Computation and Programmability at the Nano-Bio Interface

Jerzy W. Kozyra, BSc (Hons)

Newcastle University

School of Computing Science

A thesis submitted in partial fulfilment
for the degree of Doctor of Philosophy

February 2017

Declaration of Authorship

I, Jerzy Kozyra, declare that this thesis titled, 'Computation and Programmability at the Nano-Bio Interface' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____________________

Date: ____________________

“The most astounding fact is the knowledge that the atoms that comprise life on Earth, the atoms that make up the human body, are traceable to the crucibles that cooked light elements into heavy elements in their core under extreme temperatures and pressures. These stars - the high mass ones among them - went unstable in their later years. They collapsed and then exploded scattering their enriched guts across the galaxy; guts made of carbon, nitrogen, oxygen and all the fundamental ingredients of life itself. These ingredients become part of gas clouds that condense, collapse, form the next generation of solar systems - stars with orbiting planets, and those planets now have the ingredients for life itself.

So that when I look up at the night sky and I know that yes, we are part of this universe, we are in this universe, but perhaps more important than both of those facts is that the Universe is in us. When I reflect on that fact, I look up - many people feel small because they’re small and the Universe is big - but I feel big because my atoms came from those stars. There’s a level of connectivity. That’s really what you want in life, you want to feel connected, you want to feel relevant, you want to feel like you’re a participant in the goings on of activities and events around you. That’s precisely what we are, just by being alive...”

Neil deGrasse Tyson

Abstract

The manipulation of physical reality on the molecular level and construction of devices operating on the nanoscale has been the focal point of nanotechnology. In particular, nanotechnology based on DNA and RNA has a potential to find applications in the field of Synthetic Biology thanks to the inherent compatibility of nucleic acids with biological systems. Scaffolded DNA origami, proposed by P. Rothemund, is one of the leading and most successful methods in which nanostructures are realised through rational programming of short 'staple' oligomers which fold a long single-stranded DNA called the 'scaffold' strand into a variety of desired shapes. DNA origami already has many applications; including intelligent drug delivery, miniaturisation of logic circuits and computation *in vivo*. However, one of the factors that are limiting the complexity, applicability and scalability of this approach is the source of the scaffold which commonly originates from viruses or phages. Furthermore, developing a robust and orthogonal interface between DNA nanotechnology and biological parts remains a significant challenge.

The first part of this thesis tackles these issues by challenging the fundamental assumption in the field, namely that a viral sequence is to be used as the DNA origami scaffold. A method is introduced for *de novo* generation of long synthetic sequences based on De Bruijn sequence, which has been previously proposed in combinatorics. The thesis presents a collection of algorithms which allow the construction of custom-made sequences that are uniquely addressable and biologically orthogonal (i.e. they do not code for any known biological function). Synthetic scaffolds generated by these algorithms are computationally analysed and compared with their natural counterparts with respect to: repetition in sequence, secondary structure and thermodynamic addressability. This also aids the design of wet lab experiments pursuing justification and verification of this novel approach by empirical evidence.

The second part of this thesis discusses the possibility of applying evolutionary optimisation to synthetic DNA sequences under constraints dictated by the biological interface. A multi-strand system is introduced based on an alternative approach to

DNA self-assembly, which relies on strand-displacement cascades, for molecular data storage. The thesis demonstrates how a genetic algorithm can be used to generate viable solutions to this sequence optimisation problem which favours the target self-assembly configuration. Additionally, the kinetics of strand-displacement reactions are analysed with existing coarse-grained DNA models (oxDNA).

This thesis is motivated by the application of scientific computing to problems which lie on the boundary of Computer Science and the fields of DNA Nanotechnology, DNA Computing and Synthetic Biology, and thus I endeavour to the best of my ability to establish this work within the context of these disciplines.

Acknowledgements

There is a quote that bears repeating: “If you want to build a ship, don’t drum up the men to gather wood, divide the work, and give orders. Instead, teach them to yearn for the vast and endless sea.” That is a suitable portrayal of what it felt like to explore science with my supervisor, Natalio Krasnogor. His audacious visions are contagious; they have the power to alter one’s mindset and change assumptions about what is possible. It was truly a pleasure to study with Natalio. I owe him many thanks for his continuous guidance, motivation and patience but also for granting me the opportunity to work on such a challenging and mind-bending project. Regardless of whether or not we accomplish our goals, if we are allowed to dream big, these dreams will prevail strongly in our ambitions - and for that I am grateful.

I would like to thank Harold Fellermann, who always seems to understand what I am saying, even though I am not nearly as eloquent as he is. Harold’s zen-like calmness, in-depth knowledge and clear-thinking make him a great mentor and colleague. I am grateful for his priceless advice and criticism. Indeed, I was very fortunate to be able to work alongside him.

I acknowledge my examiners: Martyn Amos and Jason Steggles for a lot of constructive criticism, as well as valuable discussion and insights.

I would like to thank my colleagues without whom this dissertation would have been a much inferior piece of work: Annunziata Lopiccolo, Alessandro Ceccarelli, Paweł Widera, Daven Sanassy and Charles Winterhalter - for all your aid, advice, encouragement and valuable friendship.

I would like to acknowledge all current and former members of Interdisciplinary Computing and Complex BioSystems (ICOS) research group, especially Nicola Lazzarini, Maria Franco, Jonathan Blakes, Benjamin Shirt-Ediss, Jonathan Naylor, Birgit Koch and Omer Markovitch.

Special thanks to my girlfriend, Natalia Kwiatkowska for her affection and patience. Also, my parents Anna and Wiesław, who are always hugely supportive and, despite the copious and hazardous experiments I carried out as a kid, managed not to squander my inner scientist.

I wish to thank the institutions where I carried out this work, namely Newcastle University and University of Nottingham Computing Science departments. Also, the Engineering and Physical Sciences Research Council (EPSRC) for funding a project I have been involved with: “Towards a Universal Biological-Cell Operating System (AUdACiOuS)” (EP/J004111/1 & EP/J004111/2).

Last but not least, I would like to thank “Davy” Chien-Yi Chang for giving me a life lesson.

Contents

List of Figures

List of Tables

In memory of my brother,
Grzegorz "Tasior" Kozyra

Chapter 1

Introduction

1.1 Background and Motivation

Every cell of every living organism on Earth contains a full set of instructions needed to create, duplicate and make variations of itself. These instructions are stored in chains of polymeric molecules: deoxyribonucleic and ribonucleic acids (commonly known as DNA and RNA). There are many parallels one can draw between nucleic acids and computer programs. Both are a collection of instructions but are executed on different hardware. Computer programs run on electronic machines, while the genetic code is executed on a biochemical hardware of cellular and molecular machinery. In this mechanistic view [1] cells are microscopic processing devices (built from a mere handful of extremely versatile building blocks) each of which is seemingly programmed to express a specific, surviving phenotype. The consequence of this view was a natural desire to effect change in biological systems. It gave rise to the synthetic biology – an entirely new field of science focused on a design of artificial organisms [2, 3]. Synthetic biology views the genetic code as a collection of abstracted units or parts. These parts can be combined together into novel genetic circuits to create new devices with a predictable and controllable behaviour. These devices are able to build new systems of interactions which are very different from those occurring in nature, and potentially allowing expression of much more sophisticated phenomena than what is possible naturally. Thus, the promise of the synthetic biology is to program life that would have never be created by means of natural selection.

Programming DNA sequences allows us to regulate and express genetic information. However, many of the properties which make it so useful as an information storage molecule in nature also let us build nanoscale structures or assemble chemical systems. The growing area of DNA nanotechnology regards DNA as a building material used for arranging and controlling matter on the nanoscale – which can be programmed to self-assemble into stable structures. DNA nanotechnology is now a well-established field with DNA origami being one of the most prominent methods [4]. In DNA origami, a long single-stranded 'scaffold' DNA molecule is folded using multiple short oligonucleotides called 'staples', which bind the scaffold and hold it in place. This simple one-pot technique enables construction of versatile, custom-shaped objects; for instance, DNA nanorobot for intelligent drug delivery [5]. Another fascinating area of research is DNA computing in which DNA is used as a medium for data processing.Because of the relative ease with which molecular interactions can be designed by choosing appropriate nucleic acid sequences, DNA is a prominent substrate for designing artificial reaction networks with designed functionality. In particular, it has been shown that arbitrary chemical reaction networks can be translated into equivalent toehold mediated DNA strand displacement systems [6]. DNA nanotechnology and molecular computing solutions can be readily synthesised and tested *in vitro*. Also, both DNA origami and DNA computing methods were shown to work well in an array of biotic environments, such as insects [7], mammalian cells [8] as well as lysed human cells [9].

The effort of the scientific community is focused on developing essential tools allowing programming cells like robots – to carry out complex and coordinated tasks. The hope is that a new platform may be developed, suitable for the next generation of biological systems; systems including man-made nanodevices operating in the regime of nucleic acids [10, 11]. The ultimate goal is for these devices to be used as nanocomputers of a new kind: performing a new type of computation and information processing [12]. The premise is that a right combination of synthetic biology and DNA nanotechnology could help us achieve this novel kind of computing. Both fields are clearly in an emerging state, but their potential is evident – they hold the promise to advance not only basic science but also medicine, manufacturing and electronics [13, 14]. Major problems facing society might also be addressed via the use of these technologies in a sustainable way – problems in health, energy and environment to name a few. It is imperative that, so as to make rapid advances in those areas, we will need a mastery of both theory and experimentation.

1.2 Problem Statement

To make possible the creation of DNA-based nanocomputers operating *in vivo*, a series of challenging problems has to be tackled first. These problems lie at the intersection of three disciplines: DNA nanotechnology, DNA computing and synthetic biology (Figure 1.1).

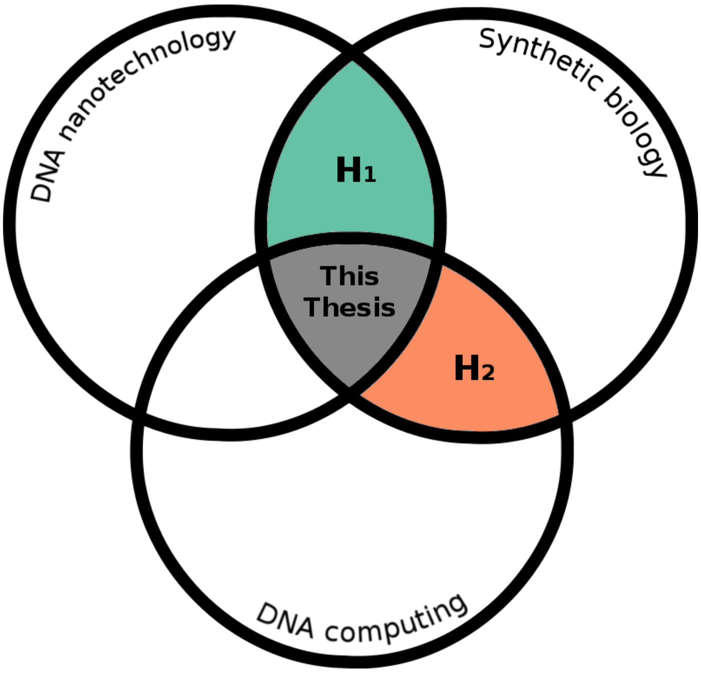


Figure 1.1: Two research hypotheses in a broader context. The intersection of all three fields represents the emerging field of nanobiotechnology (grey area).

The key limiting factor in synthetic biology is the gap between our ability to synthesise DNA and to design biological systems that work well. The problem is partially caused by genetic elements which are often incompatible with biological chassis [15]. On top of that, prior verification of synthetic devices *in vivo* is frequently intractable or simply unavailable [16].

The downside of DNA origami is the cost of synthetic DNA for *in vitro* experiments, which is still too high for viable industrial-scale applications [17]. A potential route to remove this bottleneck may involve employing bacterial cells as factories to manufacture DNA origami – seemingly similar to how viruses use bacterial cells as factories for new DNA [18]. Nevertheless, resorting to viruses lacks biological orthogonality, as bacteriophages infect bacteria, replicate and cause turbid plaques [19]. Previous attempts to fold complex structures *in vivo* may have been hindered by the adoption of viral genomes as scaffolds, thus severely limiting its future applicability as non-interfering nanotechnology platform. It seems that part of the problem is caused by the lack of rules for effective sequence design [20]. The need for a programmable construction of biologically neutral DNA sequences was emphasised recently [21]. However, the current computational methodologies are difficult to apply in the context of DNA origami.

Recent years have seen theoretical designs and molecular implementations of conventional and unconventional circuits in DNA computing. The majority of this work has been concerned with implementing devices such as Boolean logic gates [22, 23]. This approach toward molecular computing, which closely imitates electrical engineering, is somewhat disconnected from *algorithmic* computer science, where algorithms are built by composing structures and actions that operate upon them. DNA is an organic molecule for data storage, yet programmable mechanisms to read and write data *in vivo* are currently lacking. Indeed, DNA computing has so far seen few designs for DNA data structures – with Qian et al.'s theoretical design of a DNA-based stack machine being one noteworthy exception [24].

1.3 Aims and Objectives

In essence, this thesis is concerned with the nucleic acid sequence design problem which is an integral part of research in the area of DNA nanotechnology and DNA computing. The aim is to establish some of the elementary rules for sequence design in order to facilitate the incorporation of these artificial nanoscale systems into synthetic biology. We seek to design and model *in silico* as well as build and test *in vitro* two prototypes: a nanostructure and a nanodevice, both based on synthetic sequences.

In this thesis, we detail the engineering cycle walkthrough by developing computational tools and laboratory protocols for synthetic DNA nanodevices. We strive to get to a stage, where these nanodevices could be safely tested in intracellular context. Ultimately, we endeavour to use this work as a foundation of programmable architectures for next generation of molecular computers and a bridging step between DNA nanotechnologies and synthetic biology.

In biology, a structure and function, more often than not, are two sides of the same coin [25]. Here, we ponder upon two alternative hypotheses: one related to the structure (H_1), another related to the function (H_2).

The first problem we tackle lies at the intersection of DNA nanotechnology and synthetic biology (Figure 1.1 in green). The main objective is to improve DNA origami by using a synthetic scaffold. The synthetic sequence should be designed such that it does not interfere with the bacterial machinery (or interacts minimally) and which allows assembly of DNA origami structures in the most efficient way. This reasoning led me to formulate the following hypothesis:

> H_1 : It is possible to program a synthetic scaffold for DNA origami which is both bio-orthogonal and uniquely addressable.

To eliminate the ambiguity in scaffold addressability (i.e. where staples bind), and to ensure the scaffold sequences are biologically neutral (i.e. "bio-orthogonal") have been key objectives behind this research question. The aspects of biological orthogonality and addressability are covered in detail in Section 2.6.

The second problem lies at the interface of DNA computing and synthetic biology (Figure 1.1 in red). The main aim is to design a DNA device mimicking what in computer science is called a data structure. We argue that DNA self-assembly can be used as a reliable mechanism for storing a collection of elements. Hence, the second hypothesis is stated as:

> H_2 : It is possible to program a synthetic DNA structure allowing recording of data in a controllable and, in principle (albeit not physically), unlimited manner.

The key objective of the study was to design, optimise and characterise the function of such a device.

1.4 Structure of the Dissertation

Synthetic biology differs from other biological research areas – the field embraces approaches normally used in engineering disciplines (see Figure 1.2). The same engineering cycle should also apply to nanotechnology. Here, we seek to establish a similar cycle to address the two working hypotheses.

This dissertation is organised as follows.

Chapter 2 introduces the disciplines of DNA nanotechnology, DNA computing and synthetic biology. In particular, two key systems are explained: scaffolded DNA origami and toehold-mediated strand displacement. The chapter summarises current research efforts and outlook on the potential future applications. Finally, it concludes with some of the concepts which are essential to understanding the motivation behind the thesis as a whole.

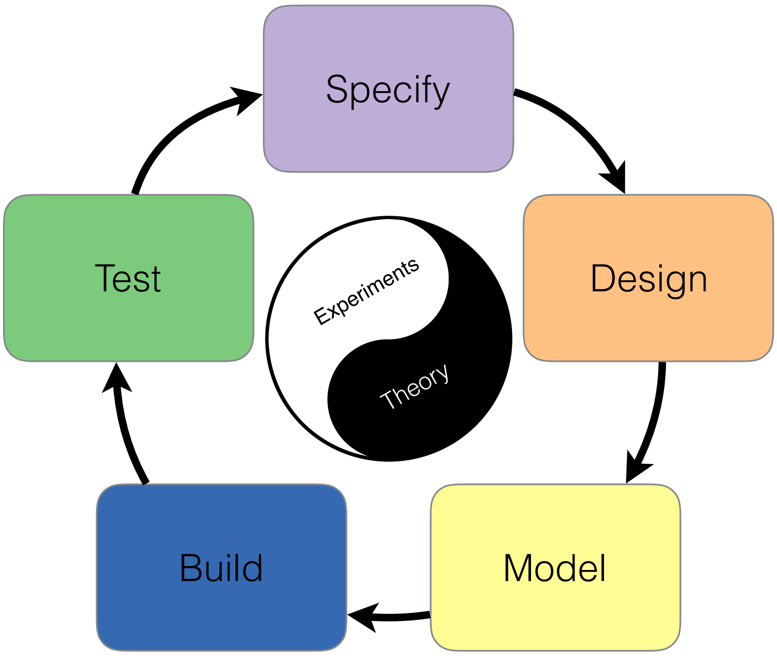


FIGURE 1.2: The engineering life cycle in synthetic biology.

The core of the thesis is composed of two parts, two chapters each. Those parts present two alternative approaches to sequence design: the former originates from combinatorics and graph theory, the latter is based on genetic algorithms. The properties of the two systems are investigated using an array of computational techniques and verified experimentally (i.e. design-build-test-learn cycle).

The first part, aiming to address H_1, begins with Chapter 3 which provides specification and design for the synthetic DNA scaffolds. Chapter 4 evaluates the aspect of biological orthogonality and measures repetitions in existing scaffolds, which affects the thermodynamic addressability. The chapter concludes with a presentation of microscopy images (AFM) of folded DNA origami using both viral and synthetic scaffolds.

The second part investigates a DNA stack machine – the data structure we have chosen as the demonstration for H_2. Chapter 5 covers the key requirements and our proposed design for this nanodevice. Chapter 6 analyses the DNA stack using secondary structure prediction (ViennaRNA) and coarse-grained DNA model (oxDNA) simulations;

finally it contains bioanalyzer spectra as well as microscopy images (TEM) of a DNA stack construct.

Chapter 7 reviews the validity of the proposed hypotheses, discusses the limitations and examines the routes for future work.

1.5 Main Contributions

This dissertation addresses the issue of DNA sequence design at the interface between nano-bio technologies using computing science methods. The proposed solutions can potentially lower the barrier of entry of these technologies into the field of synthetic biology.

In this thesis, we present the design of biologically inert (i.e. "bio-orthogonal") origami scaffolds. The synthetic scaffolds have the additional advantage of being uniquely addressable (unlike biologically derived ones) and hence are better optimised for high-yield folding. We demonstrate the fully synthetic scaffold design with both DNA and RNA origamis and set up a benchmark protocol to produce them. To the best of my knowledge, this work is the first to apply successfully an entirely synthetic scaffold in DNA and RNA origami systems.

Additionally, we present the *in vitro* implementation and experimental characterization of a DNA data structure, namely a *stack*, where data and operations form the core of the molecular interaction network. This design shares similarities with the one presented by Qian et al. [24] but has been optimised for maximal robustness among all molecular interactions and minimal occurrence of undesirable reactions. The stack data structure is here employed as a reversible, and potentially unlimited, data storage, and its recording and readout fidelity is characterised experimentally. This contribution is a stepping stone toward *in vitro* implementations of more general data structures, as well as computationally universal stack machines. We believe that this work provides the first experimental results on a DNA-based stack in particular, and DNA-based data structures in general.

This is a highly interdisciplinary work where I have conducted the computational work in its entirety. In addition, the specification of the De Bruijn system is my own while the specification of the stack data structure was done collaboratively with Annunziata Lopiccolo. The computational analysis is my own work while the laboratory work was done in collaboration with Alessandro Ceccarelli and Annunziata Lopiccolo, who performed the experimental verification of my designs.

1.6 Published Work

Journal article

- Jerzy Kozyra, Alessandro Ceccarelli, Annunziata Lopiccolo, Jing-Ying Gu, Harold Fellermann, Ulrich Stimming, and Natalio Krasnogor. "Designing uniquely addressable bio-orthogonal synthetic scaffolds for DNA and RNA origami", ACS Synthetic Biology (under review)

Conference papers

- Harold Fellermann, Annunziata Lopiccolo, Jerzy Kozyra, and Natalio Krasnogor. "In vitro implementation of a stack data structure based on DNA strand displacement", Proceedings of Unconventional Computation and Natural Computation, Manchester, UK, 2016 (see Reference [26])
- Jerzy Kozyra, Harold Fellermann, Ben Shirt-Ediss, Annunziata Lopiccolo, and Natalio Krasnogor. "Optimizing nucleic acid sequences for a molecular data recorder", Genetic and Evolutionary Computation Conference (GECCO-2017), Berlin, Germany, 2017 (under review)

Others

- Harold Fellermann, Annunziata Lopiccolo, Jerzy Kozyra, Ben Shirt-Ediss, and Natalio Krasnogor. "A DNA-based signal recorder studied in vitro and simulation", Abstract for Czech Chemical Society Symposium Series 14, 2016
- Jerzy Kozyra, Chien-Yi Chang, Alessandro Ceccarelli, Harold Fellermann, and Natalio Krasnogor. "Programming synthetic scaffolds for DNA origami", Extended abstract printed for ECAL satellite workshop: Toward Programmable Biology, pp. 12-13, York, UK, 2015

Chapter 2

Background and Related Work

This brief chapter sets the stage for the results which follow; situating the two systems investigated in this thesis. We explain the current perspectives in the literature, the computational and experimental techniques which may be applied and some of the distinctions we make between the essential concepts in the fields of nanotechnology and synthetic biology as they relate to artificial nanodevices and nanocomputers in general.

2.1 Introduction

According to the “RNA world” hypothesis, the first self-replicating RNA has formed in a primordial soup several billion years ago. It emerged, multiplied and evolved through a series of coincidences and led to the creation of DNA and proteins [27]. Although the details of this process remain unknown, certainly nucleic acids played a key role in the emergence of the complex biochemistry which we now recognise as life. Living systems utilise DNA as a stable, ”high-tech” archive of genetic information which stores blueprints of the most successful RNA and proteins. The central dogma of molecular biology describes the transfer of sequence information: DNA to DNA (through replication), DNA to mRNA (transcription) and mRNA to proteins (translation). Intrinsic self-organisation of these molecules lies at the core of the complexity and diversity of life forms on Earth. Living cells evolved to replicate expeditiously and independently. They can sense and react to external stimuli as well as their internal state often in a very intricate fashion. Another level of complexity arises in a multicellular life where specialised cells are elaborately orchestrated, for instance, in bacterial colonies or mammalian tissues. Peculiarly, what we don't yet know about biology vastly outnumbers that what we do know. Indeed, the groundbreaking research on a minimal bacterial genome syn3.0 [28] (i.e. genome containing only the genes necessary for life) has determined that almost a third of its genes have no known function; yet they are essential for life [29].

To gain a better insight into how biological entities work, it is reasonable to make tools that are similar in size. This way it will be easier to take apart these nano-assemblies and understand how they function (and the reasons why they might break). Likewise, it might be possible to use these unconventional tools to fix problems and maybe even make gradual improvements to existing systems.

It is believed that in the near future these tools (or nanodevices) will be performing variety of vital roles in the human body, including improvement of the respiratory system (respirocytes), reversing the ageing process, delivering target-specific drugs or even repairing damaged DNA which encodes our genome (to a greater extent than is allowed by modern genetic engineering)

The study of deoxyribonucleic acid, or simply DNA, came a long way since the first major discovery by Watson and Crick about its intrinsic and regular structure of double helix [31] (Figure 2.1). Based on the x-ray images provided by Rosalind Franklin they proposed a model which accurately predicts arrangement of atoms in the most common B-form of DNA. Hardly anyone could suspect at the time that study of DNA would evolve so quickly into a broad branch of science on its own and embrace multidisciplinary approaches to the subject. The same principles and mechanics researchers started to grasp half a century ago, are now used to create nanostructures and nanodevices which are thousands time smaller than the width of human hair.

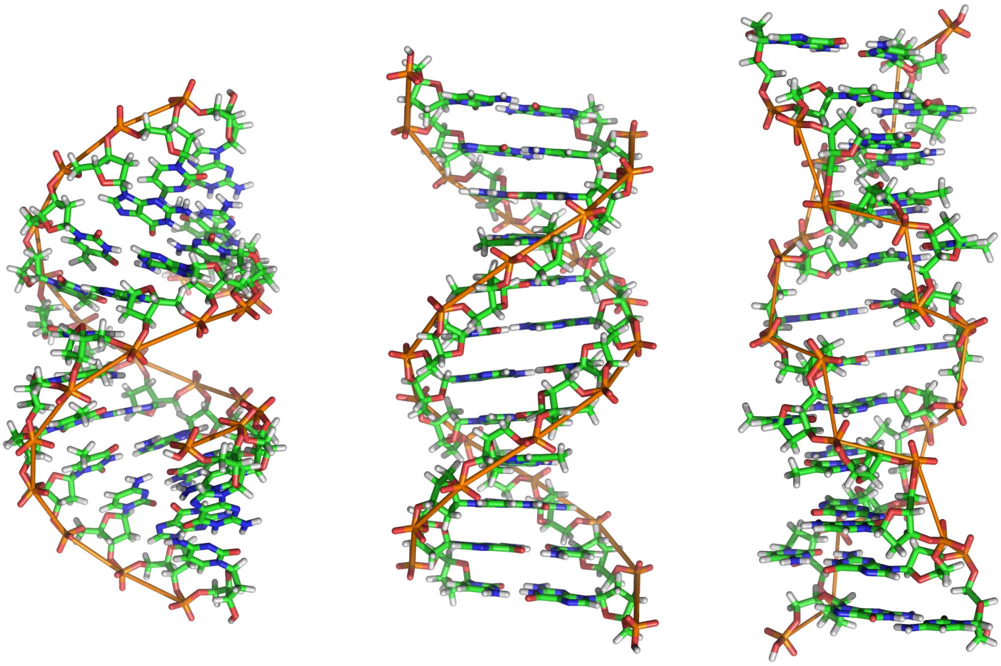


FIGURE 2.1: Structure of DNA: from left to right: A-, B- and Z-DNA. The most common structure is B-DNA which is found in cells and other aqueous environments. Dehydrated samples typically adopt A-DNA form – a similar helical structure occurs in double-stranded RNA and in DNA-RNA hybrid. The Z-DNA is rare as it occurs when DNA interacts with certain proteins. (Taken from Reference [30]).

Similarly, it is hard to predict how bionanotechnology is going to look like and what kind of applications are yet to be seen in the future.

2.2 Molecular Properties of DNA and RNA

Although DNA and RNA both carry genetic information, there are quite a few crucial differences between them (see Figure 2.2 and Table 2.1). In DNA the sugar backbone consists of deoxyribose; in RNA the sugar is ribose. The sugar type is partially responsible for the stability of the nucleic acid molecule. Typically, DNA forms a double-stranded helix, while RNA is usually single stranded.

DNA is relatively easy to work with: enzymes such us ligase, polymerase and restriction enzymes are used to glue, copy and cut sequences at desired places. Additionally, rapid and inexpensive access to short fragments of DNA is available thanks to oligonucleotide chemical synthesis. Last and probably the most important is a fundamental

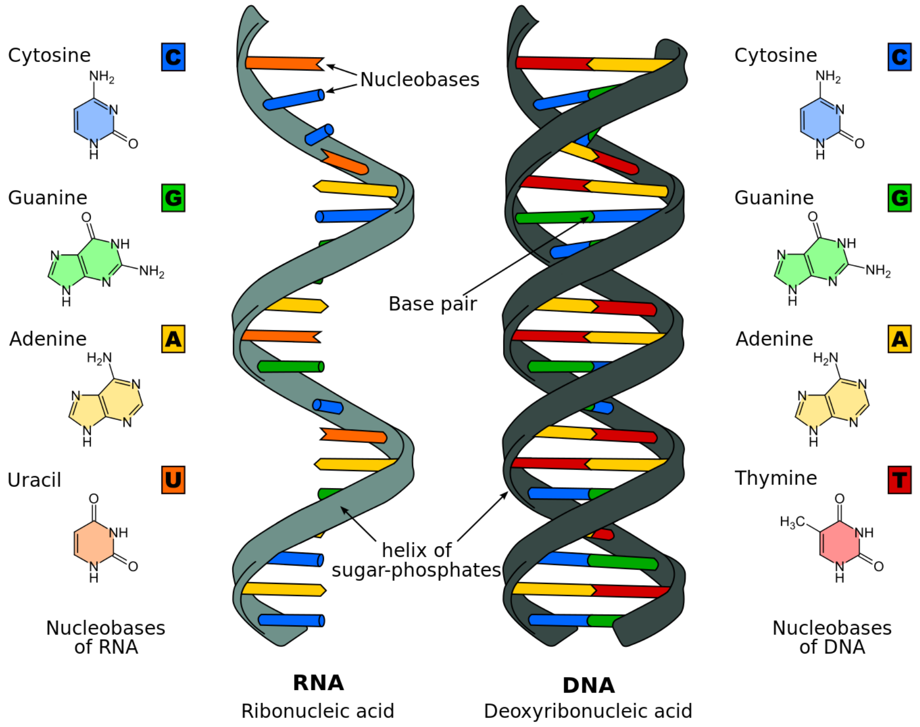


Figure 2.2: Differences in composition and structure between DNA and RNA. (Taken from Reference [32]).

principle of complementary base pairing which states that single strands of DNA self-assemble into double helices using the elegant recognition of Watson-Crick bindings. Simply put, a sequence composed of four nucleotides: A, C, G and T (standing for adenine, cytosine, guanine and thymine) create the strongest bonds with its perfect complement; the reason for that being A interacts only with T while C only interacts with G on the other strand. For instance, a sequence $S_1 = 3'$-GTAGGACTTC-$5'$ binds most strongly to complementary sequence $S_2 = 5'$-CATCCTGAAG-$3'$ and normally prefers it over less attractive $S_3 = 5'$-CATCC$\underline{\text{C}}$GAAG-$3'$. Note that strands run in opposite directions (i.e. $5'$ and $3'$ markers).

