain (named $G_{*i:i,k}$),
$P_G(r,0,1) = 3.2\%$ with an error in the T domain (named $G_{i*:i,k}$), and
$1 - P_G(r,0,0) - 3 \times P_G(r,0,1) - 2 \times P_G(r,1,0) = 7.5\%$ (inert molecules)

The threshold molecule consists of an extended toehold domain of 10 bases and a branch migration domain of 15 bases, complimentary to a top strand with 15 bases (fig. 2.10a). Thresholding is an irreversible reaction where the threshold molecule consumes a signal molecule producing waste (fig. 2.1b). An error in the top strand or the branch migration domain in the bottom strand will not affect the rate of the reaction. This is because an error in the top strand will only bias the reaction forward which is already favored. An error in the bottom strand does not introduce additional bias to the random walk steps of branch migration. Hence only the extended toehold is considered. The probability of the threshold molecule having t errors in the toehold domain is calculated as

$$P_{\mathrm{Th}}(r,t) = P(r,10,t) \tag{2.24}$$

The threshold species can be divided into three populations:

$P_{Th}(r,0) = 90.4\%$, with no synthesis errors (named $Th_{j:i,i}$),
$P_{Th}(r,1) = 9.1\%$ with an error in the T domain (named $Th_{j,*i:i}$), and
$1 - P_{Th}(r,0) - P_{Th}(r,1) = 0.4\%$ (inert molecules)

The importance of synthesis errors (mismatch, deletion, insertion) in reaction rate was informed by a previous study on the robustness of a catalytic DNA strand displacement motif [60]. The authors found that a single base mutation in an invading strand significantly slowed down (on the scale of 100 fold) a reversible strand displacement reaction that was designed with $\Delta G^\circ \approx 0$, both when the mutation was in the toehold and when it was in the branch migration domain. In contrast, an irreversible strand displacement reaction was only slowed down significantly (also on the scale of 100 fold) when the mutation was in the toehold domain, but the reaction rate remained roughly unchanged when the mutation was in the branch migration domain. Since the threshold reaction is an irreversible reaction, an error in the toehold domain plays a significant role, and the error in the migration domain need not

be considered.

Based on these observations, we concluded that a synthesis error in the toehold domain can slow down strand displacement by increasing the disassociation rate of the toehold and thus decreasing the overall reaction rate. A synthesis error in the branch migration domain can also slow down strand displacement, but only when the energy change caused by the synthesis error is significant compared to the designed standard free energy of the reaction, and not when the reaction is already strongly favored in one direction. Based on these interpretations, we estimated the rates of all five types of reactions in a seesaw network, involving all populations of defective molecules as calculated above (fig. 2.10b and Appendix Note section 2.C).

We first simulated rule 110-124 circuit assuming purified components using the compiler. We observed that with purified components, the computation was faster and completed by 8 hrs. However, the rate of completion did not agree with our experimental results. Next, we simulated the same circuit by incorporating the fraction of molecules with synthesis errors calculated using eqs. (2.21) to (2.24). Simulations done with incorporating synthesis errors showed remarkable agreement with our experimental data.

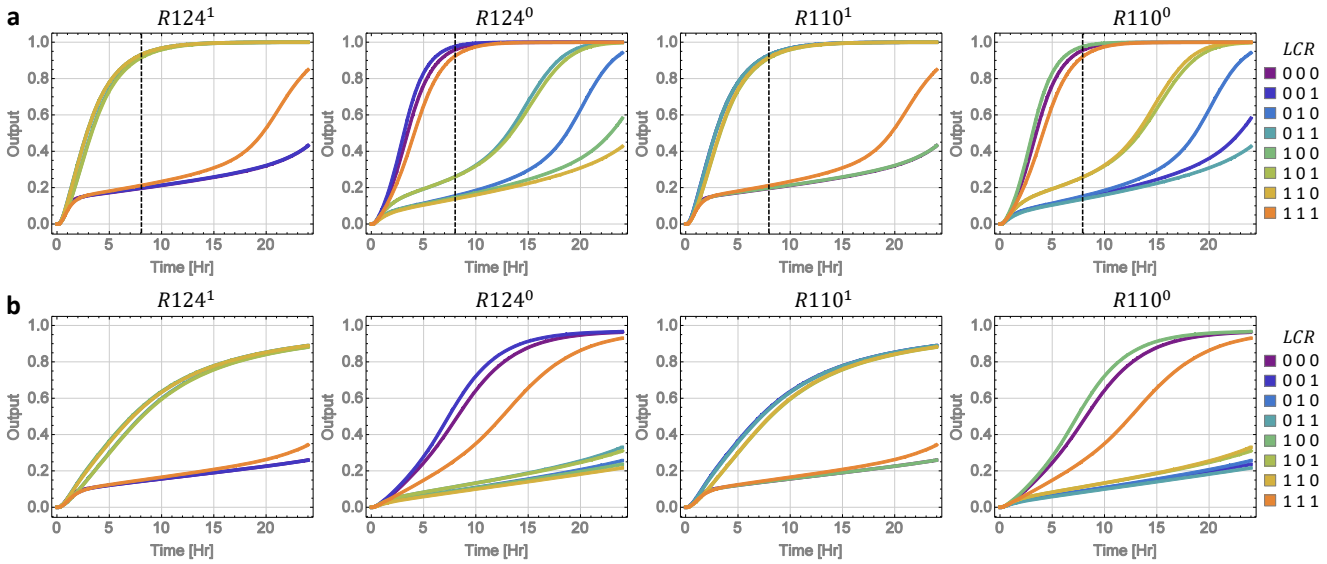

Figure 2.11: **S**imulations comparing the purified and unpurified models. **a**, Simulations of the rule 110-124 circuit using the previously developed model for purified seesaw circuits, predicting that the circuit should yield desired outputs in roughly 8 hours (shown as dotted lines) and the undesired reactions will take over in 24 hours. **b**, Simulations using the new model for unpurified seesaw circuits, predicting that the circuit should yield desired outputs in roughly 24 hours. $k_{\mathrm{f}} = 2 \times 10^6\ \mathrm{M^{-1}s^{-1}}$, $k_{\mathrm{s}} = 5 \times 10^4\ \mathrm{M^{-1}s^{-1}}$, $k_{\mathrm{l}} = 10\ \mathrm{M^{-1}s^{-1}}$, $k_{\mathrm{rf}} = 26\ \mathrm{s^{-1}}$, $k_{\mathrm{rs}} = 1.3\ \mathrm{s^{-1}}$. $1\times = 100$ nM.

Conclusions

This work initially began as a project in a teaching class at Caltech. The goal was to train budding scientists in experimental techniques of molecular programming. Building new molecular components to build new circuits is a daunting challenge. Seesaw circuit components provided an excellent opportunity to build new circuits with existing, well-characterized and simple parts. Any arbitrary feedforward circuit could be designed with the help of the Seesaw compiler using only three types of molecular components (signal, threshold and gate). The compiler uses a three letter code (A, T and C) to eliminate undesired reactions. However, introducing purification techniques involved in implementing DNA circuits was not time-effective for the class. We realized that this could also be a hurdle for other researchers for testing new circuits. Thus, we took our first steps towards building DNA circuits using unpurified components.

The biggest challenge one faces with unpurified DNA strands is synthesis errors which can cause unpredictable behavior. The simulations provided by the compiler can be used to estimate behavior in the designed circuit. Given that there are only

three types of circuit components, it becomes easier to understand the trend in behavior of these molecules as the synthesis quality mostly remains the same across molecules of same type. Finally, digital signal restoration at each step can allow simple tuning to obtain predicted circuit behavior. In order to build better molecular compilers and implementing unpurified circuits with more robust behaviors, one could also implement a system which eliminates leak reactions in DNA circuits [67], and to improve the building blocks such that they are substantially less sensitive to synthesis errors and stoichiometry errors.

Developing a systematic procedure to implement DNA circuits with unpurified components, we believe we have not only experimentally validated the Seesaw compiler, but we have also found ways to simplify experimental procedures involved, adding a step towards automation by using the compiler for high-level circuit design to be possibly implemented by liquid-handling robots.

2.7 Methods

DNA oligonucleotide synthesis.

DNA oligonucleotides were purchased from Integrated DNA Technologies (IDT). The DNA strands in gate, threshold and fuel species were purchased unpurified (standard desalting). The reporter strands with fluorophores and quenchers were purchased purified (HPLC). All strands were purchased at 100 μM in TE buffer, pH 8.0, and stored at 4 °C.

Annealing protocol and buffer condition.

Gate complexes were annealed together at 20 μM, with equal stoichiometry of top and bottom strands. Threshold and reporter complexes were annealed together at 20 μM with a 20% excess of top strands. All DNA complexes were annealed in 1× TE buffer with 12.5 mM Mg^{2+}, prepared from 100× TE, pH 8.0 (Fisher BioReagents) and 1 M $MgCl_2$ (Invitrogen). Because reporter top strands have no toehold domains and are modified with quenchers, this excess will ensure the formation of complexes even with somewhat imperfect stoichiometry, without leaving active single-stranded DNA or changing the fluorescence baseline. The top strands of the threshold complex have no toehold domains, are produced as waste and do not interfere in downstream reactions, excess of the top stand will ensure formation of threshold complexes without leaving active single stranded DNA. Annealing was performed in a thermal cycler (Eppendorf), first heating up to 90 °C for 2 minutes, and then slowly cooling down to 20 °C at the rate of 6 sec per 0.1 °C. All annealed

complexes were stored at 4 °C.

Fluorescence spectroscopy.

Fluorescence kinetics data in fig. 2.4 to fig. 2.8 and Supplementary fig. 2.14 to fig. 2.7 were collected every 2 minutes in a monochromator-based plate reader (Synergy H1M, BioTek). Experiments were performed with 100 μL reaction mixture per well, in 96-well microplates (black with clear flat bottom, polystyrene NBS, Corning #3651) at 25 °C. Clear adhesive sealing tapes (Thermo Scientific Nunc #232701) were used to prevent evaporation. The excitation/emission wavelengths were set to 497/527 nm for ATTO 488 and 597/629 nm for ATTO 590.

Fluorescence kinetics data in fig. 2.9 were collected every 4 minutes in a spectrofluorimeter (Fluorolog-3, Horiba). The spectrofluorimeter has a 4-cuvette holder for running parallel experiments. Experiments were performed with 500 μL reaction mixture per cuvette, in fluorescence cuvettes (Hellma #115F-QS) at 25 °C. The excitation/emission wavelengths were set to 502/522 nm for ATTO 488, 602/624 nm for ATTO 590, 560/575 nm for ATTO 550 and 649/662 nm for ATTO 647. All 4 fluorophores were tested and found to be spectrally well separated. They did not interfere with each other when used in the same circuit. Both excitation and emission bandwidths were set to 2 nm, and the integration time was 10 seconds for all experiments.

Data analysis.

A Mathematica Notebook file for data analysis and example data files are available to download at the Seesaw Compiler website: http://qianlab.caltech.edu/SeesawCompiler/DataAnalysis.php.

APPENDIX

2.A Additional design diagrams

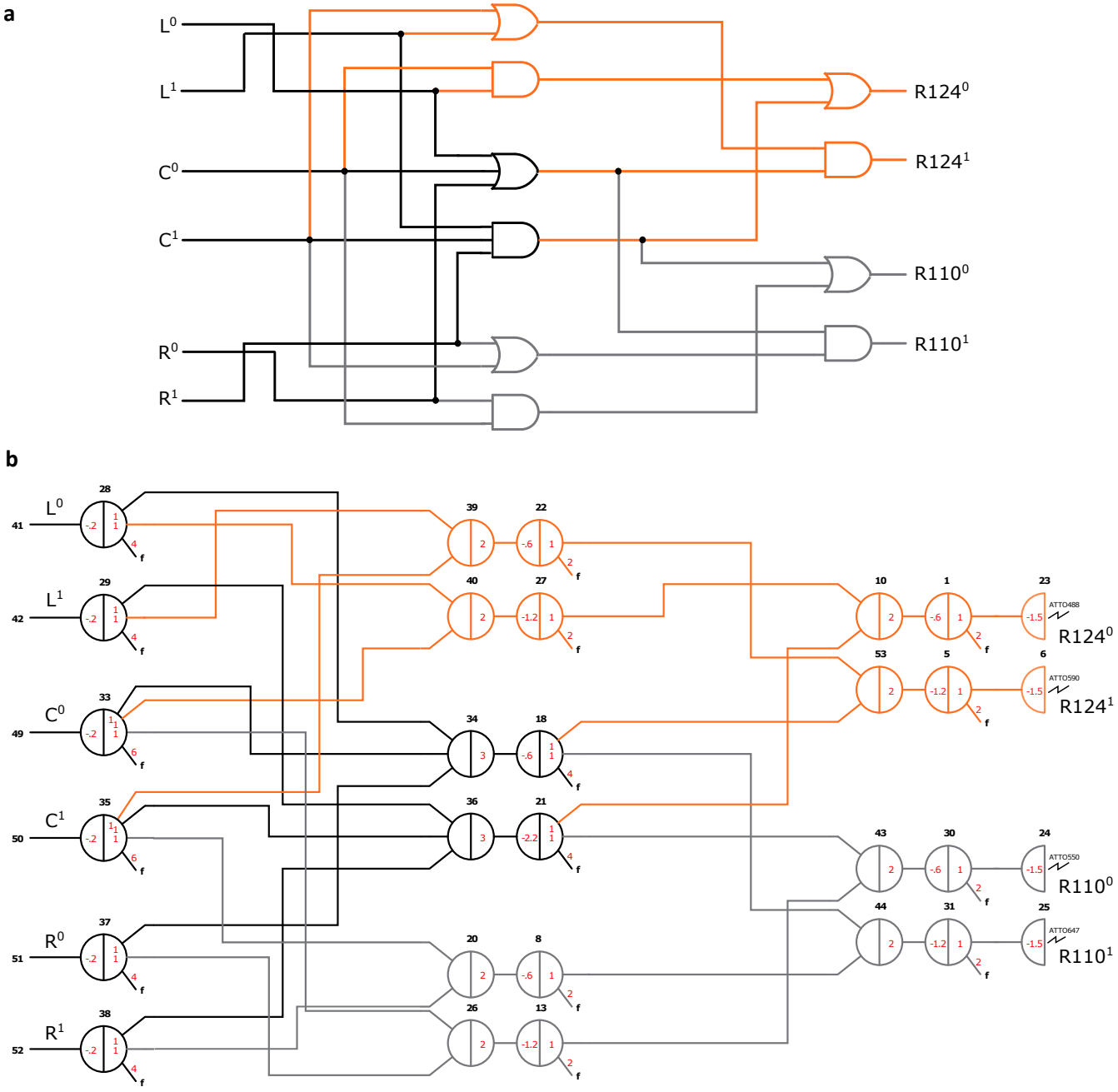

Figure 2.12: **Diagrams of the rule 110-124 circuit.** **a**, Dual-rail circuit diagram. **b**, Seesaw circuit diagram.

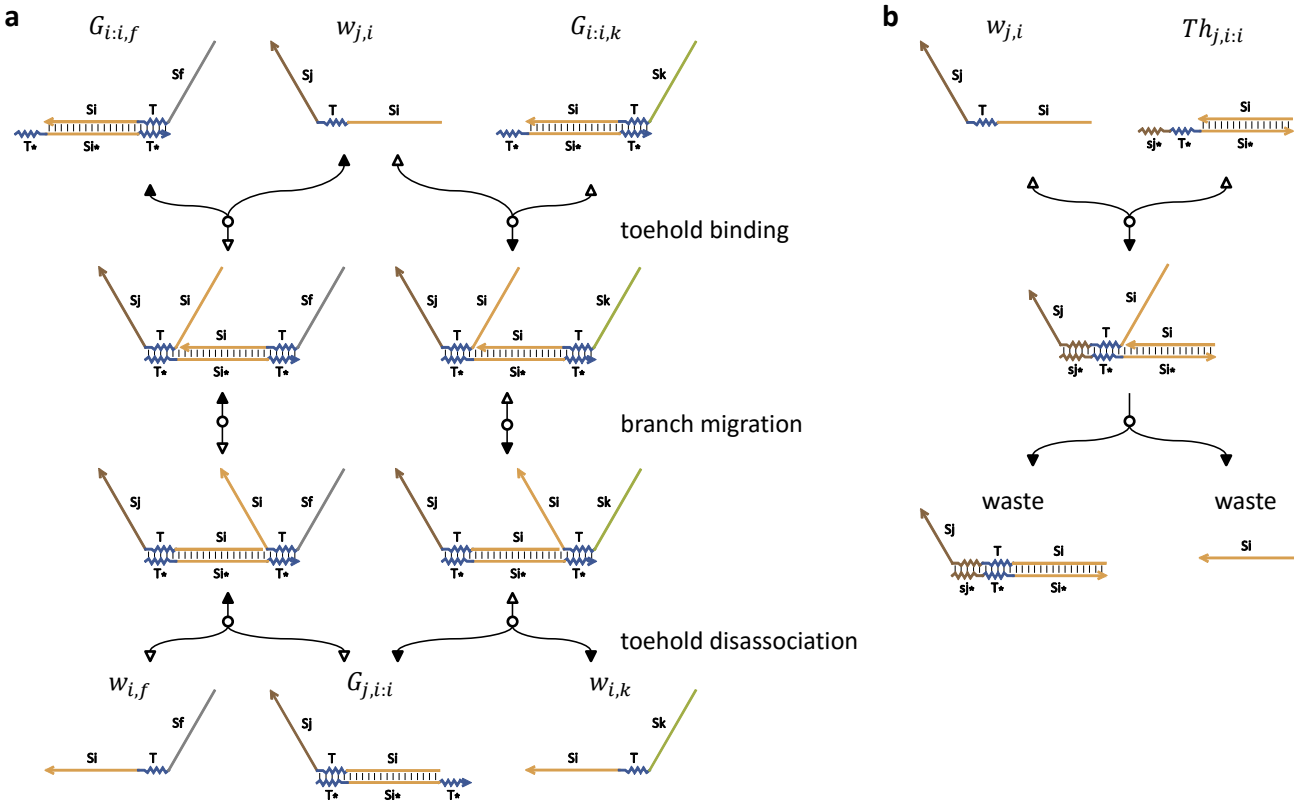

Figure 2.13: **Basic DNA strand displacement reactions in a seesaw network** (adapted from ref. [25]). **a**, Catalysis. **b**, Thresholding. Solid arrows indicate flows of the forward reactions and outlined arrows indicate flows of the respective backward reactions. Catalysis is driven forward by a high concentration of the fuel species $w_{i,f}$ and downstream irreversible reactions (i.e. thresholding or reporting reactions) that consume the output species $w_{i,k}$. Matching colors and stars in domain names suggest complementary DNA sequences. For example, the blue domains T and $T*$ are complementary to each other, the orange domains Si and $Si*$ are complementary to each other, etc. $sj*$ is complementary to the first 5 nucleotides of the Sj domain. Thresholding is much faster than catalysis because the $sj*$ domain serves as an extended toehold, which significantly decreases the rate of toehold disassociation and thus speeds up the overall rate of stand displacement [29, 30].

2.B Additional experimental data

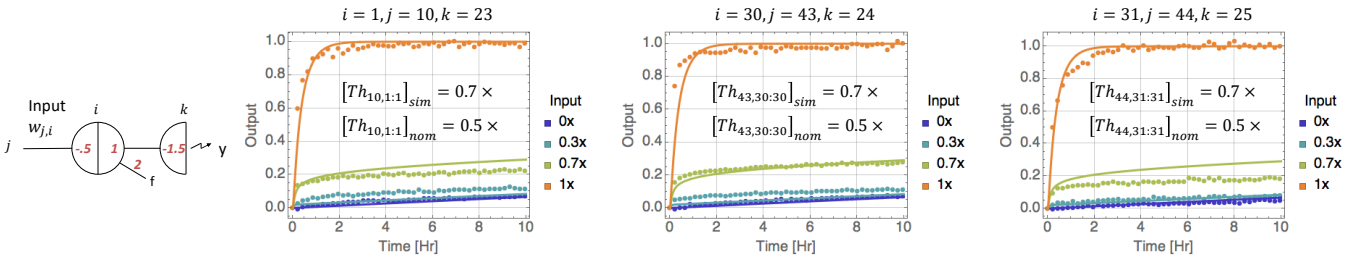

Figure 2.14: **Estimating effective concentrations of distinct thresholds.** The small differences between simulations and data for $Th_{10,1:1}$ and $Th_{44,31:31}$ are considered non-significant. We show that $\beta/\alpha = 1.4$ works well enough for four distinct thresholds, including three shown here and one shown in fig. 2.4c. $1\times = 100$ nM.

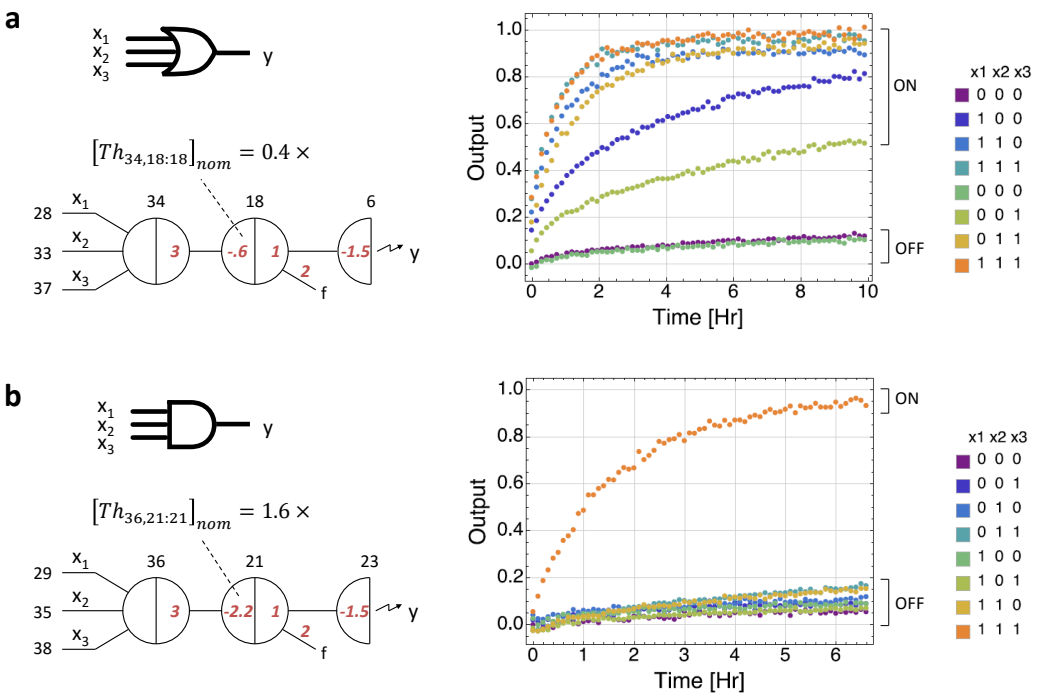

Figure 2.15: **Three-input logic gates with adjusted nominal thresholds.** **a**, OR gate. Not all possible inputs were tested here ($x_1x_2x_3$ = 000 and 111 were repeated twice), but we believe that the circuit behavior for $x_1x_2x_3$ = 010 and 101 should be similar to that for 100/001 and 110/011, respectively. **b**, AND gate. 1× = 100 nM.

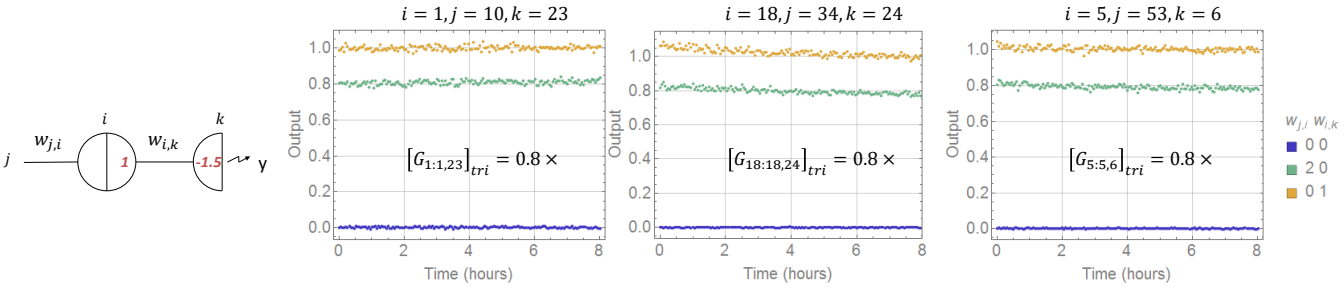

Figure 2.16: **Estimating effective concentrations of distinct gates.** Data show steady state fluorescence level, as signal strands and gate molecules were mixed together and incubated before the measurements. We show that $\gamma/\alpha = 0.8$ works well for four distinct gates, including three shown here and one shown in fig. 2.5b. 1× = 100 nM.

2.C Additional modeling details

Reactions that involve molecules with synthesis errors

Seesaw circuits can be modeled with five types of reactions [25] (fig. 2.10b): seesawing reactions that reversibly exchange two signals between inactive (i.e. bound to a gate) and active (i.e. free-floating) states, thresholding reactions that irreversibly consume a signal, reporting reactions that generate fluorescence readout, leak reactions that slowly release a signal from a gate molecule, and universal toehold binding reactions that temporarily occur between any single strand and any gate or threshold molecules. Compared to the reactions that only involve molecules without synthesis errors, there is a much longer list of reactions that involve molecules with synthesis errors, because each distinct species is now divided into multiple populations. To concisely describe these reactions, we define that reactions in the following format

$$\{R_{11}, R_{12}, \cdots, R_{1n}\} + \{R_{21}, R_{22}, \cdots, R_{2m}\} \xrightarrow{k} \{P_{11}, P_{12}, \cdots, P_{1n}\} + \{P_{21}, P_{22}, \cdots, P_{2m}\} \tag{2.25}$$

can be interpreted as the set of reactions:

$$R_{1i} + R_{2j} \xrightarrow{k} P_{1i} + P_{2j}, \forall 1 \leq i \leq n, 1 \leq j \leq m \tag{2.26}$$

This means for reactions with two reactants and two products, we always group them together in a way that the first product can be determined just based on the first reactant, and the second product can be determined just based on the second reactant. Note that a reversible reaction can be seen as two irreversible reactions that each follow the same definition.

Based on the previous results [60], we estimated the rates of seesawing and thresholding reactions that involve all populations of signal, gate and threshold molecules shown in fig. 2.10a.

(1) Seesawing reactions:

If there is no error in the domains participating in a seesawing reaction, regardless of any errors in the other domains (e.g. Sj and Sk domains for $w_{j,i}$ interacts with $G_{i:i,k}$), the rate remains the same as in the previous model for purified seesaw circuits.

$$\{w_{j,i}, w_{j*,i}\} + \{G_{i:i,k}, G_{i:i,k*}\} \underset{k_s}{\overset{k_s}{\rightleftharpoons}} \{G_{j,i:i}, G_{j*,i:i}\} + \{w_{i,k}, w_{i,k*}\} \tag{2.27}$$

If there is an error in the participating toehold or branch migration domain of the invading signal strand, or in the initiating toehold domain of the gate molecule (i.e. the toehold that binds to the invading signal strand), the forward rate is 100 times slower and the backward rate remains the same.

$$\begin{aligned} \{w_{j,*i}, w_{j,i*}\} + \{G_{i:i,k}, G_{i:i,k*}\} &\underset{k_s}{\overset{k_s/100}{\rightleftharpoons}} \{G_{j,*i:i}, G_{j,i*:i}\} + \{w_{i,k}, w_{i,k*}\} \\ \{w_{j,i}, w_{j*,i}\} + \{G_{*i:i,k}, G_{*i:i,k*}\} &\underset{k_s}{\overset{k_s/100}{\rightleftharpoons}} \{G_{j,i:*i}, G_{j*,i:*i}\} + \{w_{i,k}, w_{i,k*}\} \end{aligned} \tag{2.28}$$

Symmetrically, if there is an error in the participating toehold or branch migration domain of the bound signal in the gate molecule, or in the disassociation toehold domain (i.e. the toehold that is originally covered), the backward rate is 100 times slower and the forward rate remains the same.

$$\begin{aligned} \{w_{j,i}, w_{j*,i}\} + \{G_{i:i*,k}, G_{i:i,*k}\} &\underset{k_s/100}{\overset{k_s}{\rightleftharpoons}} \{G_{j,i:i}, G_{j*,i:i}\} + \{w_{i*,k}, w_{i,*k}\} \\ \{w_{j,i}, w_{j*,i}\} + \{G_{i*:i,k}, G_{i*:i,k*}\} &\underset{k_s/100}{\overset{k_s}{\rightleftharpoons}} \{G_{j,i:i*}, G_{j*,i:i*}\} + \{w_{i,k}, w_{i,k*}\} \end{aligned} \tag{2.29}$$

Note that gate molecules with two synthesis errors (e.g. $G_{*i:i,k*}$ and $G_{i*:i,k*}$) are not in the initial populations but can be produced by a seesawing reaction between signal and gate molecules that each have just one synthesis error. Reactions are omitted if there exist more than one synthesis error that can significantly affect the rate, because they are either too slow or do not have enough reactants to take place. For example,

$$w_{j,i*} + G_{*i:i,k} \underset{k_s}{\overset{\ll k_s/100}{\rightleftharpoons}} G_{j,i*:*i} + w_{i,k}$$

and

$$w_{j,i*} + G_{i:i*,k} \underset{k_s/100}{\overset{k_s/100}{\rightleftharpoons}} G_{j,i*:i} + w_{i*,k}$$

(2) Thresholding reactions:

Unlike a seesawing reaction, if there is no error in the toehold domains participating in a thresholding reaction, regardless of any errors in the branch migration domains, the rate remains the same as in the previous model for purified seesaw circuits.

$$\{w_{j,i}, w_{j*,i}, w_{j,i*}\} + Th_{j,i:i} \xrightarrow{k_f} \emptyset \tag{2.30}$$

Otherwise the rate is 100 times slower.

$$\begin{aligned} w_{j,*i} + Th_{j,i:i} &\xrightarrow{k_f/100} \emptyset \\ \{w_{j,i}, w_{j*,i}, w_{j,i*}\} + Th_{j,*i:i} &\xrightarrow{k_f/100} \emptyset \end{aligned} \tag{2.31}$$

Reactions are again omitted if there exist more than one synthesis error that can significantly affect the rate. For example,

$$w_{j,*i} + Th_{j,*i:i} \xrightarrow{\ll k_f/100} \emptyset$$

An error in the extended toehold but not in the universal toehold domain of a signal strand is not considered to affect the rate of thresholding, because an error more distant from the branch migration domain should affect the rate less, and considering it would complicate the classification of signal molecules.