A secondary structure of RNA and DNA is defined as a set of nucleotides that form base pairs. Strands of RNA and DNA can form various secondary structure motifs which are shown in Figure 2.3.

The ordering of binding strengths is highly dependent on the composition and length of given sequences. Although not perfect, an approximation using the Hamming distance provides an adequate indication of those attraction forces. Furthermore, it is

	RNA	**DNA**
Main function	Transfer of genetic information to produce proteins	Stable storage of genetic information
Structure	A-form, single-stranded	B-form, double-helix
Bases	**A**denine, **G**uanine, **C**ytosine, **U**racil	**A**denine, **G**uanine, **C**ytosine, **T**hymine
Base Pairing	**A-U** **C-G**	**A-T** **C-G**
Stability	Unstable More reactive	Stable Less reactive
Location	Found in cell nucleus cytoplasm and ribosome	Found in cell nucleus and mitochondria
Propagation	Synthesised from DNA when needed	Self-replicating

Table 2.1: Overview of the main difference between RNA and DNA.

worth mentioning that binding energy is correlated more or less linearly with Hamming distance, which is equal to the number of Watson-Crick mismatches between the two strands. On the other hand, the likelihood of strands binding together, as measured by the equilibrium constant, decreases exponentially when Hamming distance is increased. That property allows a creation of DNA parts of astonishing specificity. More detailed explanation of DNA thermodynamics is provided in Section 2.3.

2.3 Modelling DNA and RNA

Many processes involving DNA, including hybridisation, can be elegantly described by thermodynamic equations. Two single-stranded oligonucleotides A and B form a duplex AB:

$$\mathrm{A} + \mathrm{B} \rightleftharpoons \mathrm{AB} \tag{2.1}$$

The equilibrium constant of this reaction is given by:

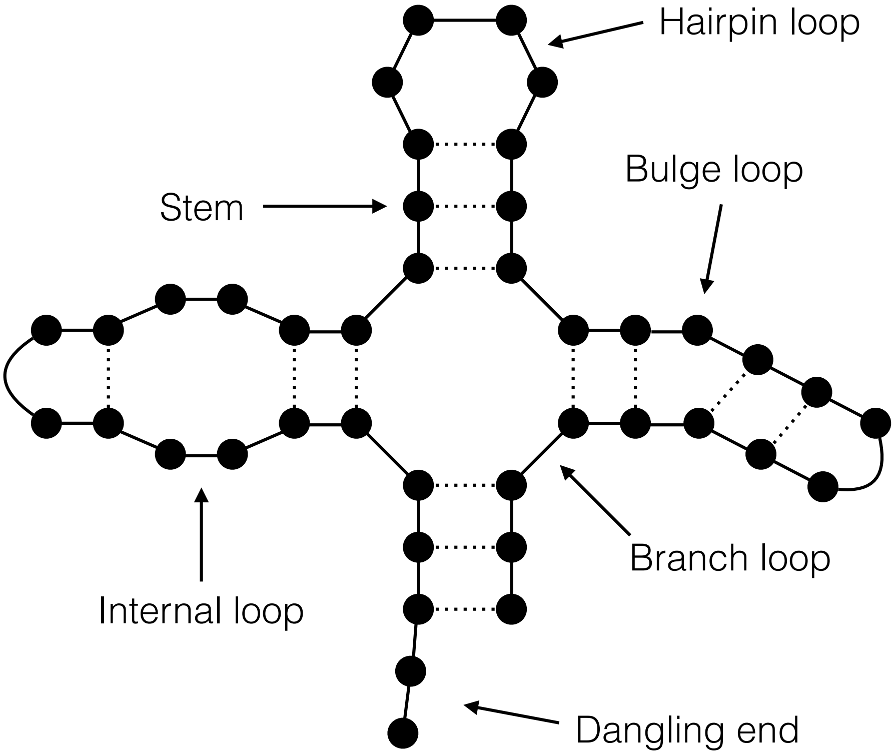


Figure 2.3: Secondary structure motifs in RNA and DNA. The dots represent bases which are connected to the backbone (solid lines) and form hydrogen bonds (dotted lines).

$$K = \frac{[A][B]}{[AB]} \tag{2.2}$$

where $[A]$, $[B]$ and $[AB]$ are concentrations of respective species in the solution.

The stability of a duplex depends on nucleic acid concentration but also on the temperature and the buffer solution.

Derived from the Van 't Hoff equation:

$$\Delta G = -RTlnK \tag{2.3}$$

where R is the ideal gas constant, and T is the temperature. This equation gives a Gibbs free energy (ΔG) – a thermodynamic potential which is commonly used as an indicator of stability for DNA and RNA complexes. The lower the free energy, the more stable the structure is. Another way to express Gibbs free energy

$$\Delta G = \Delta H - T\Delta S \tag{2.4}$$

where ΔH is the change in enthalpy, T is the temperature, and ΔS is the change in entropy.

Another indicator of stability is the melting temperature:

$$T_m = -\frac{\Delta G}{Rln/\alpha} \tag{2.5}$$

where R is the ideal gas constant, and ΔG is Gibbs free energy. Parameter α is set to one for self-complementary strands, and to four for non-self-complementary strands. Melting temperature defines a temperature at which a single and double strands are in 1:1 ratio. The higher the melting temperature is, the more stable the structure is.

Thermodynamic parameters ΔG, ΔH, ΔS for any nucleic acids sequence can be predicted with high accuracy with the nearest-neighbor model of SantaLucia [33]. The interaction between bases on different strands depends on the neighbouring bases (since the stacking interactions are stronger than hydrogen bonds). Instead of treating a DNA helix as a string of interactions between base pairs, the nearest-neighbor model treats a DNA duplex as a string of interactions between 'pairs' of base pairs. Experimental verification of the model allowed deriving all thermodynamic parameters for DNA, RNA and DNA/RNA hybrid duplexes. These parameters were, in turn, used to create various software packages for modelling nucleic acids, such as Mfold [34], NUPACK [35]; and also fully atomistic models. Here we describe two of them: ViennaRNA for secondary structure prediction and MFE calculation [36]; and oxDNA which is a coarse-grained DNA model [37].

2.3.1 ViennaRNA

The ViennaRNA Package consists of libraries and programs for the prediction and comparison of RNA and DNA secondary structures [36]. The secondary structure prediction of nucleic acids is achieved through energy minimization. Vienna RNA provides three kinds of algorithms which are based on dynamic programming: (1) the minimum free energy algorithm calculating the optimal structure for a single-stranded species [38]; (2) the partition function algorithm for the base pair probabilities in the thermodynamic ensemble [39]; (3) and the suboptimal folding algorithm [40] which finds suboptimal structures within a given range of the optimal energy.

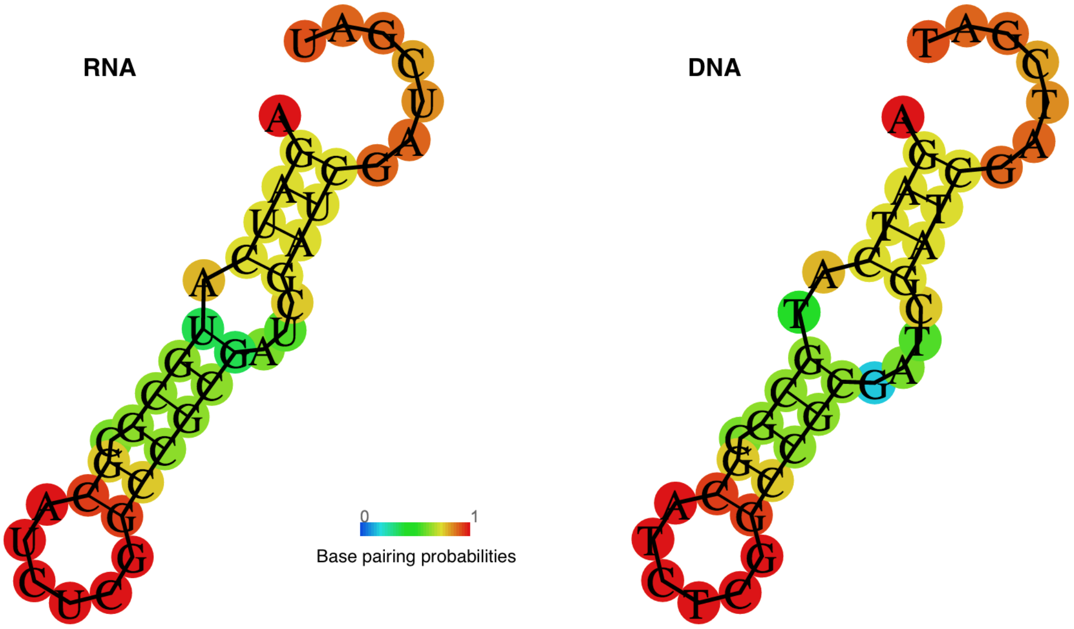


Figure 2.4: Typical output of the RNAfold program. The structures are coloured by base-pairing probabilities. For the unpaired regions, the color denotes the probability of being unpaired.

The two programs that were used most often throughout this research were: RNAfold, which predicts secondary structures of single-stranded RNA and DNA sequences; as well as RNAcofold, which predicts dimer formation between two sequences. Besides the value of a minimum free energy, the software also provides the base-pairing probability and positional entropy. The typical output of the program is shown below (i.e.

single-stranded RNA and its secondary structure prediction in dot-bracket notation):

```
AGAUCAUGCGCGCAUCUCGGCCGCGAUCGAUCGAUCGAU
.((((.((((.((......))))))...)))).......    MFE: -10.40 kcal/mol
```

In this notation, the dots correspond to unpaired nucleotides while the brackets denote the base pairs. The graphical output for the same input sequence is shown in Figure 2.4.

2.3.2 oxDNA Model

oxDNA is a simulation code which was recently developed to implement the coarse-grained DNA model introduced by Ouldridge et al. [41, 42] at the University of Oxford. It includes extendable simulation and analysis framework and natively supports DNA, RNA, Lennard-Jones and patchy particle simulations on both CPUs and NVIDIA GPUs.

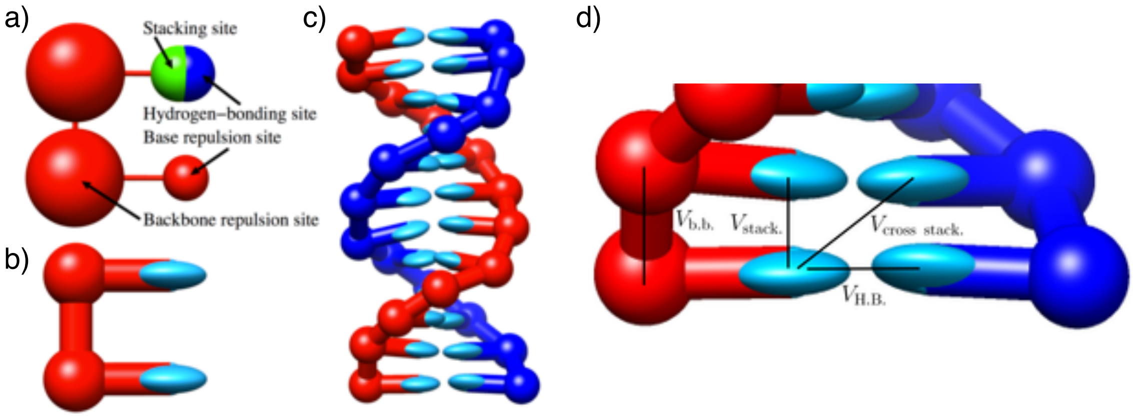


FIGURE 2.5: The oxDNA model: (a) abstracted molecules; (b) flat nucleobases which indicate stacking direction; (c) an example of a DNA duplex; (d) some of the captured interactions.

The oxDNA model offers a coarse-grained approach for simulations of DNA. In this model, DNA molecules are represented as chains of backbone molecules and nucleobases (see Figure 2.5a-c). The captured interactions (presented in Figure 2.5d) were parametrised to produce the well-known double-helical structure of DNA. Specifically, the interactions are:

1. sugar-phosphate backbone connectivity,
2. excluded volume,
3. hydrogen bonding,
4. nearest-neighbour stacking,
5. cross-stacking between base-pair steps in a duplex,
6. coaxial stacking

oxDNA focuses on interactions at the nucleotide level that allow the self-assembly processes associated with DNA nanotechnology to be studied [37]. Recently the model was extended with explicit major and minor grooves and modified coaxial stacking and backbone-backbone interactions, which allows the model to treat large (kilobase-pair) structures [43]. Moreover, the model can be parameterised to a range of salt concentrations. In this thesis, the oxDNA model is used to study the operation of DNA stack (see Section 6.3).

2.4 Experimental Techniques

An array of laboratory techniques exists which is extremely helpful to observe objects that cannot be seen with the naked eye. This section describes some of the standard investigation methods in microbiology and nanotechnology.

2.4.1 Gel Electrophoresis and On-chip Electrophoresis

Gel electrophoresis is a standard technique used in molecular biology allowing separation and analysis of DNA based on its size. Nucleic acids are negatively charged molecules – when placed in an electric field DNA migrates towards the positive pole. When the electrophoresis is performed in a gel medium such as agarose or polyacrylamide [44] the migration speed is influenced by the *size* of the migrating molecule. In a mixed-size population of DNA, shorter molecules move faster and migrate farther than longer ones, because shorter molecules migrate more easily through the dense network of pores in the gel. This phenomenon is called sieving and it allows separation and sorting of DNA fragments.

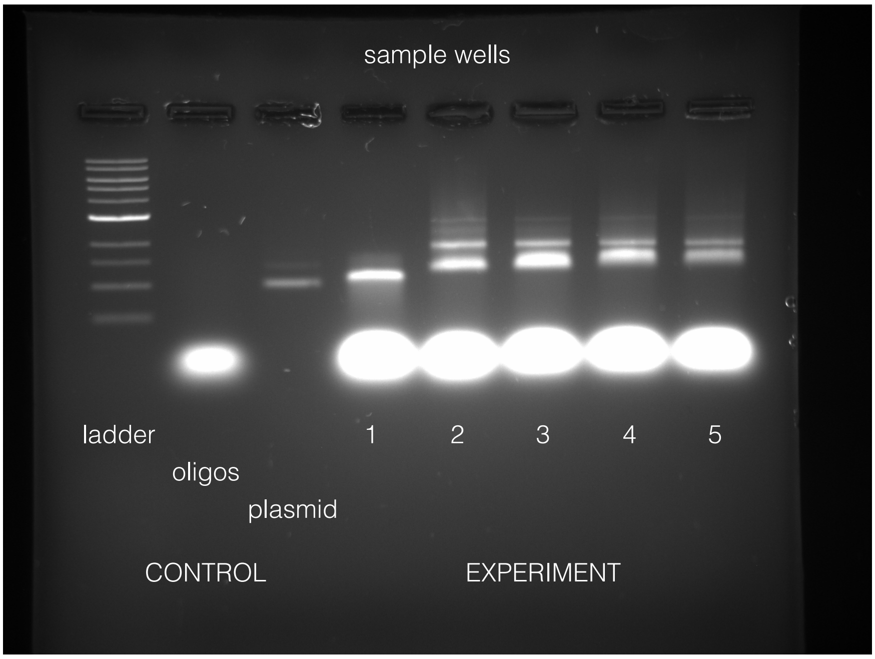


Figure 2.6: Gel electrophoresis

One example of gel electrophoresis is shown in Figure 2.6. Initially, the mixed solutions of DNA are placed in the sample wells (top of Figure 2.6) such that each lane contains one particular sample. The leftmost lane contains the DNA ladder, a standardised solution of various DNA fragments of known length, for the purpose of comparison. The remaining control and experimental samples are loaded into adjacent wells and run in parallel in their individual lanes. DNA fragments of the same length migrate together and therefore they are grouped into a single horizontal band visible on the gel. Depending on the sample, each lane shows separation of DNA as one or more distinct bands. For example, short DNA oligonucleotides travel the fastest and they appear on the bottom of the gel image, as opposed to a longer plasmid whose migration rate is lower. The brightness of a particular band depends on the amount and type of a fluorescent dye incorporated into DNA for visualisation. For example, larger concentrations of DNA absorb more dye and therefore appears brighter; also, single-stranded DNA usually absorbs less dye than double-stranded DNA and therefore appears dimmer.

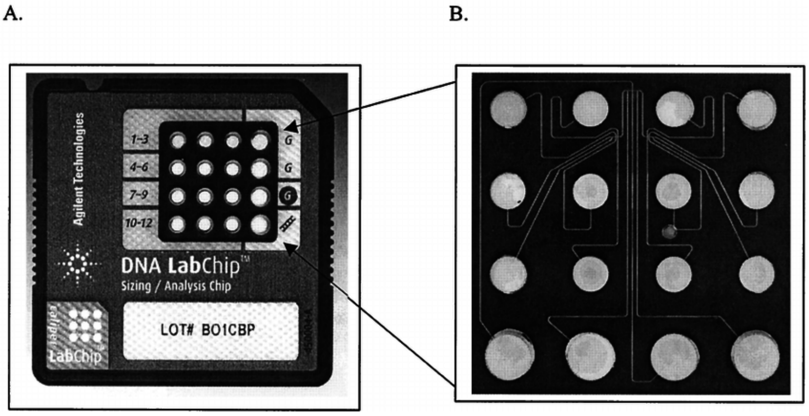


FIGURE 2.7: LabChip 7500 used in the Agilent 2100 Bioanalyzer. (A) Top side of the chip showing layout of marker and sample wells. (B) The chip performs capillary electrophoresis in a series of micro-fabricated channels (taken from [45]).

Based on the same principle, on-chip electrophoresis offers another approach towards an advanced qualitative analysis. The DNA "High Sensitivity Chip" provided by Agilent Technology allows size measurement, quantification and quality control of nucleic acids on a single platform. The Bioanalyzer system (see Figure 2.7) can be used to obtain quantitative and qualitative measurements of short DNA strands. The kit provides lower and higher markers (35bp and 10Kb) which the software uses to align sample solutions with a DNA ladder of known composition that is run in a separate lane (the ladder range is 50-7000 bp). The raw data is measured and displayed in the form of an electropherogram that plots the arbitrary fluorescent units displayed against either migration time or predicted fragment size.

In this study, the Agilent High sensitivity DNA kit was used to assay the states of the DNA nanodevice (see Section 6.4.1).

2.4.2 Atomic Force Microscope (AFM)

Atomic Force Microscope (AFM) is a type of a scanning probe microscope. AFM allows imaging with a very high resolution, typically on the order of fractions of

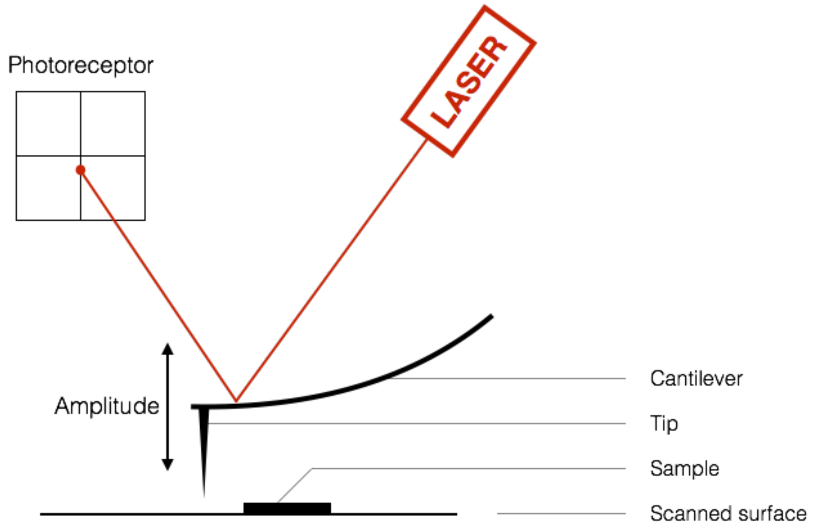


Figure 2.8: AFM schematics

a nanometre (i.e. size of individual atoms). In principle, AFM works due to the intermolecular forces between the microscope's probe and the surface being imaged. The key element of AFM is a small cantilever with a sharp tip that oscillates up and down over the sampling surface (Figure 2.8). The laser points at the tip and its beam is differentially refracted according to the cantilever oscillations. The tip can sense depth fluctuations over the background, thus creating a shift in the z-axis of the laser beam. When scanning the surface of interest, a topography software is used to reconstitute 3D images of the sampling units.

The main advantage of AFM over classic scanning microscopy is the ability to create 3D images. Its resolution can also be increased in liquid environments, but it also imposes high samples purity to minimise detection of noise over the background. Inherent from the resolution achieved via AFM, its computer processing requires more time than usual scanning microscopes and it can take hours to get a refined image of a certain sampling area.

In this thesis, AFM is used to observe folded DNA origami (see Section 4.4).

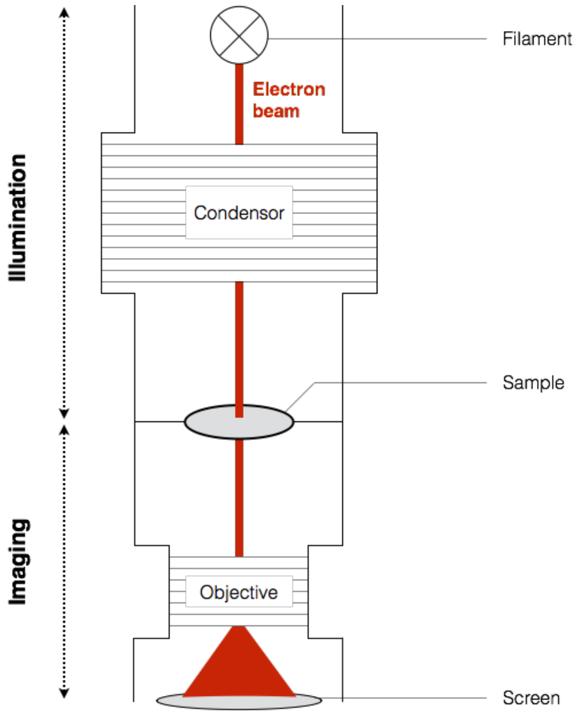


Figure 2.9: TEM schematics

2.4.3 Transmission Electron Microscope (TEM)

Transmission Electron Microscope (TEM) was inspired by a classic light microscope; however, it uses a beam of electrons instead of light. Imaging under TEM can reach the resolution of a few angstroms (10^-10 m), which corresponds to a 100,000 times magnification. In perspective, if one stands at the bottom of Mount Everest, a TEM could visualise a rock that is at the mountain highest peak. With a filament of very high intensity (about 100 kV), a powerful electron beam is generated and focused into a thinner beam via condensing lenses (Figure 2.9). In the illumination part, focused electrons travel through the sampling unit and are affected by the studied specimen (they are scattered by dense objects). Electron beams pass into the objective lenses and the unscattered electrons move towards the screen. Shadows of the objects detected in the sample can be distinguished from the background; the denser objects are easier to recognise.

TEM is a very powerful instrument that provides incredible detail of the sampling units. However, prolonged illumination of the same area within a sample may lead to its degradation due to the high capacity of the electron beam and relative dimensions of the sample. The downside of electron microscopes relies on the instrument parameterisation and maintenance; for instance, samples need to be loaded onto vacuumed carbon grids specially coated to optimise the signal detection. In addition, TEM requires a heavy set of lab instruments and targeted microscopy expertise.

In this study, we use TEM for the readout of DNA nanorecorder (see Section 6.4.2).

2.5 Literature Survey

In the late 1950s, Richard Feynman first proposed the idea of using living cells and molecular complexes for a construction of “sub-microscopic computers”. In his famous talk “There’s Plenty of Room at the Bottom” [46], Feynman considered the problem of “manipulating and controlling things on a small scale”, which established the foundation of nanotechnology. Although he focused primarily on information storage and molecular manipulation, Feynman highlighted the potential for biological systems to act as small-scale information processors:

> The biological example of writing information on a small scale has inspired me to think of something that should be possible. Biology is not simply writing information; it is doing something about it. A biological system can be exceedingly small. Many of the cells are very tiny, but they are very active; they manufacture various substances; they walk around; they wiggle; and they do all kinds of marvelous things all on a very small scale. Also, they store information. Consider the possibility that we too can make a thing very small which does what we want that we can manufacture an object that maneuvers at that level! [46].

Nowadays, scientists use DNA as a material of choice for building things on a nanoscale. The purpose of this section is to provide a concise survey about DNA as a nanomaterial and how it has been used over the years in fields of DNA nanotechnology, DNA computing and synthetic biology.

2.5.1 DNA Nanotechnology

Currently, researchers show that various capable nanotools can be created using DNA and RNA molecules. Regular structure of nucleic acids makes them an appealing building material for supramolecular assemblies (see Figure 2.10). In addition, they can be used for the realisation of molecular machines and computers (see Section 2.5.2).

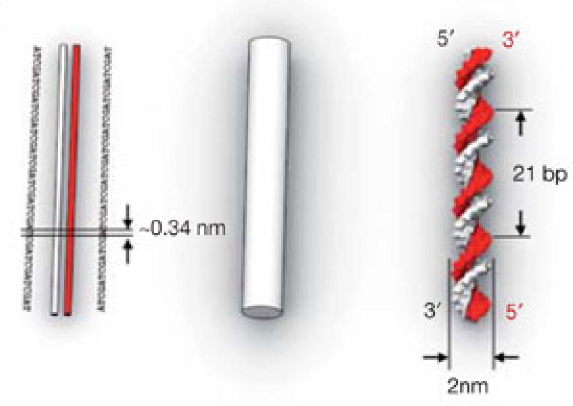


FIGURE 2.10: Regular structure of DNA B-form: complementary strands of double helices running anti-parallel to each other have a diameter of 2 nm with a twist of 10.5 base pairs (right panel). Nucleotides are roughly $\frac{1}{3}$ nm wide (left panel). As a simplified representation of double helix, a solid cylinder could be used (middle panel). Adapted by permission from Macmillan Publishers Ltd: Nature Methods [47], copyright 2011.

Brief History

In the early 1980s N. Seeman, now considered a father of DNA nanotechnology, came up with an innovative proposal to construct complex multidimensional objects and lattices using DNA [48]. Instead of the usual linear duplexes that are formed by nucleic acids, he based his design on a branched architecture. At the time, the existence of branched DNA structures, such us Holliday junctions, was already known. Inspired by the woodcut *Depth* of M. C. Escher (Figure 2.11 left), Seeman envisioned organising matter on the nanoscale in a similar manner. He designed a set of predefined sequences such that they assemble into immobile junctions with sticky ends. That, in turn, would result in the formation of more complex geometric objects, such as lattices (shown in Figure 2.11 right).

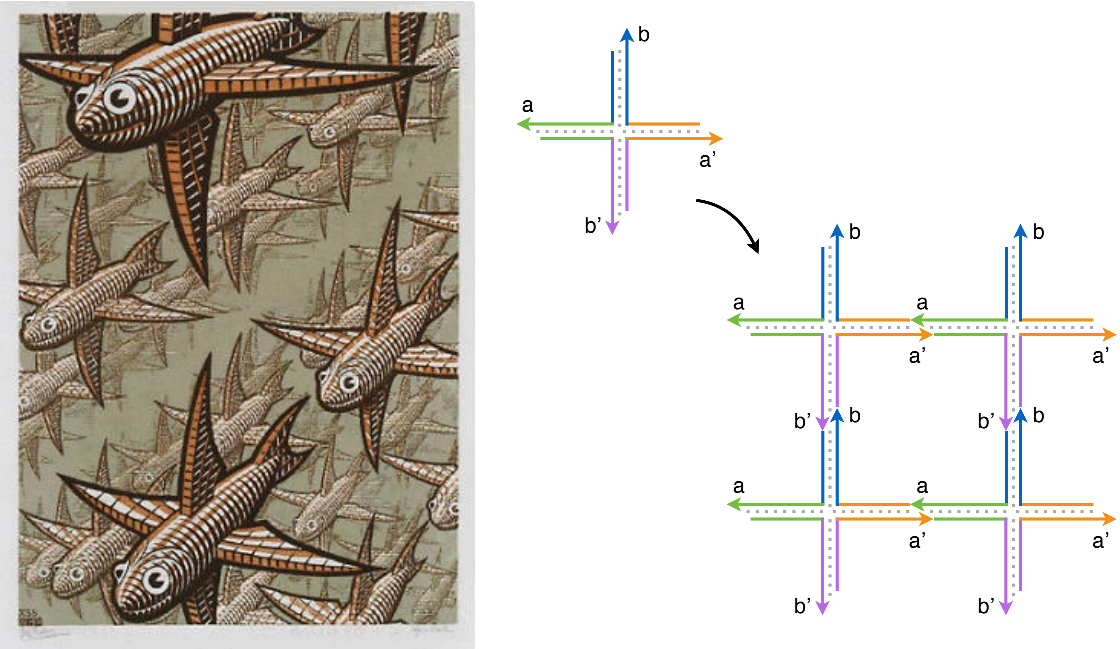


Figure 2.11: *Depth* by M. C. Esher (left) and two-dimensional lattice formed from immobile junctions with sticky ends (right). Sequences *a* and *a'*, *b* and *b'* are complementary to each other.

Since then, DNA has been used to build increasingly complex structures: two-dimensional lattices [49–51] and three-dimensional objects [52–54]. Also, the RNA tectosquares units were shown to self-assemble into arrays and patterns [55].

DNA Origami

One of the problems with DNA nanotechnology was that construction of relatively complex structures required interactions between a vast number of small DNA strands. In consequence, the yield of self-assembly was highly sensitive to stoichiometry (i.e. relative ratios of DNA strands). The synthesis of nanostructures was, therefore, a laborious process requiring multiple reaction steps and purifications.

The revolutionary idea came in 2006 when P. Rothemund presented scaffolded DNA origami. DNA origami quickly became one of the most promising methods of matter arrangement in DNA nanotechnology. In DNA origami, a long single-stranded scaffold DNA molecule is folded using multiple short DNA strands which bind it at target locations and hold it in place. A single scaffold folding a single shape has proved to provide a staggering robustness and greatly simplified the required laboratory work.

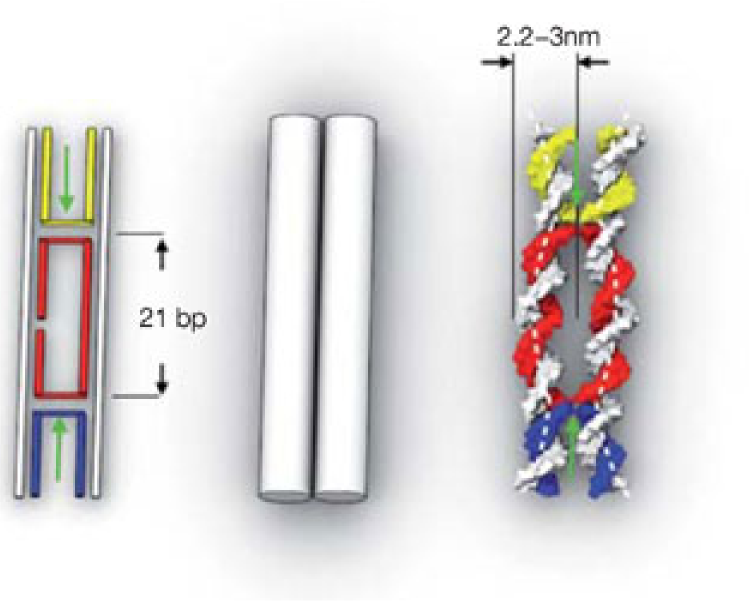


FIGURE 2.12: Basic building block in DNA origami: two double-helices running parallel to each other are joined by the staple crossovers 21 base pairs apart, here marked by green arrows (left and right panels). Electrostatic repulsion is believed to cause bending of helices (white dotted lines). The crossovers are hidden in the cylindrical representation (middle panel). Adapted by permission from Macmillan Publishers Ltd: Nature Methods [47], copyright 2011.

The elementary building block in DNA origami, which one can think of as a molecular brick of specified dimensions, is composed of double-helical domains stacked together and interlinked by anti-parallel periodic crossovers (see Figure 2.12). Those crossovers are allowed by U-turns of the phosphate backbone and could only be placed at the tangent point between parallel helices.

DNA Origami Construction Phase

The design process can be thought of as developing a blueprint for any other physical object (the difference being that instead of bricks one uses DNA domains).

All structures follow similar design algorithm, shown in Figure 2.13. The steps are as follows:

1. The scaffold strand is laid down so that an approximated shape is obtained.
2. The potential positions of scaffold double crossovers are determined. Usually, these are then placed in the centre (i.e. a seam across the shape is created).

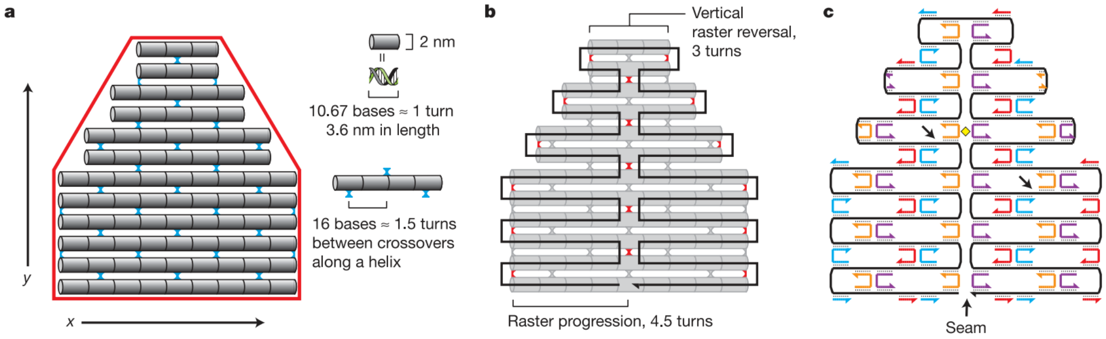


FIGURE 2.13: Design pipeline: (a) firstly a desired shape (red outline) is raster-filled with cylinders representing double helices; scaffold crossovers are placed 1.5 turns along the altering sides. (b) Scaffold path (black) is guided through the shape and staple crossover places are determined (grey) bearing in mind minor-major groves of DNA. (c) Initially, most staples binding to the backbone (coloured) are 16-mers. Adapted by permission from Macmillan Publishers Ltd: Nature [4], copyright 2006.

3. Staple strands are added. At first, most of the strands are 16-bp long and have two arms of equal length.

4. In case there is no obvious continuation (mostly because there is no neighbouring strand) a single-domain staple may be introduced.

5. The exact arrangement of staples is chosen in such a way as to maximise the number of strand crossovers: the feasible location is dictated by the tangent point between neighbouring helices.

6. A subset of staples, called “bridged staples”, may be spanning across the backbone seam and holding distant regions of the scaffold.

7. Finally, some adjacent staples may be merged to form a longer staple with multiple arms (which has an influence on the nanostructure stability).

Applications

This novel method enables versatile construction of custom-shaped objects with nanometre precision [4] (Figure 2.14a). A range of impressive nanostructures has been conceived, such as tetrahedron [56] (Fig. 2.14c, top); cube [57] (Fig. 2.14c, bottom); regular multi-layer solids [58] (Fig. 2.14d); bent bars and gears [59] (Fig. 2.14f-g). as well as tensegrity objects [60] (Fig. 2.14h). Objects constructed in that way follow slightly altered design principles (i.e. shifted crossover points - the honeycomb arrangement imposes a threefold symmetry between neighbouring helices). Furthermore, insertion and deletions of base pairs can be used to induce the global twist: bent and curved parts obtained that way allow building nonlinear objects (Fig. 2.14e-g). DNA origami being a bottom-up method of patterning can be combined with top-down methods by treating origami objects as molecular tiles [61]. In this approach, crystalline origami arrays are assembled from hundreds of smaller cross-shaped objects (Fig. 2.14b).

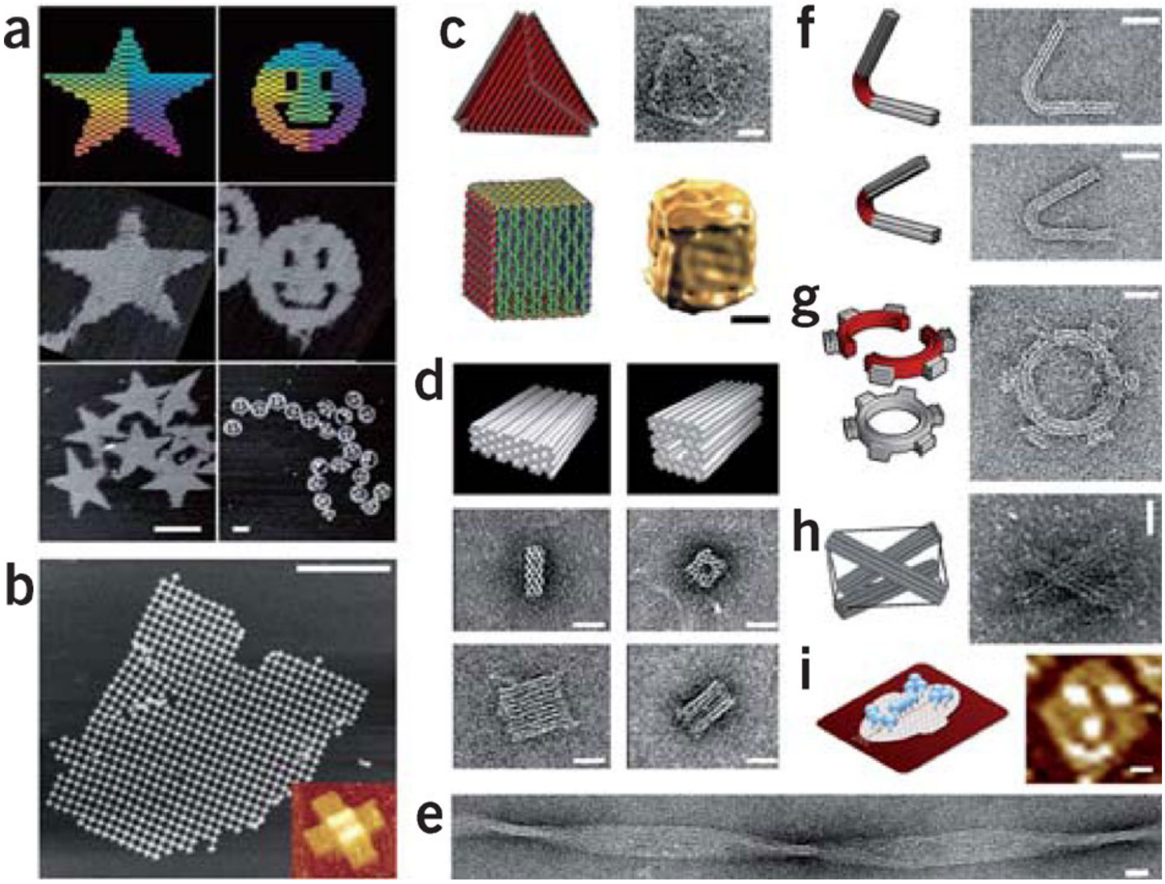


FIGURE 2.14: Examples of DNA origami: single-layer DNA origami design of star and smiley with AFM images (a); crystalline two-dimensional arrays (b); three-dimensional containers: tetrahedron and cube (c); regular multi-layer objects (d) and long object with global twist deformation (e), bent bars (f) and gears (g); tensegrity structure (h); origami with site-detected protein attachment (i). Scale bars: 100 nm (a), 1000 nm (b), 20 nm (c-i). Reprinted by permission from Macmillan Publishers Ltd: Nature Methods [47], copyright 2011.

The tetrahedron and cube, mentioned above, are composed of four triangular and six quadrangular flat faces, respectively. Those faces are held together by additional staples across their edges. As a result, molecular containers were constructed with interior compartments suggesting that DNA origami could serve as a cargo carrying nanostructure. Moreover, in the cube design one of the faces is designated to act as a lid: externally supplied DNA 'keys' could dynamically open the box through strand displacement reactions (described in Section2.5.2). Other structures with controllable behaviour include diffusive molecular cargo [62], dynamic mechanisms [63] and nanorobots [5].

This particular nanorobot is an autonomous device which can carry molecular cargo to target cells [5]; the device is essentially an open-ended barrel with two lock duplexes which could be opened in the presence of particular aptamers 'keys' (i.e. oligonucleotides or peptides binding specific DNA sequence). When the lock is opened, the nanorobot is reconfigured exposing the internal payload it is carrying. Several different lock-key combinations were implemented to show controlled functionality and specificity of the nanorobot. This nanodevice was able to deliver molecular cargo to cancerous cell lines (present in a culture of healthy cells) with astonishing specificity.

The community has also seen an implementation of mechanical parts that display rotational [64, 65] as well as the translational movement [66, 67]. These examples have laid the foundation for mechanically active nanomachines that can generate, transmit, and respond to physical cues in molecular systems [68, 69].

Biological Compatibility

Recent research on scaffolded DNA origami has been used to design a range of naturally bio-compatible materials. Assembled structures were used as two-dimensional origami landscapes for molecular robots [70]; DNA nanochips for direct analysis of single enzymes through atomic force microscopy [71]; and observation of chemical reactions involving single molecule [72]. The possibility of attaching site-directed proteins has been explored [73] (see Figure 2.14i). Apart from those applications, a cellular scale of DNA structures, actively studied chemistries and well developed enzymatic procedures on top of easily modifiable functionalities makes DNA origami an attractive candidate for cellular studies. DNA origami was shown to be stable in the cell lysate solution obtained from normal and cancerous cell lines [9] (see Figure 2.15).

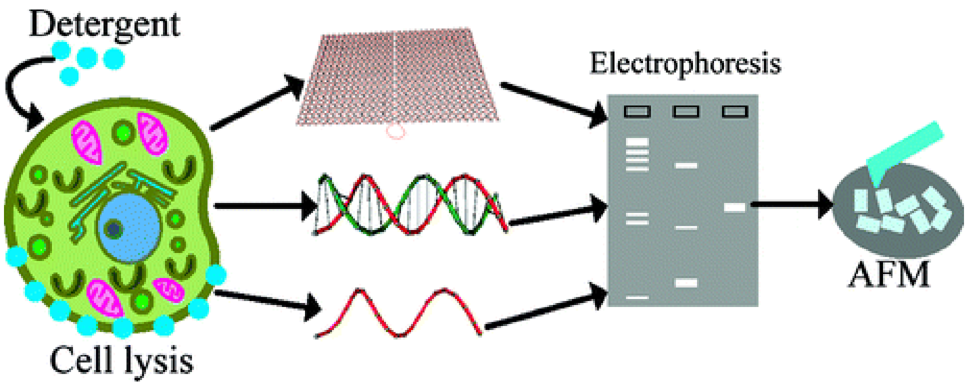


Figure 2.15: DNA origami has been shown to maintain its structure when placed in a cell lysate. Recovered objects were fully intact which indicates the method is well suited for various cellular studies. Reprinted with permission from [9]. Copyright 2011 American Chemical Society.

Also, DNA origami robots have been shown to emulate logic gates which remain functional in a living animal [7]. Expressing DNA origami objects genetically and folding *in vivo* remains an open problem.

2.5.2 DNA Computing

Besides the creation of nanodevices, DNA has been utilised for computation [74, 75]. The very first work on DNA-based computation was conducted by Adleman [76] in an attempt to solve a Hamiltonian path problem [77]. The Hamiltonian path problem requires finding a path through a graph that visits each node (or vertex) exactly once (see Figure 2.16).

Adleman solved a small instance of this NP-complete problem using the standard techniques of molecular biology. Specifically, he used the incredible storage capacity of DNA to develop a brute force method resulting in a massively-parallel and combinatorial algorithm. In his proof-of-principle experiment, DNA strands are encoding random paths (i.e. potential solutions) in such a way that a strand representing the Hamiltonian path (the correct solution) is present with high probability. By removing all strands that do not encode the Hamiltonian path and verifying that the remaining

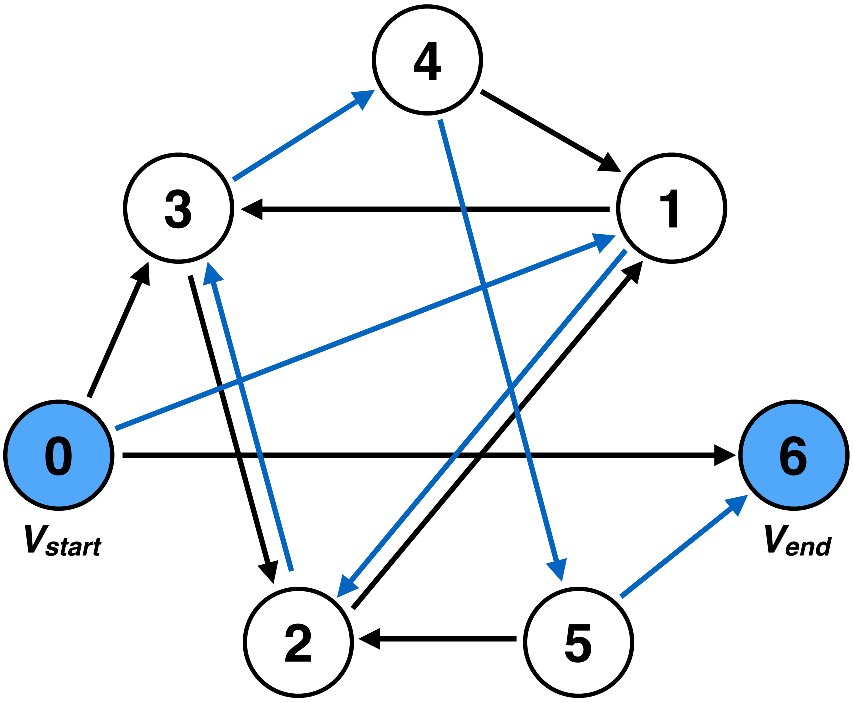


Figure 2.16: Instance of the Hamiltonian path problem solved by Adleman. The unique solution for this particular problem is denoted by blue arrows.

strands indeed encode a solution, Adleman succeeded in this first wet-lab demonstration of molecular computing.

Adleman's original experiment suffers some limitations, such as: (1) excessive volumes of DNA needed to solve problems of larger size, (2) experimental purification of the solution is error prone, and (3) the final verification step assumes a single solution. Nonetheless, Adleman's approach created a storm of excitement and galvanised scientists into considering the chemistry of DNA as a potential route to the next generation of molecular computers. Since then, a variety of examples for DNA-based information processing have been shown, including Boolean circuits [78, 79] and molecular finite state automata [80, 81]. Recently, a genetic circuit relying on DNA was engineered to mimic a biological transistor [82].

One particularly powerful branch of DNA computing is based on toehold-mediated strand-displacement (shown in Figure 2.17). These systems utilise DNA strands with partially identical sequences whose competition for common binding partners can induce dynamical changes in the configuration of the DNA/RNA assemblies. In these

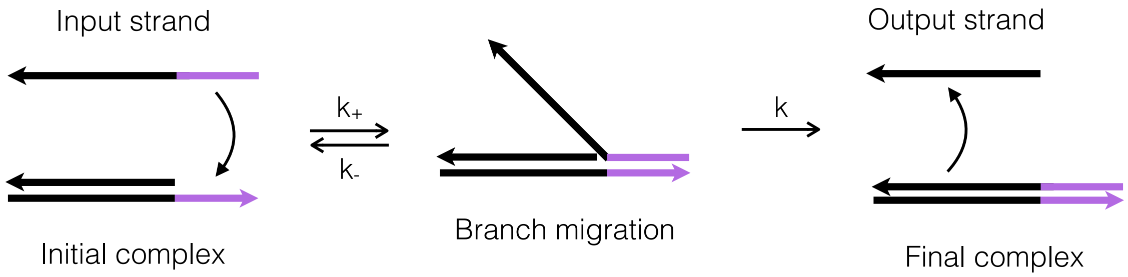


FIGURE 2.17: Schematic of toehold mediated strand-displacement. Toehold (in purple) initiates the binding between input strand and initial complex; the reaction is reversible. Once the toehold is bound, the input strand can displace the output strand through branch migration. Since the input strand forms a more stable duplex, the output strand is eventually released from the final complex (irreversibly).

designs, nanodevices feature short stretches of unpaired, single-stranded nucleotides (referred to as *toeholds*) which can capture strands with complementary regions. If the captured strand extends over the toehold and is also complementary to the adjacent sequence, it can compete for hybridization partners with any other strand that is bound to the adjacent domain. This thermally driven branch migration can result in the complete displacement of the original binding partner. This causes a potentially irreversible structural change of the nanodevice that has been used for programming dynamical behaviour such as mechanical actuation [65] and molecular computation [23, 83, 84]. These computing devices usually consist of multiple DNA strands which fold into a rationally designed structure; the configuration of a device may be switched between a number of states, often induced by addition of DNA or RNA strands. Thanks to strand-displacement reactions, the DNA nanodevices may be used, for instance, as sensors which detect input DNA sequences. This relatively simple system can lead to a surprising diversity of dynamic behaviours [85]. In particular, this mechanism allows for the precise kinetic control of reaction pathways.

2.5.3 Synthetic Biology

The advances in modern molecular biology have "led us into the new era of synthetic biology where not only existing genes are described and analysed but also new gene arrangements can be constructed and evaluated" [86]. This new engineering discipline perceives cells as microscopic processing devices built from a mere handful of extremely versatile building blocks, each evolved to express a particular phenotype. With the rise of synthetic biology came two essential revelations. The genetic code of living organisms contains abstracted units called "parts". By putting several parts together, it is possible to create new "devices" which contain novel genetic circuits with a predictable and controllable behaviour [87]. But perhaps more thrilling than just putting several genes together is the creation of new systems of interactions allowing expression of much more sophisticated phenomena than what the cells are now programmed to do. In other words, we strive to programme life that never existed before and which otherwise would not be possible. Secondly, synthetic biology poses a question (arguably one of the most profound questions in science): is it possible to create synthetic organisms or artificial life *in vitro* from organic material? And if so, what does it take to synthesise minimal life?

Creation, from the point of view of synthetic biology, amounts to being able to put together basic cellular parts, thus building a novel entity that exhibits all the characteristics of life [88]. It is important to note that this kind of creative activity is not "creatio ex nihilo", creation from nothing [89]. Even if we could, following a bottom-up approach, create a living cell from organic molecules alone, this would still be classified as creation by means of a refined combination of given parts. It may be said that in this scenario, we do not create life from scratch; we merely supply necessary conditions for the matter to actualize its potential to form living organisms [88].

As a species, we learned how to control materials, physics, conditions and, to some extent, the environment; however, we don't have nearly as much control over life. Living systems are complex things that have the appearance of having been "designed" with a purpose. Richard Dawkins refers to them as "designoids" which are complex objects that are not designed but superficially look a bit like they are [90]. Not only are "designoid objects" complex on the outside, they are also complex on the inside and perhaps vastly more complex than designed objects are. While it is true that "designoids" cannot come about by chance, evolution provides a non-random method of creation, namely, Darwinian natural selection. However, natural selection does not develop systems that are easy to understand. Living things have the quality of being able to adapt, survive and propagate their genes in reproduction. With no doubt, the transparency in the perceived design is not one of the qualities favoured by evolution.

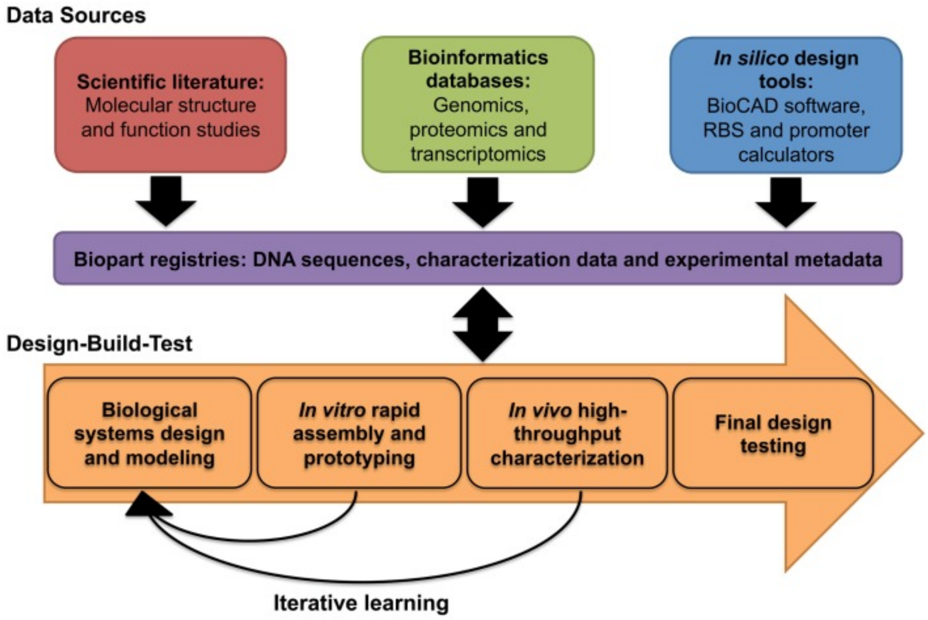

Data Sources
Scientific literature: Molecular structure and function studies
Bioinformatics databases: Genomics, proteomics and transcriptomics
In silico design tools: BioCAD software, RBS and promoter calculators
Biopart registries: DNA sequences, characterization data and experimental metadata
Design-Build-Test
Biological systems design and modeling
In vitro rapid assembly and prototyping
In vivo high-throughput characterization
Final design testing
Iterative learning

FIGURE 2.18: Systematic design of biological systems in synthetic biology. The engineering cycle describes the iterative process requiring multiple rounds of design, build and testing (taken from [91]).

As a result, there is currently a gap between our ability to synthesise DNA and create biological systems that work well. Synthetic biology is quite different from other engineering areas; many challenges remain to make living systems comply with engineering principles [88]. Unlike electronic or mechanical parts, genetic parts tend to change. Cells work for themselves and disfavour exploitation against their survival. Thus, recombinant DNA is often mutated if no selective pressure exists to maintain its function. Besides mutations, other issues exist including crosstalk, noise, cell death, environmental conditions, cellular context and incomplete models. Synthetic biology, therefore, requires entirely new engineering rules. For instance, a common practice is to create solutions that are suboptimal, learn from them and iterate over until better solutions are developed (see Figure 2.18). More than five years after its publication [92] these core 10 challenges of synthetic biology remain unresolved:

1. Reaching a consensus on synthetic and streamlined genomes

2. Cooking from scratch (bottom-up)
3. Learning from nature: naturally evolved reduced minimal genomes
4. Refine and make reality the notion of biological chassis
5. Manufacturing engineered biosystems
6. Overcoming physical and chemical constraints
7. From models to cells and back
8. Replication and reproduction
9. Towards an integrated design strategy of synthetic organisms
10. Coupling scientific development and public opinion information

In particular, cooking from scratch (bottom-up), make reality the notion of biological chassis and manufacturing engineered biosystems are the problems that are closely tied to the core of this thesis. In other words, we are interested with self-assembling nanodevices (i.e. bottom-up manufacturing) which could, in principle, be "installed" in any biological chassis.

To summarise, a primary goal of synthetic biology is to harness the inherent "biological nanotechnology" of living cells for the purposes of computation, manufacturing, or diagnosis. Advances in synthetic biology were made in three waves: modules, systems, and networks (with the last wave still yet to peak) [93]. These waves follow the hierarchy of layers, similar to computers, where each layer corresponds to a specific level of complexity and organisation. At the lowest level (i.e. the physical layer) lie fundamental components: in computer architecture, these are transistors and resistors, while in synthetic biology, these are promoters and repressors. These components can be combined to form functional devices (i.e. Boolean logic gates); and several of those devices may in turn form modules to achieve specific tasks, such as pulse generation, switching, or oscillation.

2.6 Summary and Essential Concepts

In the classical view of synthetic biology, the parts forming the physical layer are DNA sequences of defined structure and function [94]. The majority of parts are derived from various organisms and include promoters, ribosome binding sites, protein-coding sequences, terminators etc. [95–97]. However, synthetic biology is a more creative activity than genetic engineering has been before and carries with it a new level of aspiration. While genetic engineering was primarily focused on optimisation of existing systems and organisms, synthetic biology allows full play to artistry and imagination. It seems attainable that synthetic biology will take us beyond nature; “Nature 2.0,” i.e., nature with novel functions or even an orthogonal system of life, is not pure speculation anymore [88].

To explore this possibility we may branch out from systems of existing parts and create new parts, including artificial nanomachines operating in the regime of nucleic acids [11, 98]. These novel nanomachines could potentially be synthesised *in vivo* to carry out complex and coordinated tasks. The ultimate goal is for these devices to be used as nanocomputers of a new kind: performing a new type of computation and information processing [12]. Jungmann et al. [10] identify three potential merging points between DNA-based nanodevices and nanocomputers with synthetic biology. These are:

1. Biological cells could be simply used to produce RNA nanodevices by transcription
2. Gene regulatory mechanisms can be used to control the time and production of the nanodevices and naturally occurring RNA
3. Concepts from DNA nanotechnology and DNA computing can be adapted to devise novel strategies for the control of gene transcription and translation

One can envision novel “assembly lines” embedded *in vivo* that produce self-assembling nanodevices upon transcription. Those devices may even be used as controllers of their own production (i.e. autoregulation). This way, we may be able to incorporate a novel type of programmable production and control circuits into biological systems.