(3) Reporting reactions:

Reporting reactions are also irreversible, and thus are modeled similarly as the thresholding reactions, based on if there is an error in the toehold domains.

$$\begin{aligned} \{w_{j,i}, w_{j*,i}, w_{j,i*}\} + Rep_i &\xrightarrow{2k_s} Fluor_i \\ w_{j,*i} + Rep_i &\xrightarrow{k_s/50} Fluor_i \end{aligned} \tag{2.32}$$

(4) Leak reactions:

Leak reactions are essentially 0-toehold strand displacement reactions. If there is no error in the participating domains of the two competing signal strands, the rate remains the same as in the previous model for purified seesaw circuits. An error in the gate bottom strand should not affect the rate significantly, regardless of if it is in the toehold domain, because the toehold is covered and thus treated the same as the branch migration domain.

$$\begin{aligned} \{w_{i,x}, w_{i,x*}\} + \{G_{i:i,k}, G_{i:i,k*}\} &\xrightarrow{k_l} \{G_{i:i,x}, G_{i:i,x*}\} + \{w_{i,k}, w_{i,k*}\} \\ \{w_{i,x}, w_{i,x*}\} + \{G_{*i:i,k}, G_{*i:i,k*}\} &\xrightarrow{k_l} \{G_{*i:i,x}, G_{*i:i,x*}\} + \{w_{i,k}, w_{i,k*}\} \\ \{w_{i,x}, w_{i,x*}\} + \{G_{i*:i,k}, G_{i*:i,k*}\} &\xrightarrow{k_l} \{G_{i*:i,x}, G_{i*:i,x*}\} + \{w_{i,k}, w_{i,k*}\} \end{aligned} \tag{2.33}$$

Leak reactions should be faster, if there is an error in the participating toehold or branch migration domain of the bound signal strand in the gate molecule, because the forward reaction will be favored. The reaction would be roughly 10 times faster if the error occurs at either end of the double-stranded domain and opens up a 1-nucleotide toehold for the invading signal strand. However, the error should occur in the middle of the double-stranded domain, with a much higher probability, and thus serve as a much less effective 1-nucleotide toehold. Therefore, we estimate the

rate to be only twice as fast.

$$\begin{aligned}
\{w_{i,x}, w_{i,x*}\} + \{G_{i:i*,k}, G_{i:i,*k}\} &\xrightarrow{2k_l} \{G_{i:i,x}, G_{i:i,x*}\} + \{w_{i*,k}, w_{i,*k}\} \\
\{w_{i,x}, w_{i,x*}\} + \{G_{*i:i*,k}, G_{*i:i,*k}\} &\xrightarrow{2k_l} \{G_{*i:i,x}, G_{*i:i,x*}\} + \{w_{i*,k}, w_{i,*k}\} \\
\{w_{i,x}, w_{i,x*}\} + \{G_{i*:i*,k}, G_{i*:i,*k}\} &\xrightarrow{2k_l} \{G_{i*:i,x}, G_{i*:i,x*}\} + \{w_{i*,k}, w_{i,*k}\}
\end{aligned} \tag{2.34}$$

Since leak reactions are already very slow, reactions are omitted if there is an error that slows down the rate even further. For example,

$$w_{i*,x} + G_{i:i,k} \xrightarrow{\ll k_l} G_{i:i*,x} + w_{i,k}$$

(5) Universal toehold binding reactions:

Finally, the forward rate remains the same for all universal toehold binding reactions, since it is just the rate of hybridization. The backward rate remains the same if there is no error in the toehold domains, and is 10 times faster if there is an error, simply because the rate of toehold disassociation can be estimated as 10^{6-l} /s, where l is the number of bases in the toehold [29, 30].

$$\begin{aligned}
W + \{G_{i:i,k}, G_{i*:i,k}, G_{i:i*,k}, G_{i:i,*k}, G_{i:i,k*}, G_{i*:i,k*}\} &\underset{k_{rf}}{\overset{k_f}{\rightleftharpoons}} \{G_{W:i:i,k}, G_{W:i*:i,k}, G_{W:i:i*,k}, G_{W:i:i,*k}, \\
G_{W:i:i,k*}, G_{W:i*:i,k*}\} + W + \{G_{*i:i,k}, G_{*i:i,k*}\} &\underset{10k_{rf}}{\overset{k_f}{\rightleftharpoons}} \{G_{W:*i:i,k}, G_{W:*i:i,k*}\} \\
\{G_{j,i:i}, G_{j*,i:i}, G_{j,*i:i}, G_{j,i*:i}, G_{j,i:*i}, G_{j*,i:*i}\} + W &\underset{k_{rf}}{\overset{k_f}{\rightleftharpoons}} \{G_{j,i:i:W}, G_{j*,i:i:W}, G_{j,*i:i:W}, \\
G_{j,i*:i:W}, G_{j,i:*i:W}, G_{j*,i:*i:W}\}\{G_{j,i:i*}, G_{j*,i:i*}\} + W &\underset{10k_{rf}}{\overset{k_f}{\rightleftharpoons}} \{G_{j,i:i*:W}, G_{j*,i:i*:W}\} \\
W + Th_{j,i:i} &\underset{k_{rs}}{\overset{k_f}{\rightleftharpoons}} Th_{W:j,i:i} \\
W + Th_{j,*i:i} &\underset{10k_{rs}}{\overset{k_f}{\rightleftharpoons}} Th_{W:j,*i:i} \\
\{w_{j,i}, w_{j*,i}, w_{j,i*}\} + G &\underset{k_{rf}}{\overset{k_f}{\rightleftharpoons}} \{G_{j,i:G}, G_{j*,i:G}, G_{j,i*:G}\} \\
w_{j,*i} + G &\underset{10k_{rf}}{\overset{k_f}{\rightleftharpoons}} G_{j,*i:G} \\
\{w_{j,i}, w_{j*,i}, w_{j,i*}\} + TH &\underset{k_{rs}}{\overset{k_f}{\rightleftharpoons}} \{TH_{j,i:TH}, TH_{j*,i:TH}, TH_{j,i*:TH}\} \\
w_{j,*i} + TH &\underset{10k_{rs}}{\overset{k_f}{\rightleftharpoons}} TH_{j,*i:TH}
\end{aligned} \tag{2.35}$$

$$
\begin{aligned}
[W]|_{t=0} &= \sum [w_{j,i}]|_{t=0}, \forall i, j \\
[G]|_{t=0} &= \sum [G_{j,i:i}]|_{t=0} + [G_{i:i,k}]|_{t=0} + [Rep_i]|_{t=0}, \forall i, j, k \\
[TH]|_{t=0} &= \sum [Th_{j,i:i}]|_{t=0}, \forall i, j
\end{aligned} \tag{2.36}
$$

Approximation in domain lengths

The signal strand as designed in the original seesaw system by Qian et. al [25] composed of two branch migration domains flanking a toehold domain where the branch migration domain had 15 bases, a toehold domain had 5 bases. Additionally, in order to reduce undesired leak reactions between two gate species, a clamp domain consisting of two bases were included on either side of the branch migration domain or part of a toehold domain. This depended on the side on which the gate and the signal strand interacted (supplementary notes S8 of [25]). Thus the signal strand is really only 33 bases long with base duplicates. Including the clamp domains in the model would make the classification of molecules and domains significantly complicated while introducing a very small impact on the calculations performed. Hence we choose not to include the clamp domains and approximate the signal strand to have 35 bases.

Concentrations of threshold species

Our system implemented with unpurified components saw a significant decrees in reaction kinetics compared to purified circuits. The model including systhesis errors helped expalin the slow down. However, it is still not very clear as to why the effective concentration of threshold is higher than the effective concentration of the signal molecule. The calculated threshold to signal ratio $\beta/\alpha = 1.4$ eq. (2.7) was applied across the board for all threshold concentrations in the simulation. Qian et. al [25] also observed a higher threshold concentrations in their purified system. A factor $\beta/\alpha = 1.1$ was applied to all threshold concentrations in the previously-developed model for purified seesaw circuits. In our work, we experimentally measured $\beta/\alpha = 1.4$, the $1.1\times$ nominal threshold was tuned in order to get a better agreement between the model and data for large purified circuits.

The difference in effective concentrations of the threshold versus the signal strand could be cause by aspects of DNA synthesis that are not yet understood. While the difference was definitely better in the case of purified circuits, it could not be eliminated by in-house gel purification. Because this difference is brought about in the synthesis step, it is imperative that the ratio β/α is determined by the users of Seesaw Compiler as discussed in section 2.2.

2.D DNA sequences

DNA sequences for components of Rule110 - 124

Table S1: DNA sequences

Name	Domain	Sequence
w41.28 (L0)	S28 T S41	CATCTACAATTCACA TCT CAACAAACCATTACA
w42.29 (L1)	S29 T S42	CACCAATACTCCTCA TCT CACTTTTCACTATCA
w49.33 (C0)	S33 T S49	CAACTCAAACATACA TCT CATCCTTAACTCCCA
w50.35 (C1)	S35 T S50	CACTCTCCATCACCA TCT CATTACCAACCACCA
w51.37 (R0)	S37 T S51	CACCTCTTCCCTTCA TCT CACAAACTACATCCA
w52.38 (R1)	S38 T S52	CATACCCTTTTCTCA TCT CACTTCACAACTACA
Th41.28:28-t	S28	CATCTACAATTCACA
Th41.28:28-b	s41* T* S28*	TTTGTTG AGA TGTGAATTGTAGATG
w28.34	S34 T S28	CACATAACAAAACCA TCT CATCTACAATTCACA
w28.40	S40 T S28	CAATACAAATCCACA TCT CATCTACAATTCACA
G28-b	T* S28* T*	TG AGA TGTGAATTGTAGATG AGA TG
w28.f	Sf T S28	CATTTTTTTTTTTCA TCT CATCTACAATTCACA
Th42.29:29-t	S29	CACCAATACTCCTCA
Th42.29:29-b	s42* T* S29*	AAAAGTG AGA TGAGGAGTATTGGTG
w29.36	S36 T S29	CAAACTAAACAACCA TCT CACCAATACTCCTCA
w29.39	S39 T S29	CACTATACACACCCA TCT CACCAATACTCCTCA
G29-b	T* S29* T*	TG AGA TGAGGAGTATTGGTG AGA TG
w29.f	Sf T S29	CATTTTTTTTTTTCA TCT CACCAATACTCCTCA
Th49.33:33-t	S33	CAACTCAAACATACA
Th49.33:33-b	s49* T* S33*	AAGGATG AGA TGTATGTTTGAGTTG
w33.34	S34 T S33	CACATAACAAAACCA TCT CAACTCAAACATACA
w33.40	S40 T S33	CAATACAAATCCACA TCT CAACTCAAACATACA
w33.26	S26 T S33	CATTCATTACCTCCA TCT CAACTCAAACATACA
G33-b	T* S33* T*	TG AGA TGTATGTTTGAGTTG AGA TG
w33.f	Sf T S33	CATTTTTTTTTTTCA TCT CAACTCAAACATACA
Th50.35:35-t	S35	CACTCTCCATCACCA
Th50.35:35-b	s50* T* S35*	GGTAATG AGA TGGTGATGGAGAGTG
w35.36	S36 T S35	CAAACTAAACAACCA TCT CACTCTCCATCACCA
w35.39	S39 T S35	CACTATACACACCCA TCT CACTCTCCATCACCA
w35.20	S20 T S35	CAATCTAACACTCCA TCT CACTCTCCATCACCA
G35-b	T* S35* T*	TG AGA TGGTGATGGAGAGTG AGA TG

Name	Domain	Sequence
w35.f	Sf T S35	CATTTTTTTTTTTCA TCT CACTCTCCATCACCA
Th51.37:37-t	S37	CACCTCTTCCCTTCA
Th51.37:37-b	s51* T* S37*	GTTTGTG AGA TGAAGGGAAGAGGTG
w37.34	S34 T S37	CACATAACAAAACCA TCT CACCTCTTCCCTTCA
w37.26	S26 T S37	CATTCATTACCTCCA TCT CACCTCTTCCCTTCA
G37-b	T* S37* T*	TG AGA TGAAGGGAAGAGGTG AGA TG
w37.f	Sf T S37	CATTTTTTTTTTTCA TCT CACCTCTTCCCTTCA
Th52.38:38-t	S38	CATACCCTTTTCTCA
Th52.38:38-b	s52* T* S38*	TGAAGTG AGA TGAGAAAAGGGTATG
w38.36	S36 T S38	CAAACTAAACAACCA TCT CATACCCTTTTCTCA
w38.20	S20 T S38	CAATCTAACACTCCA TCT CATACCCTTTTCTCA
G38-b	T* S38* T*	TG AGA TGAGAAAAGGGTATG AGA TG
w38.f	Sf T S38	CATTTTTTTTTTTCA TCT CATACCCTTTTCTCA
w34.18	S18 T S34	CATCTTCTAACATCA TCT CACATAACAAAACCA
G34-b	T* S34* T*	TG AGA TGGTTTTGTTATGTG AGA TG
Th34.18:18-t	S18	CATCTTCTAACATCA
Th34.18:18-b	s34* T* S18*	TTATGTG AGA TGATGTTAGAAGATG
w18.53	S53 T S18	CATATCTAATCTCCA TCT CATCTTCTAACATCA
w18.44	S44 T S18	CAAAACTCTCTCTCA TCT CATCTTCTAACATCA
G18-b	T* S18* T*	TG AGA TGATGTTAGAAGATG AGA TG
w18.f	Sf T S18	CATTTTTTTTTTTCA TCT CATCTTCTAACATCA
w36.21	S21 T S36	CAACCATACTAAACA TCT CAAACTAAACAACCA
G36-b	T* S36* T*	TG AGA TGGTTGTTTAGTTTG AGA TG
Th36.21:21-t	S21	CAACCATACTAAACA
Th36.21:21-b	s36* T* S21*	TAGTTTG AGA TGTTTAGTATGGTTG
w21.10	S10 T S21	CATACAACATCTACA TCT CAACCATACTAAACA
w21.43	S43 T S21	CATCATACCTACTCA TCT CAACCATACTAAACA
G21-b	T* S21* T*	TG AGA TGTTTAGTATGGTTG AGA TG
w21.f	Sf T S21	CATTTTTTTTTTTCA TCT CAACCATACTAAACA
w26.13	S13 T S26	CACAACTCATTACCA TCT CATTCATTACCTCCA
G26-b	T* S26* T*	TG AGA TGGAGGTAATGAATG AGA TG
Th26.13:13-t	S13	CACAACTCATTACCA
Th26.13:13-b	s26* T* S13*	ATGAATG AGA TGGTAATGAGTTGTG
w13.43	S43 T S13	CATCATACCTACTCA TCT CACAACTCATTACCA
G13-b	T* S13* T*	TG AGA TGGTAATGAGTTGTG AGA TG

Name	Domain	Sequence
w13.f	Sf T S13	CATTTTTTTTTTTCA TCT CACAACTCATTACCA
w20.8	S8 T S20	CACTAACATACAACA TCT CAATCTAACACTCCA
G20-b	T* S20* T*	TG AGA TGGAGTGTTAGATTG AGA TG
Th20.8:8-t	S8	CACTAACATACAACA
Th20.8:8-b	s20* T* S8*	TAGATTG AGA TGTTGTATGTTAGTG
w8.44	S44 T S8	CAAAACTCTCTCTCA TCT CACTAACATACAACA
G8-b	T* S8* T*	TG AGA TGTTGTATGTTAGTG AGA TG
w8.f	Sf T S8	CATTTTTTTTTTTCA TCT CACTAACATACAACA
w43.30	S30 T S43	CACCATTACAATCCA TCT CATCATACCTACTCA
G43-b	T* S43* T*	TG AGA TGAGTAGGTATGATG AGA TG
Th43.30:30-t	S30	CACCATTACAATCCA
Th43.30:30-b	s43* T* S30*	TATGATG AGA TGGATTGTAATGGTG
w30.24	S24 T S30	CACTCATCCTTTACA TCT CACCATTACAATCCA
G30-b	T* S30* T*	TG AGA TGGATTGTAATGGTG AGA TG
w30.f	Sf T S30	CATTTTTTTTTTTCA TCT CACCATTACAATCCA
w44.31	S31 T S44	CAATCCACACTTCCA TCT CAAAACTCTCTCTCA
G44-b	T* S44* T*	TG AGA TGAGAGAGAGTTTTG AGA TG
Th44.31:31-t	S31	CAATCCACACTTCCA
Th44.31:31-b	s44* T* S31*	AGTTTTG AGA TGGAAGTGTGGATTG
w31.25	S25 T S31	CAATTCACTCAATCA TCT CAATCCACACTTCCA
G31-b	T* S31* T*	TG AGA TGGAAGTGTGGATTG AGA TG
w31.f	Sf T S31	CATTTTTTTTTTTCA TCT CAATCCACACTTCCA
w40.27	S27 T S40	CAAACACTCTATTCA TCT CAATACAAATCCACA
G40-b	T* S40* T*	TG AGA TGTGGATTTGTATTG AGA TG
Th40.27:27-t	S27	CAAACACTCTATTCA
Th40.27:27-b	s40* T* S27*	TGTATTG AGA TGAATAGAGTGTTTG
w27.10	S10 T S27	CATACAACATCTACA TCT CAAACACTCTATTCA
G27-b	T* S27* T*	TG AGA TGAATAGAGTGTTTG AGA TG
w27.f	Sf T S27	CATTTTTTTTTTTCA TCT CAAACACTCTATTCA
w39.22	S22 T S39	CATTCCTACATTTCA TCT CACTATACACACCCA
G39-b	T* S39* T*	TG AGA TGGGTGTGTATAGTG AGA TG
Th39.22:22-t	S22	CATTCCTACATTTCA
Th39.22:22-b	s39* T* S22*	TATAGTG AGA TGAAATGTAGGAATG
w22.53	S53 T S22	CATATCTAATCTCCA TCT CATTCCTACATTTCA
G22-b	T* S22* T*	TG AGA TGAAATGTAGGAATG AGA TG

Name	Domain	Sequence
w22.f	Sf T S22	CATTTTTTTTTTTCA TCT CATTCCTACATTTCA
w10.1	S1 T S10	CATCCATTCCACTCA TCT CATACAACATCTACA
G10-b	T* S10* T*	TG AGA TGTAGATGTTGTATG AGA TG
Th10.1:1-t	S1	CATCCATTCCACTCA
Th10.1:1-b	s10* T* S1*	TTGTATG AGA TGAGTGGAATGGATG
w1.23	S23 T S1	CAAATCTTCATCCCA TCT CATCCATTCCACTCA
G1-b	T* S1* T*	TG AGA TGAGTGGAATGGATG AGA TG
w1.f	Sf T S1	CATTTTTTTTTTTCA TCT CATCCATTCCACTCA
w53.5	S5 T S53	CACCACCAAACTTCA TCT CATATCTAATCTCCA
G53-b	T* S53* T*	TG AGA TGGAGATTAGATATG AGA TG
Th53.5:5-t	S5	CACCACCAAACTTCA
Th53.5:5-b	s53* T* S5*	AGATATG AGA TGAAGTTTGGTGGTG
w5.6	S6 T S5	CATAACACAATCACA TCT CACCACCAAACTTCA
G5-b	T* S5* T*	TG AGA TGAAGTTTGGTGGTG AGA TG
w5.f	Sf T S5	CATTTTTTTTTTTCA TCT CACCACCAAACTTCA
Rep6-t	RQ S6	/5IAbRQ/ CATAACACAATCACA
Rep6-b	T* S6* ATTO590	TG AGA TGTGATTGTGTTATG /3ATTO590N/
Rep23-t	FQ S23	/5IABkFQ/ CAAATCTTCATCCCA
Rep23-b	T* S23* ATTO488	TG AGA TGGGATGAAGATTTG /3ATTO488N/
Rep24-t	RQ S24	/5IAbRQ/ CACTCATCCTTTACA
Rep24-b	T* S24* ATTO550	TG AGA TGTAAAGGATGAGTG /3ATTO550N/
Rep25-t	RQ S25	/5IAbRQ/ CAATTCACTCAATCA
Rep25-b	T* S25* ATTO647	TG AGA TGATTGAGTGAATTG /3ATTO647NN/

Chapter 3

A RANDOM WALK MODULE FOR DNA ROBOTS

DNA nanotechnology employs DNA as an engineering material capable of performing computations and also as a structural building material. In this chapter we demonstrate a systematic approach to develop a hybrid DNA structure-strand displacement system wherein we design and characterize a simple module for a DNA robot capable of performing random walk on a DNA nanostructure.

The DNA nano-robotics community has composed a fairly large library of DNA robots capable of exploring a nanoscale surface. Yin et al. [45] designed an autonomous, unidirectional DNA walker that walks along a self-assembled track, driven by the hydrolysis of ATP (an energy currency molecule used in biological cells). At each step it ligates to the next anchorage, disengaging from the previous one by a restriction endonuclease digestion. Wickham et al. [46] implemented a DNA-based molecular motor that integrates long-range transport and information processing. The authors showed that the path of a motor through a network of tracks containing four possible routes can be programmed using instructions that are added externally or carried by the motor itself. DNA robots that are spider-like multi-legged [68] also exist. The molecular spiders have a streptavidin molecule as an inert 'body' and three deoxyribozymes as catalytic 'legs'. The spiders showcase basic robotic behavior when interacting with a precisely defined environment. The study included single-molecule microscopy observations confirming that such walkers achieve directional movement by sensing and modifying tracks of substrate molecules laid out on a two-dimensional DNA origami landscape. Other robots with multiple arms capable of carrying a cargo molecule on each as well as those capable of reconfiguring assembly lines dynamically in response to changing demands have also been designed [47, 69].

Such molecular robots offer programmable motion that allow development of computing networks, molecular systems capable of sorting and processing cargoes according to instructions they carry. The above mentioned robots require consumption of some form of "fuel" to make a step. While some of them use enzymes requiring energy buffers to enable taking a single step, others simply use fuel-based strand displacement reactions to advance.

While the previously designed systems are novel in their function, they share a common feature. The robots can perform only one iteration of their function either because of fuel exhaustion or track digestion after the robot has taken a step. The ability to perform multiple iterations of a particular task can be useful in scaling up the functionality of DNA robots. In our study, we design a robot that can continuously explore the two-dimensional surface of the DNA nanostructure and can be programmed to perform a complex task during its exploration, without requiring any form of external energy.

Identifying modular building blocks, characterizing and integrating them together successfully to perform complex functions will enable a new world of DNA robots that are sufficiently autonomous capable of performing tasks that require continuous exploration and information processing on a 2-dimensional surface. To that end, we designed our first module composed of a robot that can perform bi-directional random walk using strand displacement technology on the surface of DNA origami. The random walking DNA robot system is modular; designed with the capability to be further integrated in a variety of tasks requiring continuous exploration in a complex molecular environment (Chapter 4).

3.1 Implementing random walking DNA robot on DNA origami

A DNA robot capable of performing random walk using reversible toehold mediated strand-displacement reaction was implemented on a grid-like nanoscale surface provided by DNA origami. The staple ends on the origami are arranged in a hexagonal grid. Extending the 3' ends (or 5') on the same side of the origami provides distinct locations spatially separated by ~5.3 nm. The robot is a single stranded DNA with 2 distinct toeholds separated by a strand displacement domain. A single step of the robot uses one toehold domain for initiation of the step and the other toehold for disassociation from its previous position. The following step utilizes the previously dissociated toehold for initiation. This design of the robot along with suitable track design (explained in the next section) enables bi-directional walking mechanism.

The robot has a prefixed start and a goal location. The space between the start and the goal locations is covered with a strategic track layout for the robot. The track is linear with width three, to reduce the possibility of the robot getting stuck in any location caused by a missing track strand. The goal location is monitored using fluorescence spectroscopy. The robot carries with it a quencher molecule and the

goal location has a fluorophore attached (fig. 3.6a,b). The fluorescence at the goal location gets quenched with the arrival of a robot.

Quantitative understanding of the random walk hitting time is possible by comparing it with random walk theory [70]. In theory, the hitting time for two points on a path with length n steps is proportional n^2. We established an experimental setup where the time required for the robot to travel a fixed distance was identified using a linear track with pre-fixed start and stop locations. Different linear track lengths (l) separating robot start location and goal (fig. 3.1a) can be tested to verify random-walking nature of the robot. In our setup, time taken for the robot to reach its goal location from the start location is a function of the number of steps taken to reach the goal (n^2) (fig. 3.1b). If the robot successfully performs random walk, then the relationship between the track length and the time taken for the robot to reach the goal must be quadratic (fig. 3.1c). The system was simulated as a continuous-time Markov chain. The robot takes a step towards one of the immediate neighbors with equal probability (fig. 3.1a).

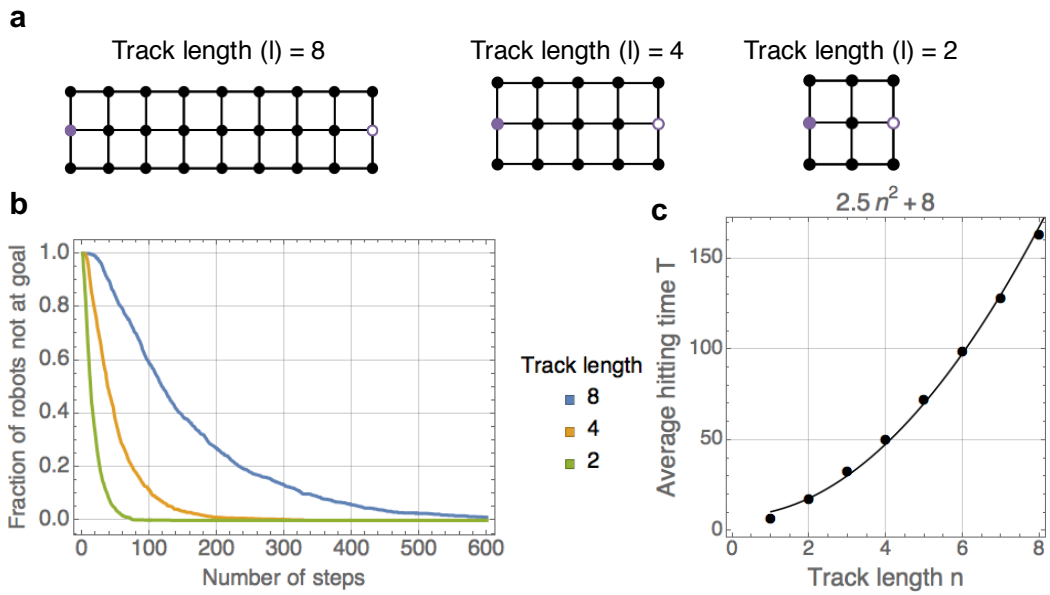

Figure 3.1: **Simulating random walk. a,** Connected nodes treated as tracks for the robot. The robot steps from one node to any neighboring node with equal probability. Start location denoted by purple dot on the left node of the grid, end point is on the right end of the grid, denoted by a purple hollow circle. The grid size can be increased or decreased to simulate a random walk using the nodes. **b,** Simulated trajectories of time taken by a population of robots to reach the goal on different track lengths denoted by the grids in fig. 3.1a. **c,** Average hitting time measured in number of steps for the robot is a quadratic function of its track length to the goal.

3.2 Track design for the random walk

The random walking robot combines dynamic DNA nanotechnology with structural DNA nanotechnology. The robot walks on the origami surface employing DNA strand-displacement technology to travel from one track localized on the origami to the next. DNA staples used to fold the origami structure have 3' extensions (all extending from the same side on the origami) acting as tracks for the robot. To ensure bi-directional walk for the robot, two types of tracks were designed, track-1 and track-2. Track-1 and track-2 are both interspersed in the design fig. 3.2a.

All staples for folding the origami get incorporated with high probability. However, there can be some staples missing when the structure is annealed. In order to account for missing tracks, we use a linear track with width 3, so the robot does not get "stuck" in a location unable to progress towards the goal. Furthermore, we hypothesize that having 3 rows of tracks makes the central row more accessible to the robot as the border rows may provide sufficient electrostatic repulsion for the

central row of tracks to remain upright, which, in our opinion, could increase the efficiency of strand-displacement of the robot on tracks.

The robot makes a step by strand-displacing from track type one to the other. The two track types can be distinguished by the toeholds encoded in them (*foot1** for track-1, *foot2** for track-2). The robot contains both toeholds enabling it to make consecutive steps. For example, if the robot is on track-1 (fig. 3.2b), its toehold domain corresponding to track-1 (*foot1**) is occupied and hence the next possible step for the robot would be onto track-2 using *foot2** as the initiation toehold (fig. 3.2c), branch migrating and finally disassociating via toehold on track-1, *foot1**. Thus, the free toehold on the robot determines the next step of the robot. The track layout for implementing random walk is a 1-dimensional path with width 3, including a "start" location and a "goal" location for the robot. The goal location is fixed for all experiments while the start location is varied to allow testing for different track lengths.

Earlier studies by Zhang et al. [71] showed that by varying the strength (length and sequence) of toehold length, the rate of strand displacement reaction (untethered in solution) can be controlled over a factor of 10^6, and begins to saturate with toehold length 6-7 nts for tested sequences. Toehold length in our study was chosen to be 6 nt to provide fast strand-displacement rate. The DNA sequences were chosen from a pool generated using a 3-base sequence (A, C and T) to reduce spurious interaction. Each track has a *foot* and a *leg* domain extending above a short poly-Thyamine (T) spacer above the surface of the DNA origami.

The spacer length was calculated based on the necessary length required for the track with robot to be able to make contact with the toehold on the adjacent track. Using length of ssDNA as 0.43 nm, dsDNA as 0.34 nm and distance between two neighboring track locations as 6 nm, a spacer length corresponding to 5 nt was calculated for the track. Track-1 has the complimentary *foot1** close to the origami surface and track-2 has the *foot2** domain further away from the origami surface. 5-Thyamines and 11-Thyamines were used as a spacer for track-1 and track-2 respectively. The six additional Ts were added to compensate for the foot position being higher with respect to the origami surface on track-2. The spacer lengths thus calculated allow the toeholds on the robot and the track to be aligned ensuring that the stands do not have to overextend in order to initiate and complete strand displacement (fig. 3.2c).

In addition to the walking domains, the robot also has *hand* and *arm* domains,

which, for the purposes of the random walk module is only used to keep the robot stationary (inactive) at its start position (fig. 3.2d). The staple location on the origami chosen as the "start" location for the appropriate track length has a 5' extension of 20 nt. The tri-molecular robot start complex is localized onto the origami post-anneal via hybridization. The tri-molecular complex consists of (a) the robot; (b) the robot-probe that hybridizes to the origami via the 20nt complementarity to the start location while also occupying *foot1* toehold; and (c) the robot-inhibitor strand, which keeps the robot locked at the start location by occupying *foot2* toehold (fig. 3.2d) The inhibitor strand also has a free toehold to aid in activation of the robot. Thus the robot in this state is stationary and inactive. Addition of a large excess of the robot-trigger molecule removes the robot-inhibitor strand via a forward-biased toehold-mediated strand-displacement reaction. The triggering step frees the *foot2* activating the robot, enabling it to make a step. The robot moves from track-1 to track-2 reversibly until it reaches the goal. The goal molecule is a modified track and has both *foot1* and *foot2*. Hence the robot reaching the goal is an irreversible strand displacement reaction. This reaction is monitored via fluorophore-quencher interaction between the robot and goal using a fluorescence spectrophotometer.

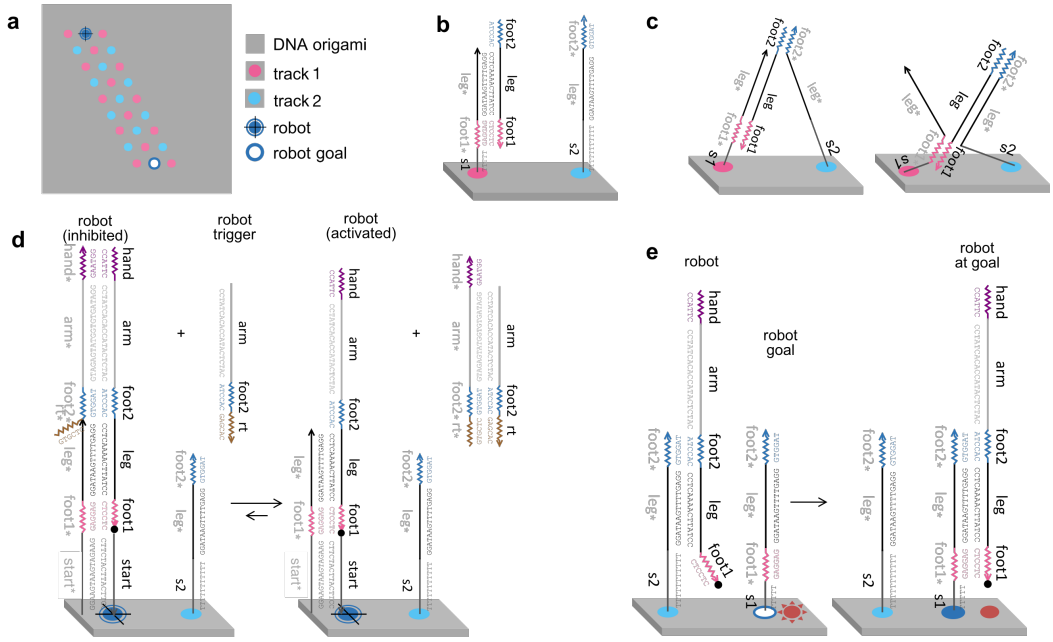

Figure 3.2: **The random walk track design. a,** Scheme of DNA origami surface with track layout for randomwalk. Pink and blue dots represent 3' extension of staples acting as two types of tracks separated by 6 nm. The robot start location is represented by blue concentric circles with a cross running through. The robot goal location is represented by the white circle outlined in blue. **b,** Sequence and domain level representation of track 1 and track 2. The toehold domains *foot1* and *foot2* are 6 nt each and the strand-displacement domain *leg* is 15 nt. Distance between two tracks on the origami surface is 6 nm and linkers *s1* and *s2* are calculated to ensure that the robot can reach the necessary toe-hold to strand-displace between track 1 and track 2. **c,** Mechanism of toe-hold mediated strand-displacement reaction of robot from track 1 to track 2. The linkers *s1* and *s2* stretch and contract to aid displacement. **d,** Robot-start complex showing the inhibited form of the robot. Addition of robot trigger strand displaces the inhibitor strand reversibly. The activated robot has *foot2* toehold to start random walk on the track. **e,** The robot goal is a modified track with both *foot1* and *foot2* so the robot is irreversibly bound to the goal.

In the process of implementing random walk, we experienced several challenges that influenced our design and experimental protocols. The observations made during this progress which eventually lead to a successful implementation of random walk were:

- *Rigidity of DNA origami as a testing ground for the robot:* Structural fluctuations of DNA origami in solution, influencing the extent of undesired reactions on its surface.

- *DNA sequence of foot domains of the robot:* influencing the rate of walking.
- *Purity of DNA origami:* Fraction of partially formed nanostructures interfering with well-formed structures, influencing the completion level of desired reactions.

Each of the above observations are explored in more detail below.

3.3 Rigidity of DNA origami as a testing ground for DNA robots

Quantitative assessment of random walk involves measuring the number of steps taken by the robot to reach its goal, and verifying that it varies quadratically with the length of track. Having free robots in solution can cause them to directly land on an origami. If the DNA nanostructure is flexible, the goal strand on the origami can pick up the robot directly after its activation. The two scenarios do not constitute random walk and are undesirable outcomes. Our measurement technique allows us to monitor the position of the robot at the goal but not its progression towards the goal. Hence the signal collected from the fraction of molecules where the robot lands directly on the goal without performing random walk in the test tube will introduce noise in the signal collected from robots performing the desired action. Free robots in solution can be removed by various purification methods (discussed later in detail). But testing the rigidity of the nanostructure is more challenging.

To asses the effect of structural fluctuations of the nanostructure on robot:goal interactions, we designed an experiment to test for robot directly reaching its goal by excluding tracks on the origami surface. We call this the "space-walk" reaction. A rectangle origami [38] was chosen as the initial test-bed for building the random walk system(fig. 3.3a). Three different start locations were chosen to asses the extent of interaction between the robot and goal tethered to the origami in the absence of tracks. Figure 3.3b shows a schematic of the rectangle with robot start location and the goal location separated by the farthest distance limited by staple position (72 nm). The blank gray area in the schematic represents the lack of tracks between the start and goal.

To prepare the nanostructure testing ground, rectangle origami was annealed with regular staples (without track extensions), robot start extension, goal extension and the fluorophore-tagged staple for labeling the goal. The robot-start complex was later localized on the origami by hybridizing it to the robot-start extension. The whole system was purified to ensure minimal free floating strands in solution. With

farthest possible spatial separation between robot and goal on the origami (72 nm), the data indicated that the interaction was minimal (fig. 3.3e). Testing two more separations with spatial distances 48 nm and 24 nm between the robot and goal, an increase in the robot-goal interaction was observed. We hypothesized the reason for such a dependence of spurious robot-goal interaction on spatial distance to be the flexibility of the DNA scaffold. We hypothesize that the flexibility of origami in solution could provide sufficient structural fluctuation (fig. 3.3d) that allows the robot to reach the goal directly without needing tracks. This behavior would increase with decreasing distance based on the axis of flexibility of the origami, as observed in fig. 3.3e.

In order to test this hypothesis, a more rigid double-layered DNA origami was designed(fig. 3.3f). As the name suggests, the double-layer origami has 2 layers with the helices of one layer traveling perpendicular to the other layer. The two layers are held in place by extra-long staples that traverse both layers. Hairpin motifs are used as edge staples to reduce stacking interactions [72]. The staples can be extended on either layer, however one layer provides higher possibilities for extensions than the other (Appendix, fig. 4.14). This was named the "top-layer". We believe the helices of the two layers being perpendicular to each other reduces structural fluctuation of the origami significantly (fig. 3.3i) and still provide sufficient surface area to make a reasonably sized test ground for the robot. Using CanDo [73], the thermal fluctuations of the single layer and double layer structures were compared. The simulation indicated increased degree of structural fluctuation in case of the single layer rectangle when compared to the double layer square (fig. 3.3d,i).

The space-walk test was repeated using the double-layer structure. The maximum separation between start and goal location accommodated along the diagonal of the double layered origami was 48 nm. Space walk was tested for lengths 48 nm, 24 nm and 12 nm. The kinetics indicated an overall decrease in signal in all three test cases. However, there was no correlation between spatial distance on the origami and the robot-goal interaction. The overall decrease in signal was comparable in all three cases (fig. 3.3j). The signal decrease could be the result of the inter-origami interaction where goal of one origami picks up the robot on another, however, the distance dependent behavior causing intra-origami interaction seemed minimal.

The result of this study implied that the double-layer origami was perhaps a better option than the single layer origami for designing a rigid nanoscale test ground for the random walking robot.

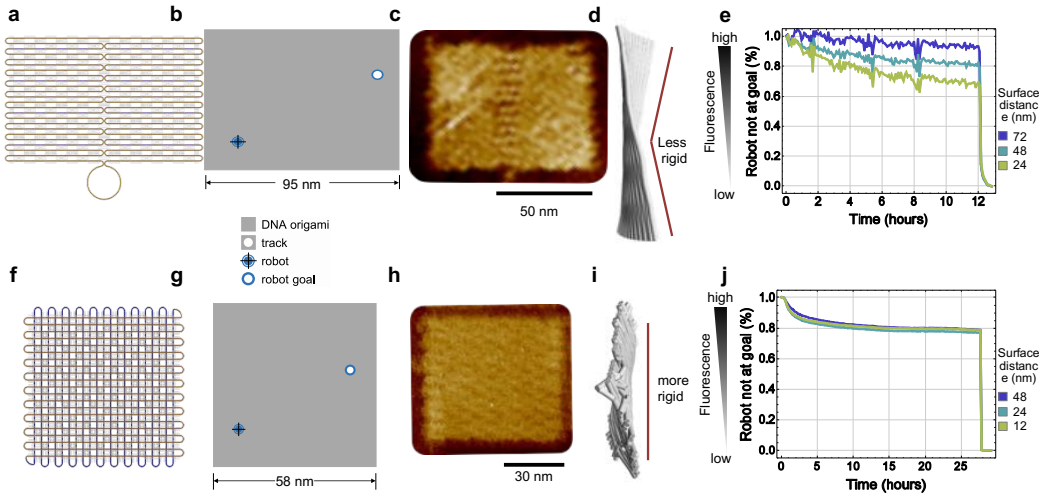

Figure 3.3: **Comparing rigidity of single layer and double layer origami. a,** Schematic representation of scaffold routing for single layer origami rectangle from caDNAo [74]. **b,** Schematic representation of the single layer rectangle origami showing robot start and goal locations separated by 72 nm. **c,** AFM image of single-layer rectangle origami with single stranded extensions representing the longest possible track length. **d,** CanDo [73] image showing the bend in single layer rectangle origami. The structural fluctuation is an indication of flexibility of the structure. **e,** Fluorescent experiment data showing the interaction between robot-start location and goal location. Moving the start location closer to the goal increases robot-goal interaction in the absence of tracks. **f,** Schematic representation of scaffold routing for double layer origami square using caDNAo [74]. **g,** Schematic representation of the double layer square origami showing robot start and goal locations separated by 48 nm. **h,** AFM image of double-layer square origami with single stranded extensions representing the longest possible track length. **i,** CanDo image showing the bend in double layer square origami. The structural fluctuation is an indication of flexibility of the structure. **j,** Fluorescent experiment data showing the interaction between robot-start location and goal location. Moving the start location closer to the goal did not change the robot-goal interaction in the absence of tracks.

3.4 DNA sequence and its effect on the rate of walking

We explored two DNA domain variations for the *foot* toehold of each track. In one case, we tested two different *foot* toeholds for track-1 and track-2 and compared the experimental results with the same *foot* toehold for both tracks.

For testing the above two cases, double layer origami was designed with a track layout that localized the robot a single irreversible step away from the goal location (fig. 3.4a). The experiment was performed with annealed origami, spin-filter

purified to remove unused strands in solution. The inactive robot was localized at the start location. Addition of trigger molecule activated the robot. A staple with fluorophore on the 5' end and without a track extension on the 3' end was used to label the goal location. The experiment is complete when the robot makes the irreversible step to the goal, quenching the fluorescence at the goal. The result from this experiment was intriguing for two reasons.

- The data suggests that the half completion time of the reaction on the origami surface is $t_{1/2} = 2.5hr$.
- The gray trajectory in fig. 3.4d shows roughly 50% completion level with respect to the total quenched signal after 14 hours. This suggests that only about 50% of the robots in the total origami population reached their respective goals.

The $t_{1/2}$ observed is slower than the rate observed by Teichmann et al. [75] wherein a single strand displacement cascade reaction was tested, the authors observed $t_{1/2} = 1000s$. The cascade was triggered by a molecule in solution, similar to the robot activation method, monitored using fluorescence spectroscopy. The triggered strand disengaged from its anchor free to float into solution and land on a docking site 21.5 nm away. So while this might be a comparable study, the system is quite different from the random walking robot. However, the $t_{1/2}$ measured could provide a rough estimate for a single reaction tethered on the origami surface.

The difference in timescale between our data and the study by Teichmann et al. [75] motivated us to investigate the cause of the slower reaction rate. It is important to note that the two types of tracks are different in the sequence of the walk-enabling toehold. The toehold on track-1 (*foot1**) consists of 4 guanines and 2 adenosines (A), while toehold on track-2 (*foot2**) consists of 3 guanines and 3 adenosines (A).

A DNA duplex is formed with a single DNA strand hybridizing with its complimentary strand via hydrogen bonds between the bases encoded in the strands. Particularly, adenine (A) and thymine (T) bases form two hydrogen bonds and cytosine (C) and guanine (G) form three hydrogen bonds. Although the hydrogen bonds between specific bases are just one of the stabilizing interactions in the DNA double helix, they plays a crucial role in designing DNA circuits. G-C interaction is strengthened by 3 hydrogen bonds and the disassociation energy (ΔG) for the G-C pair at a given

temperature is hence higher than A-T. The constituents of *foot1** toehold result in the sequence having higher ΔG making it stronger in comparison to *foot2**.

The slow reaction rate for the robot making the step towards the goal suggested that disassociation of the robot from *foot1** may be a rate limiting step.

The hypothesis was further strengthened by a coarse-grained DNA model study conducted by Šulc et al. [76] of a DNA cargo strand capable of unidirectional motion on single stranded DNA stators tethered to the origami surface [46], the authors investigated the free-energy profile of the multiple stages involved when the cargo makes a step via toehold mediated strand displacement. The cargo starts off as a DNA duplex on a stator. A nicking enzyme recognizes the site on the cargo/stator duplex and nicks the backbone of the stator a short distance from the 5' end (The system has staple extensions on 5' end different from the random walk system having 3' staple extensions). The shorter nicked strand spontaneously falls off the stator revealing a toehold on the cargo, which can then take a step towards the next stator. While measuring the free-energy profile of the cargo/stator strand-displacement reaction tethered to the origami surface, the authors noted that toehold initiation occurred with the first hydrogen bond formed with nucleotides whose contact point is far away from the nucleotides attached to the surface. Hence, it is not difficult for the strands to reach each other at contact point. With the progression of strand displacement as more hydrogen bonds are formed, the contact point moves closer to the nucleotides attached to the surface, eventually getting close enough that maintaining the structure causes considerable tension and is free-energetically undesirable.

The study implies having weaker disassociation toehold closer to the origami surface may reduce the tension in the strands making the completion of strand displacement more favorable. To that end, we replaced *foot1** toehold closer to the origami surface with the weaker *foot2** toehold, making the disassociation of the robot while moving from track-1 to track-2 free-energetically favorable.

A design iteration with all tracks with toehold *foot1** replaced with the weaker *foot2** yielded the blue trajectory in fig. 3.4d. Comparing the kinetics of the blue with gray trajectory in fig. 3.4d indicates an improved reaction rate with weaker toeholds.

The results of this test were compelling enough to chose *foot2** as the walk-enabling toehold for both track types which will hence forth be referred to as just *foot** or *f**.

Changing to a weaker toehold did not change the completion level of the experiment. The blue trajectory in fig. 3.4d still shows roughly 60% of the total number of robots at the goal. While this completion level reaches ~10% higher, it was still not clear as to why ~40% of the robots were not able to reach the goal.

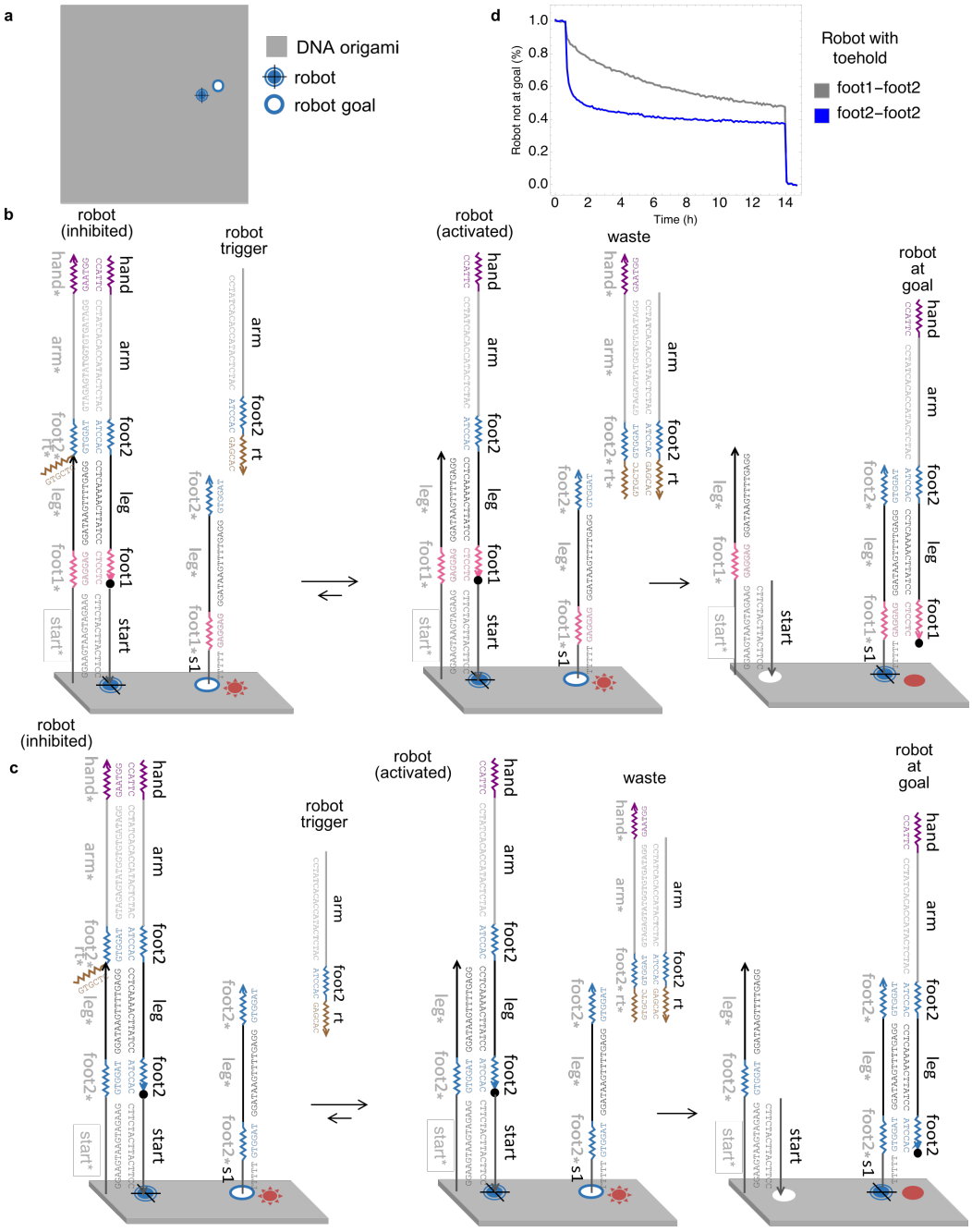

Figure 3.4: **Effect of track sequence on randomwalk. a,** Schematic of double layer DNA origami showing robot-start location single step away from the goal location. **b,** Domain and sequence specification of robot and goal with 2 different toeholds : *foot1* and *foot2*. The inhibited robot at the start location when activated takes a single irreversible step and binds to the goal. **c,** Domain and sequence specification of robot and goal with 2 same toeholds : *foot2* and *foot2*. The inhibited robot at the start location when activated takes a single irreversible step and binds to the goal. **d,** Comparing fluorescent kinetics data of the robot with 2 different toeholds (*foot1*, *foot2*) and 2 same toeholds (*foot2*, *foot2*).

3.5 Purity of DNA origami and its effect on the reaction completion level

Observing roughly 60% of the robots reaching their goal in the previous section, we hypothesized that perhaps there was fraction of malformed origami structures in the test population where the robot start location and the goal were not geometrically located as designed. To test this hypothesis, we investigated two different purification techniques and compared them with unpurified origami using the robot walking on linear tracks with varying lengths.

The first set of experiments for random walk was performed using three distinct track lengths with one control. The longest available track length on the double layer origami was 8 (separated by surface distance of 42 nm). The robot start location was moved closer to the goal allowing for testing track lengths 4 (24 nm) and 2 (12 nm). Space walk, with a surface distance that corresponds to 48 nm, was added as control to observe the fraction of robots that directly reached the goal on another origami via inter-origami interaction.

Figure 3.5a shows the layout for track lengths 8, 4, 2 and space-walk (labeled none). The data in the bottom panel (fig. 3.5b,c,d) tracks the fraction of robots reaching their goals. Each trajectory represents a particular track length. The total number of steps taken by robot to reach its goal is a quadratic function of track length (fig. 3.1c). Hence a robot would reach its goal faster on a 2-step track than on an 8-step track. This part is evident in the data (fig. 3.5b,c,d) shown. The space-walk control indicates the fraction of robots on one origami that have been picked up directly by the goal on another origami.

"Unpurified" test sample

The data in fig. 3.5b was obtained using "unpurified" sample. The nanostructures were annealed with necessary extensions (robot start location, tracks, goal) independently for the four cases shown in fig. 3.5a. For every 1x of M13 scaffold, 3x staples (with and without track extensions, goal) were added to ensure well-formed origami with relevant components. To ensure robot localization on origami (and not have them floating in solution) only 1x of the robot start location staple was added to the anneal mixture. After annealing the origami, 1x copy of inactivated robot was localized on the origami by incubating them for 5 hrs at 25 °C. All four samples were prepared simultaneously in different tubes. The robots were then activated simultaneously while continuously monitoring the signal at the goal. Qualitative observation of the data curves seen in fig. 3.5b follow the general trend seen in sim-

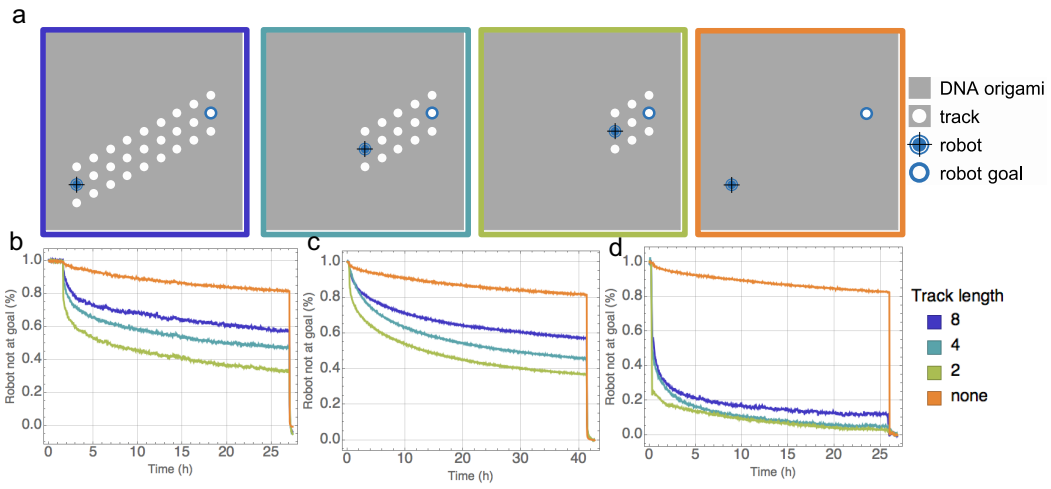

Figure 3.5: **Effect of sample purification on randomwalk. a,** Schematic representation of different track lengths on the surface of double layer origami for randomwalk. The goal location is fixed and the robot-start location is moved to implement randomwalk on track lengths 8,4 and 2. Track length 8 without tracks is used as control to account for spurious robot-goal interaction. **b,** Comparing fluorescent kinetics data of robot performing randomwalk on 8,4,2,0 track lengths in an unpurifed sample, **c,** spin-filter purified sample, and **d,** gel purified sample.

ulation (fig. 3.1b). However, the fraction of origami reaching their goal is different in each trajectory with roughly 40% of the robots not reaching the goal even in the shortest track length.

There are at least two reasons to cause the resulting completion level to be as low. Obtaining the correct stoichiometry of molecules is extremely challenging. We may not have achieved a 1:1 stoichiometry of robot to origami because of differences present in concentration measurement technique including the transfer of molecules via pipetting. The excess tracks (2x) in solution may possibly pick up the robot from an origami which may or may not land on the same/ different origami, further skewing the 1:1 robot to origami ratio.

Spin-filter purified test sample

We modified our protocol to try and obtain the necessary stoichiometry between reacting molecules and also "purify" our sample to eliminate "stray" tracks in solution. To ensure 1:1 stoichiometry of robot to origami, we annealed 1x M13 scaffold with 3x staples (with and without track extensions, goal and robot start location). After the anneal, we added 3.5x inactivated robot and incubated for 5 hrs at 25 °C to localize robot on the origami. The nanostructures were then purified using spin-filters with necessary membrane size sufficient to separate them from unused

smaller DNA components (staples, tracks, goals and robot complex). Figure 3.5c shows the result of this protocol modification. We observed the general trend of curves remain unchanged compared to fig. 3.1b, but also that the fraction of robots reaching goal did not improve even after running the experiment for a longer time than in fig. 3.5b (24hrs).

These results motivated us to revisit the system design. Figure 3.6a is an overview of the relative positions of robot, goal and tracks. If the nanostructure with its staple extensions form as desired, the track length separates the robot start location from the goal. The staple at the goal location has the goal extension on the 3' end and the adjacent staple has a fluorophore on the 5' end. When annealed, the goal extension and the flurophore are on adjacent DNA bases and are considered to be at the same location on the origami grid (fig. 3.6b). Hence the quencher on the robot quenches the signal on the fluorophore that is at the same grid location as the goal extension, but not on the goal extension itself. This simple design choice ensures that all 3 components need to be spatially constrained to observe quenching of the fluorophore signal, which cannot occur if the strands are not located on the origami as designed. The 60% completion level suggested that there may be a fractional population of origami where the above mentioned design criteria is not fulfilled.

Most origami designs used for building robot test grounds in previous studies [46, 68, 69] use the single layer rectangle [38]. The rectangles fold with high fidelity producing $>$ 90% yield. The double-layer square used for the first time in this study is a more complex structure and previously uncharacterized. The signal at the goal extension can only be quenched if the robot, goal and the staple with the fluorophore on the 5' end are all at the same location on the origami grid, but only in 'well-formed' structures. When repeated trials of the random walk experiment as mentioned above did not improve the completion level in the experiment, a more thorough investigation of the origami population was required.

Gel-purified test sample

The double-layer nanostructures were annealed with 3x staples (without extensions) for 1x M13 scaffold and visualized on agarose gel to check for target structures. Previously, complex nanostructures have been shown to require higher concentrations of cations (Mg^{++}/Na^{+}) for annealing [74] compared to 12.5 mM Mg^{++} used in [38]. Figure 3.7a is an agarose gel with the double-layer structure annealed using different Mg^{++} concentrations ranging from 0.5 mM to 20 mM. The first lane is

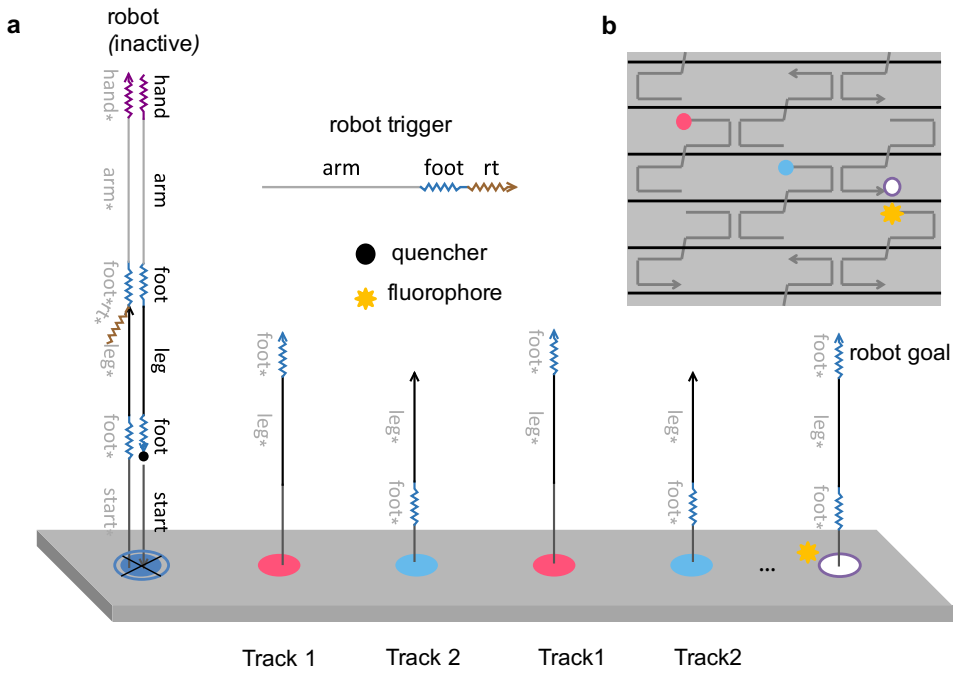

Figure 3.6: **Fluorophore labeling on origami. a,** Schematic representation of origami with robot at start location carrying quencher, separated by tracks from the goal location with the fluorophore. **b,** Schematic of staple location carrying fluorophore on the 5' end that is adjacent to the staple with goal extension on the 3' end.