Essential Concepts

When considering the idea of introducing novel nanodevices into living systems there are several essential factors to consider; among them are biological orthogonality and addressability.

The term *orthogonal* (borrowed from mathematics and computer science) is an allegory that implies a factual independence between otherwise co-existing systems [99]. While the term orthogonal means independent, when used in the synthetic biology literature it largely denotes a lesser dependence on the hosts native programs. Only a few orthogonal functions are available in the existing biological world which come from bacteriophages (for instance, the T7 RNA polymerase) and mobile genetic elements, whose genetic program has evolved to depend only minimally on the recipient cells. Paraphrasing from Boldt [88] we define this notion as:

> **Biological orthogonality** (or **bio-orthogonality**) is the property of a system whose basic structure or function are so dissimilar to those occurring in nature that they can only interact with them to a very limited extent, if at all.

Existing scaffolds for DNA origami contain genetic information; e.g. they code viral proteins and are recognised by various restriction enzymes. These inherent biological features are problematic if one tries to express and fold DNA origami that interferes minimally with a cell's machinery. Little research has focused on addressing this issue, as the phage-based scaffolds became easy to obtain and manipulate [100]. Currently, with the exception of Geary et al. [101], the sequence design and its optimisation are restricted to cyclic permutations of the existing viral scaffolds or modifications of scaffold-staples layouts [47]. On the other hand, while Geary et al. [101] present a synthetic sequence optimised for co-transcription, it requires a different sequence for each nanostructure one may want to assemble. It is a clear limiting factor, especially when one considers the potential biological (i.e. *in vivo*) production of multi-shaped (and hence multi-functional) origami.

Furthermore, and of concern not only within a synthetic biology context, is the issue of addressable complexity [102]. DNA origami and DNA brick [103] structures are the two important examples of multicomponent structures; their remarkable complexity is due to the fact that the structures are completely "addressable" [104]; other examples include single-stranded tile [105–108] and double crossover tile [109]. Since every building block within an aperiodic addressable structure (or unit cell in a periodic addressable structure) is unique, it is possible to specify exactly where a particular subunit will be located within the target structure [102, 110]. Thus, the term *addressability* when used in nanoscience has the desired connotation of being able to target specific parts of the nanostructure with a nanometre precision. Addressability was demonstrated, for example, by anchoring proteins on DNA origami nanostructures [111]. Hence, our working definition of unique addressability is slightly stronger than the established definition found in literature:

> **Addressability** is the capacity for an entity to be targeted and found within a system. To be **uniquely addressable**, such an entity must be uniquely identifiable, i.e. the association with its target must be strongly favourable – thermodynamically and kinetically – over anything else that exists within that system.

The repetition of nucleotide sequences in existing scaffolds and staple strands may cause unspecific hybridization [4]. The resulting misfolding (primarily kinetic traps) can disrupt the self-assembly process and lead to structural deformations or malfunction of folded nanodevices. The evidence of potential misfolding was explored by the previous study and prevented by a judicious design of the folding funnel [112]. The problem might also be counteracted by the cooperative nature of the folding [113] and strand-displacement reactions as an error-fixing mechanism. All these effects play a role in the self-assembly of DNA origami but are currently hard to control in a pragmatic manner [114]. On top of that, strand-displacement reactions are known to have slower kinetics compared to hybridization [115] which may be a setback in the folding process, especially in the cellular environment.

Part I

DNA ORIGAMI

Chapter 3

Synthetic Scaffolds Design

We propose a method for designing scaffold sequences as a synthetic extension of DNA origami technique suitable when reliance on viral genomes is not realistic. We develop a set of algorithms allowing a rapid construction of synthetic scaffolds and explain why they are good candidates to satisfy the criteria of biological orthogonality and unique addressability. Finally, we develop a workflow and design a number of nanostructures utilising these synthetic scaffolds. This leads to a case study which is explored further in Chapter 4.

3.1 Introduction

In this chapter, we propose a theoretical framework to address hypothesis H_1:

> H_1 : It is possible to program a synthetic scaffold for DNA origami which is both bio-orthogonal and uniquely addressable.

Addressability, in the context of DNA nanotechnology, is an essential concept which allows precise arrangement of DNA strands through the Watson-Crick complementarity. However, the same motif of nucleotides might, in general, address multiple sections of the long DNA strands (as shown in Figure 3.1). This is suboptimal from an engineering viewpoint and hence having a certainty of where it will Watson-Crick complement is crucial.

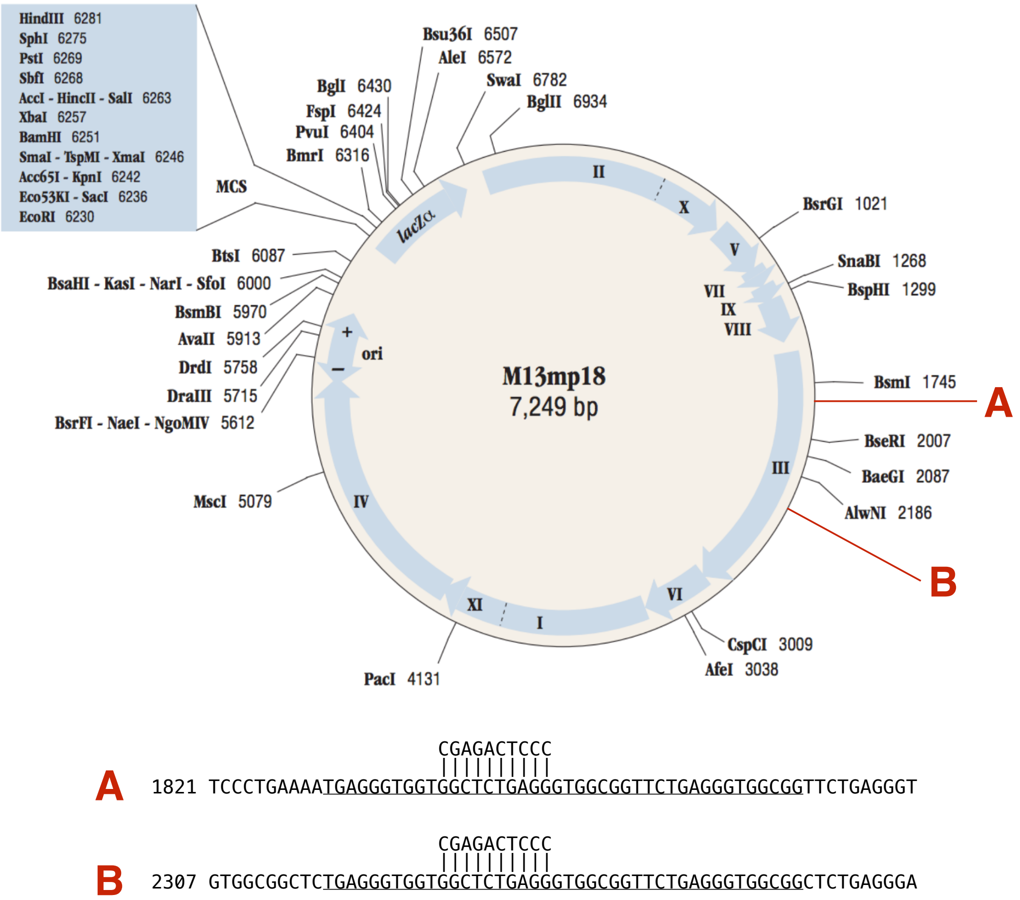


Figure 3.1: M13mp18 bacteriophage map: the most commonly used DNA origami scaffold consists of multiple genes and restriction enzyme sites (shown in bold). Additionally, it contains repeated subsequences (for example A and B; underlined) which are ambiguous and violate the addressability property. For instance, sequence '3'-CGAGACTCCC-5' designed to address a domain in A may instead bind to an incorrect domain in B (in fact, this is the case for the entire underlined sequence). (M13mp18 map is available on the NEB website).

Also, Figure 3.1 shows the genome map of the M13mp18, which is a standard scaffold in DNA origami. This DNA sequence, which is a result of viral evolution, is easy and cheap to obtain simply by infecting *E. coli* bacteria. Using M13mp18 genome as the scaffold is a double-edged sword: its availability comes with a serious drawback. Upon entry into the cytoplasm, the virus hijacks the internal machinery of the infected host and induces its own replication. M13mp18 contains 10 genes (for replication, coating proteins, and phage assembly) as well as multiple restriction enzymes cutting sites. Indeed, from a synthetic biology point of view, the viral scaffold violates the biological orthogonality. Therefore, being able to design a sequence without biological function or meaning is critical.

The approach that we chose to employ here has its grounds in combinatorics and graph theory. Namely, we are exploiting a property of a family of combinatorial objects called de Bruijn sequences (DBS). More specifically DBS of order n have no duplicate subsequences of size n or larger, thus rendering them uniquely addressable by design. This uniqueness property (i.e. lack of repetitions) makes DBS an attractive candidate for addressable scaffolds: any staple binding a specific region of such a scaffold is by design complementary only to that specific region. This in principle should favour specific hybridisation over any non-specific one (see Figure 3.2).

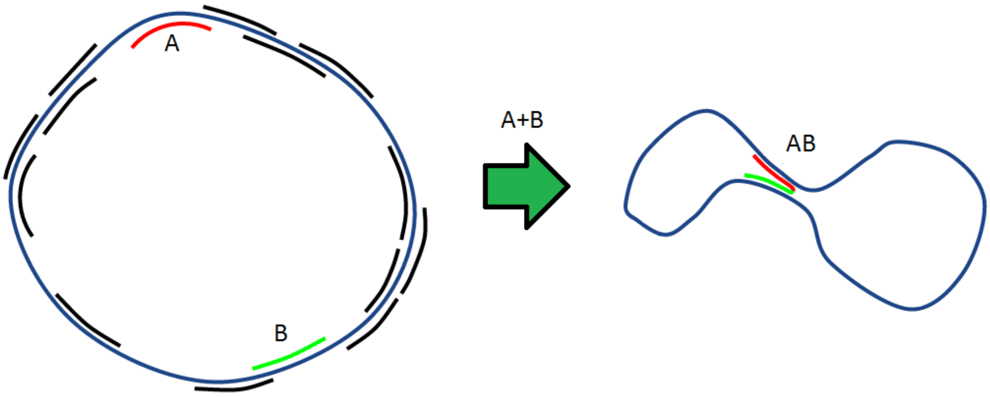


Figure 3.2: Unique addressability: DNA origami scaffold (blue) should be constructed in such a way that each of the short complementary domains (black) can potentially bind to its respective fragment and this fragment only. Following that property, joining two or more domains, here A and B, introduces a new staple strand AB that binds the scaffold in designed place only.

We explain how to generate De Bruijn sequences algorithmically and how forbidden sequences can be removed from DBS. That method allows designing DNA origami scaffolds that are orthogonal to a biological system of choice. The design pipeline is automated by software, which is explained in the following section. Optionally, generated sequences undergo a secondary structure optimisation. Lastly, the chapter presents DNA origami and DNA/RNA hybrid origami designs based on those synthetic scaffolds.

3.2 De Bruijn Sequence

De Bruijn sequence $B(k, n)$ is a cyclic sequence in which every possible n-long combination of symbols from given alphabet A of size k appears exactly once. Nicolaas Govert de Bruijn, after whom the sequences are named, has shown that construction of this type of sequences can be generalised for an arbitrary alphabet of symbols; he has also proved that the length of any $B(k, n)$ is always k^n and the total number of distinct sequences is given by $(k!)^{k^{n-1}} \times k^{-n}$ [116].

For example, consider a binary alphabet $A = \{0, 1\}$ and $n = 3$. In that case, there are only two sequences of length 8; those are 00010111(00) and 00011101(00) (note that symbols in parentheses refer to the beginning of the sequence to emphasise its cyclic nature).

It is possible to generate DBS for a DNA alphabet $A = \{A, C, G, T\}$. For example, the lexicographically smallest DBS of order 3 is shown in Figure 3.3. Careful observation shows that each substring of length 3 is a unique nucleotide string. For an alphabet with four symbols, there are $(4!)^{4^{n-1}} \times 4^{-n}$ different DBS of length 4^n. Hence, the DBS shown in Figure 3.3 is chosen from a large set containing over 189 quintillions (1.89×10^{20}) sequences in total.

De Bruijn sequences have been used in bioinformatics before, for instance, for whole genome assembly from a multitude of short reads [117–120]; also to study the structure and stability of the genetic code [121]. In the context of DNA origami, De Bruijn sequences were used to quantify the quality of the folding process by coding short DNA probes [122]. These DBS probes were used to measure the content of unpaired DNA bases in several nanostructures.

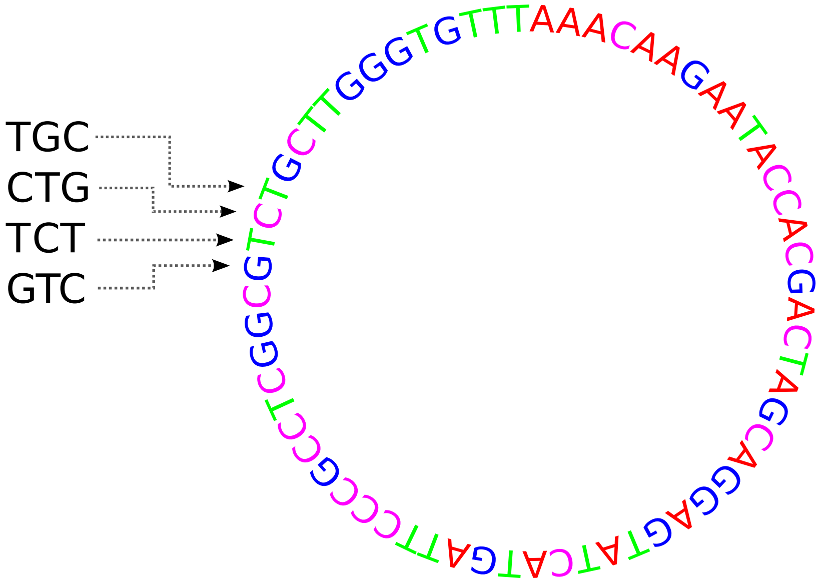

TGC
CTG
TCT
GTC
TGCTTGGGTGTTTAAACAAGAATACCACGACTAGCAGGAGTATCATGATTCCCGCCTCGGCGTCTG

Figure 3.3: De Bruijn sequence. An example of De Bruijn sequence (DBS) of order 3 which contains all nucleotide triplets (i.e. codons). DBS is uniquely addressable because the repetitions are not allowed, e.g. sequences TGC, CTG, TCT and GTC appear exactly once in the sequence.

By definition, De Bruijn sequence $B(k, n)$ contains exactly one occurrence of every n-long subsequence. It also contains k occurrences of every (n-1)-long subsequence, k^2 of (n-2)-long and so on. It is also worth mentioning that this property holds true in the opposite direction: $B(k, n)$ contains $\frac{1}{k}$ of all possible (n+1)-long sequences, $\frac{1}{k^2}$ of all possible (n+2)-long sequences and so on. That gives a very reliable way by which to estimate the probability of finding sequences of a particular length in any given De Bruijn sequence (bear in mind that probability of finding a sequence containing repetitions is zero). In other words, the probability that a specific sequence $B(k, n)$ includes subsequence s decreases exponentially as the length of s is increased.

Furthermore, two important characteristics of De Bruijn sequence are uniqueness, meaning that n-long duplicates are not allowed, and completeness, meaning that all n-long instances are included. From biological perspective, De Bruijn sequence allows obtaining a synthetic scaffold for DNA origami which is 'programmable' and uniquely addressable (see Figure 3.2). In that case, it is essential to highlight the following: for the purpose of this work, uniqueness is an essential property that should be retained. On the other hand, completeness is problematic as some of the sequences composing DBS may be biologically active and hence should be avoided (violating the completeness principle). Simply put, in programmable DNA origami what truly matters is unique addressability while in a synthetic biology context bio-orthogonality is essential. Thus, the staples are designed for one binding site only (i.e. a unique address), however having all possible addresses may be undesirable if that stands in conflict with the biological orthogonality.

3.2.1 Sequence Construction

Only a few efficient constructions of de Bruijn sequences are known [123]. In particular, there are:

- a shift generation approach based on primitive polynomials by Golomb [124],
- three different algorithms to generate the lexicographically smallest DBS (also known as the Ford sequence): a Lyndon word concatenation algorithm by Fredricksen and Maiorana [125], a successor rule approach by Fredricksen [126], and a block concatenation algorithm by Ralston [127],
- a lexicographic composition concatenation algorithm by Fredricksen and Kessler [128],
- three different pure cycle concatenation algorithms by Fredricksen [129], Etzion and Lempel [130], and Huang [131] respectively, and
- cool-lex based constructions by Sawada, Stevens and Williams [132] and Sawada, Williams and Wong [133].

Each algorithm requires only $O(n)$ space and generates their DBS in $O(n)$ time per bit, except the pure cycle concatenation algorithm by Etizon and Lempel which requires $O(n^2)$ space. The Lyndon word concatenation algorithm and the cool-lex based approaches achieve $O(1)$-amortised time per bit. There also exist interesting greedy constructions including the prefer-1 and prefer-opposite approaches by Martin [134] and Alhakim [135]; however, they require $\Omega(2^n)$ space. The drawback of these approaches is that they generate a handful of sequences (usually just the lexicographically smallest one).

In order to have a better coverage of the search space (i.e. be able to potentially generate every DBS), we chose to make use of de Bruijn graphs and Euler cycles [118]. While practically it might not be feasible to generate all possible de Bruijn sequences for large values of n, it is important that our method allows a fair sampling of the search space for a DNA alphabet. Finding Euler cycles in the de Bruijn graph is an approach that will find all de Bruijn sequences for a given n, but again, storing the graph requires $\Omega(2^n)$ space (in binary) or $\Omega(4^n)$ space (in DNA). This approach relies on a de Bruijn graph – a directed and symmetric graph which represents overlaps between sequences from any given alphabet A (for an example refer to Figures 3.4 and 3.5).

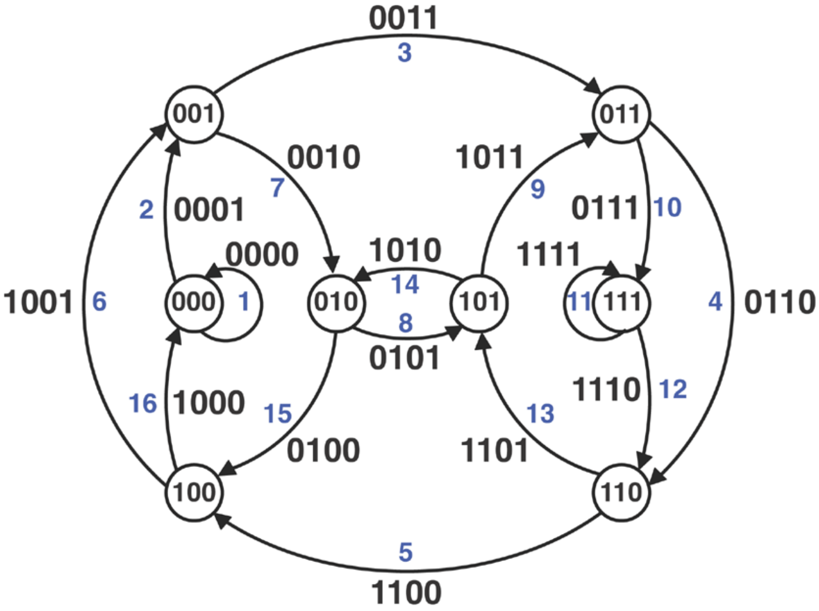


Figure 3.4: De Bruijn graph for an alphabet $A = \{0, 1\}$ and $n = 4$. Starting from the edge $e_1 = 000$ and traversing the edges in order from 1 to 16 (blue numbers) a cyclic De Bruijn sequence is produced that spells 0000110010111101(000). In total there are 16 unique sequences that can be produced based on this graph. Reprinted by permission from Macmillan Publishers Ltd: Nature Biotechnology [118], copyright 2011.

Construction of De Bruijn Graph of order n is conducted in two steps. First, all possible n-mers are listed and assigned to the graph nodes. Here, the term n-mer is simply an n-long permutation of symbols over the alphabet A. Secondly, any two nodes are connected by an edge iff the suffix of the first node is equal to the prefix of the second node (signifying overlaps). At the end of this procedure, every node should have k incoming and k outgoing edges. For instance, a node containing sequence ATG will be followed by 4 nodes: $\underline{\text{TG}}$A, $\underline{\text{TG}}$C, $\underline{\text{TG}}$G and $\underline{\text{TG}}$T, and preceded by 4 other nodes: A$\underline{\text{AT}}$, C$\underline{\text{AT}}$, G$\underline{\text{AT}}$ and T$\underline{\text{AT}}$ (a common part is underlined for clarity). Note that loops are allowed: nodes AAA, CCC, GGG and TTT all have an edge that connects the node to itself (i.e. edges AAAA, CCCC, GGGG and TTTT respectively).

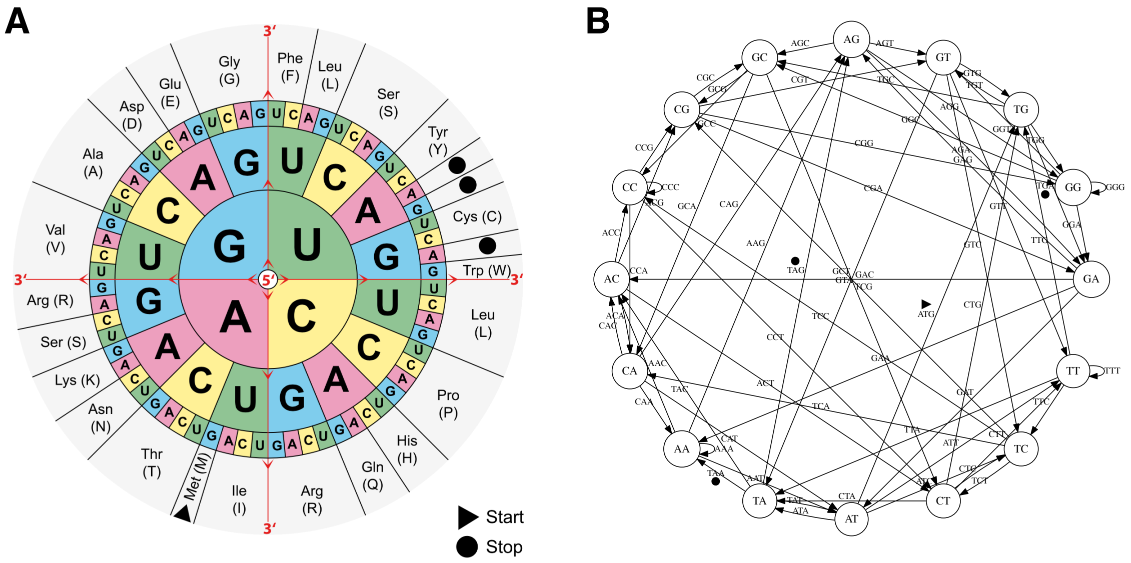


FIGURE 3.5: De Bruijn graph for the genetic code where the alphabet is a set of nucleotides, $A = \{A, C, G, U\}$ and $n = 3$. Every edge in the graph has a unique DNA codon assigned to it. Start and stop codons have been marked, with black triangle and circles respectively: (A) RNA codon table; (B) de Bruijn graph (4, 3) for DNA alphabet.

Once the De Bruijn graph is constructed, finding a DBS is equivalent to finding an Eulerian cycle in that graph [136]. This can be accomplished through graph traversal (see Figure 3.4).

Technically, producing an instance of $B(k, n)$ is equivalent to finding a Hamiltonian cycle in an n-dimensional De Bruijn graph over an alphabet A of size k (i.e. find a cycle that visits each node once). Alternatively, an Eulerian cycle in the (n-1)-dimensional graph can be found (i.e. traverse every edge once and return to the starting point). Figure 3.4 shows an example of an Eulerian cycle: all 16 edges are visited exactly once. The same graph contains Hamiltonian cycles; one instance can be obtained by traversing edges in order: 2, 7, 8, 9, 10, 12, 5, 16 which spells 00010111(00).

Note that those problems are equivalent to each other because each n-dimensional De Bruijn graph is the line graph of the (n-1)-dimensional De Bruijn graph over a similar alphabet A (i.e. there is a one-to-one relationship between edges of the base graph and nodes of the other graph). However, since the problem of finding Hamiltonian cycles is known to be NP-hard, we resort to the latter method instead.

Algorithm 1 Traverse()

```
function TRAVERSE
    current = start
    for edge in remaining do
        if edge matches current then
            current = edge
            remove current from remaining
            sequence += last symbol of current
            traverse(current, remaining, sequence)
        end if
    end for
    if remaining is empty then
        print sequence
    end if
end function
```

Two versions of the algorithm have been implemented that find Eulerian cycles in De Bruijn graphs; the former is based on a recursive procedure that simply tries all possible combinations of edges arrangement and outputs a De Bruijn sequence if one is found. This method is suitable for relatively small graphs (low values of n) as it produces all distinct permutations (see Algorithm 1). An example output (based on $n = 3$, $A = \{A, C, G, T\}$) is shown below (note the sequences differ only in the rearmost part).

The first five DBS permutations (lexicographically smallest sequences) are:

```
1) AAACAAGAATACCACGACTAGCAGGAGTATCATGATTCCCGCCTCGGCGTCTGCTTGGGTGTTT(AA)
2) AAACAAGAATACCACGACTAGCAGGAGTATCATGATTCCCGCCTCGGCGTCTGCTTGTGGGTTT(AA)
3) AAACAAGAATACCACGACTAGCAGGAGTATCATGATTCCCGCCTCGGCGTCTGCTTTGGGTGTT(AA)
4) AAACAAGAATACCACGACTAGCAGGAGTATCATGATTCCCGCCTCGGCGTCTGCTTTGTGGGTT(AA)
5) AAACAAGAATACCACGACTAGCAGGAGTATCATGATTCCCGCCTCGGCGTCTGGGTGCTTGTTT(AA)
```

Even for small values of n the number of unique sequences one can generate is colossal; for example for $n = 4$ we have a total of $8.42 * 10^{85}$ sequences (i.e. more than the number of atoms in the universe[1]). For practical reasons, one may want to sample this huge search space in a more random fashion. We implemented another version of the algorithm to find random Eulerian cycles in a De Bruijn graph. First, a random edge is selected as a starting point and marked as visited. The first symbol of that edge is written which corresponds to the first symbol in the final DBS. Then at each step of the algorithm, the next edge is selected from the current node (also at random) iff it has not been visited yet; its first symbol is appended to the DBS. This procedure repeats until the current node has no unvisited edges left (i.e. a dead-end is found). Dead-end during traversing does not necessarily complete the algorithm: it only establishes the initial cyclic sequence. It is likely that not all edges have been visited yet. If this is the case, the unvisited edges are traversed separately in a similar fashion (finding Eulerian paths) and the linear sequences they produce are inserted into the initial cyclic DBS at valid positions. This method is more suitable for larger graphs for which it is practically impossible to generate all existing permutations and a representative sample of well-shuffled sequences is more appropriate. The example output (based on $n = 3$, $A = \{A, C, G, T\}$) is shown below.

[1] The commonly accepted estimate for the number of particles in the observable universe is 10^{80}; it is derived by dividing the mass of all ordinary matter ($1.45 * 10^{53}$ kg) by the mass of a single hydrogen atom ($1.67 * 10^{-27}$ kg).

A five random DBS permutations generated by the algorithm:

```
1) AAATTATGACACGAGATAGCCCTACTTGGGCTCCGCAGTGTAAGGTCGTTTCTGCGGAACCATC(AA)
2) AAAGCCGAGGGATTCCTTTGCAATATCTAGACGTCACATGGTTACTCGCGGCTGTAACCCAGTG(AA)
3) AAACCCATGTTTACACGACTCAAGAGCTGAATAGGATTCCGCCTTGCAGTCGGTGGGCGTATCT(AA)
4) AAACGAATAGTCCGGGCTGACAAGCACCAGGAGATCGCGTGCCCTTACTCTATTTCATGTTGGT(AA)
5) AAATTTCATAAGTCTTAGATCGGCCACGCAACTATGAGCTGCGTTGTGGGTACCCTCCGACAGG(AA)
```

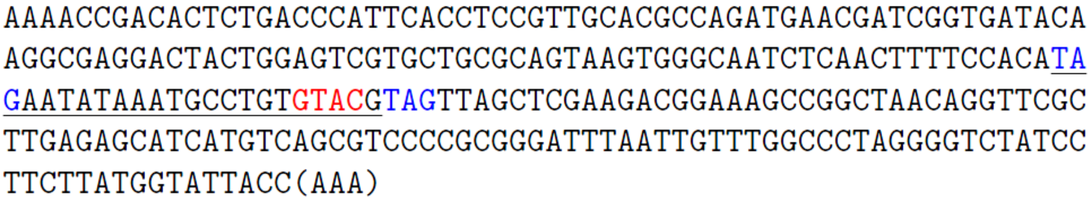
The sequences are generated using a stochastic algorithm. Much of what the algorithm does revolves around limiting repeated stretches of a forward sequence. However, it is possible to modify the algorithm to achieve additional features in the generated sequences. For instance, one immediate alteration would be to have the algorithm also minimise stretches of reverse-complementarity, which could potentially interfere with proper folding by allowing the scaffold to base pair with itself. We address this issue in Section 3.2.3. In the following section, we show how the generated scaffold sequence is curated to achieve bio-orthogonality.

3.2.2 Bio-orthogonal Filtering

As briefly mentioned before uniqueness is favoured over completeness. What is more important, sometimes completeness might even be considered undesirable as some subsequences might carry biological meaning depending on the context (e.g. an *E. coli* cell) in which they will operate. Note that the folding of DNA origami was so far performed *in vitro* – the environment pure of any entities other than DNA origami strands (and of course additional chemistry that induces the self-assembly). On the other hand, folding DNA origami *in vivo* is likely to interfere with cell internal mechanics to some extent – after all, it is a living entity, based on DNA, with numerous processes constantly targeting DNA strands [137, 138] (such as sequence-specific DNA binding proteins). Therefore, the idea here is to be able to identify sequences, which we call "taboo", that carry some biological function and either avoid them or position them at a specific place within the DNA origami shape. These taboo sequences will be context specific, that is, will depend on the cellular context the origami will be designed for (e.g., *E. coli*, *S. cerevisiae*, *CHO* cells, etc). Here, we present the method of deleting taboo sequences from the DNA origami scaffold.

The graph can be constrained with edges containing undesired sequences. If an edge contains a restricted sequence it can be removed from the graph and never visited. However, removing edges can leave the graph unbalanced in which case it will be impossible to construct the DBS (i.e. an Eulerian cycle does not exist). A solution to this is to remove a cycle which includes a restricted edge rather than the edge alone. This method maintains the balance in the underlying graph and ensures that the sequence (without restricted subsequences) can be produced.

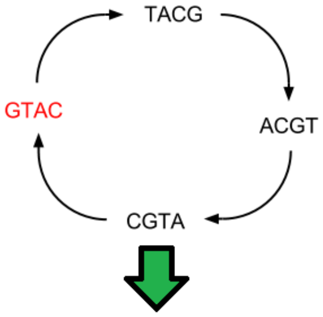
For the following filtering examples, consider this problem: from a class of De Bruijn sequences $B(4,4)$ over the DNA alphabet $A = \{A, C, G, T\}$ exclude a single taboo sequence. Suppose that the unwanted sequence is given by $s = "GTAC"$ (that depending on the reading frame might code for valine, encoded by GTA, or methionine, encoded by TAC). Since we consider $B(4,4)$ and the length of s is 4, there is exactly one instance of s to be removed.



```
AAAACCGACACTCTGACCCATTCACCTCCGTTGCACGCCAGATGAACGATCGGTGATACA
AGGCGAGGACTACTGGAGTCGTGCTGCGCAGTAAGTGGGCAATCTCAACTTTTCCACATA
GAATATAAATGCCTGTGTACGTAGTTAGCTCGAAGACGGAAAGCCGGCTAACAGGTTCGC
TTGAGAGCATCATGTCAGCGTCCCCGCGGATTTAATTGTTTGGCCCTAGGGGTCTATCC
TTCTTATGGTATTACC(AAA)
```

FIGURE 3.6: Post-generation filtering: firstly a taboo sequence $s = "GTAC"$ is found (red). Two similar (n-1)-mers surrounding s are chosen (blue) such that the distance in between them is minimised. A whole fragment containing s is removed (underlined) resulting in a 23 bp-long deletion. The remaining sequence could now be joined as is.

This can be accomplished in two alternative ways. The former is used when De Bruijn sequence is given already (i.e. post-generation filtering). Simply deleting the target subsequence from De Bruijn results in breaking the cycle and leaving a gap (similarly, any further deletions would partition the original sequence into consecutive number of shorter sequences). Note that simply joining two loose ends at that point is problematic as it introduces unwanted repetitions. To maintain a cyclic sequence a splicing technique should be used in the following way: scan left and right-hand site of target subsequence in order to find two similar (n-1)-mers. Now remove the whole sequence that falls in between the first (n-1)-mer (inclusively) and second (n-1)-mer (exclusively) (see Figure 3.6 for details). That procedure results in matching loose ends which are joined together in the final step. The distance between (n-1)-mers affects the total number of deleted symbols, therefore the aim is to find such a pair for which the distance is minimised.



```
AAAAGCATCATAATACAATGATCTCTAAGAAATCCTGCCAACCCCTTATATGCTCGTTTG
GGTCCAGGTGGAAGGCCCGAGGGGCTTGAGTGCGTGTAGCTGGCAAGTCTGAACACTACT
TCTTTTCCGCTAGACGATGTCACCACGGACTGTTGTGACCGTCGGCGCCTATTACCTCAG
AGATAGGAGCCGGTATCGACATTTAACTCCCATGGTTCAAACGCGGGATTCGCAGCGAAT
TGCACAGTTAGT(AAA)
```

FIGURE 3.7: Pre-generation filtering: before the sequence is produced a loop containing unwanted sequence $s = "GTAC"$ (red) is removed from the underlying graph. All sequences produced by traversing the adjusted graph are guaranteed to lack s. Note: resulting sequences are shorter than original De Bruijn sequence by 4 bp.

The latter method is based on pre-generation filtering, i.e. rather than the sequence itself an underlying graph is modified. The edges containing taboo subsequence are removed (and since the length of s is 4 there is exactly one edge to be removed). However, in order for this approach to work (i.e. to construct a De Bruijn-based cyclic sequence), there has to exist an Eulerian cycle in the resulting graph. Euler's theorem states that such a cycle exist iff the connected directed graph is balanced (for each node its indegree and outdegree must be equal) [118]. Since one edge has been removed this property does not hold any longer. The solution is to identify a loop containing the unwanted edge and remove all edges within that loop from the graph (see Figure 3.7). Indeed, after this operation indegree and outdegree of every node remain equal. As a consequence, each and every sequence generated from that graph will lack the taboo sequence.

It is vital to point out that this method relies heavily on the data provided. In order to design scaffold that is "perfectly" orthogonal, we would require the total knowledge of biological sequences; in particular, about sequences recognised by various biomolecules, digested by enzymes, or sequences that are functional in any other way. Although this "big unknown" may be daunting, the method presented here can be used in an iterative way. If a synthetic scaffold turns out to have some unwanted effect on the host cell, and if this effect can be linked to a specific sequence, the next iteration should be restricted with this new sequence (i.e. an improvement by trial and error). On the other hand, non-specific DNA-proteins with a general affinity for DNA could still have some interaction with the scaffold.

3.2.3 Energetic Optimisation

As mentioned earlier, because of a large number of DBS from which a designer can choose from, a scaffold for DNA origami can be selected according to certain desired characteristics. Therefore, we specified the additional criterion for the synthetic scaffold sequences based on their folding properties, namely, minimisation of secondary structures. If the scaffold can base pair with itself and form stable structures, the origami folding would be impeded. The extent to which this is problematic can be measured using a thermodynamic analysis and mitigated by the choice of scaffold.

The DNA origami scaffold (2.4 knt) was picked as a compromise between following factors: elimination of all forbidden sequences (Section 3.2.2), the stability of the secondary structure and the nucleotide composition. ViennaRNA package [36] was used to predict minimum free energy (MFE) of DNA sequences using energy parameters provided by Turner and Mathews [139]. The MFE of pUC19 scaffold is -414.6 kcal/mol (GC content: 0.52). Interestingly, it is more stable than that of a randomly generated sequence with the same nucleotide composition (see Figure 3.8 in red). We

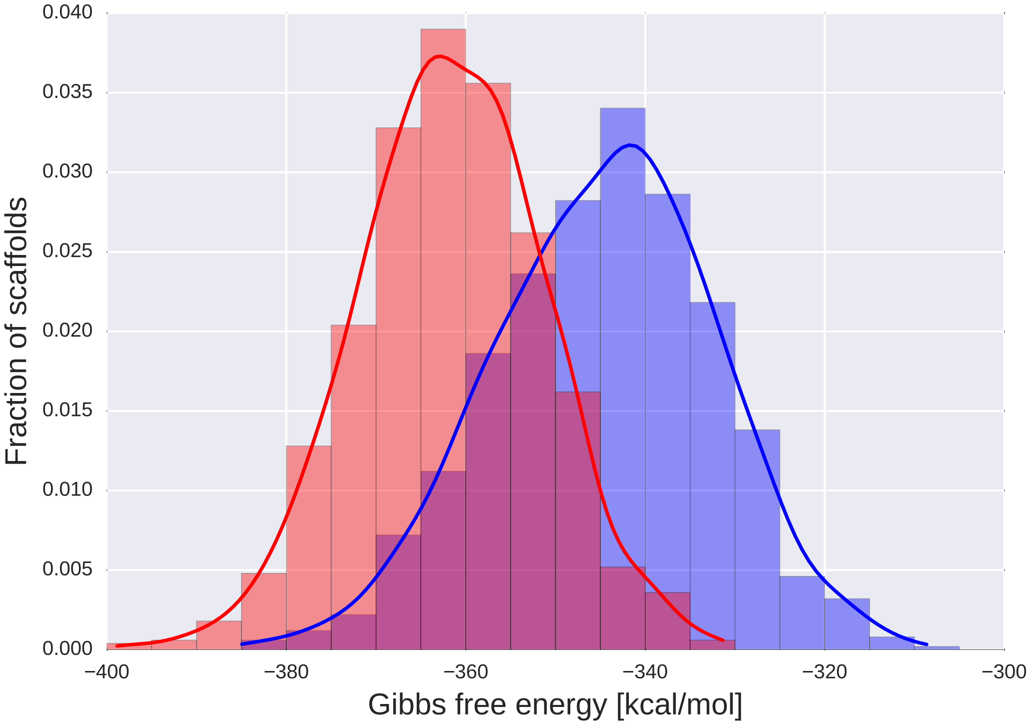


FIGURE 3.8: The MFE distribution: 1000 DBS scaffolds (blue) and 1000 randomly generated scaffolds with similar nucleotide composition as pUC19 (red), are plotted according to the secondary structure energy. Sequences are 2.4 knt long.

aimed to obtain a weaker secondary structure in DBS scaffold while preserving similar nucleotide composition (MFE of -376.4 kcal/mol, GC content: 0.5). This should facilitate the origami folding as the scaffold will hybridise more readily with the staples rather than with itself. In addition, in RNA-DNA origami both the DBS scaffold (1.1 knt) as well as the staple sequences were optimised to weaken secondary structures and avoid hairpin formation.

3.3 Scaffold Design and Analysis Software

The workflow for constructing synthetic DBS scaffolds for DNA origami is automated with a custom-made software; it is available as a plug-in for a popular open-source tool caDNAno [140]. The software features are provided in the appendix (see Section A.1).

The software automates the generation of De Bruijn sequences and uses as an input the list of restricted sequences (i.e. taboo list) provided by the user (as explained earlier in this chapter). Additionally, when the origami is designed and the scaffold is chosen, one can analyse the thermodynamic addressability in this particular design. The algorithm uses dynamic programming to find potential binding regions in the scaffold for each of the staples in the design. This is possible by aligning the scaffold sequence and the reverse-complement of the staple sequences. When such an alignment (within certain mismatch threshold) is found it is considered in the Boltzmann distribution. The tool will then calculate the probability that a staple binds its correct target in the scaffold. The results are visualised as addressability measure distribution.

This software was used to design two synthetic scaffolds as well as for the further analysis (described in Section 4.3.3).

3.4 Case Study and Diagrams

The first synthetic DBS was constructed to fold into a square DNA origami, roughly 50 nm in size, which required 2.4 knt of the scaffold (see designs in Section 2.5.1). The shortest DBS satisfying this requirement can be built from subsequences of 6 nt (i.e. DBS of order 6) and thus have a total length of 4096 nt (i.e. 4^6). However, this theoretical maximum was reduced to 3.3 knt when DBS was constrained with biological sequences (see Section 4.2 for details) and then trimmed to the length required by the square design (2484 nt).

The second synthetic DBS was constructed to fold into a triangular RNA-DNA hybrid origami, roughly 30 nm in size, which required 1.1 knt of the scaffold. Similarly, DBS of order 6 was used. Also, in RNA-DNA origami both the scaffold as well as the staple sequences were optimised to weaken secondary structures and avoid the hairpin formation of staples (data not shown).

The square DNA origami (Figure 3.9) and the triangular RNA-DNA hybrid origami (Figure 3.10) have been designed using caDNAno [140] that is now considered a standard tool for DNA design. The designs follow the architectures that were tested in previous studies [4],[141]. Note that, RNA-DNA hybrid structure assumes A-conformation (11 bp per turn).

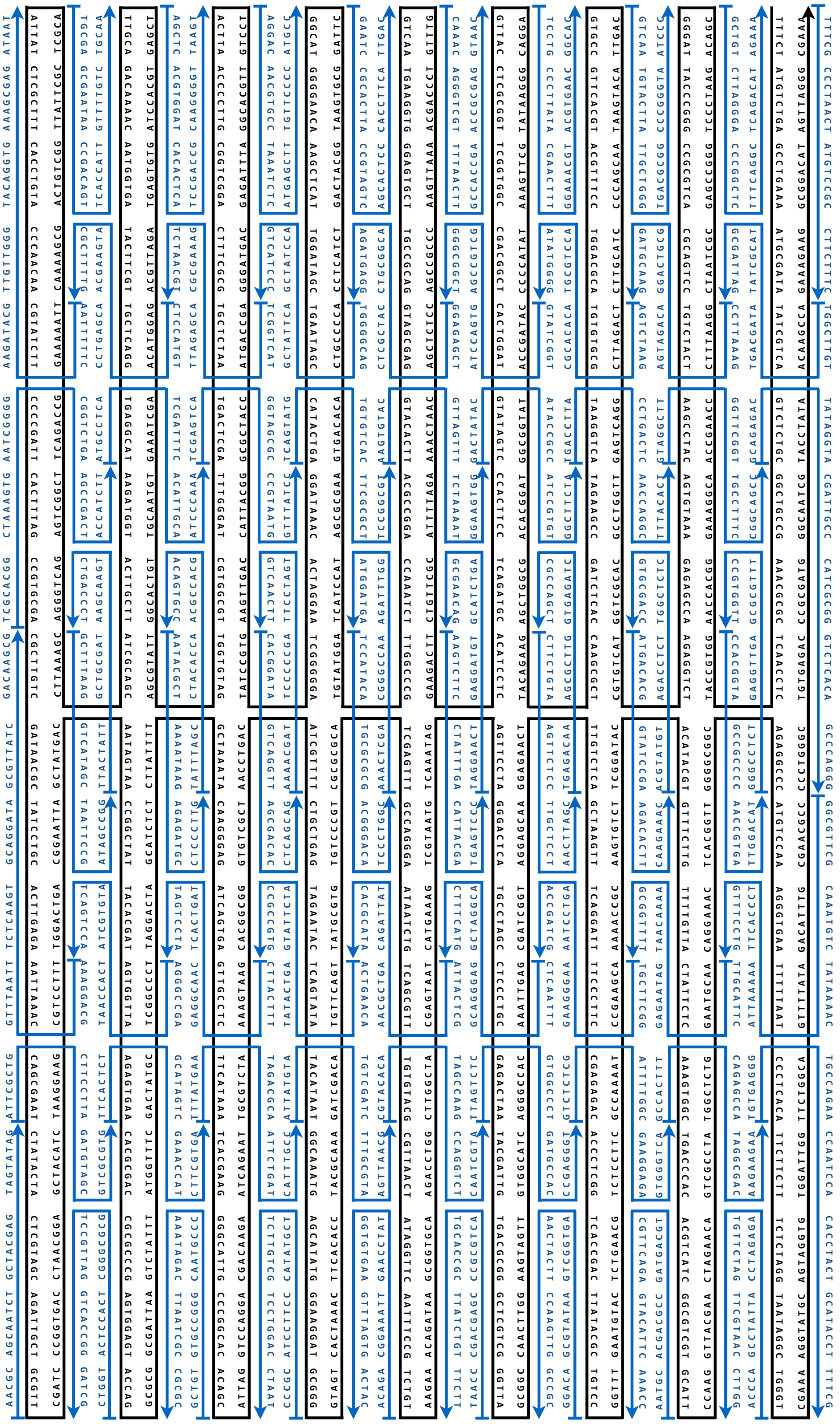


Figure 3.9: Detailed design of a square DNA origami based on synthetic DBS scaffold.

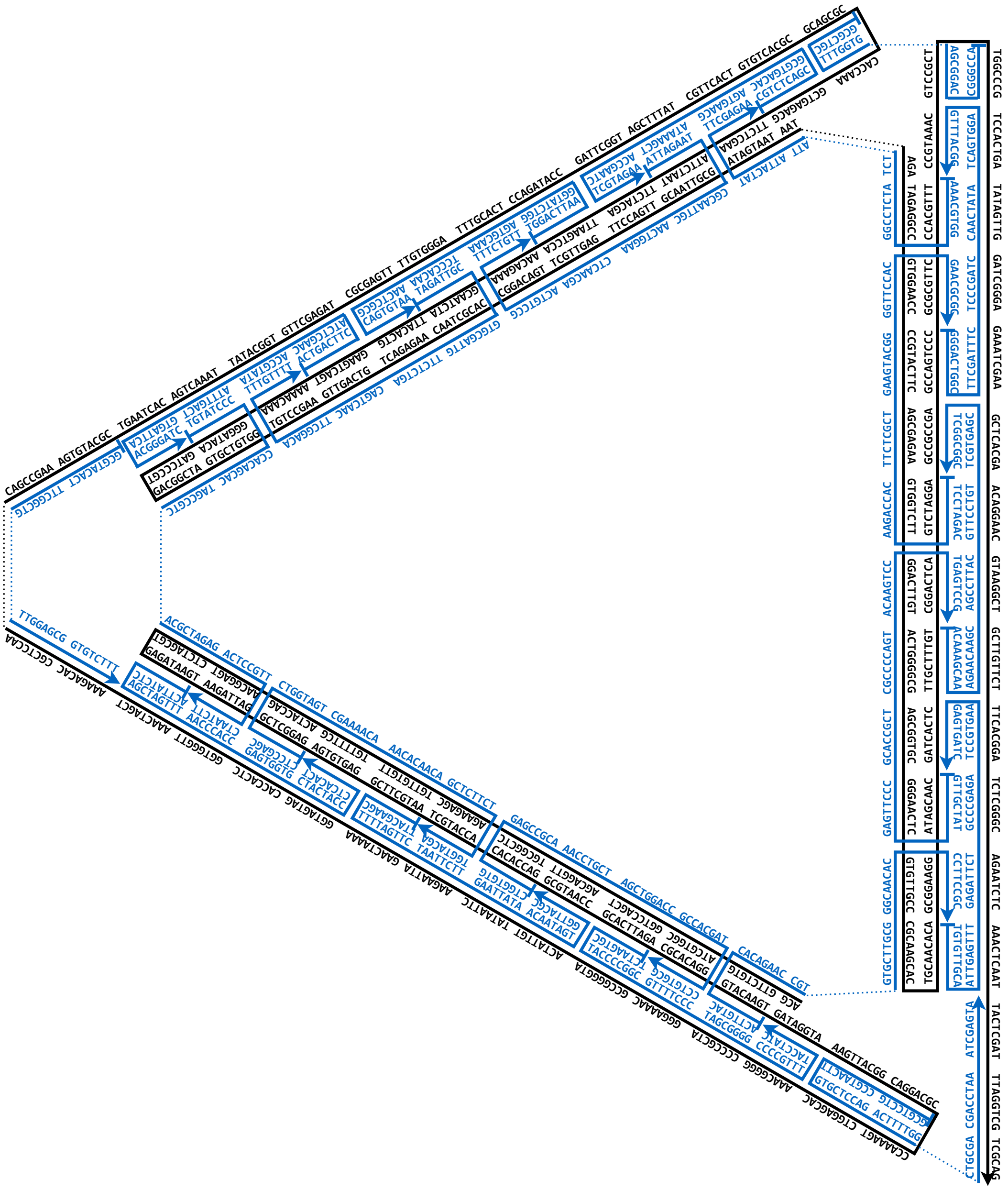

CAGCCGAA AGTGTACGC TGAATCAC AGTCAAAT TATACGGT GTTCGAGAT CGCGAGTT TTGTGGGA TTTGCACT CCAGATACC GATTCGGT AGCTTTAT CGTTCACT GTGTCACGC GCAGCGC
TGGCCCG TCCACTGA TATAGTTG GATCGGGA GAAATCGGA GCTCACGA ACAGGAAC GTAAGGCT GCTTGTTCT TTCACGGA TCTCGGGC AGAATCTC AAACTCAAT TACTCGAT TTAGGTCG TCCGA

Figure 3.10: Detailed design of a triangular RNA-DNA hybrid origami based on synthetic DBS scaffold.

3.5 DNA Origami Structure Flexibility

DNA origami structures have been analysed using CanDo online service [47]. This tool enables the prediction of the 3D equilibrium structure of programmed DNA assemblies that are designed to reside on a honeycomb or square lattice. The DNA double-helix is modelled as an elastic cylinder which bends and twists; cross-overs are assumed to be rigid links in between the neighbouring helices. The analysis ignores the sequence information.

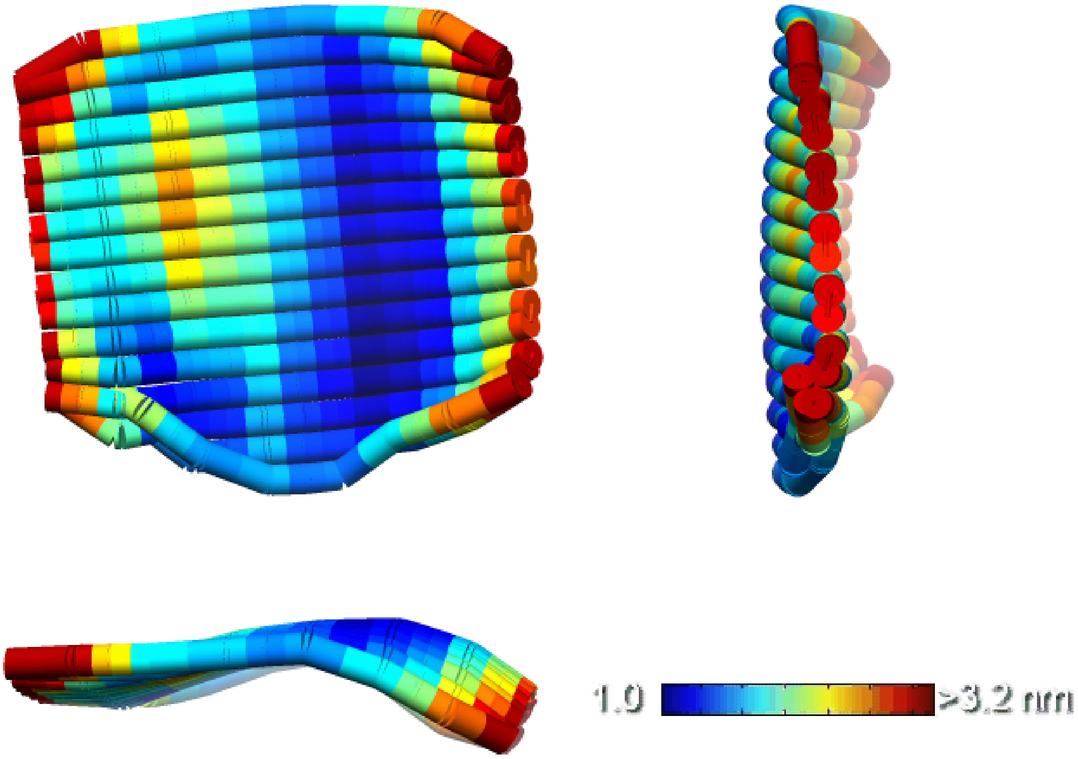


Figure 3.11: 3D equilibrium structure of a square DNA origami obtained with canDo software.

The resulting structure is shown on Figure 3.11. Deformed object is colour-coded according to the root mean square of the thermal fluctuations as calculated by the finite element modelling procedure. Also, CanDo is parametrised for the B-form structure of DNA. Since the triangular design follows A-form (i.e. RNA-DNA hybrid) the resulting prediction appears severely deformed and hence is not shown here.

3.6 Workflow and Summary

The workflow developed in this chapter to test the hypothesis is shown in Figure 3.12. The overall design pipeline is divided into four steps: (i) construction of De Bruijn graph, (ii) filtering of biological sequences, (iii) construction of alternative De Bruijn sequences and (iv) further optimisation. Currently, steps (i-iii) are fully automated, while step (iv) is semi-automated as the optimisation procedures differ depending on the criteria chosen for the specific application.

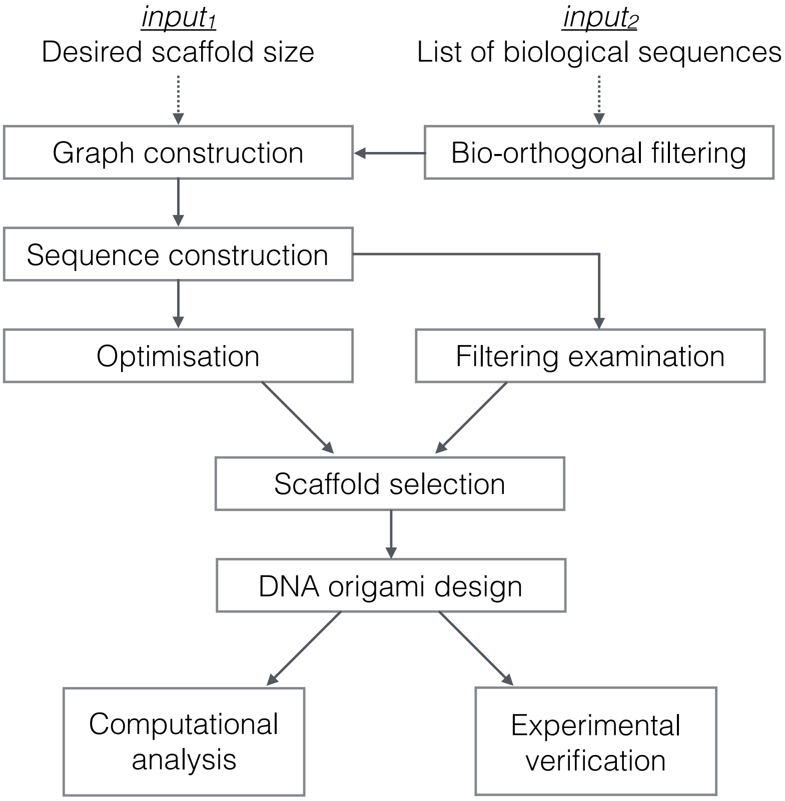


FIGURE 3.12: Workflow developed to test the hypothesis.

In this chapter, we presented the design phase of the synthetic scaffolds for DNA and RNA origami. We showed how the pipeline above was used to create a case study for these nanostructures. The computational analysis and experimental verification are the focus of the following chapter.

Chapter 4

Computational and Experimental Results

In this chapter we seek a verification of the DNA origami system with synthetic scaffolds; using both computational and experimental methods we investigate the aspects of biological orthogonality and unique addressability. We attempt to answer how "synthetic" our scaffold sequences really are using bioinformatics tools. Then, we introduce two types of addressability measures (i.e. thermodynamic addressability of staples and scaffold) which we apply to designs of Chapter 3. Comparisons are made with several of the standard viral scaffolds used in DNA origami. Finally, we pursue the experimental demonstration of the nanostructures folding.

4.1 Introduction

In this chapter, we use a variety of tools to asses bio-orthogonality and addressability. Then, we pursue experimental verification of the DNA origami with synthetic scaffolds. We explain the protocols that were used to obtain different scaffolds and fold the nanostructures. We follow with the images of the origami samples: square tile based on pUC19 (control experiment), as well as square and triangular tile based on synthetic DBS. The experiments were performed in collaboration with Alessandro Ceccarelli, Jing-Ying Gu, and Chien-Yi Chang.

4.2 Bio-orthogonality

Here, we investigate the question of biological orthogonality. We show the sources of biological data that were used to constrain the construction of DBS; then we highlight the contrast between synthetic DBS scaffolds and standard viral scaffolds used in DNA origami.

4.2.1 Sources of Biological Sequences

To demonstrate the site-specific sequence constraining, the sequence data related to *E.coli* (K12 strain) was fetched from the PRODORIC[142] database together with a list of common restriction endonucleases provided by New England Biolabs (NEB) and eliminated in the scaffold generation.

Prodoric (Prokaryotic Database Of Gene Regulation) aids the search for DNA binding sites of various organisms including bacteria such as *Escherichia coli.* Records from the database contain information such as site name, region, element, DNA sequence and its position in the genome.

Querying Prodoric for the complete chromosome of *E. coli* (K-12 strain) returns 1698 entries: 26 of those contain empty sequences (virtual sites) and among the remaining entries a total of 1384 unique DNA sequences could be identified. Their length ranges from 4 to 69 bp, with an average being $\overline{x} = 21.81$ bp and standard deviation $\sigma = 10.56$ bp. Figure 4.1 shows the distribution of those sequences.