M13 DNA only. The band formed by M13 functions as a control to compare the annealed nanostructures. Given that the origami have a more compact geometric shape compared to the floppy single stranded M13 DNA, they should run "faster" in the agarose gel despite being double the molecular weight (as they contain staples making them double stranded). The agarose gel showed a band that ran faster than the M13 band in all lanes, each lane had multiple bands. We also noted that while 0.5 mM Mg^{++} concentration was insufficient to form nanostructures (evident from the smeared lane), 12.5 mM seemed sufficient and further increasing the cation concentration resulted in aggregation of structures large enough to not enter the gel (evident from the bright wells and absence of bands in the gel corresponding to the lanes).

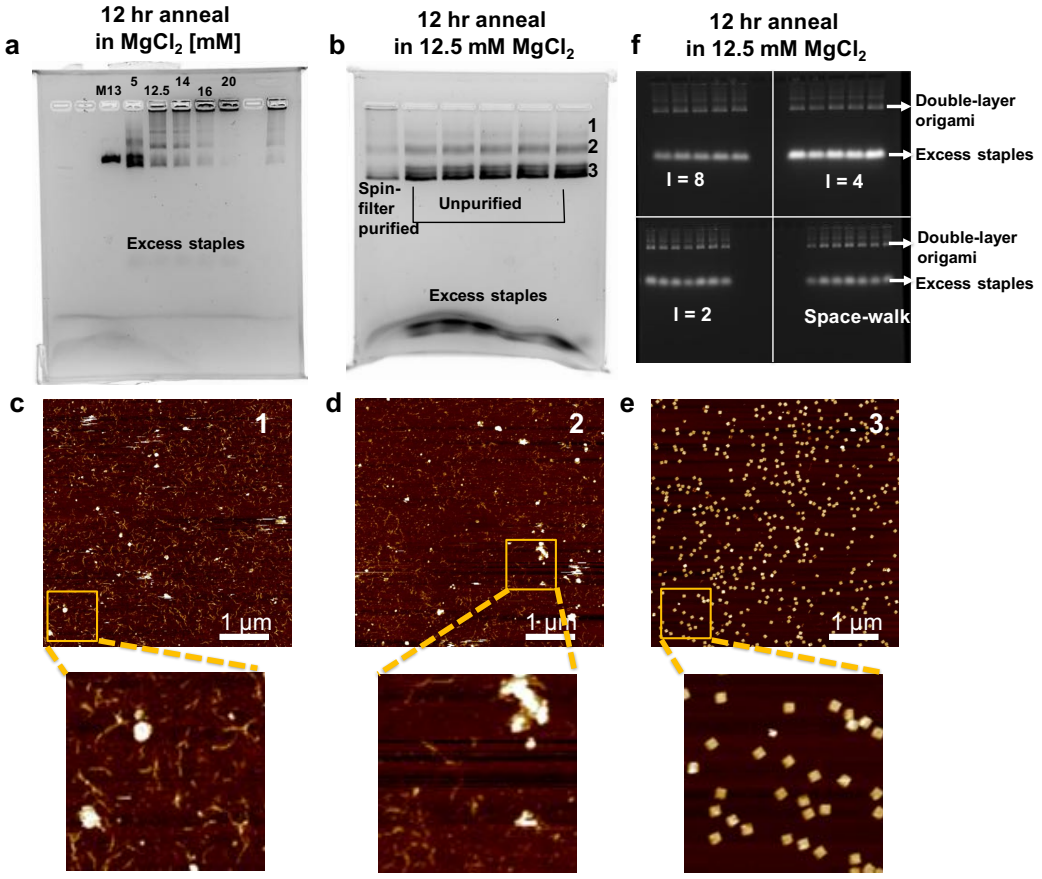

Figure 3.7: **Gel purification of DNA origami.** 2% agarose gel characterizing double layer origami annealed over 12 hrs. **a,** Double layer origami annealed at different [mM] of Mg^{++} concentrations as shown above each lane. **b,** Gel comparing spin-filter purified with unpurified origami. Excess staples were efficiently removed in spin-filtered sample. Bands labeled 1,2,3 are excised from unpurified lane and the DNA extracted individually. **c,d,e** The extracted DNA from three labeled bands were scanned using AFM, individually. Bands 1 and 2 mostly contained hairy DNA structures while band 3 mostly contained the expected target structures. The small rectangle area indicated in each scan is magnified below each scan. **f,** Gel showing double layer origami with track lengths (l) 8,4,2 and space-walk control; annealed and purified with gel extraction for collecting data shown in fig. 3.5d.

Noting that 12.5 mM Mg^{++} concentration was mostly sufficient to fold these nanostructures, we focused on the presence of the multiple bands in each lane. Running this sample through a spin-filter column did not remove the bands as evident from fig. 3.7b. The bands labeled unpurified in fig. 3.7b show three dominant bands. The two bands with very small separation between them were grouped together to

form band-3 for this exercise. Given that band-1 and band-2 ran slower with respect to the M13-only band, we conclude that these are heavier on account of having staple strands that are not present in the reference band. Further, the presence of staples does not ensure formation of the target structure. It only suggests that the single stranded M13 is perhaps double stranded and hence is at a higher molecular weight. To verify this theory, the three bands were excised from the gel, DNA extracted and imaged separately using atomic force microscopy (AFM).

The fig. 3.7c,d,e show the contents of each band. The top two bands showed partially folded structures (fig. 3.7c,d). Band-2 specifically also had what looked like multimers of the double-layer origami. One such aggregation is shown magnified in fig. 3.7d. AFM of band-3 (fig. 3.7e) showed nanostructures that closely resemble the targeted double-layer origami. While AFM is a great tool for directly visualizing target and off-target structures, it is difficult to quantify or even detect the different populations that exist in the mixture with good confidence. Gel electrophoresis method separates each of the populations in this case providing an efficient technique for extracting target structures.

The results of this exercise underlined the importance of building and characterizing the nanostructures that form an important component for the robot to perform its intended function. The less than optimal completion level, we believe, resulted from the structures observed in band-1 and and band-2 of the gel. These partially formed structures while clearly incorporating the fluorophore strand (gel in fig. 3.7b is unstained and visualized solely using fluorophore labeled staple strand) are geometrically different from the target double-layer nanostructures, hence the robot start site, tracks and goal may not be connected as intended. Thus, robots localized on these structures do not perform random walk and may perhaps not even leave the start site.

With the agarose gel showing a mixed population of target and off-target structures, band excision from the gel followed by purification is required to separate out target structures. Figure 3.7f shows the samples that were annealed and run on gel for extracting the target structures. These gel-extracted structures were then used for the random walk experiment with three different track lengths. Note that the data shown in fig. 3.5d is the result of removing the partially formed structures, while retaining well-formed structures. Gel purification technique dramatically improved our results, making it possible to observe a major fraction of robots reaching their goal.

3.6 Demonstration of the random walk

Encouraged by the results from the previous section, we set up a more intensive examination of the random walk system. We demonstrated random walking on linear tracks of lengths 1 through 8. Space-walk was used as control to estimate inter-origami interaction. Figure 3.8a is a schematic representation of the double layer origami indicating the robot start location and the goal location for linear track lengths 1 through 8 including space-walk control.

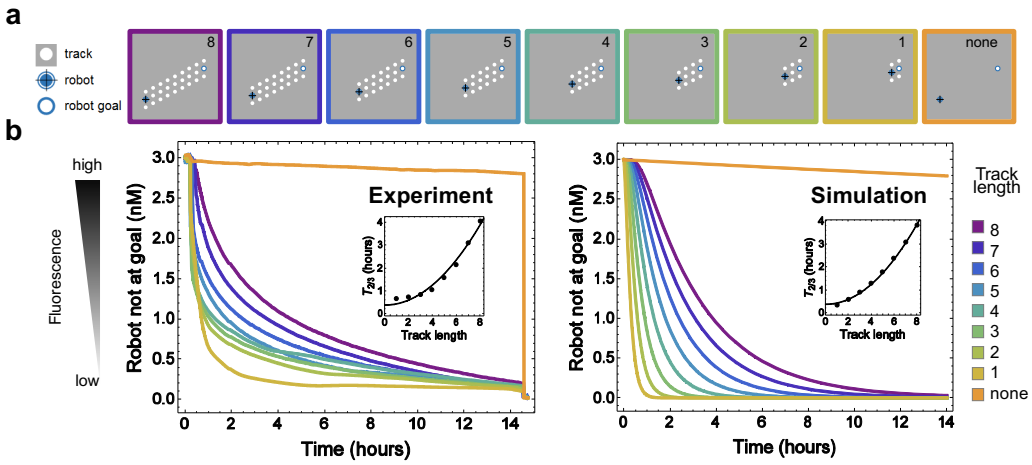

Figure 3.8: **Experiments and simulation of the random walk. a,** Schematic representation of different track lengths on the surface of double layer origami for randomwalk. The goal location is fixed and the robot-start location is moved to implement randomwalk on different track lengths ranging from 8-1 in decrements of 1step. Space-walk control is labeled as none. The outline color of each schematic corresponds to the data curve shown below. **b,** Comparing fluorescent kinetics data of space-walk control and robot performing random walk on 8-1 track lengths in a gel-purified sample with simulation. The insets in bot sets of plots show the quadratic relationship between track length and 2/3 completion time ($T_{2/3}$).

The two sets of trajectories in fig. 3.8b compare experimental data with simulation data as labeled. Double-layer origami were annealed with 10x staples, tracks and goal strands for every 1x M13 scaffold strand. After gel extraction of the target species, the fluorescence signal of each extracted sample was measured in the spectrofluorometer and the concentration was determined by using a calibration curve established with the same fluorophore. Once the concentration was established, a standard concentration of 3 nM (1x) was chosen for data collection. 1x robot start complex (PAGE purified to remove unused strands and malformed complexes) was

incubated with all nine samples for 5 hours at 25 °C. The robot was then activated and the fluorescence signal from the robot-goal reaction was monitored continuously (30 sec intervals) until its fluorescence signal equilibrated (~14 hrs). A large excess of robots with quenchers were added at this point to hybridize any unoccupied goals. This step in the protocol is useful to estimate the fraction of goals that were unoccupied, providing quantitative estimate into the fraction of robots not reaching the goal.

The goal signal can remain unquenched either because the robot has not completed random walk or our estimate of having equal robot to origami stoichiometry was skewed to lower side. From the data in fig. 3.8b, 92.54% of robots on origami with track length 8 reached the goal, and 95.17% of robots on origami with track length 1 reached the goal at the 14 hr time point. The remaining test cases had robots reach their goal at a fraction inbetween this range. The large separation in the completion level between space-walk control and the test cases indicate very low inter-origami interaction at the chosen concentration (3 nM). From the data, 7.2% (14 hr time point) of the robots reached the goal via inter-origami interaction.

It is noteworthy that space-walk control data from fig. 3.5d differs from the data in fig. 3.8b. Figure 3.5d shows roughly 10% of the robots picked up by 14 hrs and this interaction continued (18% at 24 hrs) until the experiment was stopped by adding a large excess of robots. Differences in experimental protocol may have resulted in the higher fraction of robot space-walk in fig. 3.5d. The concentrations of the origami were not measured in the same elaborate manner as in fig. 3.8b. For data in fig. 3.5d, after gel-extraction, a large excess of robot start complex was incubated with the samples and spin-filtered to remove any unbound complexes. The robots on these samples were then activated and the signal monitored. There was no quantification of concentration. Space-walking defined as a bimolecular reaction, where robot on one origami is picked up by the goal on another, is concentration dependent. Higher concentration of origami increases the probability for collision, thus increasing inter-origami interaction. This hypothesis can be tested by titrating different concentrations of origami to check for increase in interaction with increase in concentration. However, even with the existing dynamics of origami collision at a seemingly higher rate, the clear separation between space-walking and random-walking to the goal made the above suggested titration test unnecessary. Also, if the concentration of the origami can be quantified and the space-walking rate calculated, the reaction rate can be incorporated in a model to simulate the other eight

cases to better understand the effect of space-walking. We hence measured the concentration of the origami for all following experiments including data in fig. 3.8b.

3.7 Model

The two-thirds completion time measured from the experimental data was plotted against track length (l) (fig. 3.8b, inset). We found the best quadratic fit to be

$$T_{2/3} = 0.38 + 0.055 \times l^2$$

The constant term is attributed to the delay caused by the triggering reaction.

In order to draw a quantitative understanding of the DNA-based random walk, we examine the various reactions involved in the experimental process.

if $(x, y) \neq (x_{stop}, y_{stop})$,

$$Robot_{x,y} : Inhibitor + Trigger \underset{k_t}{\overset{k_t}{\rightleftharpoons}} Robot_{x,y} + Trigger : Inhibitor \tag{3.1}$$

if $(x, y) \neq (x_{stop}, y_{stop})$,

$$Robot_{(x,y)} \xrightarrow{k_w} Robot_{(x*,y*)} \tag{3.2}$$

$\text{Robot}_{x,y}$ is the robot at an arbitrary location (x, y) and $\text{Robot}_{x*,y*}$ is the robot at a neighboring location of (x, y). (x_{stop}, y_{stop}) is location of the goal. The robot reaching the goal is an irreversible reaction and hence the robot cannot continue its walk on reaching the goal. Initially the robot tethered to the origami is inactive. Adding a large excess (20x) trigger molecule, the robot is activated (fig. 3.2d,e) and can begin walking. The triggering rate k_t and the walking rate k_w defined the above reactions where k_t is the bimolecular reaction rate of the trigger-activation of the robot and k_w is the unimolecular rate of robot walking.

By comparing the quadratic fit of the two thirds completion time versus the track length in the simulation and in the experimental data, k_w and k_t were derived,

$$k_w = 0.0035 s^{-1} \tag{3.3}$$

$$k_t = 3.2 \times 10^4 M^{-1} s^{-1} \tag{3.4}$$

A previous estimate of stepping rate was performed on the robotic system characterized in [46]. The working of the system is explained in section 3.4. The study derived a stepping rate of $0.009 s^{-1}$ for a step of 6.2 nm [37], similar to the walking rate of our robot.

k_t is a bimolecular strand displacement reaction between the trigger strand in solution and the robot on the origami using a 6 nt toehold with average GC content. Previous studies have shown the rate constant for a strand displacement reaction with similar toehold strength (although both reactants were free in solution) to be in the order of $10^6 M^{-1} s^{-1}$ [29].

Our hypothesis to explain the slow reaction rate is the following: the trigger molecule in solution is reacting with a molecule tethered to the origami surface, which could be significantly slower than two free molecules interacting in solution. In addition to the diffusion times for the two types of molecules being different based on their size, the single stranded trigger may also experience an electrostatic repulsion from the origami surface with single stranded tracks. Previous studies by [75] also reported a slower rate of $10^4 M^{-1} s^{-1}$ when the signal molecule in solution reacts with a molecule localized on the origami surface. We hence conclude that the bimolecular triggering step may be 2 orders of magnitude slower when the interaction is between a free floating DNA strand and the DNA origami when compared to the reaction rate between two free strands in solution.

Further, the "space-walk" control where the robot is separated by 48 nm from the goal without tracks has a slow but considerable declining slope. From our previous experiments shown in fig. 3.3j we eliminate intra-origami interaction. Hence the space-walk control was modeled as a bimolecular reaction with rate k_s, where the robot localized on one origami is picked up by the goal localized on a different origami.

If $(x, y) = (x_{start}, y_{start})$ and $(x*, y*) = (x_{stop}, y_{stop})$,

$$Robot_{x,y} + Goal_{x*,y*} \xrightarrow{k_s} Robot_{x*,y*} \tag{3.5}$$

$$k_s = 5 \times 10^2 M^{-1} s^{-1} \tag{3.6}$$

The goal and the robot are both localized on the origami and the inter-origami interactions are occurring very close to the origami surface. This could potentially lead to the slow down of the space walk reaction by two orders of magnitude compared to k_t, as it requires collision of two large structures (compared to single strands of DNA) in solution while also requiring the necessary orientation of the reactant molecules to facilitate such a reaction.

Conclusions

The main focus of this chapter was to define a simple module for molecular robots capable of performing random walk. Designing and implementing the testing ground involved designing a more rigid DNA origami. To that end, we designed a double-layer origami that provides a more rigid platform for hosting localized reactions than the more commonly used single-layer rectangle. Further, the rigidity of the nanostructure successfully reduced spurious interactions on the surface. The origami purity also played an important role in the success of implementing the test ground. A fraction of partially folded structures resulted in a significant portion of robots not completing their assigned task. Agarose gel purification helped in obtaining fully formed structures enabling better experimental conditions for characterizing the random walk performed by the robot.

The random-walking system comprising of a single stranded DNA robot and tracks uses the sequence space of DNA to reuse components at the domain level. The robot is autonomous and functions solely on the energy from base-pair hybridization without requiring an additional energy source.. A distinctive labeling method to track the end-point of the robot allowed for recording of goal-robot reaction occurring only on the origami surface. This technique could potentially be used to track the progression of the robot by placing fluorophores along the track in multiple locations.

A mass-action model was developed to derive reaction rates for robot activation, robot walking and possible spurious interactions as in the case of space-walk. Additionally, with the information obtained during experimental troubleshooting of major components of this system, we believe that the robot is capable of successfully performing its designated task of random-walk on a one-dimensional track. The scope of our system can easily be scaled to include other functions for the

robot, including programming it to explore the entire surface of the origami while now performing a two-dimensional random walk.

Chapter 4

A CARGO-SORTING DNA ROBOT

The previous chapter focused on design and implementing a robot capable of performing one-dimensional random walk on a DNA nanostructure. The random walk behavior on linear tracks can easily be expanded to explore a larger area of the nanostructure by performing a 2-dimensional random walk. The versatile design of the system allows for integrating additional modules that facilitates programming an additional task for the robot while not hindering its random walk function.

Previously designed DNA robots have mostly been stand-alone systems with bipedal [49, 77, 78] or more complex multi-pedal conformations [47, 68]. Many robot systems implemented thus far are primarily focused on the design of robot walking mechanism, with few exceptions. Gu et al. [47] discusses a four-footed, three-armed walker based on tensegrity-triangle organization capable of carrying cargo in the form of 5 nm goal particles. The assembly-line analogy used for describing the function of this robot discusses the ability of the robot to use its three arms to pick up three different cargos, consecutively. The authors speculate on the usage of this method to allow DNA robots to construct new chemical species perhaps not readily constructed by other means. Douglas et al., [42] designed a clamp-shell DNA origami that encapsulated antibodies as cargo molecules and remained locked until a DNA "key" opened and released cargo. The released cargo can bind to cellular receptors to initiate immune response and perhaps even be used to specifically target cancer cells. These studies have been seminal in the field of cargo-carrying robots, but given the complexity in engineering such systems, have been few.

Our study includes integrating the previously designed behavior of the random walking robot with a new function; demonstrating a sophisticated example of programmed motion – sorting a mixed population of molecules into distinct piles at designated locations. The robot can perform multiple iterations of the cargo sorting task, without requiring reloads of relevant components. We demonstrate the sorting of two distinct cargo types to their designated goals.

Engineering a sorting robot at any scale requires designing an algorithm that can be programmed into the robot for implementation. And in this study involving a robot

at the nanoscale, implementing an algorithm was an engineering task incorporating tools and design space available in DNA nanotechnology.

4.1 A simple cargo-sorting algorithm

Our aim is to implement a DNA robot to sort randomly distributed molecular objects (cargos) into respective individual piles (goals) on a DNA nanostructure.

Controlling complex biochemical environments is challenging. Controlling the behavior of molecular robot programmed to perform a mechanical task at the nanoscale in such an environment is quite demanding. By designing a simple algorithm and modular building blocks, we believe that our robot system can accomplish cargo sorting of randomly distributed cargo on a nanoscale surface.

A general algorithm calls for a robot capable of systematic exploration of space. If the robot bumps into a cargo, it picks up the cargo. The robot , by identifying the type of cargo, chooses a specific path to the goal location based on the cargo type. On reaching the goal, the robot needs to recognize the goal location to drop off the cargo. However this algorithm is quite complex as the robot needs to be intelligent enough to distinguish between cargoes and goals, have a memory module tracking its path and an additional module for choosing distinct paths to goals based on the cargo type, requiring a complex signal classification system embedded in the robot.

We consider another algorithm where the robot architecture is simpler, it performs fewer functions and still sorts successfully. The robot capable of performing random walk continuously explores the space around. If it bumps into a cargo, it picks up the cargo. The robot then continues its exploration until it reaches the designated goal location, and drops off the cargo at its goal. The robot repeats this process until all the cargos have been sorted.

The function of a single robot sorting a single cargo type is emphasized in the algorithm (fig. 4.1b) and may be applicable to multiple sorting robots in parallel for sorting multiple cargo types. We introduce a cargo design that encodes its type as well as the identity of its designated goal. Hence, the complexity of the robot remains unchanged for sorting multiple cargo types. Increase in the number of cargoes need not require an increase in the number of robots. The complexity of the cargo and goal molecules remain unchanged as long as the there is sufficient specificity in the recognition between them. A robot is capable of performing its task iteratively until all cargo have been sorted. The algorithm can thus be expanded easily to sort multiple cargo types using the same robot. Multiple copies of the robot

introduced in the system may speed up the process (limited by possible crowding) as they work in parallel performing a common task.

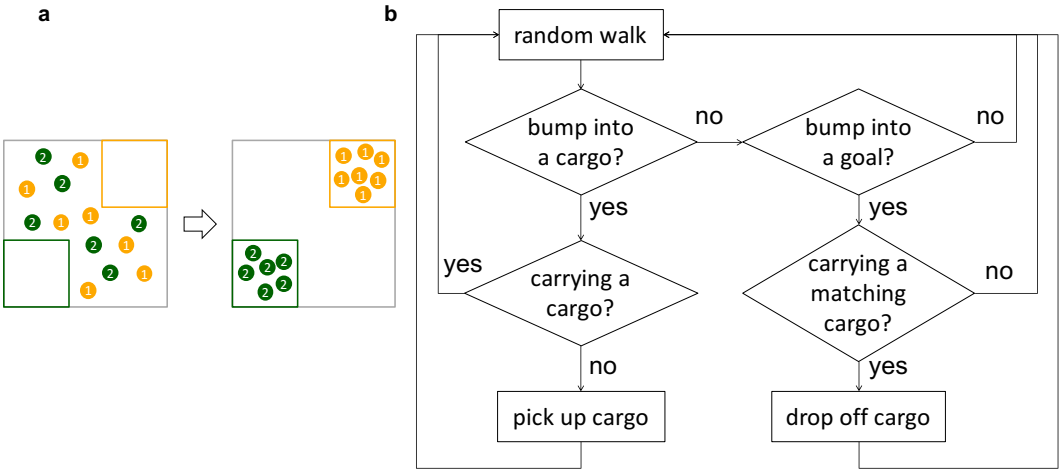

Figure 4.1: **Algorithm for cargo-sorting. a,** Schematic representation of randomly distributed cargo molecules of two different types sorted to their determined destinations. **b,** A simple algorithm for implementing a cargo-sorting robot.

4.2 Molecular implementation of the algorithm

This section describes implementing cargo sorting system using DNA-based molecular components. Using schematics, domain level descriptions and AFM characterization, we hope to paint a picture of the robotic system designed to implement cargo-sorting.

The cargo-sorting robot is a single stranded DNA (53 nt) and addition of track locations covering a larger area of the nanostructure and addition of cargo and goal locations changes the system previously used for random-walking to a cargo-sorting system. The robot has a leg and two *foot* domains for walking and a *hand* and *arm* domain for picking up and dropping off cargo. Figure 4.2a is a schematic representation of the distribution of two different types of cargos and their respective goals. The scheme only shows the the top-layer of the double layer DNA origami (Appendix, fig. 4.14). Staple ends on either layer of the double layer origami can be extended to form tracks and localize molecules. However, the staple design of the double-layer allows for one layer to have more possibilities of extensions than the other. We call this the top-layer. The top-layer has 95 3' staple extensions, of which 6 locations are used to localize cargo molecules, 8 locations are used to localize goal molecules, 1 location to localize the robot allowing 70 locations to be used as tracks. The edge staples have hairpin motifs used to reduce stacking interactions

between nanostructures, and hence are not part of the cargo-sorting test ground.

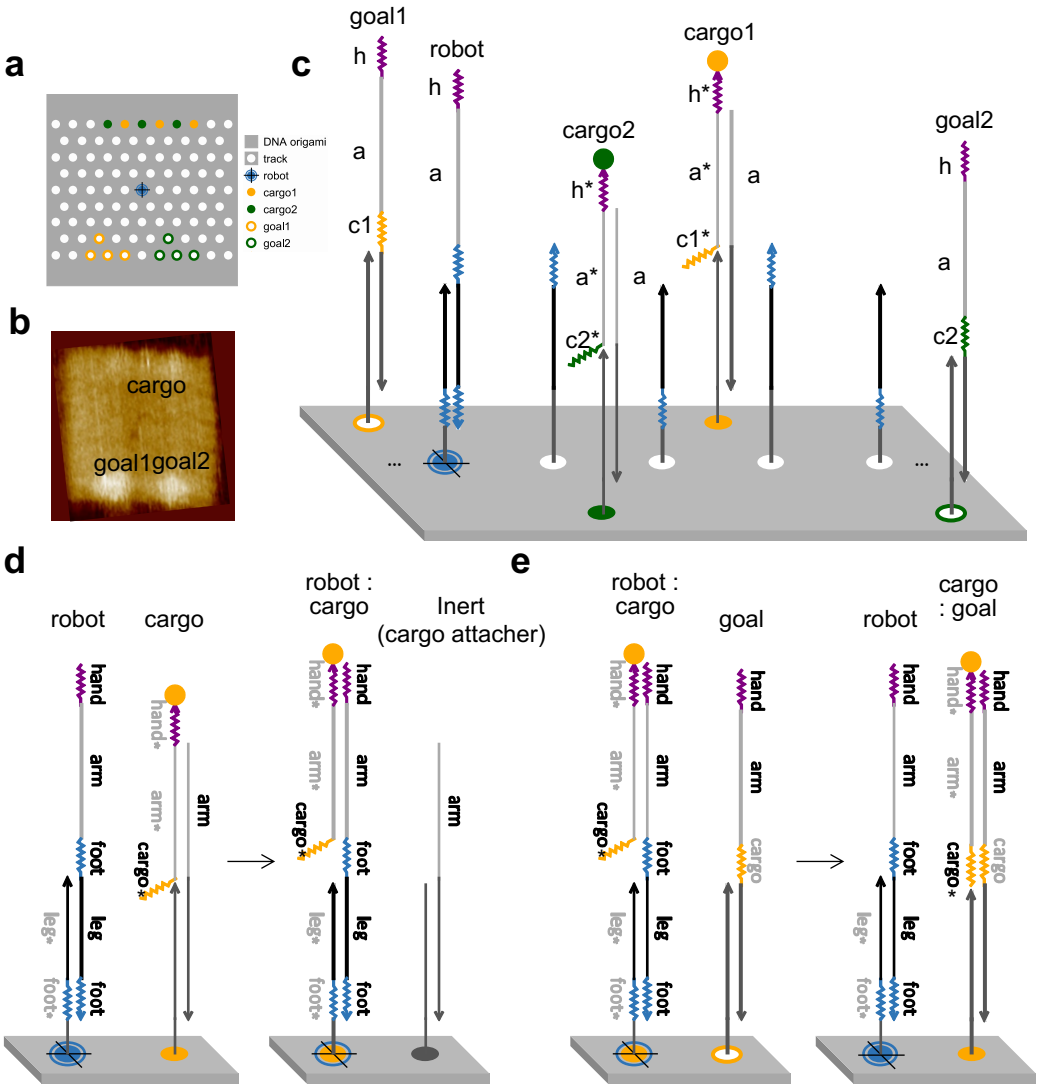

Figure 4.2: **Mechanism of cargo-sorting.** **a,** Schematic representation of cargo-sorting playground on DNA origami. Two different types of cargos and their respective goals are distributed on the opposite edges of the origami and the robot is fixed in the center at start. **b,** AFM of the DNA origami showing cargo and the two different goals along the opposite edges of the structure. **c,** Domain level details of the two types of cargo and goals shown bound to the 3' extensions of the origami staples. **d,** Basic mechanism of robot picking up cargo via the hand toehold from the cargo attacher and **e,** the basic mechanism of cargo drop off by the robot at the goal via the cargo toehold in an irreversible strand-displacement reaction.

The two types of cargo molecules are distributed interspersed with each other, along the top edge of the origami (fig. 4.2a). The corresponding types of goal molecules are distributed on the opposite edge of the origami from the cargo and are separated

into individual piles as shown. While there are three cargo molecules of each type, there are four goal locations for each cargo type. The goal redundancy was included to ensure each cargo molecule has a goal, and that lack of a goal position (maybe due to lack of a staple incorporation into the origami) would not be the reason for incomplete sorting. The robot starting position was fixed symmetrically between the cargo and goal locations. The white dots on the schematic are 3' staple extensions forming track1 and track2 as defined in the random walk system and cover the 2-dimensional origami surface as opposed to a linear track in the random walk system. The edge staples are omitted in the schematic to only show usable tracks.