Following conclusion from section 3.2 we can divide the data into two segments when confronted with the order of De Bruijn sequences $B(k, n)$:

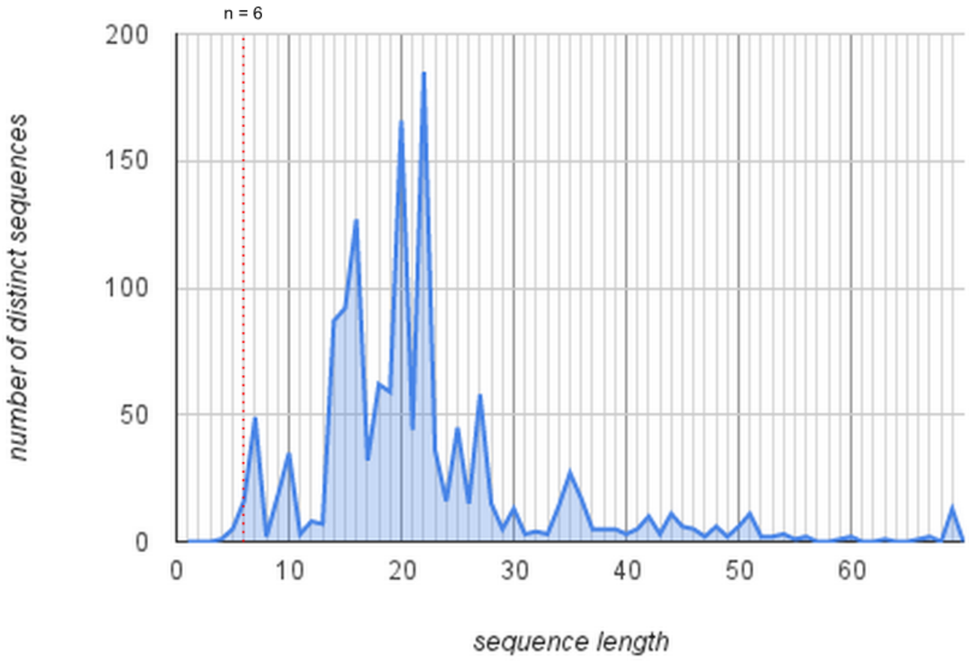


FIGURE 4.1: PRODORIC collective data: histogram of known binding sites for *E. coli* (K-12 strain). Red dotted line set at $n = 6$ is an example of separation between sequences *guaranteed* to be found in $B(4, 6)$ from those that *might* be found.

1. set of sequences of length $x \leq n$
2. set of sequences of length $x > n$

The former includes subsequences that are certain to be found; in a single $(x = n)$ or multiple $(x < n)$ places across the $B(k, n)$ sequence. The latter contains sequences that are probable, yet the probability that they occur decreases exponentially - proportional to their length.

The second DBS was additionally constrained to exclude RNA-specific sequences (and their reverse-complements). These were[1]: starting codon (ATG), Shine-Dalgarno sequence (GGAGG), T7 promoter (TAATAC...), lac operon (GGAATT...), PacI restriction enzyme (TTAATT...), EcoNI restriction enzyme (CCT...), ClaI restriction enzyme (ATCGAT), and two custom linkers sequences (CGATCC, CGCGAA).

[1] the sequences given in parentheses were removed, note that some coding sequences are longer, however, it is sufficient to remove first $(n = 6)$ bases to prevent the sequence occurring in the final scaffold

After construction, the T7 promoter was inserted upstream of the second DBS to enable transcription *in vitro* (similar to the previous study [141]). Deeper explanation is provided in Section 4.4. The analysis of the filtering procedures is provided in the following section.

4.2.2 Validation

A DBS may contain some undesirable sequences, such as restriction enzymes binding sites. One may want to constrain the scaffold construction with certain site-specific sequences. Thus, in the second step, the user specifies a set of forbidden DNA sequences which will be excluded from DBS scaffold.

The synthetic DBS scaffold construction was restrained by a set of forbidden sequences. As mentioned in Section 3.2.2, we fetched the sequence data related to *E.coli* K12 from the PRODORIC[142] database together with a list of restriction endonucleases provided by New England Biolabs (NEB). The removal of taboo sequences is summarised in Table 4.1.

Scaffold	PRODORIC	NEB	
(length in knt)		common	all
DBS (1.1)	0	0	27
DBS (2.4)	0	0	63
pUC19 (2.6)	28	9	66
M13mp18 (7.2)	42	9	89
λ-phage (48.5)	69	12	145

TABLE 4.1: Number of hits in databases for scaffold sequences. The data is from: PRODORIC database for *E.coli* strain K12 (1686 entries), NEB list of restriction endonucleases (280 entries; 13 selected as common). Note that both PRODORIC and common NEB were used to constrain the generation of DBS and thus no hits were found for those databases.

In addition, several other databases and software tools for short motifs predictions have been used, including: Pfam [143], CATH [144], tRNAscan [145], Glimmer [146], TMHMM [147] and miRBase [148, 149]. No hits were obtained for the two DBS that were tested. The lack of matches can be explained by the fact these databases contain relatively long sequences (usually proteins) and thus are extremely unlikely to appear in any DBS of order 6.

To further validate the bio-orthogonality, we used the *Reciprocal Best Hits* (RBH) method. NCBIs BLAST has been used to find alignments of DBS against known genetic sequences. Significant hits were found when adjusting advanced options of BLASTN to word size 16. The analysis revealed six alignments in the two sequences we designed for this study (Table 4.2). The low scores confirm the synthetic nature of the DBS thus further supporting it as a novel bio-orthogonal method for designing DNA origami, as these few hits can easily be added to the taboo sequences for filtering purposes.

Scaffold	Genome	Score	E Value	Identities	Accession No.
DBS (1.1)	Halichoerus grypus	43.6	7.6	100%	`JX218922.1`
DBS (2.4)	Spirometra erinaceieuropaei	51.0	0.11	100%	`LN056044.1`
DBS (2.4)	Thelazia callipaeda	51.0	1.4	94%	`LK979655.1`
DBS (2.4)	Ovis canadensis canadensis	47.3	1.4	94%	`CP011893.1`
DBS (2.4)	Ovis Aries (predicted)	47.3	1.4	94%	`XR_001042372.1`
DBS (2.4)	Protopolystoma xenopodis	45.4	5.2	96%	`LM741928.1`

Table 4.2: BLAST alignment results.

4.3 Addressability

Here, we investigate the question of addressability. Two methods are presented, the former is based on the sequence composition alone while the latter is taking into consideration the thermodynamics of the DNA hybridisation.

4.3.1 Repetitions in Natural Scaffolds

First, we quantify the number of repeated sequences in the different scaffold. This can be determined using suffix trees. A suffix tree is a compressed, ordered data structure containing all suffixes of the given text [150]. Once a suffix tree is generated, the count of longest repeats can be obtained in a trivial way. Every node in that tree with at least two children corresponds to a repeated sequence; by counting the number of the deepest nodes (that have at least two children) we obtain the count of the longest repeats. Note that we count from the longest to the shortest, so as to avoid double counting the subsequences of previously found repeats.

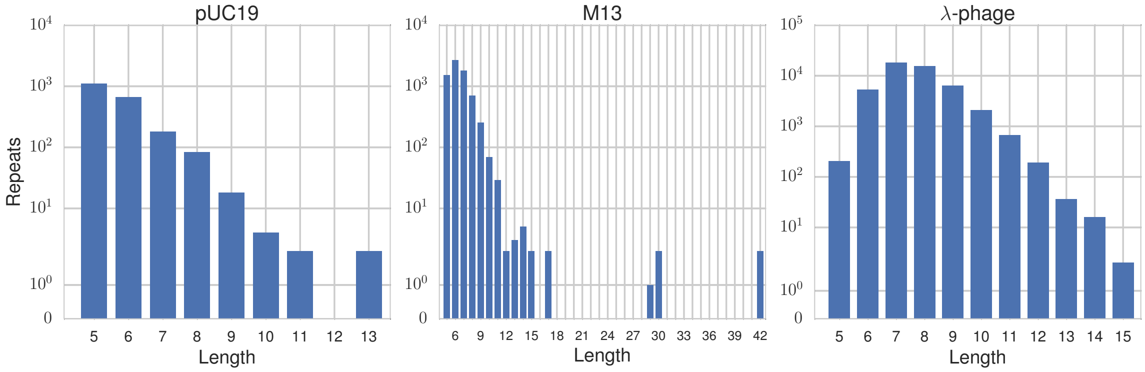


FIGURE 4.2: Repetitions in various scaffolds. 2.6kb pUC19 vector (left), 7.2kb M13mp18 bacteriophage genome (middle) and 48kb λ-phage genome (right). A number of the repeated sequences (k-mers) in the scaffold is plotted according to the length k. Note that only the longest repeats are shown and their respective subsequences are excluded. DBS is not included because it contain by design no repeats longer than 5 nt.

Using this method, we analysed statistical redundancy of the three common scaffolds for DNA origami: pUC19, M13mp18 and λ-phage. The number of the repeated sequences (k-mers) is plotted according to the length k (Figure 4.2). Note that only the longest repeats are shown and their respective subsequences are excluded. Existing scaffolds contain many repetitions which are longer than the typical binding domains of staples. In 2-dimensional structures, staple domains are usually composed of 8 nt (or multiples of it). In 3-dimensional structures (build on honeycomb lattice) domains are shorter - typically multiples of 7 nt. For example, the most frequently used scaffold, M13mp18 has over 10^3 repeats of length ≥ 8 nt, while λ-phage has over 10^4 of them. How many of these repeats occur at staple binding domains depends on the particular design and choice of corresponding staple set. Generally, the number, as well as length of repeating sequences grows proportionally to the scaffold length. M13mp18 has the longest repeats spanning 29, 30 and 42 nt which are representative examples of ambiguity in staple addressability. Interestingly, they appear as outliers in the underlying distribution of repeats and are not present in the other two scaffolds, for which the longest repeats span 13 and 15 nt respectively. In comparison synthetic DBS scaffolds of length 4 kb, 16 kb and 64 kb can be constructed such that they have no repeats longer than 5 nt, 6 nt and 7 nt respectively.

4.3.2 Sequence Addressability

In this section, we examine the sequence addressability in DBS and pUC19 scaffolds; and how they influence addressability of staples in the square design. This is achieved by listing all possible k-mers (here, of length 6) and arranging them into an address map (see Figure 4.3).

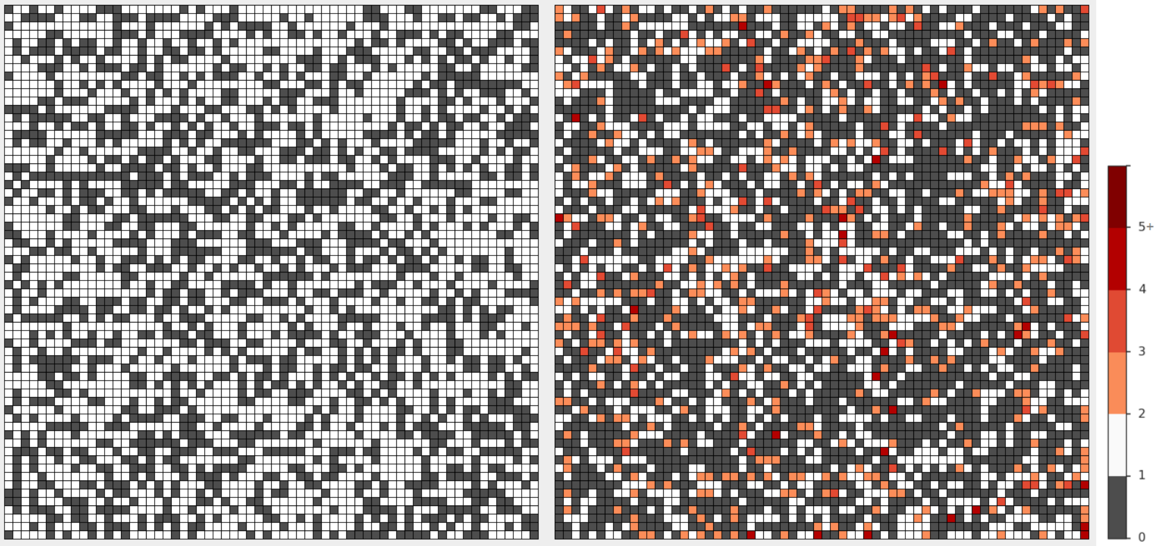


FIGURE 4.3: Address maps ($k = 6$) of De Bruijn scaffold (left) and pUC19 scaffold (right). Colours indicate the number of k-mers in the scaffold.

In the map every k-mer has a unique position; for instance, sequence 'AAAAAA' occupies the top-left address, 'CCCCCC' occupies top-right address, while 'GGGGGG' and 'TTTTTT' are in bottom-left and bottom-right addresses, respectively. The colour indicates the number of particular k-mer occurrences in the scaffold sequence. For DBS the sequences are unique, hence every k-mer either appears once or was removed through filtering. For pUC19, there are many k-mer which are repeated.

This repetitions influence staple addressability. For every staple in the design, we checked all k-mers that are included in that staple primary sequence. All k-mers are

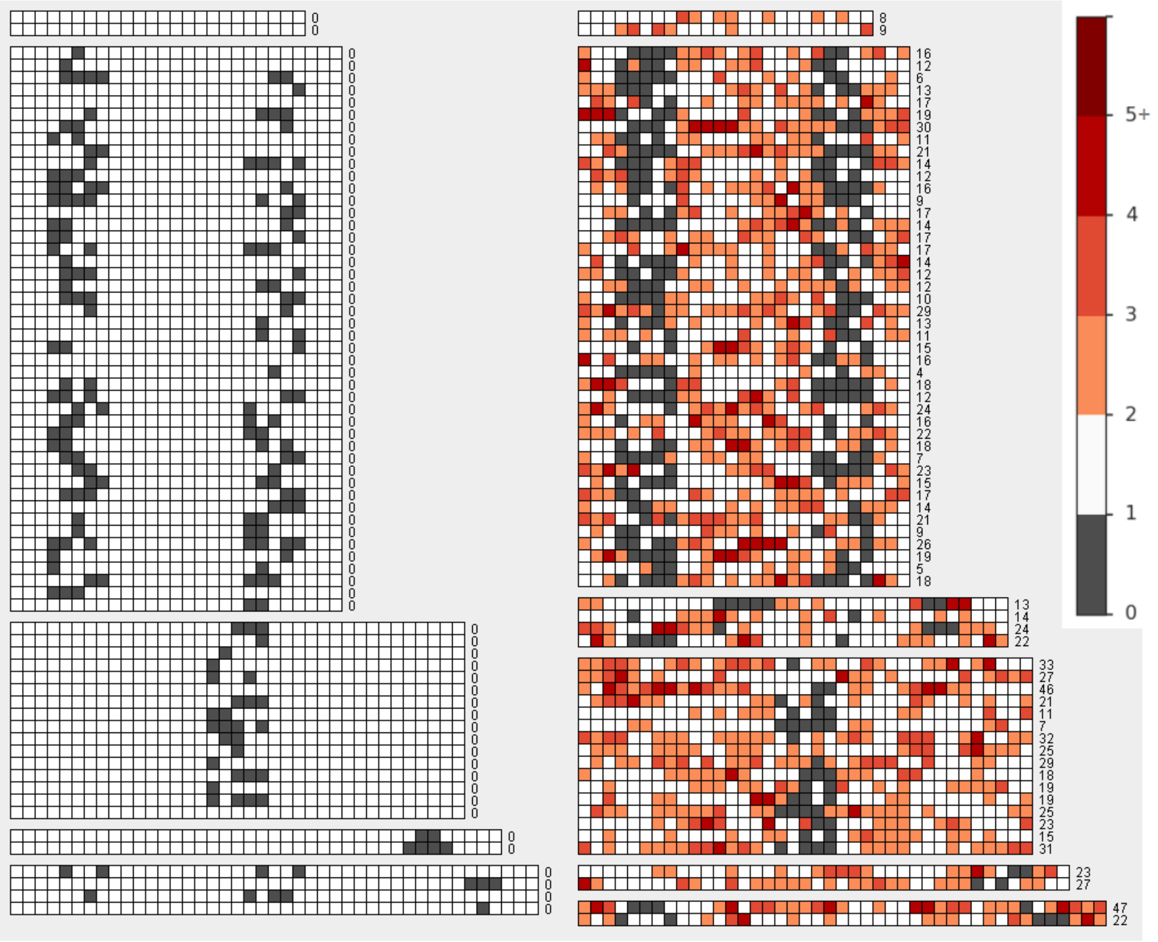


FIGURE 4.4: Staple chains ($k = 6$) from a DNA origami square design based on DBS scaffold (left) and pUC19 scaffold (right). Colours indicate the number of reverse complement k-mers in the scaffold. The number following each staple chain indicates the sum of k-mers hits.

colour-coded based on the number of reverse-complementary k-mers present in the scaffold sequence indicating a potential binding site (see Figure 4.4).

This analysis presents a basic summary of repeated sequences. The following section provides a much more comprehensive analysis of addressability.

4.3.3 Thermodynamic Addressability

Sequence uniqueness alone does not suffice to guarantee a unique binding at the target location. Since the scaffold might contain slight alternations of the sequence to which the staple might bind with mismatches (although with smaller binding affinity). That is why we investigated unique addressability based on binding energies in different designs and scaffolds configurations.

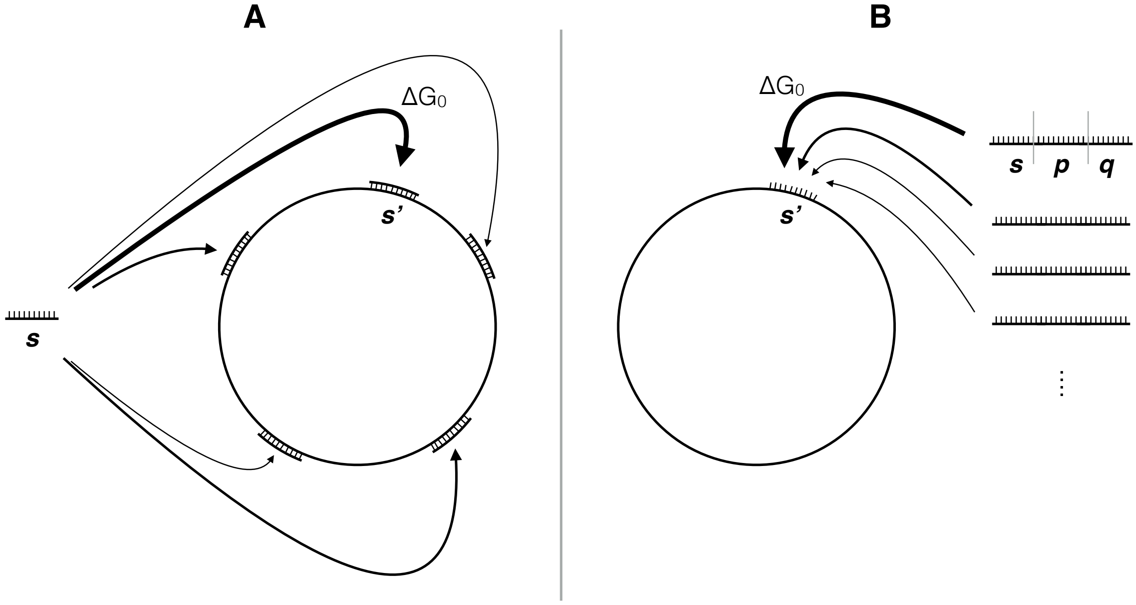


FIGURE 4.5: Two types of addressability measures: (A) how likely is the staple domain (s) to bind the correct region of scaffold (s'); (B) how likely is the scaffold domain (s') to accept the correct staple (spq).

A custom-built algorithm is used to calculate the addressability measure for each staple (see Figure 4.5A). First, a simple heuristic based on Levenshtein distance finds all possible regions of the scaffold to which a staple can hybridise. When a possible binding site is detected the associated thermodynamic potential is derived using ViennaRNA package[36] (with appropriate energy parameters provided by[139]). The resulting thermodynamic potentials (measured as Gibbs free energy) are used to establish the relative probability of staple hybridising at the specific location according to the Boltzmann distribution. In other words, as the addressability measure increases the staple is more likely to bind to the correct target.

Similarly, an algorithm is used to calculate the addressability measure for each domain in the scaffold designed to bind a staple (see Figure 4.5B). This procedure is more straightforward, as for each scaffold domain, the relative probability is derived by measuring thermodynamic potential with the whole repertoire of staples. In this case, as the addressability measure increases, the domain in scaffold is more likely to accept the correct staple.

Varying the DNA origami designs affects the addressability of staples. The most common designs for two-dimensional shapes contain staples which are composed of 3 domains: 8-16-8 nt in length; these types of designs were tested here (see Figure 4.6a).

We found that longer staple domains ($>$ 8 nt long) have nearly perfect addressability measure (the probability is approximately 1) in all examined designs. For short domains ($\leq$ 8 nt long) there is a strong tendency for DBS scaffolds to have a higher addressability measure than their biological counterparts (Figure 4.6b-c). It is the case not only for pUC19 and DBS (2.4 knt) which fold into a small DNA origami tile (presented in this study) but also for the theoretical medium tile design (85x85 nm) based on M13mp18 and DBS (order 7). Although the addressability measure in the large tile designs (200x200 nm) is generally low, the synthetic DBS (order 8) still outperforms the λ-phage scaffold (for the first measure; there is little difference for the second measure). These results suggest that longer scaffolds have a higher probability of mismatching, which is partly caused by the repeats in the scaffolds, and therefore ambiguity of staple addressing. Moreover, it might explain the difficulties of folding larger DNA origami using λ-phage scaffold [151].

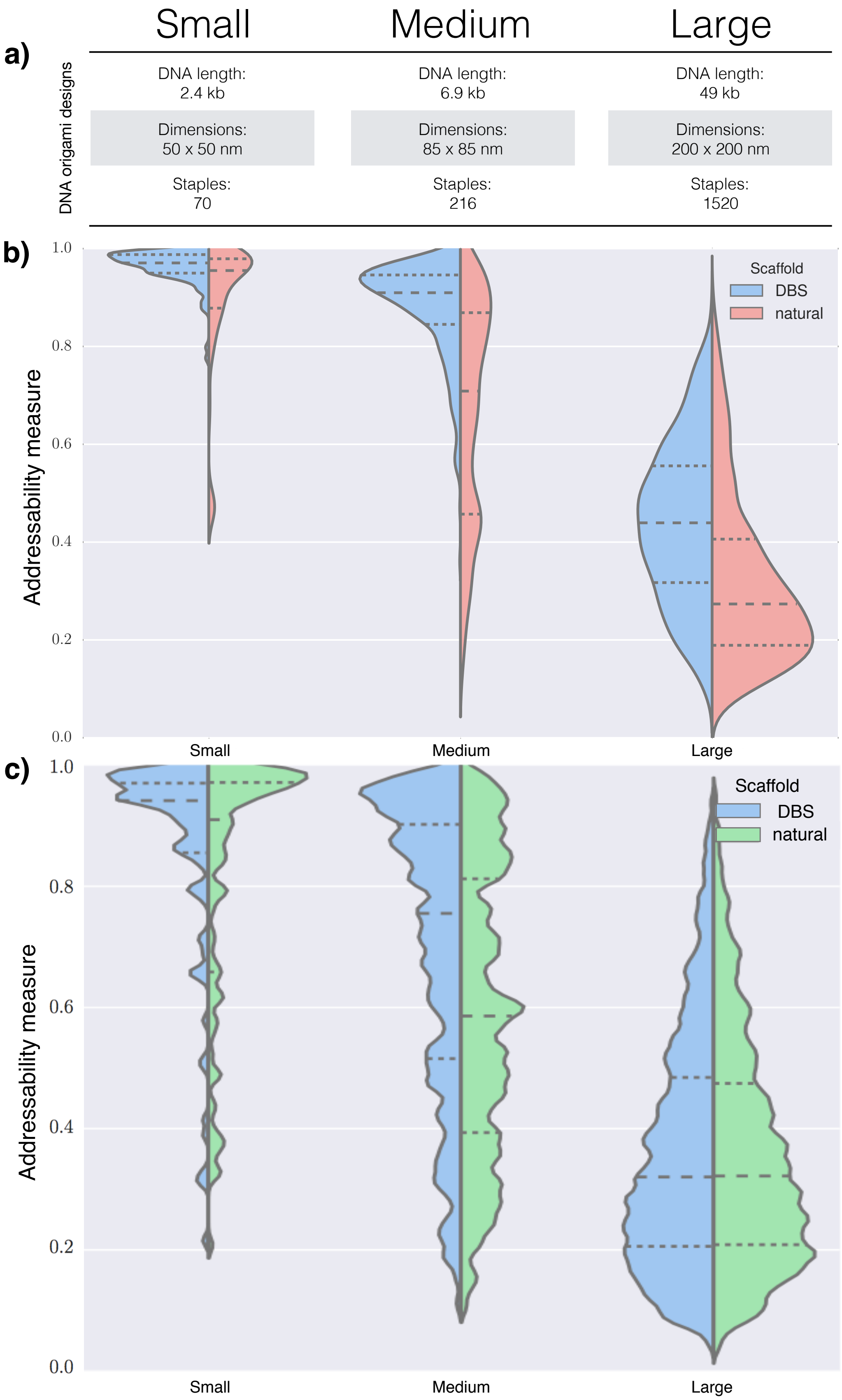


Figure 4.6: Thermodynamic addressability in different designs. DNA origami designs utilising scaffolds of increasing size (a); Small (pUC19), medium (M13mp18) and large (λ-phage) compared with DBS in terms of staple (b) and scaffold (c) addressability; smoothing is applied for representational purposes.

4.4 Experimental Validation

The first synthetic DBS was constructed to fold into a square DNA origami, roughly 50 nm in size, which required 2.4 knt of the scaffold. The second synthetic DBS was constructed to fold into a triangular RNA-DNA hybrid origami, roughly 30 nm in size, which required 1.1 knt of the scaffold. The construction of single-stranded scaffolds from double-stranded plasmids is illustrated in Figure 4.7.

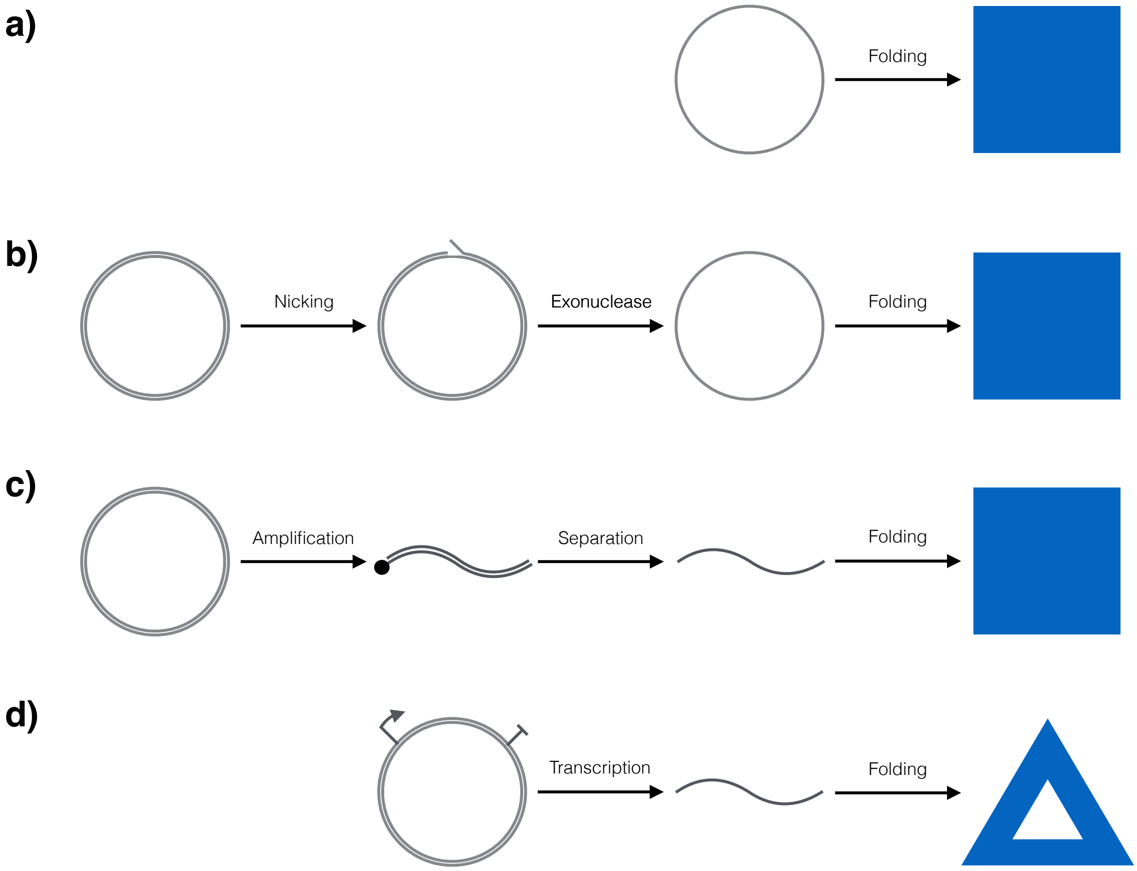


FIGURE 4.7: Scaffold preparation protocols: a) the common folding protocol assumes ready-to-use viral ssDNA; b) removal of anti-scaffold strand from pUC19 via enzymatic reactions; c) removal of anti-scaffold of de Bruijn PCR product using magnetic beads; d) transcription of RNA de Bruijn scaffold

The 2.6 Kb long pUC19 cloning vector was subjected to enzymatic reactions to obtain single-stranded DNA scaffold. This reaction nicked the anti-scaffold strand of DNA and allowed for the digestion of the anti-scaffold strand (Figure 4.7b).