Figure 4.2b is the AFM image of the cargo-sorting playground. While depositing the origami sample on a mica substrate for imaging, the structures can land face-down (with extensions facing mica) or face-up (with extensions facing solution, away from mica). Structures that land face-down have staple extensions trapped between the mica and the origami. When imaging such structures, the DNA extensions produce higher contrast compared to the nanostructure. Structures that land face-up have staple extensions facing the solution resulting in some mobility of the extensions, thus may not register as high contrast as the trapped extensions. The AFM in fig. 4.2b shows origami lying "face-down" on the mica substrate. This particular DNA origami sample was annealed without tracks so the position of cargo and goals could be identified. The schematic in fig. 4.2a shows cargo locations distributed along the top-edge of the origami, with the first cargo location 12 nm from right edge of the origami, and the last cargo location 9 nm from the left-edge. The asymmetric distribution of cargo along the top-edge serves as a label to identify the right and left edges of the nanostructure while analyzing AFM images. Goal1 location is on the left bottom edge and goal2 on the right bottom edge of the origami.

Figure 4.2c is a domain level description of the cargo-sorting system. Each track has a *foot* (*f*) as the toehold domain, and *leg* (*l*) as the branch migration domain. The robot moves from track1 to track2 reversibly using the mechanism previously described (fig. 3.2). The DNA sequence of the cargo-sorting robot is identical to the random-walking robot, without the quencher molecule on the 3' end. The domains used for triggering the random-walking robot (*hand (h),arm (a)*) are also used for cargo-pickup. The h^* (initiation toehold) and a^* (branch migration) domain on both cargo-types are identical and hence indistinguishable by the robot. The two types of cargo have two spectrally different fluorophores on the 3' end. Since the fluorophores are chemically different molecules, they may influence cargo-pickup.

At the domain level, the cargos are differentiated by the *cargo1** (*c1**) and *cargo2** (*c2**) domains on the 5' ends. These domains encoded in the cargo help identify the corresponding goal with its complementary domain. Both types of goal molecules have the *h*, *a* domains but differ by having *c1* domain for goal1 (complementary to $cargo1*$) and *c2* domain for goal2 (complementary to $cargo2*$). The two domains act as identifiers for ensuring desired cargo-goal interaction.

The *hand* (*h*) toehold domain on the robot initiates strand-displacement reaction that picks up cargo (fig. 4.2d). The cargo binds to the robot via the *hand* and *arm* complimentary domains in an irreversible strand displacement reaction. The cargo-attachment site is left inert after transfer of cargo. In principle, the cargo should not interfere with the robot walk as the domains required for random walk are independent of the domains carrying the cargo molecule. Robot carrying a cargo molecule cannot pick up another until the cargo being carried is dropped off at the goal. The cargo dropping off mechanism is shown in fig. 4.2e. During its random walk exploration when the cargo carrying robot bumps into a goal location, the cargo molecule on the robot identifies the goal via the *cargo1 (c1)* or *cargo2 (c2)* domain and binds irreversibly to its corresponding goal using complementary domain. After the cargo is dropped off, the robot continues its exploration and is free up to pick up another cargo. There is minimal interaction of robot and goal as they do not have complementary domains.

4.3 Demonstration of the robot picking up cargos

To implement a complex task like cargo-sorting, it is efficient to break down the main task to smaller ones and build the system bottom-up. This section demonstrates cargo-pickup by the robot. It is one of the major modules for our system. We experimentally demonstrate that the robot is unbiased in picking up both types of cargo.

The two distinct cargo types are tagged with spectrally different fluorophores. The two cargo molecules are localized on the origami surface via corresponding cargo-attachment strands. The cargo-attachment strands are hybridized to the 3' origami staple extensions using 15 nt domains (fig. 4.3). The 15 nt domains are different in DNA sequence, thus ensuring unique binding sites for cargo-1 and cargo-2. In order to observe the picking up event, we designed the cargo-attachment strands to have quenchers on the 5' end. When the cargo with flurophore on 3' end is bound to the attachment strand with quencher on 5' end, the fluorophore is in a quenched

state. One inactive robot per origami is also localized on the origami. All goal locations were inert in this setup, meaning the origami was annealed with staples without extensions for the goal location.

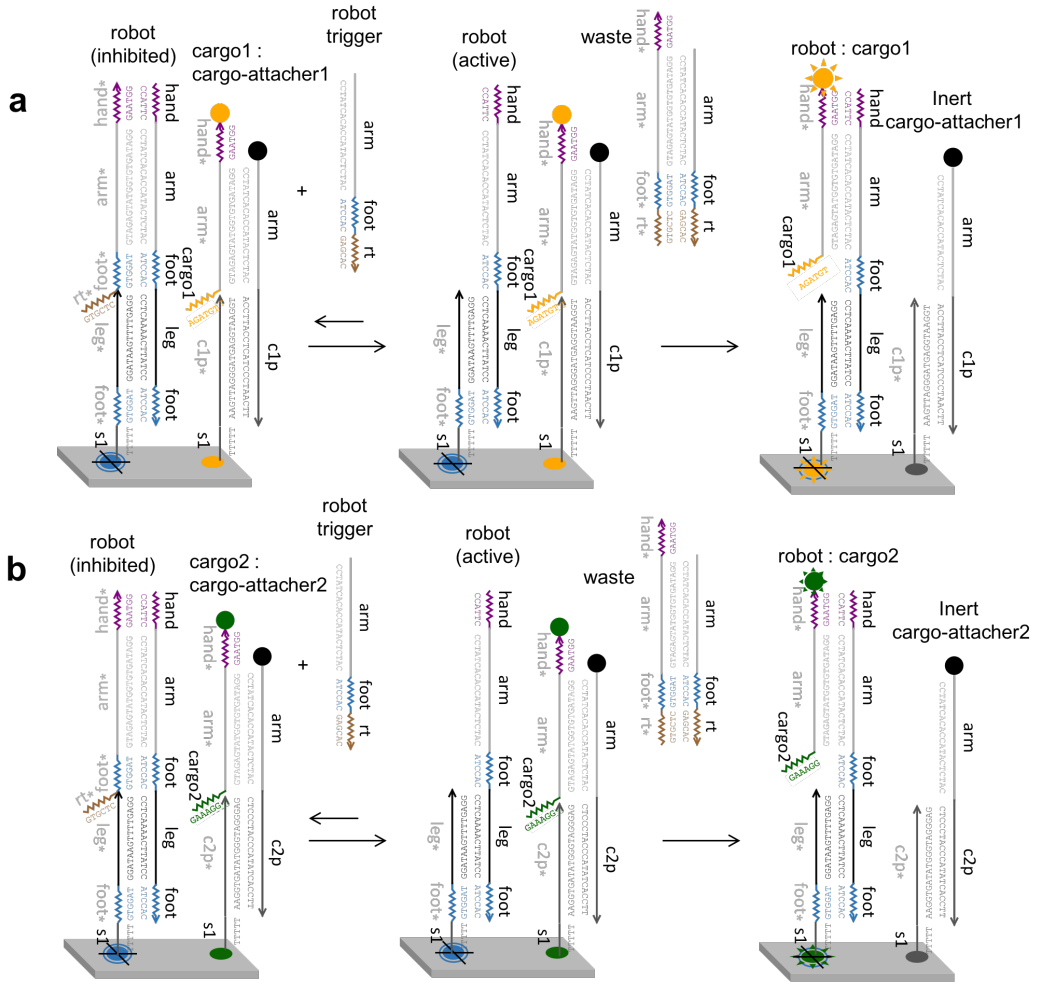

Figure 4.3: **Domain level representation for cargo pickup. a,** Basic mechanism of activating the robot using a robot trigger. The active robot then picks up cargo1 via toehold *hand* **b,** and similarly cargo2 via toehold *hand*, separating the cargo from the quencher molecule on the attacher strand.

Figure 4.3a is a domain level diagram showing an inactive robot and cargo1-F in a quenched state bound to its attachment strand with a quencher. Once the robot trigger is added, *rt* toehold binds to the robot inhibitor strand and strand-displaces it from the robot. The triggering reaction is a reversible strand-displacement reaction as the *hand* toehold is still available on the "waste" complex to re-initiate the robot inhibition reaction. Addition of a large excess of trigger strands (20x trigger strands to 1x robot), ensures the reaction is forward biased. The active robot uses *hand* toe-

hold domain to initiate strand-displacement reaction separating the cargo1-F from its attachment strand. The cargo pickup is an irreversible strand-displacement reaction via a 6 nt toehold. Once the robot and cargo are bound, the attachment strand is left inert. Figure 4.3b shows the domain-level design of cargo2-F pickup by the robot. The mechanism is the same as cargo1-F with the *hand* acting as the initiation toehold for the pickup reaction.

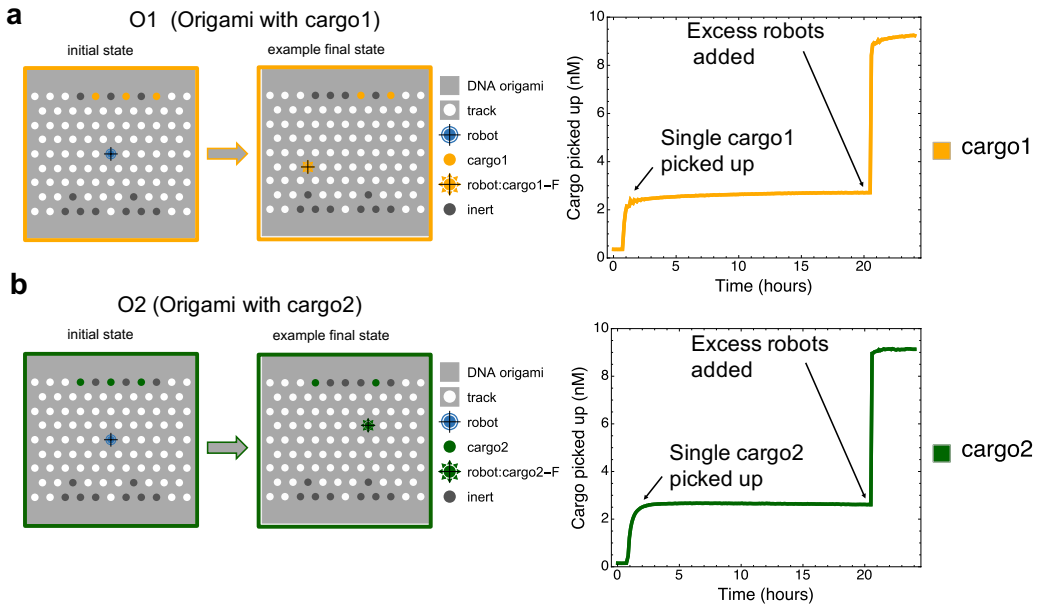

Figure 4.4: **Demonstration of cargo pickup. a,** Layout of position of cargo1, cargo2 and the robot on a schematic of the DNA origami. **b,** Fluorescence kinetics data of robot picking up cargo1 and cargo2 performed as separate experiments. Each origami has a single robot that picks up a single cargo leaving behind 2 quenched cargoes per origami.

Figure 4.4a shows the layout of quenched cargo1-F on the origami. Goal positions are inert. The robot is fixed at the center, initially inactivated. The robot is stationary and inactive, as the domains necessary for walking and cargo-pickup are inhibited. Once activated, the robot can explore the entire testing ground of the origami surface and can pick-up cargo at any location. When the robot reaches a cargo molecule, the cargo gets picked-up and is no longer in the quenched state (as indicated by the star-like fluorescence signal on the robot). Each origami carries a single copy of the robot, and in the absence of goal locations, picked-up cargo cannot be dropped-off and hence remains on the robot. In this setup, the robot is incapable of performing further iterations of pick-up and hence two of the three available cargo on each origami remain quenched as seen in the schematic.

Figure 4.4a also shows the fluorescence kinetics data for cargo1-F being picked up by the robot after being activated by the trigger molecule. Y-axis represents cargo1-F concentration. As seen in the schematic, each origami carries three copies of cargo1-F and thus the total cargo concentration is 9 nM. The kinetics data shows three stages of equilibrium. The first stage is with all copies of cargo1-F quenched while bound to its attachment strand. After establishing the quenched state fluorescence of cargo1-F for ~1hr, 20x (60 nM) of robot trigger was added to activate the robot.

Once the robot is activated, it begins its random walk exploration to pickup cargo1-F which is seen as the fluorescence signal increase until an equilibrium is reached at ~3 nM. The increase in fluorescence is due to the fluorophore on the cargo1-F molecule separating from the quencher on its attachment strand. With the absence of goal molecules, the robot and cargo1-F remain bound and hence 2x of cargo1-F molecules remain at their initial location, in a quenched state (hence only increasing fluorescence signal to 3 nM, instead of 9 nM). A large excess (200x) of free-floating active robots was added at approx 20 hr time point to unbind remaining cargo from their attachment strands, resulting in the increase in signal to 9 nM. The final completion level is helpful in estimating the fraction of cargo that was picked-up by the random walking robot and confirming the presence of roughly three copies of cargo molecules on each origami.

The above test was repeated to observe pick-up reaction for cargo2-F (fig. 4.4b). The schematic shows the location of cargo2-F initially bound to its attachment site with a quencher molecule and an inactive robot. When activated, the robot performs its exploratory walk of the test ground and picks up a single copy of cargo2-F. The absence of goal molecules leaves the picked-up cargo2-F bound to the robot, leaving two copies of cargo2-F bound in their initial quenched state, evident from the fluorescence kinetic data.

The kinetics of pick-up of cargo1-F and cargo2-F show qualitative similarity in reaction rate. The overall pickup rate is similar in both cases for two reasons, the first where the mechanism of cargo pick-up being initiated via *hand* toehold and completed via *arm* domains for both cargo types. The second being the distance between the cargo in both cases, where the robot maybe within reachable distance from either cargo location. With these design considerations, we expected to see no bias in the robot performance for either case. Taking the total concentration of the cargos from the data into consideration, we observe 30.54% of cargo1-F and 29.1%

of cargo2-F being picked-up by the robot in the two separate tests conducted, thus demonstrating unbiased pick-up mechanism by the robot.

4.4 Inhibition and activation mechanism of cargo destinations

A brief experimental protocol for performing cargo-sorting experiments with multiple steps is illustrated in fig. 4.5. Additional details of each step can be found in the Methods section.

The origami is annealed and gel purified with localized cargo molecules. Inhibited goal and robot complexes are localized after gel purification as shown in Step 3 and 5 of fig. 4.5. The goal and robot have domains involved in walking and cargo pick-up (*hand,arm,foot* for robot) and drop-off (*hand,arm,c1/c2* for the goals) and need to be protected while they localize on the origami. The protection of these domains is imperative to prevent cargo-robot and cargo-goal interaction in solution. Excess unbound molecules are removed from solution using spin-filtration method (Step 4).

The robot and goals are activated by adding large excess of trigger molecules to the sample before kinetics data is collected (Step 6, fig. 4.5). The robot is inactivated using an inhibitor strand complementary to domains *hand, arm, foot* so that the robot is inhibited from walking or picking up cargo. The inhibitor strand has an additional toehold domain, *rt*. The trigger strand used to activate the robot uses complimentary *rt** toehold to initiate de-protection of the robot strand. The *hand, arm* domains are used for cargo pick-up by the robot. A trigger strand with *hand,arm* domains can also pick-up cargo, hence interfering with the pick-up function of the robot. Hence, the trigger strand by design lacks *hand* domain. The robot-trigger waste molecule produces an exposed *hand* domain which could re-initiate deactivation of robot temporarily, making the activation reaction reversible. Adding large excess of trigger strands will bias this reaction forward, hence favoring activation of robot.

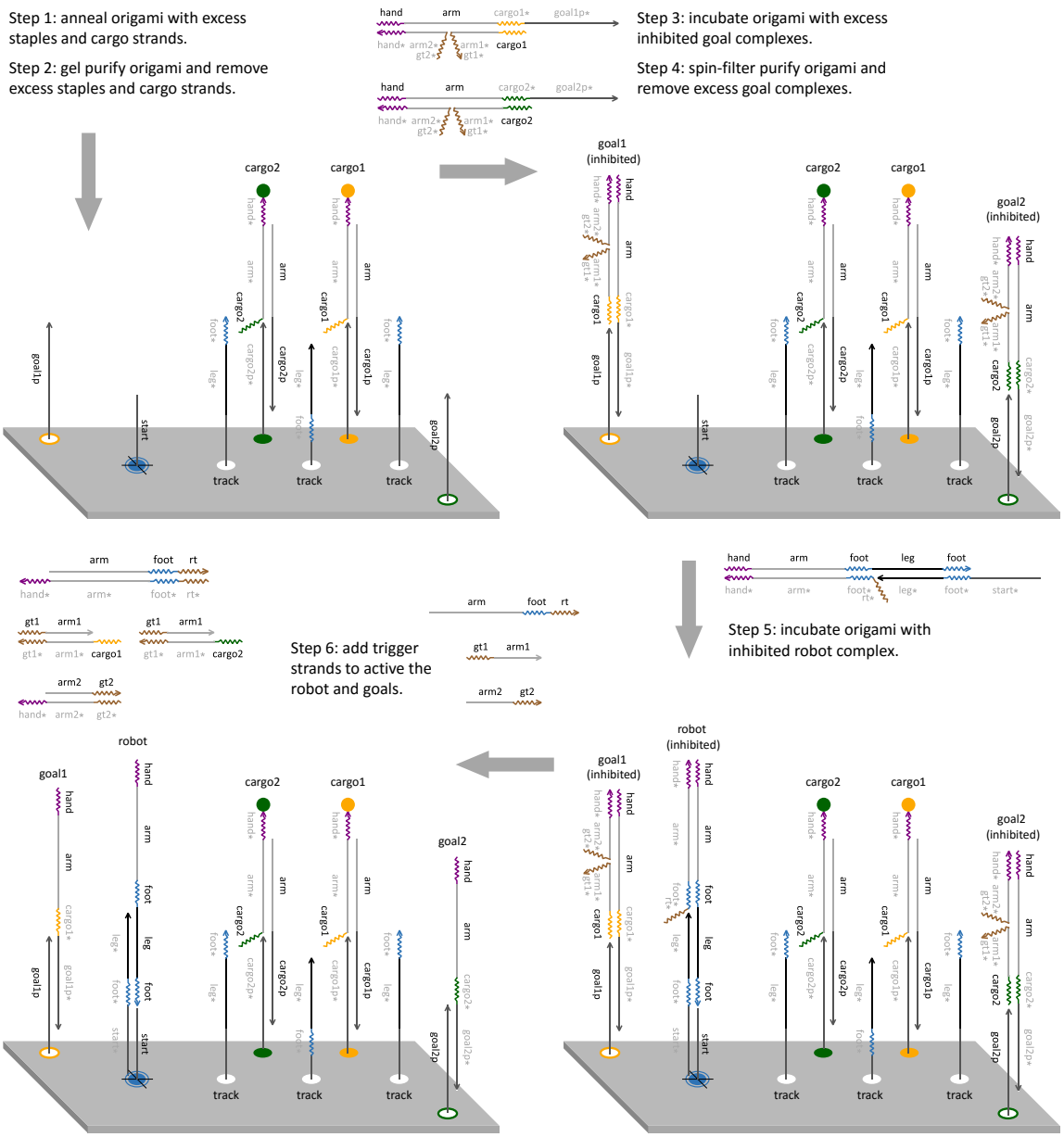

Figure 4.5: **Experimental protocol for cargo-sort sample preparation.** Step 1: anneal origami with a 10-fold excess of the regular, track, robot start, cargo and goal staples, 11-fold excess of the cargo attacher strands, and 12-fold excess of the cargo strands. Step 2: gel purify origami to remove malformed structures, and excess staples and cargo strands. Step 3: incubate origami with an approximately 2-fold excess of gel-purified inhibited goal complexes, assuming a 50% yield from gel purification. Step 4: spin-filter purify origami to remove excess goal complexes. Step 5: incubate origami with gel-purified inhibited robot complex at a 1 to 1 stoichiometry, after measuring the origami concentration using fluorescence spectroscopy. Step 6: add a 20-fold excess of trigger strands to active the robot and goals. More details see materials and methods.

Like the robot, goal molecules in the cargo-sorting setup also need to be inactive at the start of the experiment to ensure complete control over the starting of the experiment, and to prevent cargo-goal interaction while incubating to localize goals on origami (Step 3). Domains *hand, arm* are used for cargo-pickup by the robot while also used by the goal for cargo drop-off (in addition to *c1/c2*). Hence both robot and goal use the same domains complementary to cargo.

Hence, the goal / robot trigger molecules added in solution can be complementary to more than one type of molecule on the surface (for example, a goal trigger with *hand* domain can also pick up cargo), it is important to protect relevant domains before data collection. Because several criteria had to be met simultaneously, it was challenging to design inhibitor strands for the goal molecule.

- Pickup (*h*), branch migration (*arm*) and drop-off (*c1* for goal1, *c2* for goal2) domains need to be protected to ensure minimal interaction with cargo on the origami surface during incubation time required to localize the goal molecules on origami (Step 3 in fig. 4.5).
- Having a common single inhibitor for both goal1 and goal2 would not be possible because of domains *c1* and *c2* differentiating them.
- The goal trigger cannot contain both *h*,*a* or both *a* and *cargo* (*c1* for goal1 or *c2* for goal2) . Addition of a toe-hold at either the 5' or 3' end of the goal inhibitor strand would need the trigger strand to contain either pair of these domains leading to spurious interaction between cargo and goal trigger.

To meet the above criteria, we designed a two-strand inhibitor system for the goals. As seen in fig. 4.6 each goal has 2 inhibitor strands, one covering the *cargo* and half of the *arm* domain, i.e, *arm1*. Another strand covering the *hand* domain and the other half of the *arm* domain, i.e, *arm2*. Each inhibitor strand has a different toehold *gt1* and *gt2* to initiate goal activation. Two different trigger strands are necessary for each goal activation. As both goal1 and goal2 use *hand* and *arm1* domains, a common trigger strand with *gt1* is used forming one type of goal-trigger waste. Goal1 contains *cargo1* and goal2 contains *cargo2* domains that need protecting, requiring two different protection strands for goal1 and goal2. However, a single trigger strand with *gt2* and *arm2* de-protects both goals, leaving *c1* and *c2* domains unprotected in the generated goal-trigger waste. Hence there are three waste molecules generated during activation of two goals (fig. 4.6, lower schematic).

The robot activation mechanism shown in fig. 4.6 is discussed previously (fig. 3.2d). The *c1*, *c2* and *hand* domains on the goal and robot trigger waste molecules can interfere with cargo sorting temporarily. The solution is to add large excess of trigger molecules to bias activation reaction.

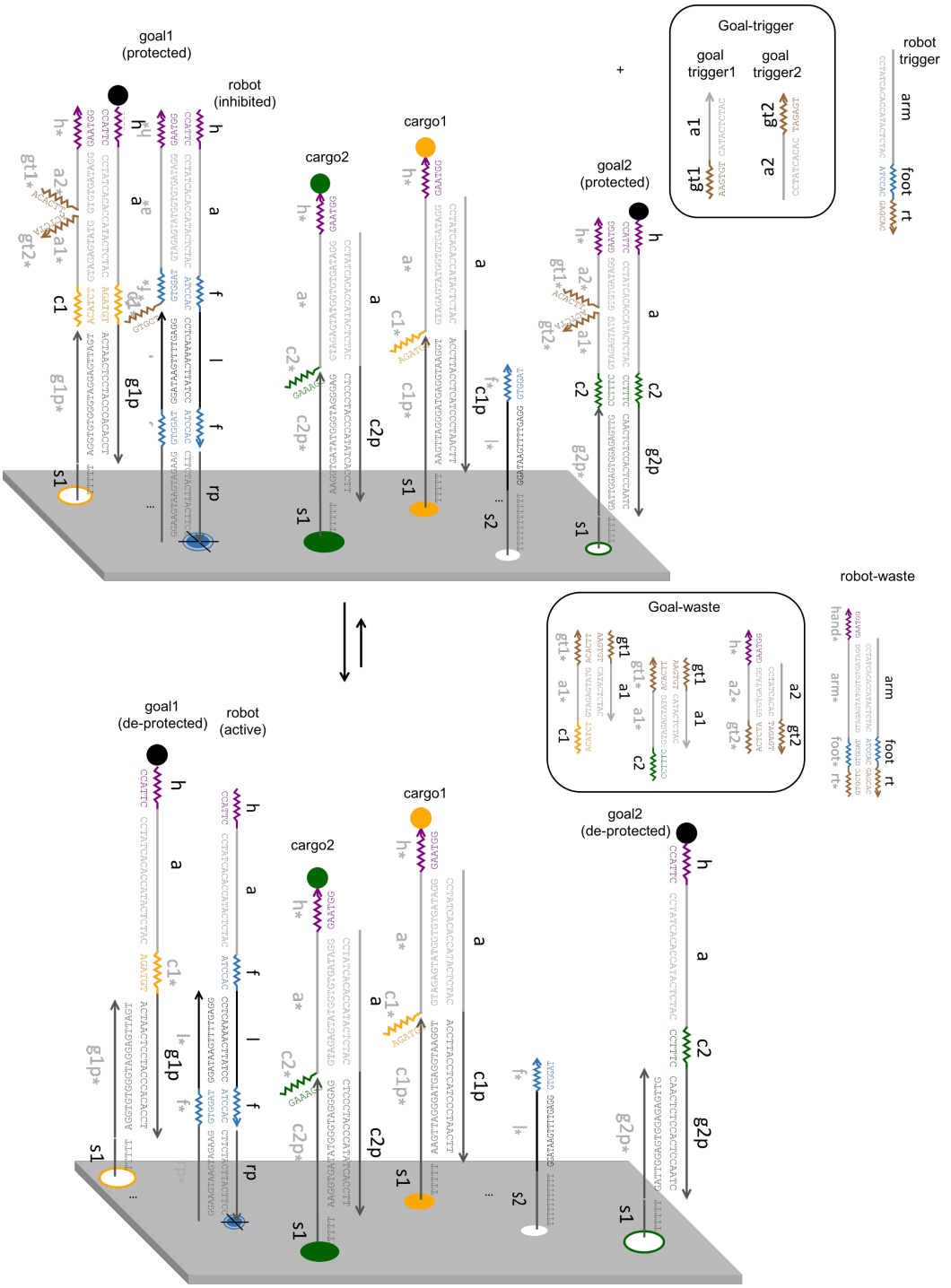

Figure 4.6: **Robot and goal deprotection scheme.** All domains on the robot are protected by a single strand with toehold *rt* and the goals are protected by two strands with toeholds *gt1* and *gt2*. Addition of robot trigger strand and the goal trigger strands deprotects the robot and goals respectively in a reversible reaction. The reaction is forward biased by the large excess of the trigger strands in solution.

4.5 Negative control for cargo-sorting without a robot

The robot is designed to act as a catalyst in the cargo-sorting system, hence there is little to no interaction between localized cargo and goals without a robot. The schematic in fig. 4.7a shows cargo1-F and cargo2-F localized interspersed along one edge of the origami on their respective attachment strands (without quenchers). Goal1-Q and goal2-Q are localized on the opposite edge of the origami (with quenchers). Figure 4.7b shows the fluorescence kinetics trajectory for cargo-goal interaction without robot. The start of the experiment has cargo-F on their respective attachment strands and both goal-Q inhibited. Once the goals are activated, the fluorescence trajectories for both cargo1-F and cargo2-F are monitored.

The data show an insignificant decrease in signal for both cargo1-F and cargo2-F. Intra-origami and inter-origami interactions between the cargo and goals are two possible reasons for the decrease in signal. But intra-origami interactions are unlikely, because in the previous experiments on random walk, we showed that there was no significant relationship between localized robot-goal spatial distance on double layer origami (fig. 3.3j).

Inter-origami interactions are the likely cause of signal decrease, where goal-Q of one origami can pick up cargo-F from another origami in close proximity. Inter-origami reaction is independent of cargo and goal type as both types have sufficient sequence similarity for such a spurious interaction. From the data, 8% of cargo1-F and 7.9% of cargo2-F have been directly picked up by the goal showing that inter-origami interactions (at a low origami concentration of 3 nM) is insignificant.

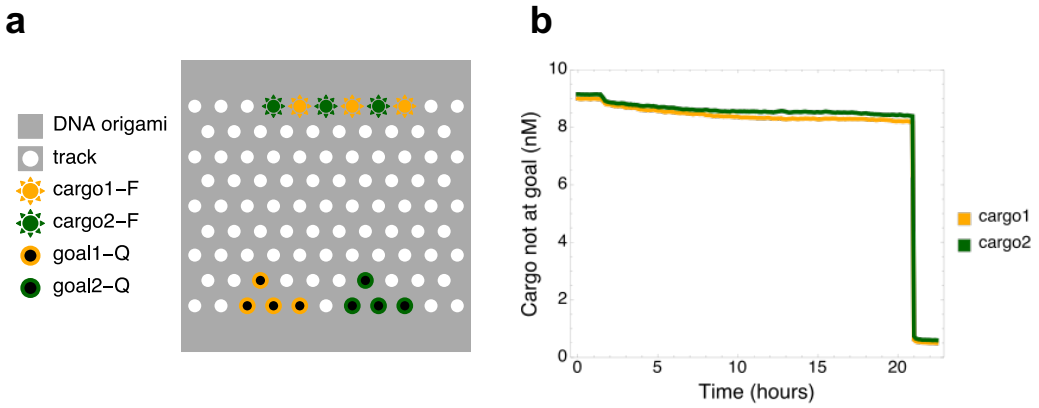

Figure 4.7: **Cargo-goal interaction without robot. a,** Schematic representation of DNA origami with the two different cargos and their respective goals on opposite edges but no robot. **b,** Comparing fluorescent kinetics data of the cargos picked up directly by the goals in an inter-origami interaction.