The 2.4 Kb De Bruijn DNA sequence encoded in a commercial plasmid was amplified through a PCR. The reverse primer was modified with a biotin molecule linked through a triethylene glycol (TEG) spacer-arm (IDT). The sequence was then attached to the magnetic beads and denatured. The complementary strand was finally removed through magnetic beads (Figure 4.7c). The single-stranded sequence was purified through agarose gel electrophoresis [152].

Finally, the 1.1 Kb long De Bruijn RNA sequence was synthesised with a standard T7 transcription kit and purified (Figure 4.7d).

The detailed protocols for scaffold preparation are explained in the Appendix A.

DNA Origami of the Square Tile

First, as a control experiment we folded a pUC19 scaffold into a square (Figure 4.8 bottom panel). The square design follows closely the design shown for DBS square (Section 2.5.1). The 2.6 kb long cloning vector was subjected to enzymatic reactions and folded into a square DNA origami tile following protocols described in Reference [47]. Because the folding of a square required only 2.4 knt of the scaffold, the remaining 200 nt were left unpaired and formed a dangling loop at the corner of the shape.

The De Bruijn origami was then folded into a square following the rapid isothermal protocol described by Sobczak et al. [113] This method grants a more stable product with a lower rate of misfolding, reducing the folding time from hours to minutes. The samples were finally analysed by AFM to compare the quality against the pUC19 DNA origami.

For AFM imaging, 5 μL of purified origami sample solution was applied onto freshly-cleaved mica. Subsequently, 10 μL of $NiCl_2$ (10 mM) was applied to stabilise the sample on the substrate. AFM imaging was performed on a Bruker multimode 8 AFM in ScanAsyst mode, using Bruker Scanasyst-Fluid+ tip. Atomic force microscopy (AFM) imaging confirmed the correct folding of the nanostructures (Figure 4.8).

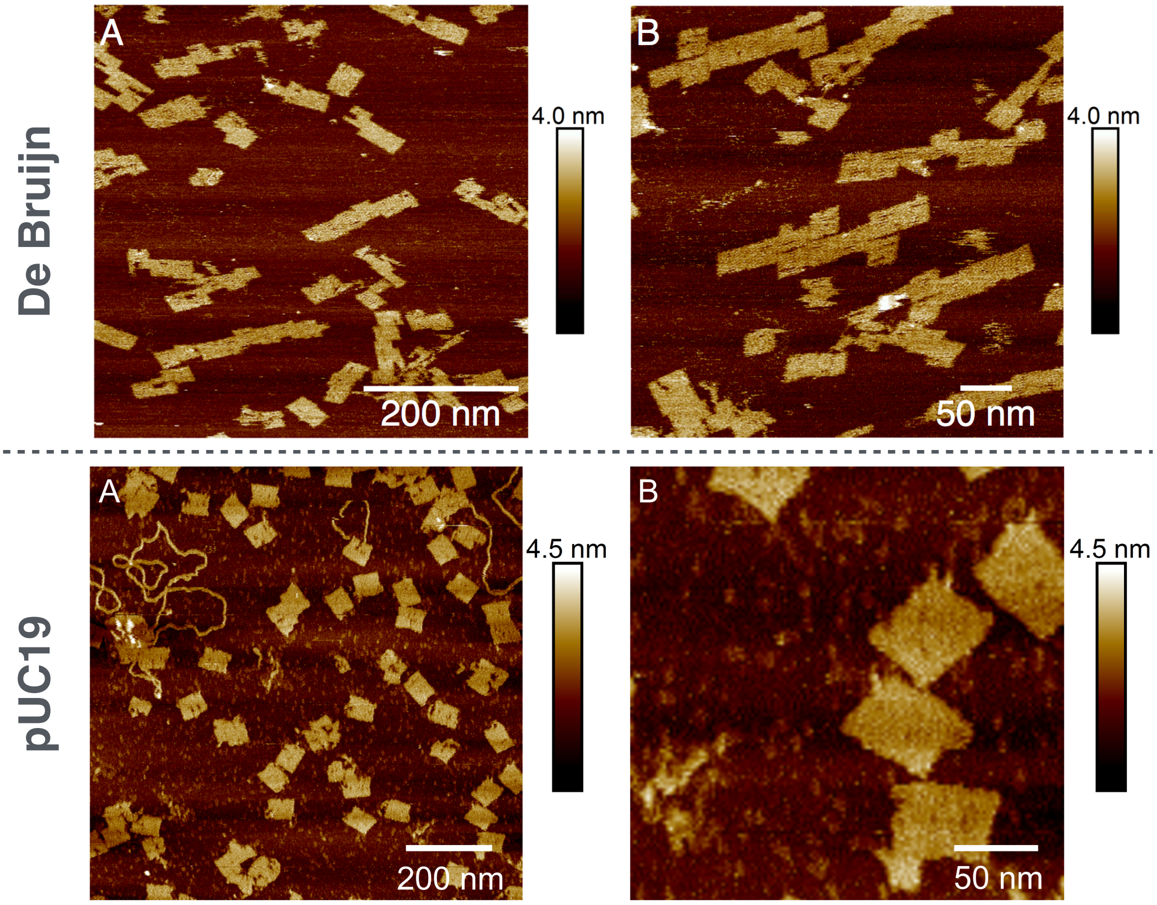


FIGURE 4.8: AFM imaging of the square DNA origami based on DBS scaffold (top) and pUC19 scaffold (bottom).

RNA-DNA Hybrid Origami of Triangular Tile

Furthermore, in order to test our design for RNA-DNA hybrid origamis, we constructed a 1.1 knt long DBS scaffold which was designed to fold into a triangular tile with a hole. Again, a DBS satisfying this length requirement was built from 6 nt long unique subsequences.

The RNA-DNA hybrid origami follows the protocol similar to the previous study [141] (see Figure 4.7d).

These two experiments demonstrate that DBS scaffolds can be utilised in the same manner as viral ones without folding protocol change. The bottleneck was a single-stranded DNA scaffold production, which is long-drawn and labour-intensive for synthetic DBS: it requires excessive laboratory handling in order to produce the volume

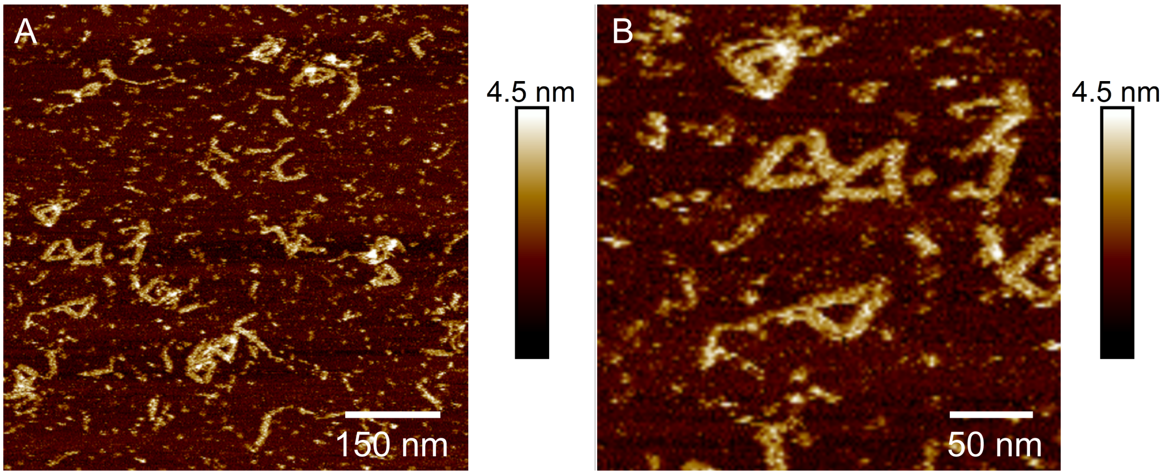


Figure 4.9: AFM imaging of the triangular RNA-DNA hybrid origami based on DBS scaffold.

of DNA which is comparable to that of the viral scaffold. This is not the case for RNA-DNA origami, where the scaffold is an RNA transcript and can be easily synthesised. However, the drawback of RNA scaffolds is that once produced they are subject to degradation by RNases (and hence require special lab conditions).

4.5 Summary

Establishing the nonspecific sources of interactions (or interference) within a given biological system is a challenging task. Regardless of this difficulty, we explained here how our method allows explicit exclusion or inclusion of sequence specific sites from the synthetic scaffold. In consequence, the final DNA nanostructure can be programmed for bio-orthogonality (provided that one has sufficient knowledge about the given biological system).

Our computational analysis shows that the repetition of sequences in natural scaffolds has a negative impact on the staple specificity. This problem is magnified for longer scaffolds because the number of potentially stable targets for a staple grows proportionally with the scaffold length. (An obvious solution would be to use only long staples, however, the exact hybridisation kinetics of longer sequences might be not-trivial; also, sparse double-crossover motifs may compromise the rigidity of nanostructures.) Thus, the use of natural sequences does not scale well for the creation of large objects based on a single scaffold. Further, we show that scaffolds based on DBS provide more specificity and are therefore uniquely addressable.

Also, we showed experimentally that DBS scaffolds can be utilised for DNA and DNA/RNA hybrid origami without folding protocol change.

Part II

DNA COMPUTING

Chapter 5

DNA-based Stack Machine

> We propose the *in vitro* implementation of a DNA data structure, where data and operations form the core of the molecular interaction network. We demonstrate how an evolutionary algorithm can be used to optimise the system for maximal robustness among all molecular interactions and minimal occurrence of undesirable reactions. The *stack* data structure is here employed as a reversible, and potentially unlimited, data storage. The following Chapter 6 evaluates the design we propose and develop here.

5.1 Introduction

In this chapter we propose a conceptual framework to address hypothesis H_2:

> H_2 : It is possible to program a synthetic DNA structure allowing recording of data in a controllable and, in principle (albeit not physically), unlimited manner.

To tackle this hypothesis, we have decided to implement a DNA-based stack machine. Stack is among the most elementary data structures – operations on stack occur only at one end of the structure [153]. The simplicity of operation and sufficient computational power were the two key features motivating our prototype creation. The specification of the DNA stack data structure, presented here, was done collaboratively with N. Lopiccolo. The sequence optimisation (Section 5.6) and computational analysis (Chapter 6) is my own work. In addition, the laboratory work was done in collaboration with N. Lopiccolo who performed the experimental verification of my designs (Chapter 6).

This chapter describes how to implement a stack machine using DNA strands. We explain the recording and reading operations (Section 5.4); these two modes of operation are realised through toehold-mediated strand displacement reactions. We chose DNA strand displacement because this mechanism allows for the strict kinetic control of reaction pathways; moreover, it is enzyme-free as opposed to Benenson et al. [81]. To design robust system favouring desired operations and minimising unspecific interactions we resorted to a genetic algorithm. Evolutionary algorithms have been successfully utilised for evolving nano-scale and self-assembling systems in the past [154–157]. We conclude the chapter with the proposed solution: a set of bricks generated by our algorithm.

The stack was optimised for RNA interaction as the end goal is to use this system *in vivo*. However, here we verify its feasibility using DNA strands. For this reason, when DNA is mentioned it does apply to both DNA and RNA. When we speak about RNA we specifically mean RNA only.

5.2 Stack Data Structure

A stack is an abstract data structure that stores as a sequential collection of elements, with two principal operations: *push* adds a new element to the top of the stack, and *pop* removes an element from the top of stack [153]. Formally, the stack is implemented as follows:

$$\begin{aligned} push &: \text{stack} \times \text{element} \longrightarrow \text{stack} \\ pop &: \text{stack} \longrightarrow \text{stack} \times \text{element} \end{aligned}$$

with the invariant

$$pop(push(\text{stack}, \text{element})) = \text{stack}, \text{element}$$

to guarantee *last-in-first-out* operation.

Further common but non-essential operations such as *peek* (return the last element without removal) and *empty* (return true if the stack experienced at least as many *pop* as *push* operations) are not provided in our implementation.

Fully implementing this data type in DNA requires molecular realisations of the assembled stack, all potential elements, as well as the push and pop operations. We achieve this by associating each data element and each operation with a single-stranded DNA (ssDNA) oligonucleotide with partial secondary structures. We call those strands "DNA bricks", or simply "bricks". An abstract schematic of the operation of a molecular stack machine is depicted on Figure 5.1.

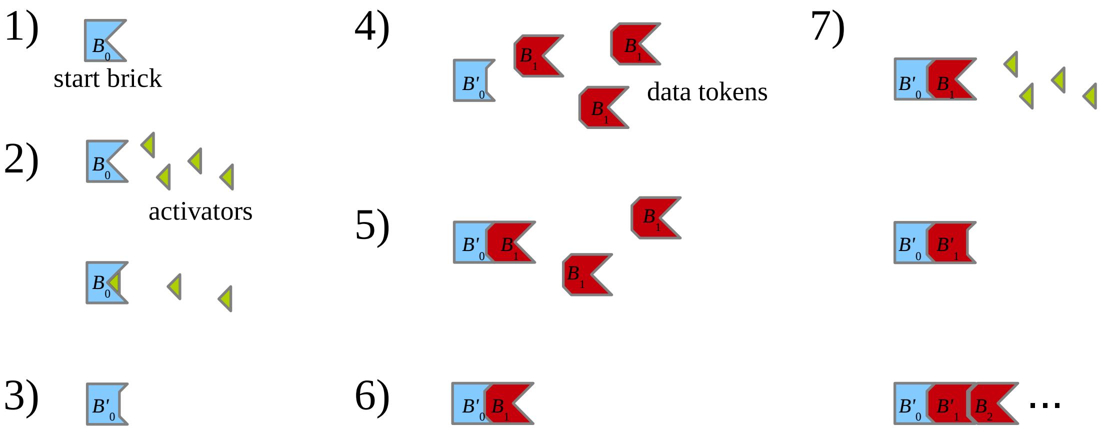


FIGURE 5.1: An abstract depiction of molecular stack machine. Here, the stack is initialised with the start bricks (light blue); which are then triggered by activators (green) to accept data tokens (red). This cycle allows, in principle, an unlimited data storage.

Core specification:

1. The data structure is implemented by a growing chain of DNA "bricks".
2. There is exactly one site at which the chain can grow by addition of write strands.
3. Toggling between an activator strand and a write strand ensures that only one data token is recorded. Prevents run-away process.
4. Surplus activators and unbound write bricks should be removed or degraded between the individual steps.
5. Addition of specific readout strands should release recorded information (last in, first out)

Additional desired features:

6. Recording followed by readout should leave the stack state (effectively) invariant.
7. Binding sites could be addressable by specific sequences to allow for multiple stacks.

The last two features would allow the implementation of (Turing-universal) stack machines as suggested in previous studies [24, 158].

5.3 DNA Bricks Design

The stack data structure is built from bricks via hybridization of complementary DNA domains. More precisely, the stack forms a double stranded DNA (dsDNA) assembly with essentially no single-stranded regions but one active toehold domain, that offers an entry for operation. This design aspect was chosen to ensure bio-orthogonality (see Section 5.5 for details). Data bricks form the top strand and push bricks form the bottom strand of this dsDNA assembly.

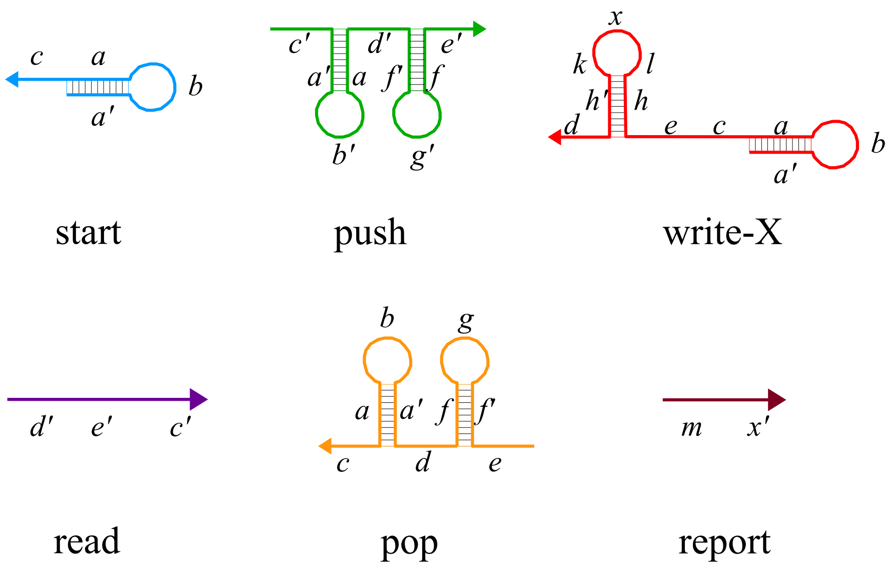


Figure 5.2: Schematic of the different bricks involved in the DNA data structure. Arrows indicate 5' → 3' direction.

brick	**label**	**domains**
start	S	$a'bac$
push	P	$c'a'b'ad'f'g'fe'$
write$_X$	X	$a'baceh_xlxk_xh'_xd$
write$_Y$	Y	$a'baceh_ylyk_yh'_yd$
read	R	$d'e'c'$
pop	Q	$ef'gfda'bac$
report$_X$	T$_X$	mx'
report$_Y$	T$_Y$	my'

Table 5.1: Specification of bricks in the design. The ssDNA strand sequences have been divided into domains. The strands are given in 5' to 3' direction.

domain	**length [nt]**	**domain**	**length [nt]**
a	6	h_x	10
b	4	h_y	25
c	11	k_x, k_y	10
d	10	l_x, l_y	10
e	10	x	11
f	6	y	11
g	5	m	11

Table 5.2: Specifications of individual domain lengths.

The stack operates with six distinct DNA bricks and is able to store combinations of two different data tokens, encoded by two types of data elements. Two further bricks are added for experimental analysis. See Figure 5.2 for a schematic representation of the employed bricks and their interactions (note the sequence notation: an apostrophe marks reverse complements, i.e. a' is a reverse complement sequence of a).

- *Start* (S): data brick initialising the data structure. It features a toehold domain for interaction with *push* and a hairpin motif at the 5' end. This hairpin undergoes branch migration with a complementary hairpin in *push* but is otherwise not functional in the current design.
- *Push* (P): operator brick to initiate subsequent recording of data tokens. The brick contains the complementary toehold for interaction with *start*, a hairpin motif complementary to the one in *start*, the second hairpin for structural reasons that does not participate in branch migration, and two toehold domains, one on each side of the structural hairpin, to bind *write* bricks.
- *Write* (X/Y): data bricks that can be stored in the data structure. These bricks contain two toehold domains complementary to the push toeholds, a structural hairpin that does not undergo branch migration, plus the same toehold domain and 5' hairpin that form the *start* brick. Toehold domains and branch migration hairpins are identical for all types of *write* bricks. Thus, they can only differ in their structural hairpin motif. Since these hairpins do not participate in hybridization or branch migration, they can be functionalized to host any desired functionality such as recognition sites for DNA binding proteins.

  We employ two different types of *write* bricks, denoted as *write-X* and *write-Y*. *Write-Y* features a longer hairpin stem than *write-X* (twenty-five base pairs against ten base pairs) and has a different sequence in its stem-loop. Although we currently employ binary data (X or Y), the approach is intrinsically n-ary.
- *Pop* (Q): data brick that undoes the rightmost *push* operation. This brick is the exact complement of *push*
- *Read* (R): operator brick that removes the rightmost *write* operation. The brick is the complement of all toehold domains used in *write*'s. Notably, it does not contain any domains that interact with the structural hairpin of *write* bricks.

- *Report* (T): non-essential bricks for experimental analysis. Report bricks do not participate directly in the operations of the stack data structure. Instead, they interact with the data domains of structural hairpins in the *write* bricks. *Report* bricks can be added to the device in any configuration since their binding sites in the data hairpins are always accessible and since they do not interfere with the operating modes of the device.

  In this study, we use linear *report* strands that are 5' biotinylated via a 2.6 nm tetraethylene glycol (TEG) spacer. We functionalized these *report* bricks with streptavidin coated gold nanoparticles of different diameters, which allows for easy recognition using transmission electron microscopy (TEM).

We introduce single-letter nomenclature for the bricks to easily denote the state of the stack. For instance, an empty stack is denoted by single start brick: S, while a stack with a single push attached is denoted by: SP. Each of the configurations SPX and SPY denote stacks with one data token recorded. In a similar way, there are four configurations of two data tokens recorder: SPXPX, SPXPY, SPYPX, and SPYPY.

5.4 Modes of Operation

DNA hybridization, branch migration and strand displacement are the three processes governing all DNA interactions involved in the system. All reactions are energetically downhill, driven by the binding energy of the closing toehold domains.

5.4.1 Recording

A schematic of the recording process is shown in Figure 5.3. Starting from an empty stack, which is represented by the *start* brick (S), the device is toggled into its data state by providing a *push* operator (P). The *start-push* interaction begins by irreversibly binding toehold *c* and continues via branch migration among the two complementary *aba'* domains. The stack is now in its data state (SP), where a single

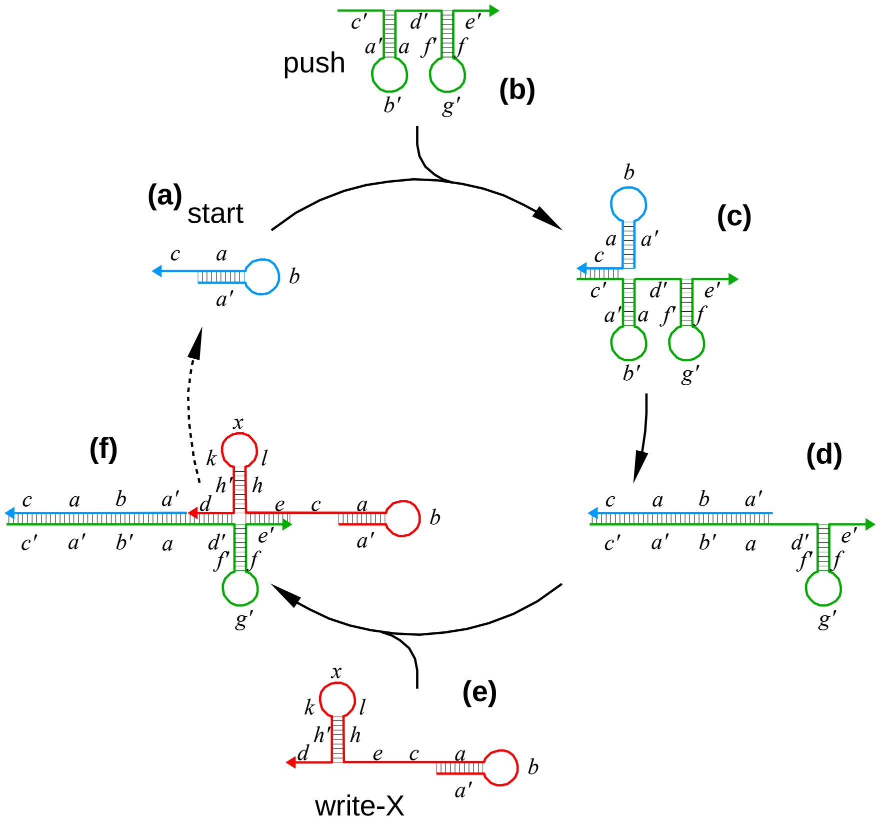


FIGURE 5.3: Schematic of the recording process. Start (a) and push (b) bricks collide and hybridise partially (c). Through the branch migration, they hybridise entirely (d). The DNA stack can now accept a free write brick (e). The final configuration (f) is analogous to (a) and the cycle can be repeated.

open toehold region (*d'e'*) can recruit a *write* brick (X or Y). The *write* will partially hybridise with the *d'e' push* toeholds, thus toggling the stack back into its operator state (SPX). In this state, the stack exposes the same toehold-hairpin interface that characterises the *start* brick, which allows the device to undergo subsequent rounds of recording.

Note that the assembled stack is essentially double stranded with a single exposed toehold domain. Because the structural hairpins of neither the *push* nor the *write* participate in branch migration, the stack will form holiday junctions for each recorded data element. As data specific domains are encoded in the loop regions of this holiday junction, the recording cycle is independent on the actual data written.

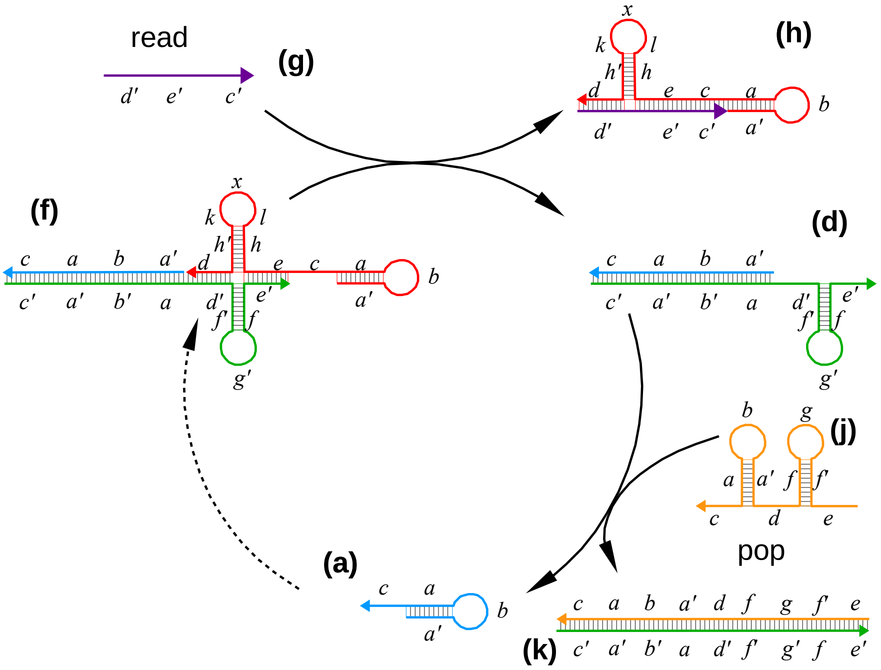


Figure 5.4: Schematic of the reading process. The stack containing one write brick (f) hybridises with the read brick (g). Through strand displacement reaction the data token is released (h) and the stack (d) can potentially accept a new write brick. However, if the pop brick (j) is provided instead, the stack is reset to its original configuration (a). This process produce a waste duplex (k).

5.4.2 Reading

While recording elongates the stack, the read-out will shorten the stack by undoing any recording in the last-in-first-out manner required by the stack specification. The read-out cycle is schematically presented in the Figure 5.4.

In operator state (SPX), providing a *read* brick (R) will peel the last recorded *write* brick off the stack, thereby toggling the device back into the data state (SP). This reaction proceeds in two steps: first, the *read* brick hybridises to the stack at its unique exposed c domain. Secondly, the dangling $d'e'$ domains of the *read* brick initiate a three-way branch migration with the $d'e'$ domains of the adjacent *push* brick against the de domains of the *write* brick, until the *push* strand is completely displaced.

Note that the data hairpin of the *write* brick does not participate in the branch migration. This ensures that a unique *read* brick can interact with any *write* brick, ensuring that data elements can be read from the data structure without a need to know which information has been stored. The resulting *read-write* complex (RX) does not expose any single-stranded domains and will not participate in further DNA interactions.

In its data state (SP), the stack can either be extended again with another data element by switching to the recording operation, or reading can be completed by toggling the stack back into its operator state. The latter is done by providing a *pop* brick (Q) that will interact with and peel off the exposed *push* brick. Analogue to the previous reaction, *pop-push* interactions are composed of their initial irreversible toehold hybridization, subsequent branch migration and eventual strand displacement. Again, the resulting *push-pop* complex (PQ) is completely double stranded and will not participate in further DNA interactions.

It is important to point here the issue of synchronisation. For example, it is possible that after a read action a write action can occur immediately (without a push having occurred first). While this is not really a problem *in vitro*, where one have control over the order of adding the bricks, it is a possible source of error *in vivo*. The synchronisation of operations has to be ensured if the stack was placed in a noisy cellular context.

5.5 Molecular Structure and Other Requirements

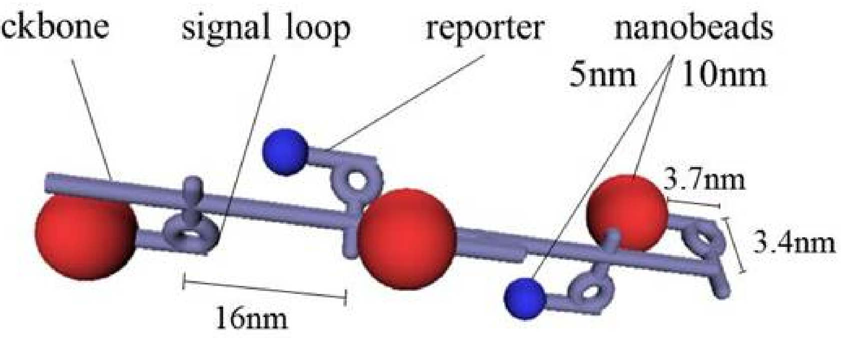


Figure 5.5: 3D conformation of an assembled DNA stack. The visualisation assumes a B-DNA form.