4.6 Demonstration of the robot sorting cargos

The previous section established that cargo-goal interaction on the origami without the robot is insignificant. This section investigates whether cargo-sorting is possible using the robot, i.e., if a single robot can sort multiple types of cargos and multiple cargos per type as designed. The robot uses *hand,arm* domains to pick up a cargo and drops it off at the goal having complimentary *cargo* domain (fig. 4.8a). Both cargo get dropped-off following the same mechanism and the *cargo* toehold distinguishes cargo1 and cargo2 from their respective goals. After the goals are activated, the waste molecules in solution containing the *hand* and *cargo* toehold can undergo reversible strand-displacement reaction to temporarily inhibit the goal again. A large excess (20x) of trigger strands in solution biases this reaction forward so as to minimize this occurrence.

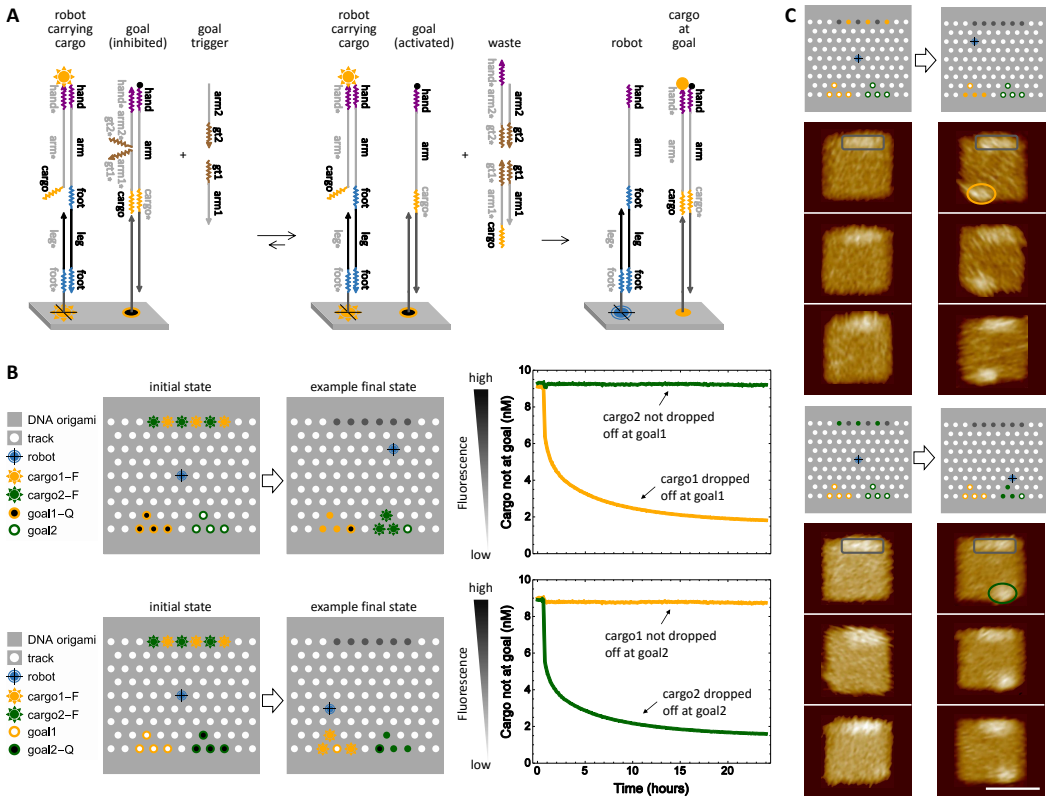

Figure 4.8: **Cargo-sorting with robot. a,** Domain level mechanism of goal activation and cargo drop off. **b,** Schematic representation of DNA origami with the two different fluorescent tagged cargos reaching their respective goals on opposite edges. **c,** AFM images confirming that cargo1 is dropped off at goal1 and cargo2 is dropped off at goal2. **d,** Comparing fluorescent kinetics data of the cargo1-F being dropped off at goal1-Q and cargo2-F being dropped off at goal2-Q by the robot.

The origami test ground for this experiment has cargo1-F and cargo2-F interspersed on one edge of the origami and goal1-Q and goal2-Q are localized on the opposite edge. A single robot is localized in the middle of the origami (fig. 4.8b). Initially, both robot and goals are inactive (fig. 4.6). Two sets of experiments were performed independently to monitor the drop-off of a single cargo type. In both experiments, cargo1 and cargo2 are tagged with spectrally different fluorophores. However, the first setup had quencher tagged goal1 and an untagged goal2. When cargo1-F is dropped-off at goal1-Q, we expect its fluorescence signal to decrease. Since goal2 is untagged, fluorescence signal from cargo2-F should remain steady. If cargo2-F get dropped-off at the wrong goal, i.e., at goal1-Q, the signal from cargo2-F will also decrease. This set-up monitors both cargo types for desired sorting of one type (cargo1-F dropped off at goal1-Q) and undesired/ spurious interaction for another (e.g. cargo2-F dropped-off at goal1-Q). The second experiment was repeated with both cargo1-F and cargo2-F tagged with fluorophores. Goal2-Q was tagged with a quencher and goal1 was untagged.

The fluorescence kinetics data in lower panel of fig. 4.8b, shows the fluorescence signal from cargo1-F channel decrease over time. This decrease in signal can only occur if the fluorophore on cargo1-F is quenched. And since in the first setup only goal1-Q is tagged with a quencher, we conclude that cargo1-F is dropped off at the desired goal location. The fluorescence signal from the cargo2-F channel remained constant over time implying that cargo2-F did not get dropped-off at goal1-Q, but got dropped-off at goal2 as designed. The cargo-goal interaction without robot (fig. 4.7) is undetectable in this setup. We think this may be because the desired sorting reaction kinetics are much faster than the undesired inter-origami interaction.

From the kinetics data in fig. 4.8b, 78.98% of cargo1-F have been sorted to goal1-Q. The addition of large excess of goal1-Q and goal2-Q in solution after 24 hrs quenched the remaining fluorophore tagged cargo helping us estimate the fraction of unsorted cargo. Failure to sort 20.02% cargo1-F could be due to multiple reasons. (1) Impurity/ synthesis errors in the robot or goal strands inhibiting the desired pick-up, drop-off or walking mechanism, (2) local malformation of origami such as missing track staples (missing one or two may not matter, but missing several could get the robot stuck at a location). (3) Origami is well formed, but could be missing a goal location (missing one may not matter, missing more than one would result in incomplete sorting) (4) some origami maybe malformed (agarose

gel electrophoresis may not be able to detect a few missing staples in origami that get lumped in the same band as target structures).

The second experiment had both cargo with fluorophore tagged but an untagged goal1 and quencher tagged goal2-Q. Both cargo1-F and cargo2-F are monitored. On observing the fluorescence kinetics data from fig. 4.8b, we conclude that cargo2-F was dropped off at goal2-Q and cargo1-F was not dropped-off at goal2-Q. Combining the results from the two separate experiments we conclude that a major fraction of the cargo is sorted correctly in both cases with cargo1-F rightly at goal1-Q and cargo2-F is rightly at goal2-Q.

Fluorescent kinetics experiments are done for observing bulk behavior of the system. Atomic force microscopy was used to observe the system at the single-molecule level. While fluorescence kinetics experiments tell us the interactions between molecules, AFM experiments directly tell us where the molecules are located on the origami surface. Figure 4.8c shows the schematic and AFM data for cargo1 drop-off. In our design, the asymmetrical distribution of cargo along the top-edge of the origami (as seen in schematic), was used to label the left and right edges of the nanostructure. The rightmost cargo molecule is ~12 nm from the right edge and the leftmost cargo molecule is ~9 nm from the left edge. The AFM image below the schematic confirms the asymmetrical distribution of the cargo molecules along the top edge is distinguishable. The asymmetrical pattern of cargo molecules tell if the origami landed right side up or upside down on mica, and the images were flipped if needed. With right side up images, goal1 is closer to the left edge of the origami with respect to the cargo distribution pattern and goal2 is closer to the right edge of the origami.

To individually verify the sorted position of the two cargo types, three copies of cargo1 were localized along the top edge of the origami with four copies of both goal1 and goal2 along the opposite edge and one copy of robot in the center. The goals and robot were activated using a large excess of trigger molecules and left undisturbed for 36 hrs (to match the end point of the bulk experiments performed before). In AFM imaging, it is difficult to distinguish double stranded DNA extensions from single stranded DNA extensions on the the origami, but much easier to distinguish between double stranded DNA extensions and no extensions. Therefore, to image the double-stranded goal locations with cargos, it is necessary to remove single-stranded tracks and partially double-stranded goal locations without cargos.

To remove partially double stranded goal locations without cargos, deblocking strands

were used (Step1 in fig. 4.9). The deblocking strands displace the partially double stranded goals without cargos from their attachment sites on the origami. To remove single-stranded extensions (tracks, unused goal attachment sites), the nanostructures were incubated with exonuclease-1 enzyme. The enzyme selectively digests single stranded DNA from its 3' end (Step 2 in fig. 4.9). The origami structures thus treated should have double stranded cargo-goals and partially double stranded cargo-attachment sites. Figure 4.8c shows the AFM images for both cargo1 and cargo2 at their respective locations. Some fraction of origami landed face-up on mica and the remaining were face-down in all scans. Manual inspection of origami that had landed face-down in multiple large scans showed cargo at the desired goal location (Appendix, fig. 4.15).

4.7 Demonstration of distinct cargo-sorting tasks in parallel

In the previous section we showed a DNA robot capable of sorting multiple copies of two different cargo types to their designated destinations. However, in each instance of the two experiments, all species in the test tube were designed to be identical copies with identical localized reactants. Theoretically, each copy of origami can be treated as an individual compartment, making it possible to use each scaffold as a test bed for different types of reactions. For implementing such a system experimentally, one needs to understand the interaction between different origami substrates, and if in reality, separate experiments can be compartmentalized on each origami substrate.

To that end, we designed a test system consisting of mixed populations of origami. Because of the electrostatic nature of the designed nanostructures, we believe well formed double-layer origami do not aggregate in a well-mixed solution. However, there are several components localized on the origami that can interfere with each other causing spurious/ undesired reactions.

For example, if two nanostructures come in close proximity to one another, the robot or the goal have the capability to pick up cargo from the other nanostructure via *hand,arm* domains that can freely interact with each other in solution, relatively far from the origami surface. A robot on one nanostructure may also "hop" to another. Robot hopping would occur while making a step and one of the *foot* domains is free. The robot will have to bind the *foot* domain on a nearby origami to hop which may result in one origami losing a robot and the other gaining two. Hopping may also result in the robot leaving the surface and escaping into solution in

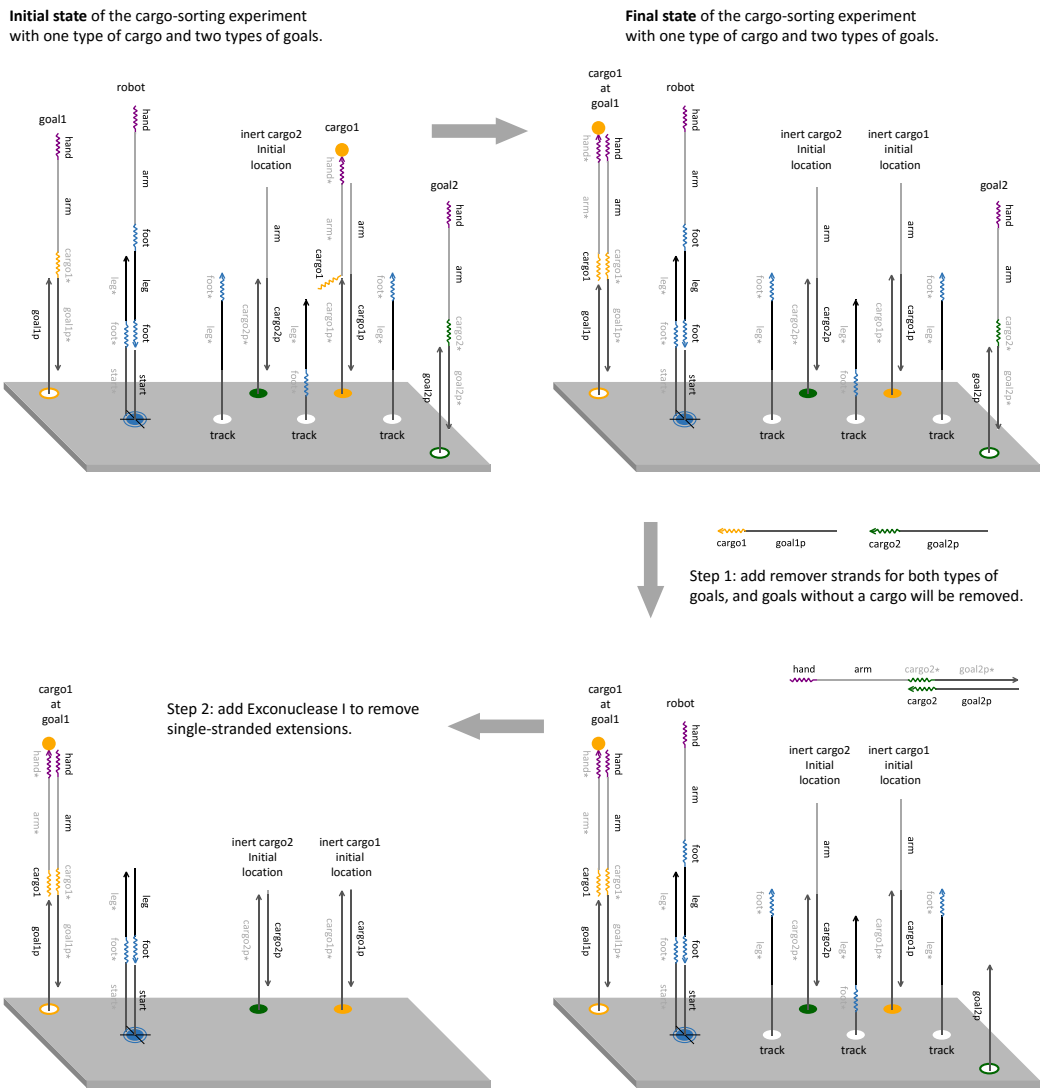

Figure 4.9: **Sample preparation for AFM imaging.** The asymmetry of cargo distribution is used as a reference for recognizing the orientation of origami in AFM images. Step 1: add a 20-fold excess of de-blocking strands for both types of goals to remove goals without a cargo. Step 2: add exonuclease-1 to digest single stranded extenstions and create a clean background ofor imaging the locations of cargos at goal.

which case, it may land on another origami. But the walking reaction occurs much closer to the surface, so we believe that robot-hopping may occur at a much lower probability than the cargo getting picked up by the robot or the goals on different origami.

Figure 4.10a, shows the schematic of two origami test-grounds for the robot, O1 and O2. O1 and O2 have both types of cargo, although O1 has fluorophore tagged cargo1-F and untagged cargo2. O2 has fluorophore tagged cargo2-F and untagged

cargo1. Both O1 and O2 has quencher tagged goal1-Q and goal2-Q. Only O1 has the robot. We have previously shown that the probability of the robot dropping cargo off at the wrong goal is minimal (fig. 4.8b,c). Therefore, we expect cargo1-F sorted to goal1-Q (fluoroscence signal decrease) and cargo2 sorted to goal2-Q (no fluoroscence signal) on O1 with the robot. O2, consisting of untagged cargo1 and fluorophore tagged cargo2-F does not have a robot. Therefore, even with quencher tagged goals on O2, there should be no sorting of either cargo. The fluorescence signal from mixture of O1 and O2 should hence only decrease in cargo1-F signal due to sorting.

O1 and O2 were prepared separately using the protocol outlined in fig. 4.5 with relevant reactants as mentioned above. O1 and O2 were later mixed in 1:1 ratio for the purposes of this experiment. Goal1-Q, goal2-Q and the robot in the mixture were activated after ~30min of initial signal monitoring. The fluorescence signal from cargo1-F channel decreases over time, implying that cargo1-F is being sorted to goal1-Q on O1 (fig. 4.10a). O2 having untagged cargo1 does not contribute to this signal decrease directly. As cargo2-F is not sorted on O2, the fluorescence signal from cargo2-F is not expected to decrease. However, we observed a trajectory in the cargo2-F channel with slow changing slope. O1 with untagged cargo2 should not contribute to this trajectory.

75.16% of cargo1-F are sorted to goal1-Q in the mixture containing O1, O2. Earlier bulk measurements from cargo-sorting where all origami were identical and consisting one copy of robot each (fig. 4.8b) resulted in 79.04% of cargo1-F sorted to goal1-Q. We see a 4.9% reduction in cargo1-F sorted when O1 and O2 are mixed. We also observe 12.1% of cargo2-F being quenched in O1, O2 mixture.

The experiment was repeated with the robot on O2. The data shows 74.07% of cargo2-F getting sorted to goal2-Q (fig. 4.10b). We also observe 14.09% of cargo1-F getting quenched. We consider the following reasons for understanding the interactions involved between the two populations of origami.

- The robot moves from one origami to another.
- In general, the cargo from one origami may be picked up by a robot on a different origami via *hand, arm* domains. In such a case, the cargo picked up from one origami may get dropped-off at a goal on a different origami.

- From the data in fig. 4.7b, we saw goal and cargo interaction without a robot being present. It is possible that some fraction of goal-Qs on one origami may pick up cargo-F on a different origami in an inter-origami interaction.

The experiments performed in this section has two populations of origami: only one with robot and one without robot. One type of cargo is fluorophore-tagged in each population. If a robot moves from O1 to O2, the type of desired cargo sorting on O1 will decrease because the same type of cargo on O2 does not have a fluorophore. A robot or goal on O1 picking up a cargo on O2 should not affect the completion level since we have four goals and three cargos per type per origami, an additional cargo would just occupy the extra goal location. A goal on O2 picking up a cargo on O1 should not affect the completion level either, because the goal on O2 also has a quencher and it is no different from the goal on O1. The observed type of undesired cargo sorting on O2 will increase if a robot moves from O1 to O2 and start sorting the cargos, or if a robot or goal on O1 picks up a fluorophore-labeled cargo on O2.

Inter-origami interactions as mentioned above can in principle be reduced by reducing the concentration of the origami. Intra-origami interactions are independent of the concentration of origami. Practically, instrument sensitivity sets the detection limit.

Parallelism

In the previous section, we showed that with small amount of inter-origami interaction, the robot is able to perform its sorting function on the designated origami. The experimental setup can be expanded by adding a robot on both O1 and O2 at the same time. Such a system can be used to implement parallel cargo-sorting on two separate origami substrates present in the same mixture.

Figure 4.11a shows a schematic of O1 and O2 origami species tested simultaneously in the same test tube. In order to differentiate between events occurring on O1 and O2, O1 was prepared with fluorophore tagged cargo1-F and quencher tagged goal1-Q and untagged cargo2, goal2. O2 was prepared with fluorophore tagged cargo2-F and quencher tagged goal2-Q and untagged cargo1, goal1. Both types of origami have a single robot initially located at the center. All goals and the robot are initially inactive. When activated with 20x of robot, goal1 and goal2 trigger strands, the sorting function is initiated independently on both populations as seen in fig. 4.11b.

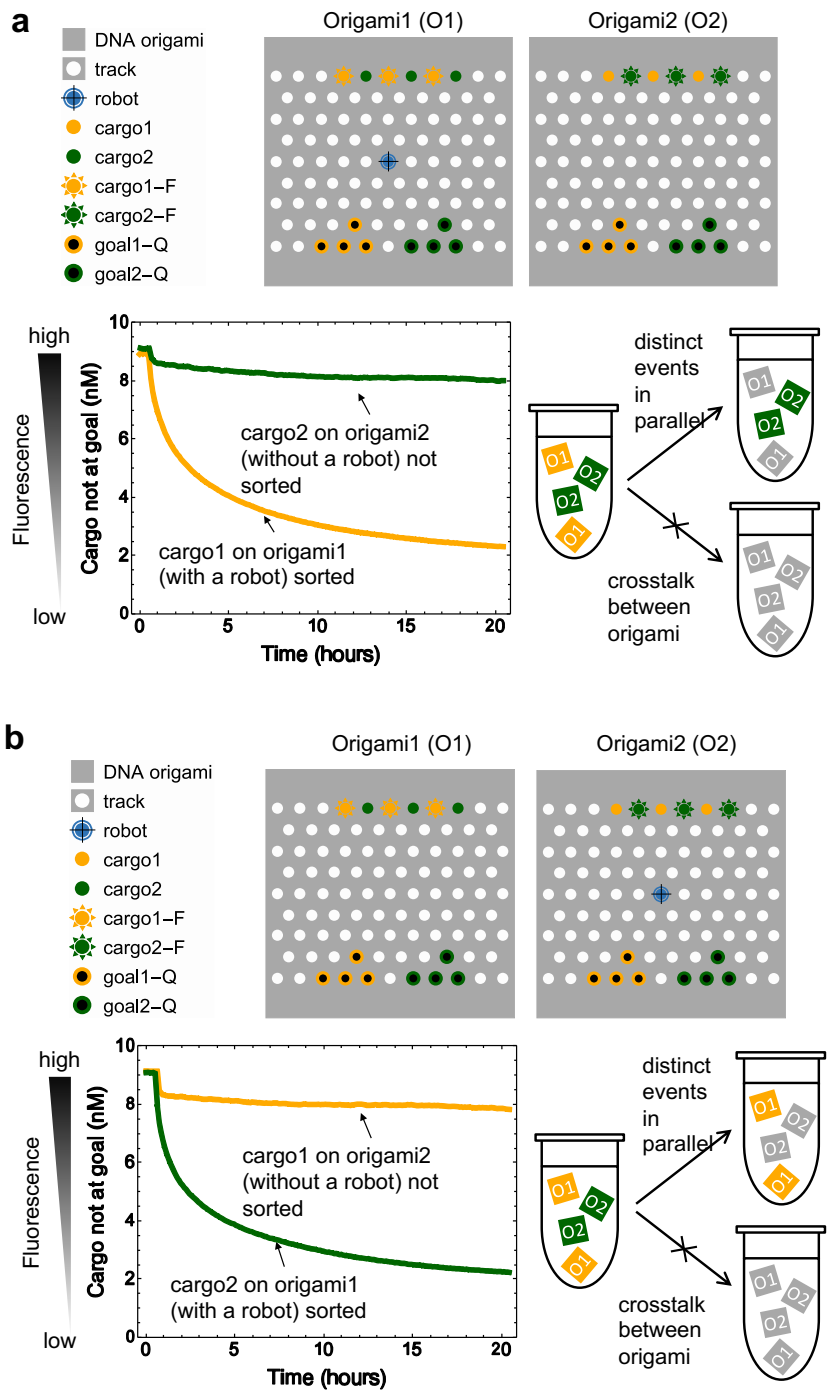

Figure 4.10: **Cargo-sorting in mixed populations of DNA origami.** Fluorescence kinetics experiment with two mixed populations of DNA origami, both having two types of cargos, two types of quenched goals and **a,** robot only on origami with cargo1-F and untagged cargo2 and **b,** robot only on origami with cargo2-F and untagged cargo1.

The data shows similar half-completion times for the sorting of both cargo types. The experiment was ended after 24hrs by adding a large excess of goal1-Q and

goal2-Q to the test tube to bind any unsorted cargo. The data shows that 62.29% cargo1-F was sorted while 64.49% of cargo2-F was sorted. We saw 75.16% cargo1-F and 74.07% of cargo2-F getting sorted in the previous mixed population setup (fig. 4.10a,b).

In the experiments performed in this section, there are two populations of origami: only one type of cargo is labeled with a fluorophore on each type of origami, and only the corresponding type of goal has a quencher. If a robot or goal on one type of origami picks up a fluorophore-labeled cargo on the other type of origami, there will be no signal change since the corresponding goal has no quencher. Because both types of origami already have a robot, an additional robot moved from another origami should not affect the completion level of the desired cargo sorting. Loosing a robot to another origami should occur much slower than the desired local cargo sorting reactions, and will not affect the completion level if the event takes place after the cargos are already sorted on the original origami.

We observed a 17% decrease in cargo1-F sorting and 12.9% decrease in cargo2-F sorting between data in section 4.7 and section 4.7. In principle, these results show that more than two distinct cargo-sorting tasks can take place simultaneously in one test tube. The decrease in sorted fraction of cargo may be the result of undesired inter-origami interactions and can be reduced by reducing the concentration of origami in solution.

Given that measuring low concentrations of origami is a limitation of the instrument, single molecule analysis of cargo-sorting may be performed using DNA-PAINT [10] which is a high-resolution microscopy technique involving the localization of individual origami on pre-treated glass slides hence allowing isolated origami to be measured thus reducing inter-origami interaction seen in solution.

4.8 Model

Cargo-sorting involves a complex system with multiple reactions necessary for successful sorting of cargo molecules to their designated goals. We developed a model that helps quantify the system by detailing some of the key reactions. We first define the coordinates of tracks that the robot uses to make a step. The robot takes a single step from location (x, y) to a neighboring location $(x*, y*)$, represented by the reaction,

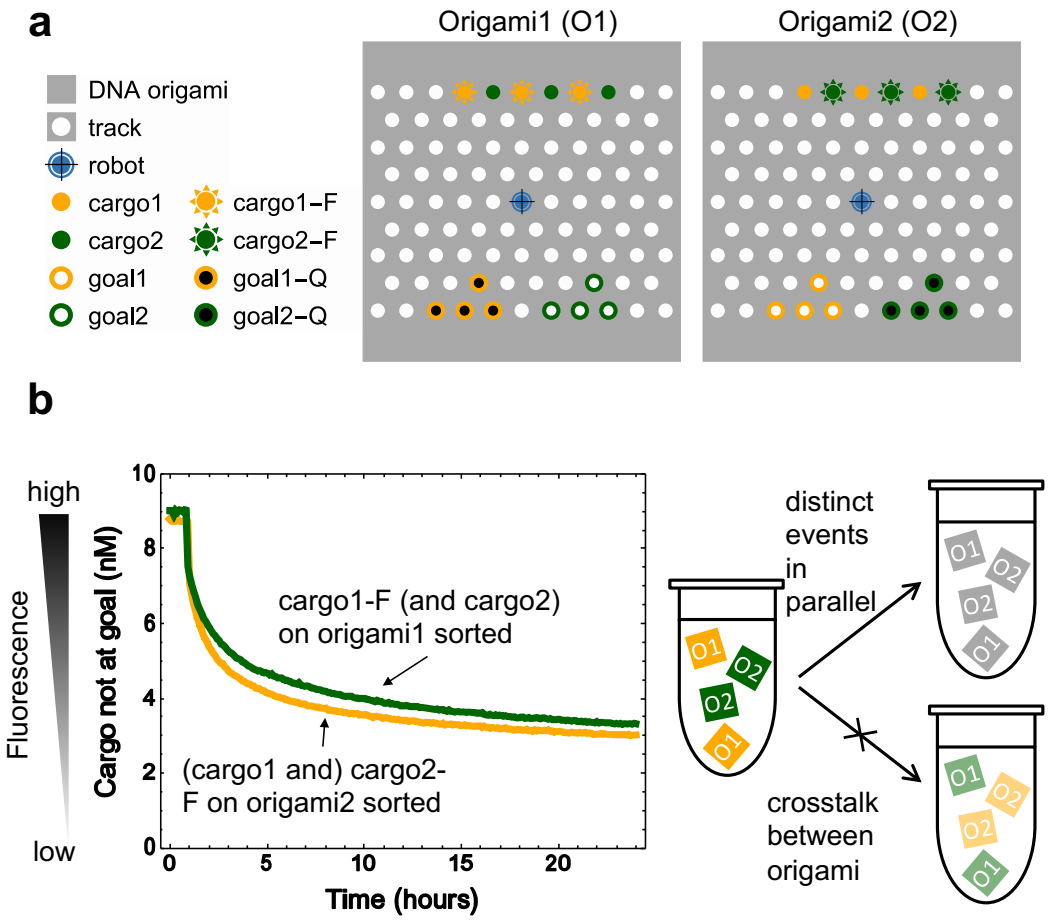

Figure 4.11: **Parallellism. a,** Schematic representation of two different origami present in the same test tube. Both origami have two types of cargos, two types of goals and a single robot each. Each type of origami has a fluorophore tagged cargo and a corresponding quencher tagged goal. The other type of cargo and goal are untagged. **b,** Fluorescence kinetics data with the afore mentioned mixed populations of DNA origami. The cartoon representation shows minimal cross talk while distinct events in parallel occur without restriction.

$$Robot_{x,y} \xrightarrow{k_w} Robot_{x*,y*} \tag{4.1}$$

$$\text{where } (x*, y*) = (x-1, y),\ (x+1, y),\ (x, y-1) \text{ or } (x, y+1)$$

The walking rate $k_w = 0.0035 s^{-1}$ was previously derived (eq. (3.2)). Equation (4.2) is the reaction of robot walking with cargo and is defined as,

$$Robot : Cargo^i_{x,y} \xrightarrow{k_w} Robot : Cargo^i_{x*,y*} \tag{4.2}$$

Cargo^i specifies a specific type of cargo (cargo1 or cargo2). A robot carrying cargo also walks at rate k_w. Because the cargo is carried by the robot and the walking domains *leg* and *foot* are unaffected by bound cargo domains (*hand, arm*); the rate of walking is not affected by the cargo being carried and hence occurs at the same rate as the random walk without cargo in eq. (4.1).

We then define the reactions used for cargo pick-up and drop-off by the robot;

If $d \leq d_{max}$,

$$Robot_{x1,y1} \,\&\, Cargo^{i}_{x2,y2} \xrightarrow{k_c \times \frac{d^2_{min}}{d^2}} Robot : Cargo^{i}_{x1,y1} \,\&\, Inert_{x2,y2} \tag{4.3}$$

$$Robot : Cargo^{i}_{x1,y1} \,\&\, Goal^{i}_{x2,y2} \xrightarrow{k_c \times \frac{d^2_{Min}}{d^2}} Robot_{x1,y1} \,\&\, Cargo^{i} : Goal^{i}_{x2,y2} \tag{4.4}$$

$k_c = 100 \times k_w$

Here k_c is the rate constant for pick-up and drop-off when cargo or goal is an immediate neighbor of the robot (i.e., $d = d_{min}$).

Robot picking up cargo is an irreversible reaction. The attachment strand that binds the cargo to the origami surface consists of the *arm* and *cargo-probe* domains. The *cargo-probe* domain anchors the attachment strand on the origami and the *arm* domain anchors the cargo on the attachment strand. Therefore once the cargo has been picked up, the *arm* domain alone, without any toehold domains, cannot interact with other molecules in the system and is labeled "inert" (fig. 4.3). The cargo drop-off reaction at the goal is initiated by the *cargo* toehold. Once the cargo drop-off reaction is complete, there are no available toeholds for the cargo to bind to the robot, practically making this reaction also irreversible.

Previously, we defined two immediate neighboring locations as (x, y) and $(x*, y*)$. For cargo pick-up and drop-off to occur, the robot may or may not be an immediate neighbor to the cargo (or goal). If the robot and cargo (or goal) are immediate

neighbors, the Euclidean distance d_{min} is calculated using the distance between the closest staple extensions on the twist-corrected DNA origami. There are 16 base pairs between two staple extension locations and twist correction is applied by deleting 1 bp every 3 staple columns.

$$d_{min} = 0.34 \text{ nm / bp} \times (16 \times 3 - 1)/3 \text{ bp} = 5.32 \text{ nm} \tag{4.5}$$

The robot can pick-up a cargo from (or drop off a cargo to) a location that is not its immediate neighbor, if the distance is within the defined reachable distance (d_{max}). We assume that the maximum rates for picking up and dropping off are when the cargo or goal is an immediate neighbor of the robot, and for any further distance, the rate decreases quadratically with the distance.

The reachable distance is calculated by using the total number of base pairs in the double-stranded foot, leg and cargo (or goal) attacher domains (41 bp), and the total number of nucleotides in the single-stranded foot and linker domains (16 nt). The distance between base pairs is established as 0.34 nm, stabilized by the base-pairing and the base-stacking. The distance between single bases is between 0.5 nm as the single strand DNA do not have the same stabilizing forces as double stranded DNA.

$$d_{max} = (41 \text{ bp} \times 0.34 \text{ nm / bp}) \ + (16 \text{ nt} \times 0.5 \text{ nm / nt}) = 21.94 \text{ nm} \tag{4.6}$$

With k_c being 100 times the rate of walking k_w, our model semi-quantitatively reproduced experimental data (fig. 4.12).

A pick-up/ drop-off rate higher than the rate of walking for the robot is possible because both interactions between the robot and a cargo and between a cargo and its goal occur further away from the origami surface. In comparison, the walking reaction occurs much closer to the origami surface. Our understanding of the rates involved in these two reactions follows from the previously described coarse-grained DNA model study conducted by Šulc et al. [76] (section 3.4). The results of the study imply that the tension in DNA strands and the free-energy undesirability of a reaction occurring in domains tethered closer to the origami surface would be greater than in reactions that occur farther away from the surface, hence making walking of the robot slower than the cargo pick-up and drop-off reactions.

Additionally, walking reaction of the robot is a reversible reaction requiring toehold disassociation while the cargo pickup and drop off are irreversible reactions hence making them relatively faster.

Figure 4.12a shows the result of a stochastic simulation of cargo1 and cargo2 pick-up (using eqs. (4.1) to (4.3)) on a testbed with three cargo molecules of each type and only one robot. No goal locations were added in this case. The results of the simulation agree well with the experimental results for cargo pick-up (fig. 4.4).

Next, we simulated cargo sorting on a single origami with two types of cargos (using eqs. (4.1) to (4.4)). We compare the results of the simulation in fig. 4.12b to experimental data shown in fig. 4.8b. In the simulation, cargo1 denoted by the yellow trajectory shows marginally faster kinetics than cargo2, denoted by the green trajectory as also observed in experiment results. On close observation of the origami layout schematic in fig. 4.8b, we notice that cargo1 can be picked up and dropped off immediately after the robot has been triggered without taking a step. This is possible because the distance between the initial position of the robot and the one of the three cargo1 molecules is less than d_{max}. The robot, on leaving its initial location by making a step in any direction, is at a distance greater than d_{max} from either a cargo or a goal location.

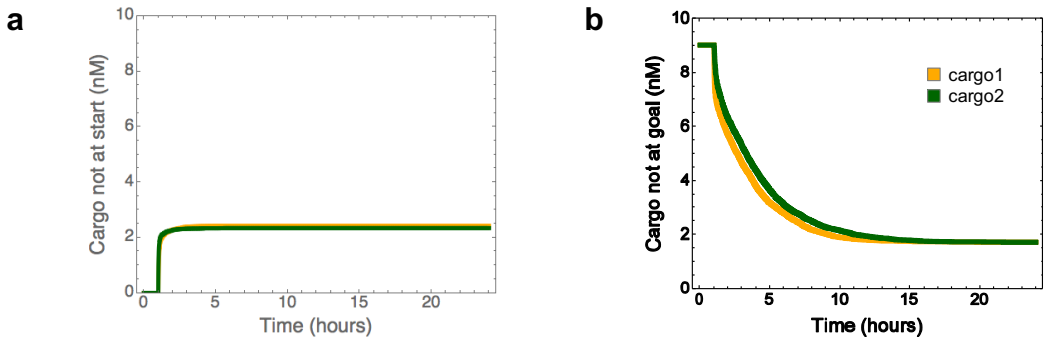

Figure 4.12: **Simulation of cargo-sort. a,** Stochastic simulation of cargo1 and cargo2 pickup on origami test ground shown in schematic fig. 4.4.**b,** Stochastic simulation of parallel sorting of cargo1 and cargo2 on origami test ground shown in schematic fig. 4.11a.

The fraction of cargo sorted in each of the above mentioned tests differed from each other (fig. 4.13a). One possible explanation for this is the type of inter-origami interactions occurring in each of these systems. The inter-origami interactions can be be classified into three main types:

- robot "hopping" from one origami to another

- robot on one origami picking up cargo from another origami
- goal on one origami picking up cargo from another origami

To quantify the extent of inter-origami interactions, we use the data obtained from section 4.5. As there is no robot involved in this experiment, the goal picking up cargo is the only reaction that can cause a decrease in fluorescence signal. The data in section 4.6 is used to obtain the maximum fraction of cargo1-F and cargo2-F that are correctly sorted under our experimental conditions.

Using a linear least-squares fit value with the completion level from the experiments, the completion level for each type of the modeled inter-origami interaction was determined using four variables and twelve data points (fig. 4.13b,c). We found that robot hopping from one origami to another was the least possible interaction (1.59%). This could be because of the higher probability of interacting with neighboring tracks or perhaps because interacting with tracks on another origami is not very accessible. A robot picking up cargo on another origami was a more significant reaction (4.56%). The cargo strands are away from the origami surface, more so than the tracks and hence more easily accesible to the robot pick up domains ($hand, arm$). The goal molecules on one origami picking up cargo from another origami was the most significant reaction (8.25%). Goal interaction was possibly more significant because the number of goal molecules per origami with respect to one robot per origami is much higher. The linear model and data agreed with each other with experimental noise less than 3.7%.

Conclusions

The biggest challenge in designing and building molecular robots is that troubleshooting is difficult when the system consists of multiple components. We followed a systematic approach to defining and designing modular building blocks. We first constructed a test ground for the robot to explore via random walk. We incrementally added cargo pick-up and finally cargo drop-off, by verifying that cargo-goal interaction without the robot was tolerable.

Because the cargo-sorting system required sequential localization of molecules (fig. 4.5), protection of domains involved in the reactions was imperative. The protection scheme for goals and the robot were tested in several iterations before implementing the final design that lead to minimal cross-talk between molecules and maximum protection of domains.

a

	Completion level of cargo1 (%)	Completion level of cargo2 (%)	Linear model
Fig. 4.7b	8.65	7.85	z
Fig. 4.8b	79.02	81.15	max
	2.72	0.88	0
Fig. 4.10ab	75.16	74.07	$max - x$
	12.11	14.09	$x + y + z$
Fig. 4.11b	62.29	64.49	$max - y - z$

b

		Completion level (%)
Robot moving from one origami to another	x	1.59
Robot on one origami picking up cargo on another	y	4.56
Goal on one origami picking up cargo on another	z	8.25

c

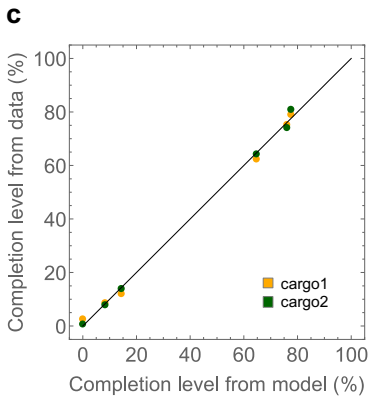

Figure 4.13: **Analysis of completion level. a,** List of completion levels at the 20 hr time point from experimental data, and our interpretation of completion levels in a linear model for analyzing inter-origami interactions. **b,** Definition of variable sin the model and the corresponding values from a linear least-squares fit. **c,** Comparison between model and data.

The designed robot is versatile enough that it can be generalized to sort multiple cargo types as the information for sorting is embedded in the recognition between cargos and their destinations. Since the robot is capable of exploring the entire two-dimensional surface of the nanostructure, cargo molecules can be randomly positioned and need not be allocated specific locations (as designed in our demonstration). The cargo picking up and dropping off are driven by the energy of base-pair formation and the robot is autonomous does not require additional energy supply. Multiple robots can be made to perform the same task in parallel on a single substrate, which may decrease sorting time. Finally, multiple instances of distinct tasks can be performed on individual origami molecules as demonstrated by the parallelism observed in our system.

4.9 Methods

DNA oligonucleotide synthesis

DNA oligos were purchased from Integrated DNA Technologies (IDT). The DNA strands in staples were purchased unpurified (standard desalting). The DNA strands that had track extensions were also purchased unpurified (standard desalting). The staple strands that were used for localizing cargo, goal and robot were purchased HPLC purified. All strands that had fluorophore and quenchers were also purchased HPLC purified. The trigger strands were purchased unpurified as they are short and were used in a large excess. All strands were purchased at 100 μM in TE buffer, pH

8.0, and stored at 4 °C.

Annealing protocol, buffer conditions and purification

Double layer origami was annealed with 1x (30 nM) M13 scaffold strand (Bayou Biolabs, Newhampshire) and 10x (300 nM) staples. The origami was annealed in 1x TAE/12.5 mM Mg buffer prepared from 50x TAE, pH 8.0 (Fisher BioReagents) and 1 M $MgCl_2$ (Invitrogen). Annealing was performed in a thermal cycler (Eppendorf), first heating up to 90°C for 5 min and then slowly cooling down to 20°C at the rate of 60 sec per 0.1°C. The origami after anneal was gel purified using 2% agarose gel, run on ice for 2 hrs at 80 V in 1x TAE/12.5 mM Mg buffer. The bands were then cut out of the gel and purified using freeze 'n squeeze (Biorad) following manufacturers instructions.

The goal complexes were annealed with 1x goal and 1.2x protection strands. The robot was annealed with 1x robot molecule and 1.2x protection strand. Annealing was performed in a thermal cycler (Eppendorf), first heating up to 90°C for 5 min and then slowly cooling down to 20°C at the rate of 6 sec per 0.1°C. After anneal, the complexes were purified using 15% PAGE.

Fluorescence Spectroscopy

Fluorescence kinetics data was collected every 2 minutes in a spectrofluorimeter (Fluorolog-3, Horiba). The spectrofluorimeter has a 4-cuvette holder for running parallel experiments. Experiments were performed with 50 μL reaction mixture per cuvette, in fluorescence cuvettes (Hellma #115F-QS) at 25 °C. The excitation/emission wavelengths were set to 602/624 nm for ATTO 590 and 560/575 nm for ATTO 550. Both fluorophores were tested and found to be spectrally well separated. They did not interfere with each other when used in the same circuit. Both excitation and emission bandwidths were set to 2 nm, and the integration time was 10 seconds for all experiments.

Atomic Force Microscopy

Samples for AFM imaging of double layer origami were prepared by diluting the origami to 1 nM in 1x TAE/ Mg buffer. After dilution, 40 μL of the sample was deposited onto freshly cleaved mica (SPI Supplies, 9.5 mm diameter, LOT 1170204). AFM images were taken in tapping mode in fluid on a Dimension FastScan Bio (Bruker) using FastScan-D tips (Bruker). All images were scanned at a resolution of 1024 lines.

APPENDIX

4.A Additional design diagrams

Double-layer square DNA origami design

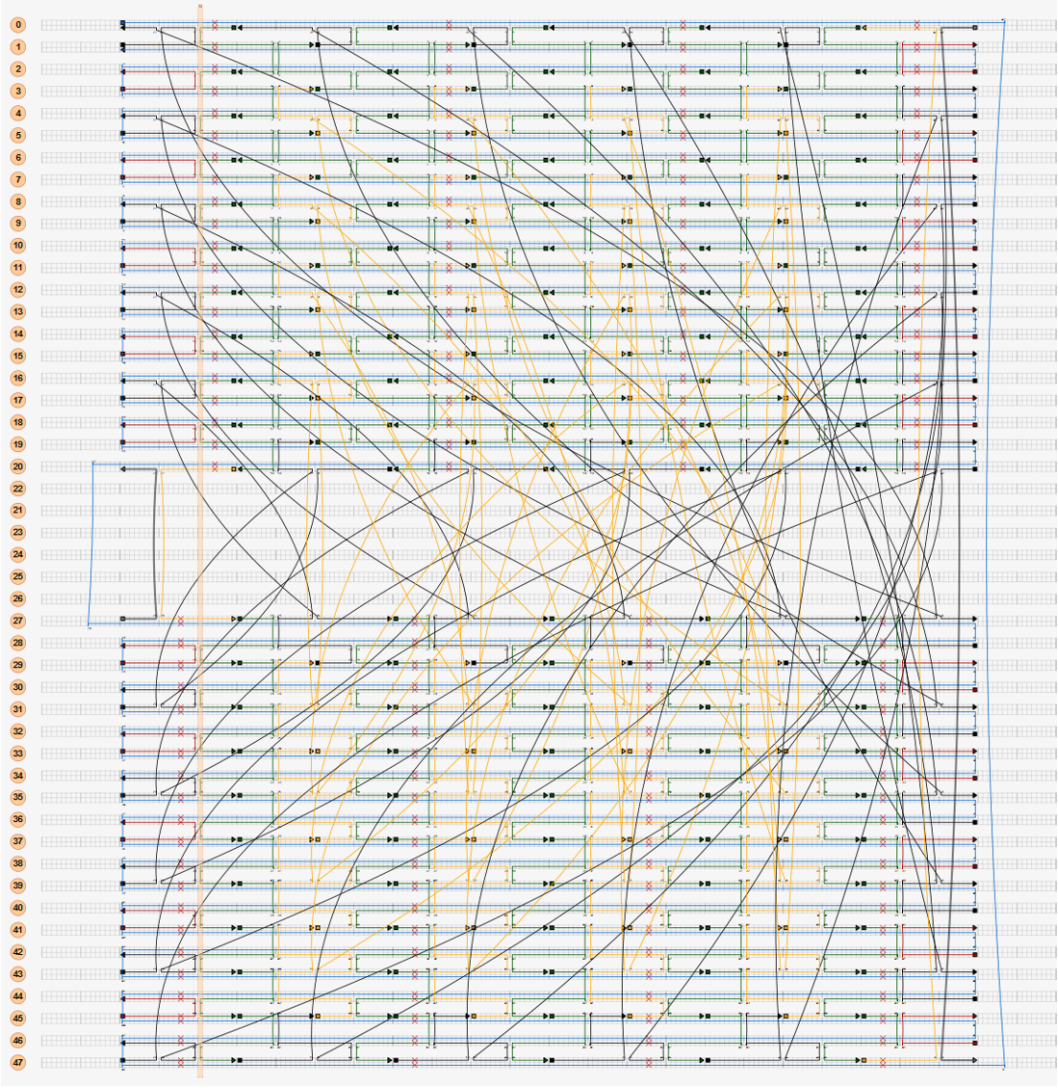

Figure 4.14: Double-layer square DNA origami design using caDNAo. A single M13 scaffold strand is show in blue. The two squares are held together using staples traversing both layers forming the double-layer square.

4.B Additional experimental results

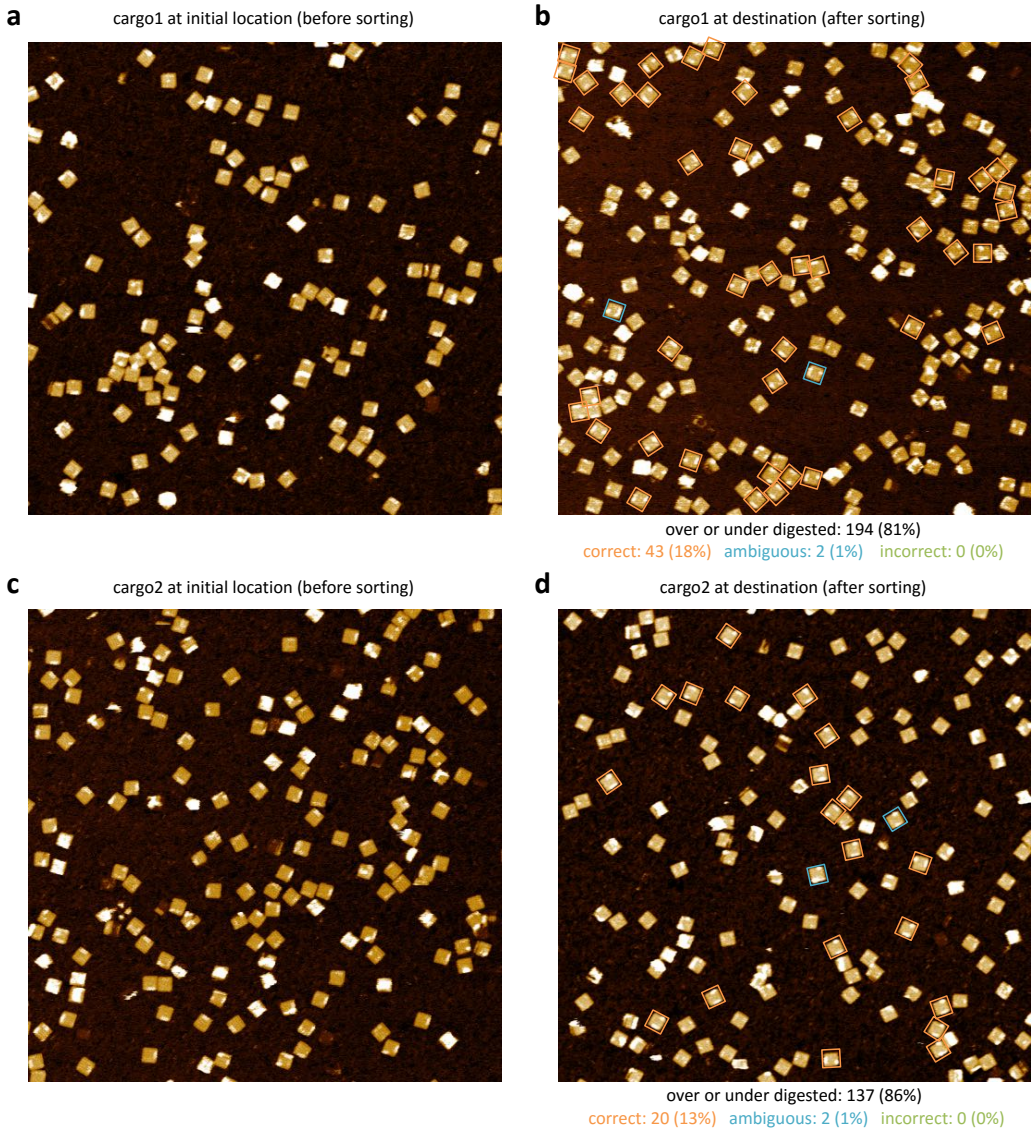

Figure 4.15: **A**FM images of the two types of cargos at their initial locations and destinations. **a**, Cargo1 at initial location. **b**, Cargo1 at destination. **c**, Cargo2 at initial location. **d**, Cargo2 at destination. All AFM images are 2 by 2 μm in size. If an origami does not have clearly recognizable molecules both at the initial location and at either of the two destinations, it is classified as *over or under digested* by Exonuclease I. Otherwise there are three situations: If the destination looks like the correct one (i.e. goal1 for cargo1 or goal2 for cargo2), it is classified as *correct*. If it is hard to tell whether the destination looks correct (for example, due to the lack of clear asymmetry of the molecules at the initial location), it is classified as *ambiguous*. If the destination looks like the incorrect one (i.e. goal2 for cargo2 or goal1 for cargo2), it is classified as *incorrect*.

4.C DNA sequences

Table S1: Staples in the double-layer square DNA origami.

Name	Sequence
T C01 R01	GGGCGATGGCCCACTAGAAAAACCAACGGGGT
T C01 R02	AACGGTACGCCAGAATAGGGATTTTAGACAGG
T C01 R03	AAGAATACGTGGCACATCTGACCTATGATACAGGAGTGTA
T C01 R04	TTGAGGATTTAGAAGTTCAATAGATAATACAT
T C01 R05	ATAACGGATTCGCCTGTACATCGGCCGTTCCAGTAAGCG
T C01 R06	TAGGTTGGGTTATATATTTTTAACCTCCGGCT
T C01 R07	GTACCGACAAAAGGTAAATAAGAGAGCCAGAATGGAAAGC
T C01 R08	CGGGAGGTTTTGAAGCCGAACCTCCCGACTTG
T C01 R09	AACAATGAAATAGCAAAATAATAATGATATTCACAAACAA
T C01 R10	GGTGAATTATCACCGTGAAATTATTCATTAAA
T C02 R01	AAAGGGCCGTGAACCATCACCCCAGGAGGC
T C02 R02	CGATTAACCTGAGAAGTGTTTTCAGAGATA
T C02 R03	GAACCCTGACAATATTTTTGAAAATAGATT
T C02 R04	AGAGCCGATTAGACTTTACAAAAGTAACAG
T C02 R05	TACCTTTATTGCTTTGAATACCGTCTGAGA
T C02 R06	GACTACCACTATATGTAAATGCCATTTTCG
T C02 R07	AGCCAGTAAGTAATTCTGTCCACGCGAGGC
T C02 R08	GTTTTAGCTTAAATCAAGATTAAATTGAGT
T C02 R09	TAAGCCCTAGCTATCTTACCGAAGGTAAAT
T C02 R10	ATTGACGCACCGACTTGAGCCACACCCTCA
T C02 R11	GAGCCGCCACCAGAAAGGAGGTTGAGGCAG
T C03 R01	TTTTTTGGTTAAAGAATTCGGTCG
T C03 R02	GTGAGGCCCAGAGCGGGAGCTAAAAAATCAAG
T C03 R03	AGTCTTTAGGGACATTCAACAACCCCTCATAGTTAGCGTACAATAGGA
T C03 R04	CAACTCGTAGGAGCACTAACAACTTGGCTATT
T C03 R05	AAATCGCGCGTCAGATAGCTTGATCGTCTTTCCAGACGCCACCACC
T C03 R06	ATCCAATCTTTATCAAAATCATAGAAGTTACA
T C03 R07	CAATAAACGTAATTTAATCAGCTTCTGTATGGGATTTTGCAACCGCCA
T C03 R08	TTTGCACCTCCGGTATTCTAAGAAGACGACGA
T C03 R09	TTAAGAAAAGAGATAACTCCAAAAAGTTTCAGCGGAGTGATACTCAGG
T C03 R10	TTAGAGCCCCGATTGAGGGAGGGAAGCCCTTT
T C04 R01	GTCCACTAGGTCGAGGTGCCGTAACTTTCCTC
T C04 R02	GTTAGAATACCGAGTAAAAGAGTCACGACCAG

Name	Sequence
T C04 R03	TAATAAAAATGCGCGAACTGATAGTATCTAAA
T C04 R04	ATATCTTTATTAAATCCTTTGCCCAGATTTTC
T C04 R05	AGGTTTAACAGAGGCGAATTATTCAAGAGTCA
T C04 R06	ATAGTGAAGCAAGACAAAGAACGCTTAACAAC
T C04 R07	GCCAACATAACATGTTCAGCTAATAGATATAG
T C04 R08	AAGGCTTACAGCTACAATTTTATCTGAGCGCT
T C04 R09	AATATCAGAGTAAGCAGATAGCCGAAAGGGCG
T C04 R10	ACATTCAAAGCAAAATCACCAGTAGCCACCCT
T C04 R11	CAGAACCGCCACCCTCTTTTTCACTAAAGGAATTGCGA
T C05 R01	AATCGGAATGTTGTTCATTACAGG
T C05 R02	CACGCAAAAGCACGTATAACGTGAGCACTA
T C05 R03	ACATCGCCAGATTCACCGTTAATAGAAAGAGGACAGATGATCCGCGAC
T C05 R04	ATTAATTTGGAATTGAGGAAGGTCCCTAAA
T C05 R05	TTACCTGAATAAAGAATACCAGTGCGCATAGGCTGGCTTTGTATCA
T C05 R06	CTTTTTCAGATTAAGACGCTGAGATTTCAA
T C05 R07	GCGCCTGTTGAGAATCTACCTTATAATCTTGACAAGAACCATACCAAG
T C05 R08	TTACCAACAATAGCAAGCAAATCGCAGAAC
T C05 R09	TTACCAGAAAGTCAGATGGTTTAAAAATCAACGTAACAAACTAAAACA
T C05 R10	TACCATTACCAGCGCCAAAGACAAACAAAG
T C06 R01	GGGTTGAGCCCTAAAGGGAGCCCCTATGGTTG
T C06 R02	CTTTGACGTTAACCGTTGTAGCAAGATTATTT
T C06 R03	ACATTGGCATTAAAAATACCGAACAAATCAAC
T C06 R04	AGTTGAAATAAAAGTTTGAGTAACTGCACGTA
T C06 R05	AAACAGAAGCAAAAGAAGATGATGACATAGCG
T C06 R06	ATAGCTTAAATATATTTTAGTTAAAACAGTAG
T C06 R07	GGCTTAATTTATCAACAATAGATAAATCATTA
T C06 R08	CCGCGCCCGCTAACGAGCGTCTTTCTGAACAC
T C06 R09	CCTGAACAAGGAAACCGAGGAAACAAATTCAT
T C06 R10	ATGGTTTAGCAAGGCCGGAAACGTAGCCACCA
T C06 R11	CCGGAACCGCCTCCCTCACCAGAATAAGGCTTGCCCTG
T C07 R01	AGCTTGACAAATCAAATTTGGGGC
T C07 R02	TGATTAGTGCTACAGGGCGCGTACCGATTTAG
T C07 R03	AGCAGAAGTCAATCGTATGGTCAATCAAATATCGCGTTTTTCAGGTCT
T C07 R04	TTTGCGGATATCTGGTCAGTTGGCGAACCACC
T C07 R05	ATCAAGAACCATATCAATTTAGTCGAACCAGACCGGAATCAGAAAA

Name	Sequence
T C07 R06	TCTGACCTCTTAGAATCCTTGAAAAAACAAAC
T C07 R07	ACAAGAAATATAAAGCAACAGTTGCAGGATTAGAGAGTACTCATTGAA
T C07 R08	TAATTTGCTTATTTTCATCGTAGGAGTCCTGA
T C07 R09	AACGGAATTAGACGGGAAGTACGGTTGATAAGAGGTCATTGCGTCCAA
T C07 R10	AAACCATCTGTCACAATCAATAGAGCAATAAT
T C08 R01	CCCTTATGGGGAAAGCCGGCGACGCGCTTA
T C08 R02	ATGCGCCAATAACATCACTTGCCCTACATT
T C08 R03	TTGACGCATAAAACAGAGGTGACAAACCCT
T C08 R04	CAATCAAACAAAGAAACCACCAAAGGGTTA
T C08 R05	GAACCTAAACAAAATTAATTACATTAATTA
T C08 R06	ATTTTCCAAATTTAATGGTTTGACAAATTC
T C08 R07	TTACCAGAATAATATCCCATCCACAAGCAA
T C08 R08	GCCGTTTCAGTTACAAAATAAAAACAGGGA
T C08 R09	AGCGCATACCCAAAAGAACTGGGGAATAAG
T C08 R10	TTTATTTGATAGCAGCACCGTAATCTTTTC
T C08 R11	ATAATCAAAATCACCTGTAGCTAGCTTAATTGCTGAA
T C09 R01	AGAAAGGATGGTGGTTAACAAACG
T C09 R02	AAGAACTCTAACCACCACACCCGCACGTGGCG
T C09 R03	GTATTAACAACGCTCAGCGAGTAAGTCATTGCCTGAGAGTATGATATT
T C09 R04	GGAATTATGCTGAACCTCAAATATGGCGGTCA
T C09 R05	ATTTCATTTACTTCTGTGTAGCCAATCGATGAACGGTAAAAGGCCG
T C09 R06	ACCGTGTGTCTGTAAATCGTCGCTATTTAACA
T C09 R07	GAGCATGTGTATCATACGCCATCAGTCAATCATATGTACCGTAATGTG
T C09 R08	TTATTTATACCGCACTCATCGAGATAATTTAC
T C09 R09	AGACTCCTGAGAGAATTTAAATCAAAAAGCCCCAAAAACAAACCCTCA
T C09 R10	CGACAGAAAACGCAAAGACACCACCATGATTA
T C10 R01	CTGTTTGAAGGGAAGAAAGCGAAAGCGGTCAC
T C10 R02	GCTGCGCGAAACTATCGGCCTTGCCCATTGCA
T C10 R03	ACAGGAAAACCGCCTGCAACAGTGTCTAAAGC
T C10 R04	ATCACCTTCATCATATTCCTGATTCTGATTGT
T C10 R05	TTGGATTATGAATTACCTTTTTTAAGTGAATA
T C10 R06	ACCTTGCTATAAATAAGGCGTTAAAGAAAAAG
T C10 R07	CCTGTTTAAGAAACCAATCAATAAGGGTATTA
T C10 R08	AACCAAGTCCCAATCCAAATAAGAAATAGCAG
T C10 R09	CCTTTACATATTACGCAGTATGTTGGCAACAT

Name	Sequence
T C10 R10	ATAAAAGATCAAGTTTGCCTTTAGTTTCGGTC
T C10 R11	ATAGCCCCCTTATTAGATATTTTAAATATTTAAATTGT
T C11 R01	GTTTGCCCGCAGCAAG
T C11 R02	GCGCTGGCAAGTGTAGGAGCGGGCGCTAGG
T C11 R03	CAATATTATTGCCCTTTCGTAATCATGGTCATGTTGTAAA
T C11 R04	GCAGCAAATGAAAAACCACGCTGAGAGCCA
T C11 R05	TTCATCAATCTTTTCAAATTGTTATCCGCTTGGGTAAC
T C11 R06	AAATCAATATATGTGATGGAAACAGTACAT
T C11 R07	GGAATCATGCGGTTTGATACGAGCCGGAAGCACGAAAGGG
T C11 R08	TATCATTCCAAGAACTCGGCTGTCTTTCCT
T C11 R09	AACGTCAACATTAATGGGGGTGCCTAATGAGTGTGCGGGC
T C11 R10	TACATACATAAAGGTAGCAAACGTAGAAAA
T C11 R11	CGTTTTCATTTCCAGATTGCGTTGCGCTCA
B C01 R01	GCATTGACCCACCACC
B C01 R02	ATAAGTATAGCCCGGGTCGAGAGGGTTGAT
B C01 R03	ATAATAATAGAGCCACTTTGGGAA
B C01 R04	CAACCTAAAACGAAACCACTACGAAGGCAC
B C01 R05	ACGAGAAACAGAGCCGCACCAT
B C01 R06	GGGGGTAATAGTAAAAGAAGTTTTGCCAGA
B C01 R07	TATAATGCGGAACCAGCACCAATG
B C01 R08	TTTATTTCAACGCAATTTGCGGGAGAAGCC
B C01 R09	AAACGTTACGTTTGCCATCAGTAG
B C01 R10	CAGGCTGCGCAACTGGCGCCATTCGCCATT
B C01 R11	CTGCCCGCTCGGCATCGTCAGACTGTAGCG
B C02 R01	GTCAGACGCAGGCGGATAAGTGCCAATAGGTG
B C02 R02	TATCACCGGAATAGAAAGGAACAACGTTGAAA
B C02 R03	ATCTCCAAGGTAAAATACGTAATGGAGGCAAA
B C02 R04	AGAATACAGCTGCTCATTCAGTGAACGAGTAG
B C02 R05	TAAATTGGGAGAGGCTTTTGCAAAATGTTTAG
B C02 R06	ACTGGATATTTGCGGATGGCTTAGCAACATGT
B C02 R07	TTTAAATAATGACCCTGTAATACTGGATAAAA
B C02 R08	ATTTTTAGGGAAGATTGTATAAGCGTTAAAAT
B C02 R09	TCGCATTACCGGAAACCAGGCAAATTGGGAAG
B C02 R10	GGCGATCGGAGCTAACTCACATTATCGGGAAA
B C02 R11	CCTGTCGTGCCAGCTGAAATGAAAACGATTTTTTGTTT

Name	Sequence
B C03 R01	AGGTTTAGGGGTTTTGCTCAGTACATTGGCCTGAGCAAGA
B C03 R02	CTCATCTTAAGTTTCCATTAAACGAAAAAAGGCCCACAAGGTTGCTAT
B C03 R03	TACTGCGGTAAAAACCAAAATAGCGCTTGAGAGGGTAATCTGAATC
B C03 R04	TATATTTTTTGTACCAAAAACATTTGCAACTAAGAATTAACCAGAGCC
B C03 R05	CTCTTCGCGGCACCGCTTCTGGTGAATTTTTGAACATAAACAGCCATA
B C04 R01	ATAAATCGGATTAGGATTAGCGTACCGCCA
B C04 R02	CCCTCAGTAAACAACTTTCAACGGAGCCTT
B C04 R03	TAATTGTGACTTTTTCATGAGGTGACCCCC
B C04 R04	AGCGATTGGATATTCATTACCCTTTCAACT
B C04 R05	TTAATCATTTACCAGACGACGAAATCGTCA
B C04 R06	TAAATATCTTTAATTGCTCCTTTGTCTGGA
B C04 R07	AGTTTCAATAAAGCTAAATCGGAAATGCAA
B C04 R08	TGCCTGACCGGTTGATAATCAGGCTCATTT
B C04 R09	TTTAACCTCCAGCCAGCTTTCCTATTACGC
B C04 R10	CAGCTGGTAAAGTGTAAAGCCTAATCGGCC
B C04 R11	AACGCGCGGGGAGAGAATTACTATAAGAATAAACACC
B C05 R01	CCCTCAGAAGACTCCTCAAGAGAACTCATTAAAATATAAA
B C05 R02	CGCGAAACGGCTTTGAGGACTAAAATCGGTTTGGCAGAGGTGATGCAA
B C05 R03	TCCCCCTCACTATCATAACCCTCGTTGTGAATGCCATATGAGAAAA
B C05 R04	TAGGTAAAAATAAAGCCTCAGAGCTTCCATATCAACGCTCTTTCATCT
B C05 R05	GGATGTGCCTCAGGAAGATCGCACAATAGGAATGCGTTATAAATACCG
B C06 R01	GCAGTCTCAAGTATTAAGAGGCTGACCGCCAC
B C06 R02	CCTCAGAGTTAGTAAATGAATTTTGCTTTCGA
B C06 R03	GGTGAATTTAGCAACGGCTACAGAAAAGTACA
B C06 R04	ACGGAGATGACCTTCATCAAGAGTGCGATTTT
B C06 R05	AAGAACTGGCATAGTAAGAGCAACAAATGCTT
B C06 R06	TAAACAGTGCAAACTCCAACAGGTATTCCCAA
B C06 R07	TTCTGCGAATTAGCAAAATTAAGCGATTCAAA
B C06 R08	AGGGTGAGATCGTAAAACTAGCATAAAATAAT
B C06 R09	TCGCGTCTCGACGACAGTATCGGCTGCAAGGC
B C06 R10	GATTAAGTCACAATTCCACACAACCGTATTGG
B C06 R11	GCGCCAGGGTGGTTTTTATAATCATCAGATGATGGCAA
B C07 R01	CTCATTTTATTCTGAAACATGATGAATTTAGAGAAACA
B C07 R02	TCGCCTGGCATCGGAACGAGGGTCTTAAACGAATATACCAATTCGA
B C07 R03	CGAGAATAAAAGGAATTACGAGGCTCATTAATTGCGTGAACGTT

Name	Sequence
B C07 R04	GAGACAGCAGGCAAGGCAAAGAACGAGTAGAAATTATTATTATCAT
B C07 R05	GCCAGGGGCCAGTTTGAGGGGAGGCCTTCCAATAATGGGAAGGAGC
B C08 R01	TCATACATCTATTTCGGAACCTATTCAGGGAT
B C08 R02	AGCAAGCCACGATCTAAAGTTTTGTACCGATA
B C08 R03	GTTGCGCCTCAGCAGCGAAAGACAATAAATTG
B C08 R04	TGTCGAAAACGGTGTACAGACCAGCAGGACGT
B C08 R05	TGGGAAGAGCAGATACATAACGCCGACCATAA
B C08 R06	ATCAAAAAAATTCGAGCTTCAAAGTTGACCAT
B C08 R07	TAGATACACATCCAATAAATCATATCAAATCA
B C08 R08	CCATCAATCTGGAGCAAACAAGAGAGCTTTCA
B C08 R09	TCAACATTTCGTAACCGTGCATCTTTTTCCCA
B C08 R10	GTCACGACAGCTGTTTCCTGTGTGACCAGTGA
B C08 R11	GACGGGCAACAGCTGACCGCCAGTGGTAATATCCAGAA
B C09 R01	ACCCATGTAGTTAATGCCCCCTGCGGCTTTTGGAAAGCGT
B C09 R02	CTGCTCCATGCGGGATCGTCACCCGACAATGACTGGCCAATATAATCA
B C09 R03	TTACCCTGACCACATTCAACTAATAAAATCTACAGTCACTGTCCAT
B C09 R04	CAACCGTTATAGTAGTAGCATTAATTTCGCAACTGAAATGTACTTCTT
B C09 R05	ACGACGGCGGTGTAGATGGGCGCAAAATGTGATGGAAATACTGAGTAG
B C10 R01	CTGGTAAAGTGCCCGTATAAACACCGTAAC
B C10 R02	ACTGAGTCATTCCACAGACAGCATCGCCCA
B C10 R03	CGCATAAGTTAAAGGCCGCTTTTGTTACTT
B C10 R04	AGCCGGAGAACTGACCAACTTTAAACGAAC
B C10 R05	TAACGGATTGAGATTTAGGAATACTATTAT
B C10 R06	AGTCAGAGAAGCCCGAAAGACTTAACCTGT
B C10 R07	TTAGCTAGCATCAATTCTACTACTAGCTGA
B C10 R08	TAAATTATACAAAGGCTATCAGCAACCCGT
B C10 R09	CGGATTCGGGATAGGTCACGTTCAGTGCCA
B C10 R10	AGCTTGCGTACCGAGCTCGAATCACCGCCT
B C10 R11	GGCCCTGAGAGAGTTCAGCAGGCGAAAATC
B C11 R01	CAGTGCCTTGAGTAACTAAGTTTTGTCTATCA
B C11 R02	ACTACAACGCCTGTAGTTCGTCACCAGTACAA
B C11 R03	CTGAGGCTTGCAGGGACCGATATACGTGGACTCCAACGTC
B C11 R04	CAATCATAAGGGAACCACGAGGCGCAGACGGT
B C11 R05	TAGAAAGATTCATCAGACAACATTCAGTTTGGAACAAGA
B C11 R06	TCAAAAAGATTAAGAGAGCAAAGCGGATTGCA

Name	Sequence
B C11 R07	GCGAGCTGAAAAGGTGTATTTTCAAGAATAGCCCGAGATA
B C11 R08	CTATTTTTGAGAGATCATGCCGGAGAGGGTAG
B C11 R09	GCGGATTGACCGTAATTCCGTGGGCCGAAATCGGCAAAAT
B C11 R10	TCTAGAGGATCCCCGGATGCCTGCAGGTCGAC

Table S2: Robot, track, cargo and goal strands.

Name	Sequence
robot	CCATTC CCTATCACACCATACTCTAC ATCCAC
	CCTCAAAACTTATCC ATCCAC
robot inhibitor	GTGCTC GTGGAT GTAGAGTATGGTGTGATAGG
	GAATGG
robot trigger	CCTATCACACCATACTCTAC ATCCAC GAGCAC
robot start attacher	GGAAGTAAGTAGAAG GTGGAT
	GGATAAGTTTTGAGG
robot start staple	CTTCTACTTACTTCC TT - staple
track1 staple	staple - TTTTT GTGGAT GGATAAGTTTTGAGG
track2 staple	staple - TTTTT TTTTTT GGATAAGTTTTGAGG GTGGAT
robot goal staple	staple - TTTTT GTGGAT GGATAAGTTTTGAGG GTGGAT
cargo1	AGATGT GTAGAGTATGGTGTGATAGG GAATGG TT
cargo1-F	AGATGT GTAGAGTATGGTGTGATAGG
	GAATGG TT /3ATTO532N/
cargo1 attacher	CCTATCACACCATACTCTAC
	ACCTTACCTCATCCCTAACTT
cargo1 attacher-Q	/5IAbRQ/ CCTATCACACCATACTCTAC
	ACCTTACCTCATCCCTAACTT
cargo1 staple	staple - TTTTT AAGTTAGGGATGAGGTAAGGT
cargo2	GAAAGG GTAGAGTATGGTGTGATAGG GAATGG TT
cargo2-F	GAAAGG GTAGAGTATGGTGTGATAGG
	GAATGG TT /3ATTO590N/
cargo2 attacher	CCTATCACACCATACTCTAC
	CTCCCTACCCATATCACCTT
cargo2 attacher-Q	/5IAbRQ/ CCTATCACACCATACTCTAC
	CTCCCTACCCATATCACCTT
cargo2 staple	staple - TTTTT AAGGTGATATGGGTAGGGAG
goal1	CCATTC CCTATCACACCATACTCTAC ACATCT
	ACTAACTCCTACCCACACCT
goal1-Q	/5IAbRQ/ CCATTC CCTATCACACCATACTCTAC ACATCT
	ACTAACTCCTACCCACACCT
goal1 staple	staple - TTTTT AGGTGTGGGTAGGAGTTAGT
goal2	CCATTC CCTATCACACCATACTCTAC CCTTTC
	CAACTCTCCACTCCAATCAA

Name	Sequence
goal2-Q	/5IAbRQ/ CCATTC CCTATCACACCATACTCTAC CCTTTC CAACTCTCCACTCCAATCAA
goal2 staple	stape - TTTTT TTGATTGGAGTGGAGAGTTG
goal1 inhibitor	AGATGT GTAGAGTATG ACACTT
goal2 inhibitor	GAAAGG GTAGAGTATG ACACTT
goal inhibitor	ACTCTA GTGTGATAGG GAATGG
goal trigger1	AAGTGT CATACTCTAC
goal trigger2	CCTATCACAC TAGAGT

Chapter 5

CONCLUSIONS

The introduction to this thesis highlighted our efforts to build artificial molecular machines, taking inspiration from the natural molecular machines in biology. For example, a compartmentalized cell includes complex biochemical reactions for processing information inside cells as well as molecular motors that help in transporting cargo across the cell. In this thesis, we used DNA both as a computational material to implement complex circuits and as an engineering material capable of implementing robots that can perform complex behavior of cargo-sorting.

In chapter 2, we demonstrated a systematic way of building large scale DNA circuits capable of Boolean computation. We also demonstrated that such large circuits could be built using cheap, 'unpurfied' DNA strands using simplified experimental procedures.

Most DNA circuits to date have been constructed using purified circuit components. 'Purified' components can refer to two different things. Vendor-purified DNA oligos can be purchased, although they are currently priced 10 fold more than unpurified strands. Here, single-stranded oligos are passed though an HPLC column or gel-purified. Such purification techniques remove truncated DNA strands that do not contain the complete desired sequence. In addition to truncation, purification also removes DNA strands with synthesis errors including insertion and mutation. Another level to the purification process is the removal of unused strands while annealing DNA complexes. Such methods ensure correct stoichiometry in molecules. While these established practices help demonstrate DNA computation, they are time and labor intensive. Gel-purification techniques help reduce stoichiometry errors in DNA complexes, there are still synthesis errors in the oligos that need to be addressed. The state-of-the-art DNA synthesis currently provides less than 1% error per base coupling reaction, but ideally we want the the process to be error-free for predictable computation. In order to build sophisticated circuits that rival biochemical processes inside biological cells, we hope to implement liquid handling systems that can help build DNA circuits quickly and efficiently. However, even with perfectly synthesized DNA strands, the stoichiometry errors (caused not only by pipetting errors in mixing multiple strands but also by errors in quantifying the

concentrations of individual strands) will still exist, preventing the experiments to be completely performed by the present day liquid handling systems. But until such time that these technical issues can be resolved, we hope that our work encourages researchers to efficiently design and construct large DNA-based circuits, using our systematic experimental procedure that takes synthesis and stoichiometry errors into consideration.

In addition to implementing DNA for large-scale computations and information processing, the DNA nanotechnology community has also been developing sophisticated tools for building complex two-and three-dimensional DNA structures. Using these tools in cahpters 3 and 4, we demonstrated the sophisticated task of cargo-sorting on a two-dimensional DNA origami by a DNA robot. Using modular building blocks and a simple algorithm, we were able to construct a nano-robot system which can be scaled up easily to sort multiple types of cargoes using the same robot design. The system can also be used as a platform for multiple robots to perform collective tasks. The modular building blocks developed in our study can be employed for performing other diverse nanomechanical tasks like finding the shortest path for efficient molecular transportation. The robots performing nanomechanical tasks can be used in diverse applications including chemical synthesis.

The "RNA-world" hypothesis suggests that an all nucleic acid world is possible [79]. By using modular building blocks for constructing sophisticated systems capable of demonstrating complex behaviors, could we then envision building a "DNA-world" with significant information processing capabilities on its own?

BIBLIOGRAPHY

[1] Erwin Schrödinger. *What Is Life? the physical aspect of the living cell and mind*. Dublin, 1943.

[2] John Von Neumann and John C Oxtoby. "John von neumann". In: *American Mathematical Soc* (1988).

[3] Alan Mathison Turing. "The chemical basis of morphogenesis". In: *Bulletin of mathematical biology* 52.1-2 (1990), pp. 153–197.

[4] D Elson and E Chargaff. "On the desoxyribonucleic acid content of sea urchin gametes". In: *Cellular and Molecular Life Sciences* 8.4 (1952), pp. 143–145.

[5] James D Watson and Francis HC Crick. "The structure of DNA". In: *Cold Spring Harbor symposia on quantitative biology*. Vol. 18. Cold Spring Harbor Laboratory Press. 1953, pp. 123–131.

[6] Timothy J Richmond and Curt A Davey. "The structure of DNA in the nucleosome core". In: *Nature* 423.6936 (2003), pp. 145–150.

[7] D. Yu. Zhang and G. Seelig. "Dynamic DNA nanotechnology using strand-displacement reactions". In: *Nature Chemistry* 3.2 (2011), pp. 103–113.

[8] Adrien Padirac, Teruo Fujii, and Yannick Rondelez. "Nucleic acids for the rational design of reaction circuits". In: *Current opinion in biotechnology* 24.4 (2013), pp. 575–580.

[9] Robert Carlson. "The changing economics of DNA synthesis". In: *Nature biotechnology* 27.12 (2009), p. 1091.

[10] Ryosuke Iinuma et al. "Polyhedra self-assembled from DNA tripods and characterized with 3D DNA-PAINT". In: *science* 344.6179 (2014), pp. 65–69.

[11] Toshio Ando. "High-speed atomic force microscopy coming of age". In: *Nanotechnology* 23.6 (2012), p. 062001.

[12] J. N. Zadeh et al. "NUPACK: analysis and design of nucleic acid systems". In: *Journal of Computational Chemistry* 32.1 (2011), pp. 170–173.

[13] M. Lakin et al. *Visual DSD*. http://research.microsoft.com/en-us/projects/dna. 2009.

[14] L. Qian. *Seesaw Compiler*. http://www.qianlab.caltech.edu/SeesawCompiler. 2011.

[15] Harish Chandran et al. "Localized hybridization circuits". In: *International Workshop on DNA-Based Computers*. Springer. 2011, pp. 64–83.

[16] Paul J Hagerman. “Flexibility of DNA”. In: *Annual review of biophysics and biophysical chemistry* 17.1 (1988), pp. 265–286.

[17] John SantaLucia, Hatim T Allawi, and P Ananda Seneviratne. “Improved nearest-neighbor parameters for predicting DNA duplex stability”. In: *Biochemistry* 35.11 (1996), pp. 3555–3562.

[18] John SantaLucia Jr and Donald Hicks. “The thermodynamics of DNA structural motifs”. In: *Annu. Rev. Biophys. Biomol. Struct.* 33 (2004), pp. 415–440.

[19] Nadrian C Seeman. “Nucleic acid junctions and lattices”. In: *Journal of theoretical biology* 99.2 (1982), pp. 237–247.

[20] Leonard M Adleman. “Molecular computation of solutions to combinatorial problems”. In: *Nature* 369 (1994), p. 40.

[21] Bernard Yurke et al. “A DNA-fuelled molecular machine made of DNA”. In: *Nature* 406.6796 (2000), pp. 605–608.

[22] Georg Seelig et al. “Enzyme-free nucleic acid logic circuits”. In: *science* 314.5805 (2006), pp. 1585–1588.

[23] Paul WK Rothemund, Nick Papadakis, and Erik Winfree. “Algorithmic self-assembly of DNA Sierpinski triangles”. In: *PLoS Biol* 2.12 (2004), e424.

[24] David Yu Zhang et al. “Engineering entropy-driven reactions and networks catalyzed by DNA”. In: *Science* 318.5853 (2007), pp. 1121–1125.

[25] L. Qian and E. Winfree. “Scaling up digital circuit computation with DNA strand displacement cascades”. In: *Science* 332.6034 (2011), pp. 1196–1201.

[26] Niranjan Srinivas et al. “Enzyme-Free Nucleic Acid Dynamical Systems”. In: *bioRxiv* (2017). DOI: 10.1101/138420. eprint: http://biorxiv.org/content/early/2017/05/16/138420.full.pdf. URL: http://biorxiv.org/content/early/2017/05/16/138420.

[27] P. Yin et al. “Programming biomolecular self-assembly pathways”. In: *Nature* 451.7176 (2008), pp. 318–322.

[28] Bernard Yurke and Allen P Mills. “Using DNA to power nanostructures”. In: *Genetic Programming and Evolvable Machines* 4.2 (2003), pp. 111–122.

[29] D. Y. Zhang and E. Winfree. “Control of DNA strand displacement kinetics using toehold exchange”. In: *Journal of the American Chemical Society* 131.47 (2009), pp. 17303–17314.

[30] N. Srinivas et al. “On the biophysics and kinetics of toehold-mediated DNA strand displacement”. In: *Nucleic Acids Research* 41.22 (2013), pp. 10641–10658.

[31] Andrew Phillips and Luca Cardelli. “A programming language for composable DNA circuits”. In: *Journal of the Royal Society Interface* 6.Suppl 4 (2009), S419–S436.

[32] M. R. Lakin et al. “Visual DSD: a design and analysis tool for DNA strand displacement systems”. In: *Bioinformatics* 27.22 (2011), pp. 3211–3213.

[33] Cal Green and Clark Tibbetts. “Reassociation rate limited displacement of DNA strands by branch migration”. In: *Nucleic Acids Research* 9.8 (1981), pp. 1905–1918.

[34] Luis P Reynaldo et al. “The kinetics of oligonucleotide replacements”. In: *Journal of molecular biology* 297.2 (2000), pp. 511–520.

[35] L. Qian and E. Winfree. “A simple DNA gate motif for synthesizing large-scale circuits”. In: *Journal of the Royal Society Interface* (2011), doi:10.1098/rsif.2010.0729.

[36] Richard A Muscat et al. “DNA-based molecular architecture with spatially localized components”. In: *ACM SIGARCH Computer Architecture News*. Vol. 41. 3. ACM. 2013, pp. 177–188.

[37] Frits Dannenberg et al. “DNA walker circuits: Computational potential, design, and verification”. In: *International Workshop on DNA-Based Computers*. Springer. 2013, pp. 31–45.

[38] P. W. K. Rothemund. “Folding DNA to create nanoscale shapes and patterns”. In: *Nature* 440.7082 (2006), pp. 297–302.

[39] Shawn M Douglas et al. “Rapid prototyping of 3D DNA-origami shapes with caDNAno”. In: *Nucleic acids research* (2009), gkp436.

[40] George Amoako et al. “3D DNA origami designed with caDNAno”. In: *Chinese Science Bulletin* 58.24 (2013), pp. 3019–3022.

[41] Wibowo Adi and Kosuke Sekiyama. “A component-based framework for molecular robotic development as smart drug system”. In: *Micro-NanoMechatronics and Human Science (MHS), 2015 International Symposium on*. IEEE. 2015, pp. 1–5.

[42] Shawn M Douglas, Ido Bachelet, and George M Church. “A logic-gated nanorobot for targeted transport of molecular payloads”. In: *Science* 335.6070 (2012), pp. 831–834.

[43] Alexander E Marras et al. “Programmable motion of DNA origami mechanisms”. In: *Proceedings of the National Academy of Sciences* 112.3 (2015), pp. 713–718.

[44] Martin Langecker et al. “Synthetic lipid membrane channels formed by designed DNA nanostructures”. In: *Science* 338.6109 (2012), pp. 932–936.

[45] Peng Yin et al. “A unidirectional DNA walker that moves autonomously along a track”. In: *Angewandte Chemie International Edition* 43.37 (2004), pp. 4906–4911.

[46] Shelley FJ Wickham et al. “A DNA-based molecular motor that can navigate a network of tracks”. In: *Nature nanotechnology* 7.3 (2012), pp. 169–173.

[47] Hongzhou Gu et al. “A proximity-based programmable DNA nanoscale assembly line”. In: *Nature* 465.7295 (2010), pp. 202–205.

[48] Ye Tian et al. “A DNAzyme That Walks Processively and Autonomously along a One-Dimensional Track”. In: *Angewandte Chemie International Edition* 44.28 (2005), pp. 4355–4358.

[49] Jong-Shik Shin and Niles A Pierce. “A synthetic DNA walker for molecular transport”. In: *Journal of the American Chemical Society* 126.35 (2004), pp. 10834–10835.

[50] Michael JB Krieger, Jean-Bernard Billeter, and Laurent Keller. “Ant-like task allocation and recruitment in cooperative robots”. In: *Nature* 406.6799 (2000), pp. 992–995.

[51] Anton Kuzyk et al. “A light-driven three-dimensional plasmonic nanosystem that translates molecular motion into reversible chiroptical function”. In: *Nature communications* 7 (2016).

[52] Dominik J Kauert et al. “Direct mechanical measurements reveal the material properties of three-dimensional DNA origami”. In: *Nano letters* 11.12 (2011), pp. 5558–5563.

[53] Emanuel Pfitzner et al. “Rigid DNA Beams for High-Resolution Single-Molecule Mechanics”. In: *Angewandte Chemie International Edition* 52.30 (2013), pp. 7766–7771.

[54] Jürgen J Schmied et al. “DNA origami nanopillars as standards for three-dimensional superresolution microscopy”. In: *Nano letters* 13.2 (2013), pp. 781–785.

[55] Kevin Montagne et al. “Programming an in vitro DNA oscillator using a molecular networking strategy”. In: *Molecular systems biology* 7.1 (2011), p. 466.

[56] David Soloveichik, Georg Seelig, and Erik Winfree. “DNA as a universal substrate for chemical kinetics”. In: *Proceedings of the National Academy of Sciences* 107.12 (2010), pp. 5393–5398.

[57] J. M. Schaeffer, C. Thachuk, and E. Winfree. “Stochastic simulation of the kinetics of multiple interacting nucleic acid strands”. In: *Lecture Notes in Computer Science* 9211 (2015), pp. 194–211.

[58] Seung Woo Shin, Chris Thachuk, and Erik Winfree. *Verifying chemical reaction network implementations: A pathway decomposition approach*. Preprint at https://arxiv.org/abs/1411.0782. 2014.

[59] M. R. Lakin et al. “Design and analysis of DNA strand displacement devices using probabilistic model checking”. In: *Journal of the Royal Society Interface* (2012), doi:10.1098/rsif.2011.0800.

[60] D. Y. Zhang and E. Winfree. “Robustness and modularity properties of a non-covalent DNA catalytic reaction”. In: *Nucleic Acids Research* 38.12 (2010), pp. 4182–4197.

[61] L. Qian, E. Winfree, and J. Bruck. “Neural network computation with DNA strand displacement cascades”. In: *Nature* 475 (2011), pp. 368–372.

[62] D. E. Müller. *Asynchronous Logics and Application to Information Processing, switching Theory in Space Technology*. 1963.

[63] D. Soloveichik. *CRNSimulator*. http://users.ece.utexas.edu/~soloveichik/crnsimulator.html. 2009.

[64] S. Wolfram. “Statistical mechanics of cellular automata”. In: *Reviews of Modern Physics* 55.3 (1983), p. 601.

[65] M. Cook. “Universality in elementary cellular automata”. In: *Complex Systems* 15.1 (2004), pp. 1–40.

[66] Integrated DNA Technologies. *Chemical Synthesis and Purification of Oligonucleotides*. https://www.idtdna.com/pages/docs/technical-reports/chemical-synthesis-of-oligonucleotides.pdf. 2005.

[67] C. Thachuk, E. Winfree, and D. Soloveichik. “Leakless DNA strand displacement systems”. In: *Lecture Notes in Computer Science* 9211 (2015), pp. 133–153.

[68] Kyle Lund et al. “Molecular robots guided by prescriptive landscapes”. In: *Nature* 465.7295 (2010), pp. 206–210.

[69] Yu He and David R Liu. “Autonomous multistep organic synthesis in a single isothermal solution mediated by a DNA walker”. In: *Nature nanotechnology* 5.11 (2010), p. 778.

[70] László Lovász. “Random walks on graphs”. In: *Combinatorics, Paul erdos is eighty* 2 (1993), pp. 1–46.

[71] D. Y. Zhang. “Cooperative hybridization of oligonucleotides”. In: *Journal of the American Chemical Society* 133.4 (2010), pp. 1077–1086.

[72] Sungwook Woo and Paul WK Rothemund. “Programmable molecular recognition based on the geometry of DNA nanostructures”. In: *Nature chemistry* 3.8 (2011), pp. 620–627.

[73] Carlos Ernesto Castro et al. “A primer to scaffolded DNA origami”. In: *Nature methods* 8.3 (2011), pp. 221–229.

[74] Shawn M Douglas et al. “Self-assembly of DNA into nanoscale three-dimensional shapes”. In: *Nature* 459.7245 (2009), pp. 414–418.

[75] Mario Teichmann, Enzo Kopperger, and Friedrich C Simmel. “Robustness of localized DNA strand displacement cascades”. In: *ACS nano* 8.8 (2014), pp. 8487–8496.

[76] Petr Šulc et al. “Simulating a burnt-bridges DNA motor with a coarse-grained DNA model”. In: *Natural Computing* 13.4 (2014), pp. 535–547.

[77] William B Sherman and Nadrian C Seeman. “A precisely controlled DNA biped walking device”. In: *Nano letters* 4.7 (2004), pp. 1203–1208.

[78] Jonathan Bath et al. “Mechanism for a directional, processive, and reversible DNA motor”. In: *Small* 5.13 (2009), pp. 1513–1516.

[79] John Fuller Gesteland Atkins and F Raymond. *The RNA world: the nature of modern RNA suggests a prebiotic RNA world*. Tech. rep. 1993.