Domain sizes have been chosen with the following objectives: toeholds are long enough to span a single helical turn when hybridised with their complements (10 nt) which should promote irreversible hybridization. Hairpin loops that participate in branch migration are long enough to promote stable stems (6 base pair stems with 4-5 nt loops) but short enough to obtain quick branch migration times.

The structural hairpin loop of *write* bricks (containing data tokens) together with the unpaired domain of *report* are long enough to accommodate 5 nm and 10 nm diameter nanoparticles in close vicinity to the device. When assembled, subsequent data tokens are separated by the domains $ecaba'd$ (47 bp). Data-carrying hairpins are orthogonal and formed by domain h_x (10 bp) or h_y (25 bp). The biotin spacer is formed by m (11 bp) and k (10 bp) or part of the latter.

Assuming A-DNA conformation (raise 0.24 nm/bp, rotation 33.6°/bp) data-carrying hairpins are separated by about 11 nm and lie in a 139° degree turn. The data-carrying hairpin is orthogonal and 2.4 nm long. The biotin is separated from the recording site by an up to 2.5 nm spacer.

Assuming B-DNA conformation (raise 0.34 nm/bp, rotation 35.9°/bp) data-carrying hairpins are separated by about 16 nm and lie in a 247° degree turn (-112.7^0). The data-carrying hairpin is orthogonal and 3.4 nm long. The biotin is separated from the recording site by an up to 3.7 nm spacer. See Figure 5.5 for an example.

In order to facilitate future implementation of our device *in vivo*, all strands (except the report strands) have to begin with a sequence that encodes a promoter. That is because for the transcription to take place, the enzyme that synthesises RNA, namely RNA polymerase, must attach to the template DNA containing the brick "blueprint". Promoters contain a specific DNA sequence recognised by the polymerases which provide a secure initial binding site and regulate the transcription of RNA [159–161]. While the entire promoter sequence is necessary for a successful recognition and initiation of the transcription, only the last few nucleotides (underlined in the Table 5.3) are included in the transcription product. And thus, every brick should start with the

promoter	sequence
T3	AATTAACCCTCACTAAAGGGAGA
T7	TAATACGACTCACTATAGGGAGA
SP6	ATTTAGGTGACACTATAGAAGNG

Table 5.3: Potential promoter sequences. Information from the MEGAscript© Kit user guide 1330M(G).

sequence of the specific promoter that regulates its expression. The sequences here were chosen in such a way as to maximise the transcription efficiency [162].

In addition, binding sites of (unbound) push and pop strands are dangling single strands that allow for a $3'$→$5'$ ribonuclease (RNase) digestion. Similarly, the dangling strand of write bricks is oriented opposite to that of push and pops, and would thus be susceptible to RNAse digestion when unbound; however, write bricks bound to the stack should be protected from enzymatic attack.

Since all hybridization should be irreversible, we require $\mathrm{len}(c)$ and $\mathrm{len}(d) \geq 11$ (implying one full turn of an RNA A helix). To support stability of the $a'ba$ hairpin, $\mathrm{len}(a) = \mathrm{len}(a')$ should be at least four, and $\mathrm{len}(b)$ between four and eight nucleobases. The aba' motif should not constitute a terminator.

When assembled, subsequent data tokens are separated by the domains $ecaba'd$. Following the architecture of Reference [163], we would like this sequence to be 44 nucleotides long, implying 11 nm separation and 105° turn between two subsequent data tokens.

5.6 Genetic Algorithm for Sequence Optimisation

Designing the nucleotide sequences that make up the DNA data structure, based on the various specification and constraints mentioned previously, is a complex combinatorial design problem. Thus, an evolutionary algorithm is used to optimise the various sequences; namely, we employ a variant of a genetic algorithm [164]. Genetic algorithms are a way of solving problems by mimicking the processes of natural selection.

They rely on selection, recombination and mutation to quickly evolve a solution to an optimisation and search problems. The key advantage of using genetic algorithms is that they are most effective in handling a large and complex search space for which little is known.

Our custom-build genetic algorithm is based on the free and open-source inspyred[1] framework. For an overview of the main loop see Algorithm 2.

Algorithm 2 Evolve()

```
function EVOLVE
    initialise random population $P$
    evaluate fitness of $P$
    while terminator not satisfied do
        select parents $F_0 \leftarrow$ selector($P$)
        create offspring $F_1 \leftarrow$ variator($F_0$)
        evaluate fitness of $F_1$
        update population $P \leftarrow$ replacer($P$, $F_1$)
    end while
end function
```

In our representation, an individual (i.e. candidate solution) is described by its genotype, such that each gene corresponds to a separate domain from Table 5.1. Thus, a gene codes a domain sequence of a given length and initially consists of random nucleotides. A phenotype is expressed as a complete design of DNA bricks assembled from these genes; its fitness is evaluated using a multiple objective function (see Table B.1). The fitness function calculation (i.e. the evaluator) and other genetic operators are described in the following sections.

5.6.1 Evaluator

The fitness of an individual is evaluated based on two factors: desired secondary structure and binding energies. We implemented the following partial scoring functions: (i) hairpin loop formation S_{hlf}, (ii) intermolecular hybridisation S_{ih} and (iii) energy gain S_{eg}.

[1] Software available at https://pypi.python.org/pypi/inspyred

Hairpin loop formation evaluates a single RNA strand using RNAfold to predict its minimum free energy structure (MFE). The result is compared to a secondary structure imposed by the design. Similarly, intermolecular hybridisation uses RNAcofold to evaluate two RNA sequences which are forming a dimer structure. These functions are given by:

$$S_{hlf}(s, s') = len(s) - (D_H(s, s'))^2 \tag{5.1}$$

$$S_{ih}(s_1, s_2, s'_1, s'_2) = len(s_1 s_2) - (D_H(s_1 s_2, s'_1 s'_2))^2 \tag{5.2}$$

where $len(s)$ is length of a sequence s; D_H is the Hamming distance between two sequences s and s' denoting the predicted and desired secondary structures in a dot bracket notation (as described in Section 2.3.1). Furthermore, $s_1 s_2$ denotes a dimer structure of two molecules.

Thus, each function penalises individuals which are further away from the desired structure; its score decreases quadratically with an increasing number of mismatches. The algorithm aims to maximise the value of these functions.

Energy gain function measures the net free energy gain obtained from dimerisation of two RNA strands at a cost of breaking the secondary structure of the individual strands. It is given by:

$$S_{eg}(s_1, s_2) = \Delta G(s_1 s_2) - \Delta G(s_1) - \Delta G(s_2) \tag{5.3}$$

where ΔG is a minimum free energy (in kcal/mol). For all specific interactions (as shown in recording and readout processes) the energy gain is contributing positively to the total score function (denoted S_{eg+}), but it penalises non-specific interactions (denoted S_{eg-}). The detailed list of all evaluator functions is provided in Appendix B.1.

The absolute fitness of an individual is then evaluated by a weighted sum of all the individual scores in the following manner:

$$Fitness = \Sigma(S_{hlf} \times w_{hlf}) + \Sigma(S_{ih} \times w_{ih}) + \Sigma(\pm S_{eg\pm} \times w_{eg}) \tag{5.4}$$

where weight values w_{hlf}, w_{ih}, w_{eg} are set to $3.0, 2.0$, and 1.0 respectively.

5.6.2 Variator

The variator combines existing solutions (from the parental population) into others, possibly unexplored solutions (the offspring population). We defined 3 genetic operators which are applied to individuals with a certain probability and independently of one another. These are:

- **single-gene mutation:** a gene is picked at random[2] and assigned a new random nucleotide sequence – similar to reinitialising the domain ($prob = 0.01$).
- **single-nucleotide mutation:** similar to above, but rather than mutating the entire gene a nucleotide at random position is mutated into another type of nucleotide ($prob = 0.14$).
- **crossover:** is a standard one-point crossover in which a crossover point is set to a random nucleotide position at the random domain. All nucleotides beyond that point are swapped between the two parents ($prob = 0.8$).

5.6.3 Terminator, Selector and Replacer

The main loop is guarded by the terminator which stops the genetic algorithm when a total of 10^5 individuals have been evaluated. The population size is set to 1000 individuals, and thus the evolution is complete after 100 generations.

[2] In our case "picked at random" imply sampling with a discrete uniform distribution (i.e. each outcome is equally likely to happen).

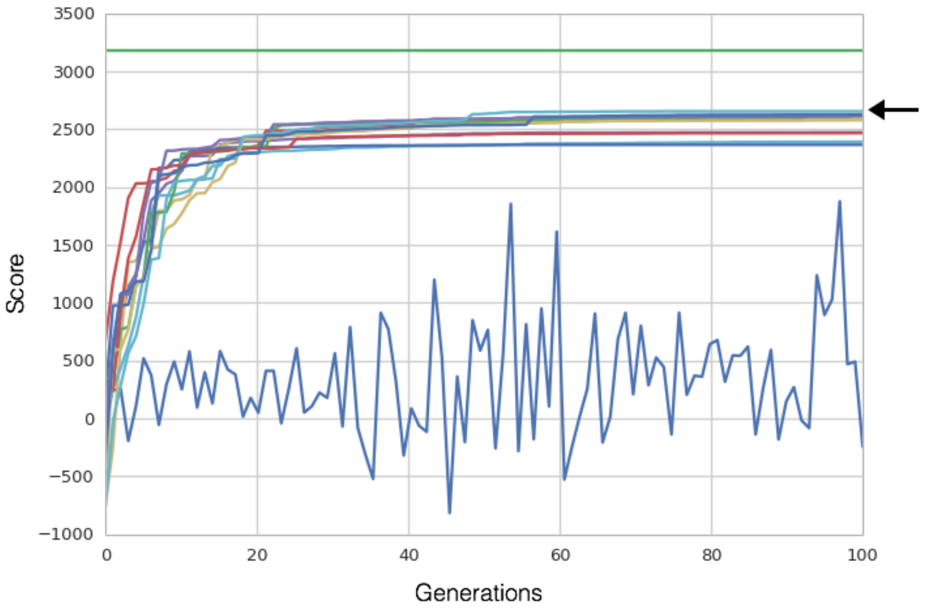


FIGURE 5.6: Trajectories of the best individual in each of the 11 populations. Also, the theoretical maximum fitness (green) and random search (blue) is shown.

domain	sequence	domain	sequence
a	$\underline{\text{TCTCCC}}$	h_y	GCACGCTCGAGCTCGTATCGCAGTA
b	GCCA	k_x	CTCTAATCAC
c	GCACA$\underline{\text{C}}$A$\underline{\text{CTTC}}$	k_y	CATCCCTATA
d	ACAC$\underline{\text{C}}$A$\underline{\text{CTTC}}$	l_x	AGACAAAAAA
e	$\underline{\text{GGGAGA}}$CCAA	l_y	ATTTTTTTCC
f	CGGCGG	m	TATGACTGCAA
g	CTGCC	x	AGACCGCTAAA
h_x	ATTAGTAGGT	y	ATACTGCTTTA

TABLE 5.4: Sequence specification of domains in the design (i.e. the "winning genotype"). Sequences are indicated in 5'→3' direction. Underlined nucleotides were set as constants (imposed by the promoter choice).

The selector is a default tournament selector provided by the *inspyred* framework. It pulls 2 individuals from the population using random sampling without replacement and selects the best one. This procedure is repeated until 1000 parents are selected for reproduction.

In the last step, the replacer discards the worst 2% of the offspring population and retains the top 2% of parents population as survivors (i.e. elite individuals).

5.7 The Proposed Solution

Table 5.4 lists the domain sequences of the highest-scoring genotype found by the genetic algorithm. The trajectories of the highest-scoring individuals in each of 11 independent populations is shown on Figure 5.6. The best individual (marked with an arrow) was chosen from a set including 10^5 contenders. The theoretical maximum was calculated by assigning maximum score value to all individual evaluator functions (i.e. perfect structures, maximum energy gains). In reality this optimal solution is not reachable (i.e. suboptimal energy gains are required to achieve desired structures).

5.8 Summary

In this chapter, we presented the design phase of the DNA-based stack data structure. The primary sequence of DNA strands was obtained using a genetic algorithm. One of the prohibitive factors in our approach is repeated fitness function evaluation; finding the optimal solution to this multi-objective optimisation problem requires computationally expensive evaluation. To study the behavior of the DNA stack one may want to use, for example, molecular dynamics simulation which may require several hours to several days to complete. We resorted to this type of simulation as a validation of the final design (see Section 6.3), while the genetic algorithm is based on fitness evaluation that is computationally more efficient, namely, dynamic programming for secondary structure prediction. Our algorithm optimised the DNA stack design for the individual structures, dimer structures as well as energies of desired and undesired reactions. The computational modeling of the stack and experimental verification are the focus of the following chapter.

As our design is based on ssDNA bricks, our entire data structure could – in principle – be expressed *in vivo* by a living cell as an RNA data structure and post-transcriptionally controlled. As we store data in a double-stranded fashion rather than in dangling single strands, an *in vivo* realization is likely to suffer less from enzymatic attack.

Chapter 6

Computational Modelling and Experimental Results

> In this chapter, we present results of the operations performed by the DNA stack *in silico* and *in vitro*. We model the stack computationally using secondary structure prediction software as well as molecular dynamics and Monte Carlo methods. We build the stack and test it in the lab using both standard molecular biology procedures (gel electrophoresis) and a sensitive DNA quantification method (on-chip electrophoresis). Finally, we visualise the structure using Transmission Electron Microscope (TEM).

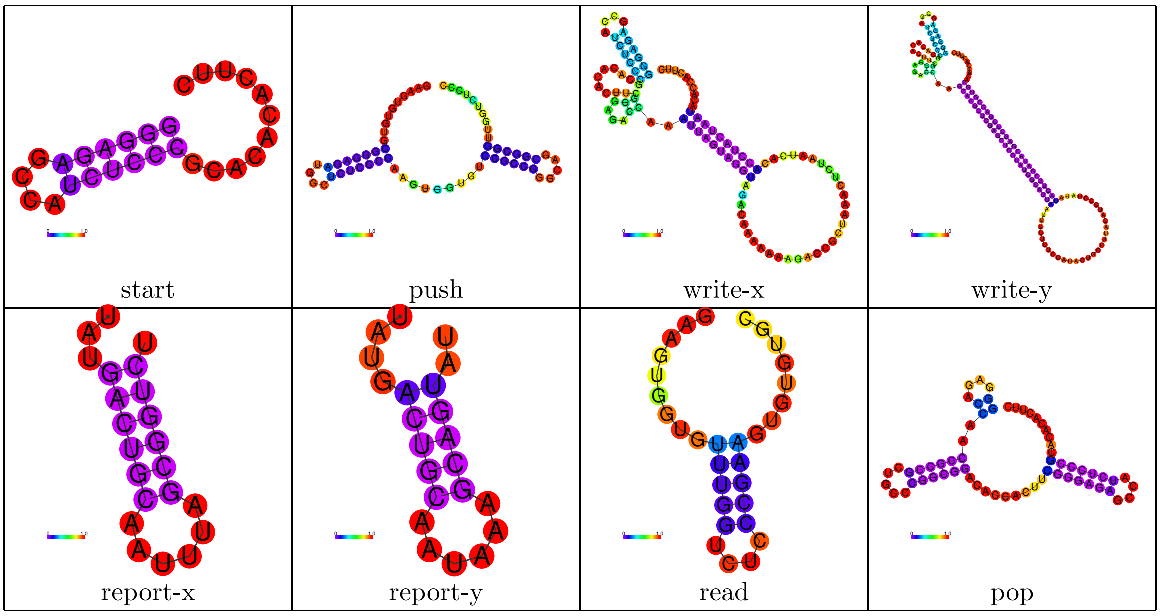
6.1 Introduction

This chapter analyses the DNA-based stack using ViennaRNA package for secondary structure prediction; the predictions were made using RNA sequences and parameterised accordingly; the correct folding was also verified for DNA parameters. In the section that follows, we show the oxDNA simulations of the DNA stack in room temperature. Brief descriptions of ViennaRNA and oxDNA model were provided in

Chapter 2. Finally, we follow up with the experimental results: electrophoresis of DNA stack in different states and microscopy of the assembled stack decorated with report bricks. The experiments were performed in collaboration with Annunziata Lopiccolo.

6.2 Secondary Structure Prediction

Here, we analysed the individual RNA bricks that are used to assemble molecular data structure. As shown on Figure 6.1 start, push, both writes, and pop bricks all form desirable secondary structures. However, both reports form undesired secondary structures. Similarly, read brick form a minor stem-loop but both 5'- and 3'-end are dangling and thus should not impede the reading operation.



start

push

write-x

write-y

report-x

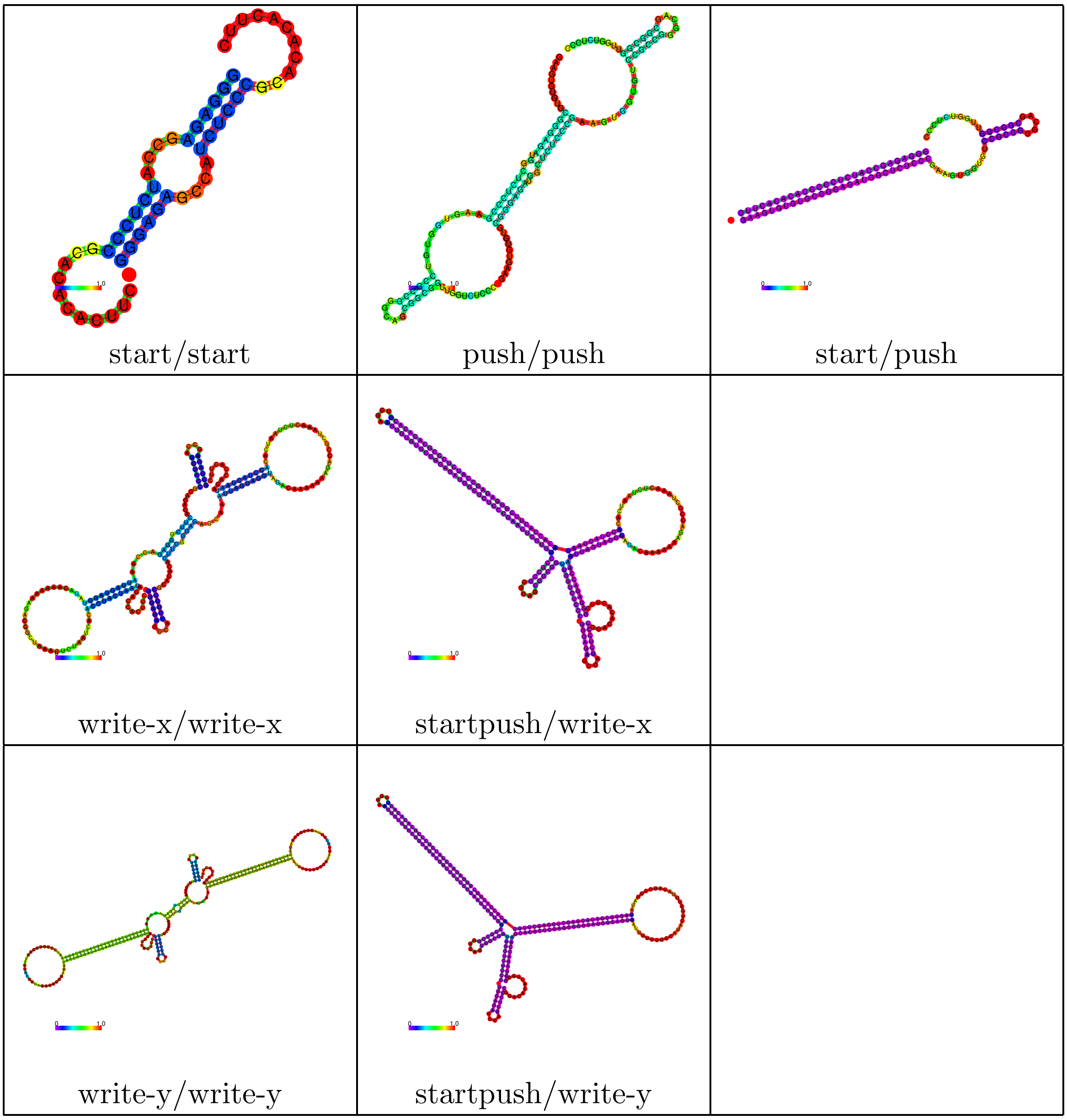
report-y

read

pop

Figure 6.1: RNAfold secondary structure prediction for individual strands

Next, we investigated RNA bricks involved in recording (Figure 6.2) and reading (Figure 6.3). The software predicts that the bricks form stable dimer structures (i.e. start-start, push-push, etc.), however, all these dimers are a result of self-complementarity. There is no free energy gain when comparing secondary structures of two single bricks and one dimer complex, meaning that dimer formation process is not spontaneous. Moreover, the prediction indicates the correct formation of the start-push complex as well as binding of two write bricks. Note that, for the last two predictions we introduced a special 'startpush' brick since the software cannot make predictions for more than two strands.



start/start

push/push

start/push

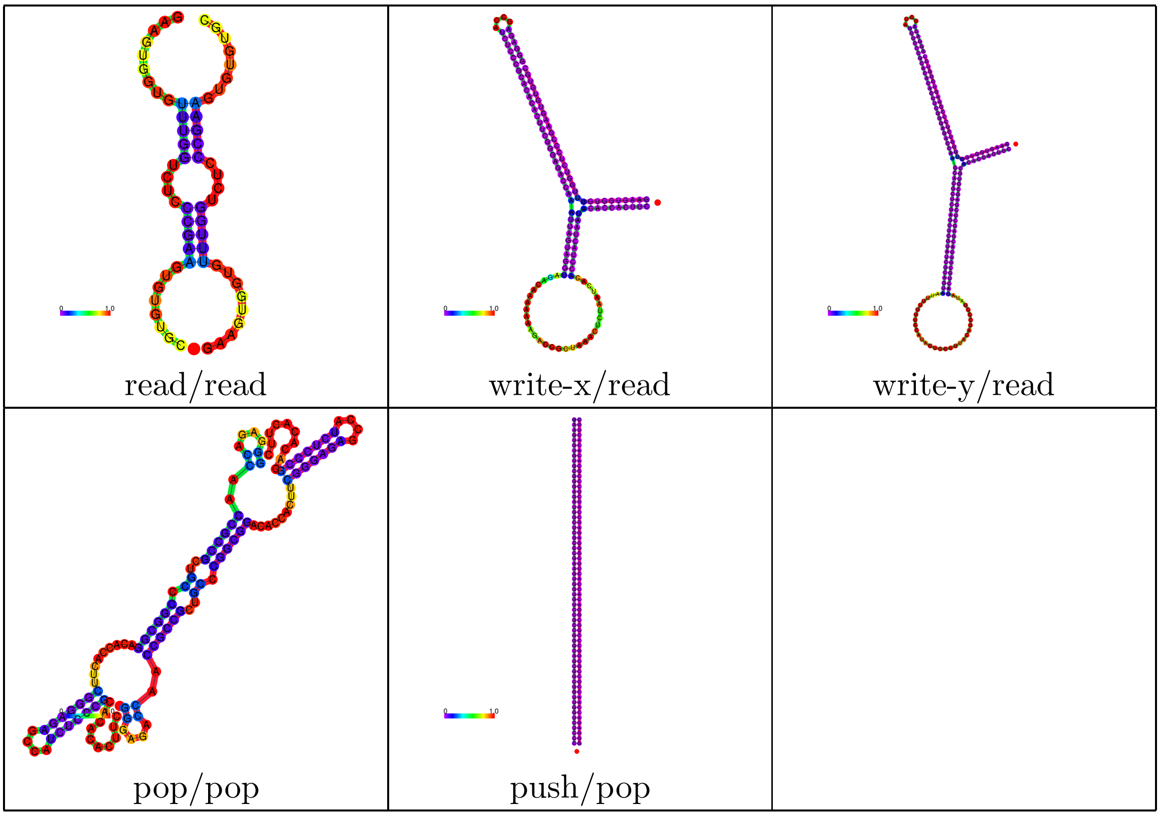
write-x/write-x

startpush/write-x

write-y/write-y

startpush/write-y

FIGURE 6.2: RNAcofold secondary structure prediction of strands involved in recording process.



read/read

write-x/read

write-y/read

pop/pop

push/pop

Figure 6.3: RNAcofold secondary structure prediction of strands involved in reading process.

6.3 Coarse-grained DNA Simulations

Here, we simulate all individual bricks and focus on the assembly of the growing DNA stack; the read-out process is not simulated. Two types of simulations were run: Monte Carlo (running on CPU) and molecular dynamics (running on CUDA/GPU). The simulations were parameterised to run at constant room temperature (23 °C for CPU simulation, 295 K for CUDA/GPU simulation). For details of the simulation setup refer to Appendix B.2.

6.3.1 Individual Bricks

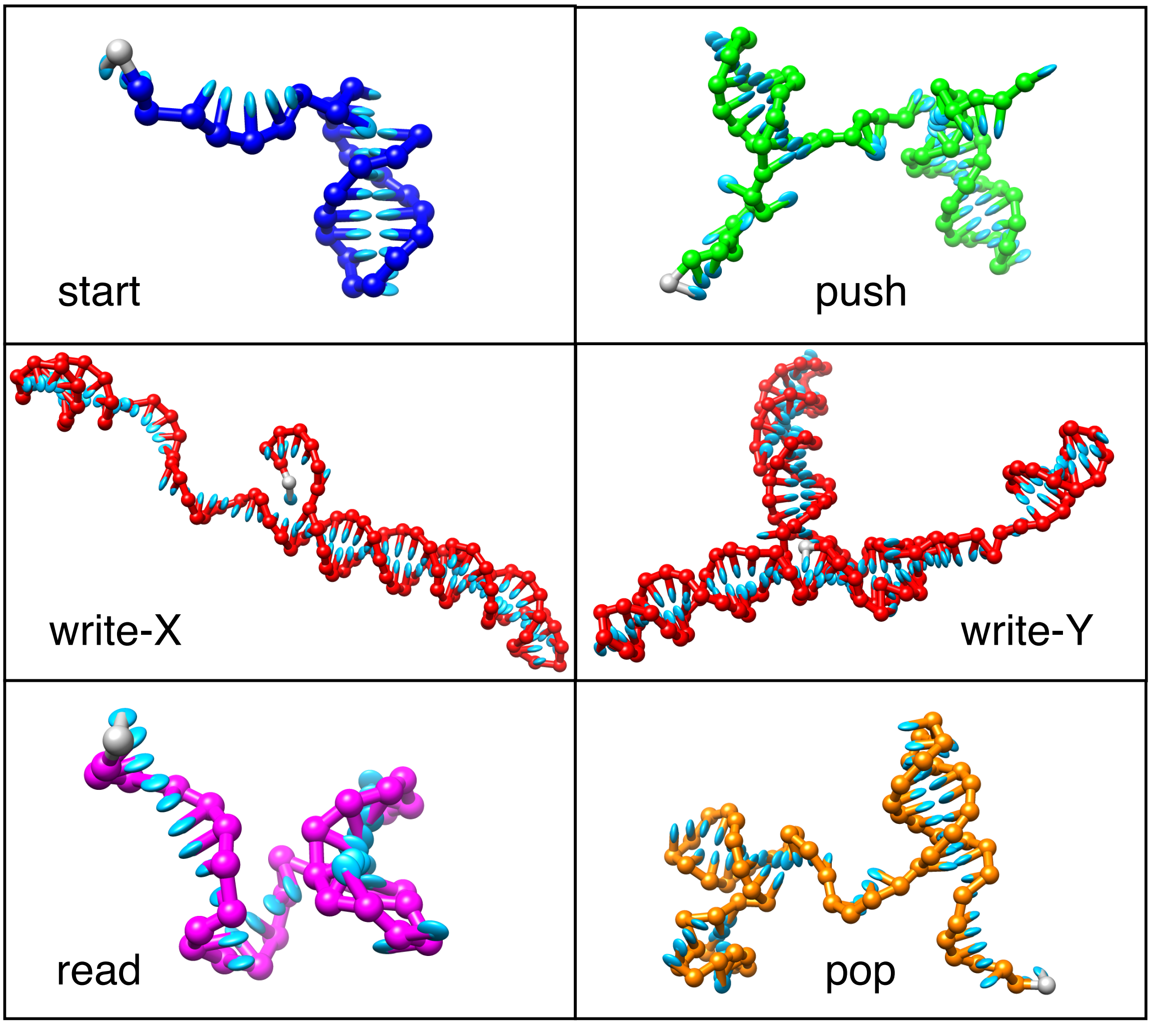


FIGURE 6.4: Individual bricks at the equilibrium state. The 5'-end of each brick is marked in grey.

All brick from the DNA stack design were simulated as separated systems until they reach the equilibrium state (i.e. no further change in the configuration is observed). The individual structures (shown on Figure 6.4) are close to the predictions obtained with ViennaRNA package, with the exception of the write$_Y$ brick. For that single brick, we failed to observe a long hairpin stem. Instead, the simulation indicates a formation of two shorter hairpin stems. The tertiary structure is non-trivial and somewhat difficult to analyse: here, write$_Y$ resembles a three-way junction with an additional pseudoknot at 5'-end. Whether the write$_Y$ structure as predicted by oxDNA is also realised in reality remains a matter of further investigation.

6.3.2 Start-Push Complex

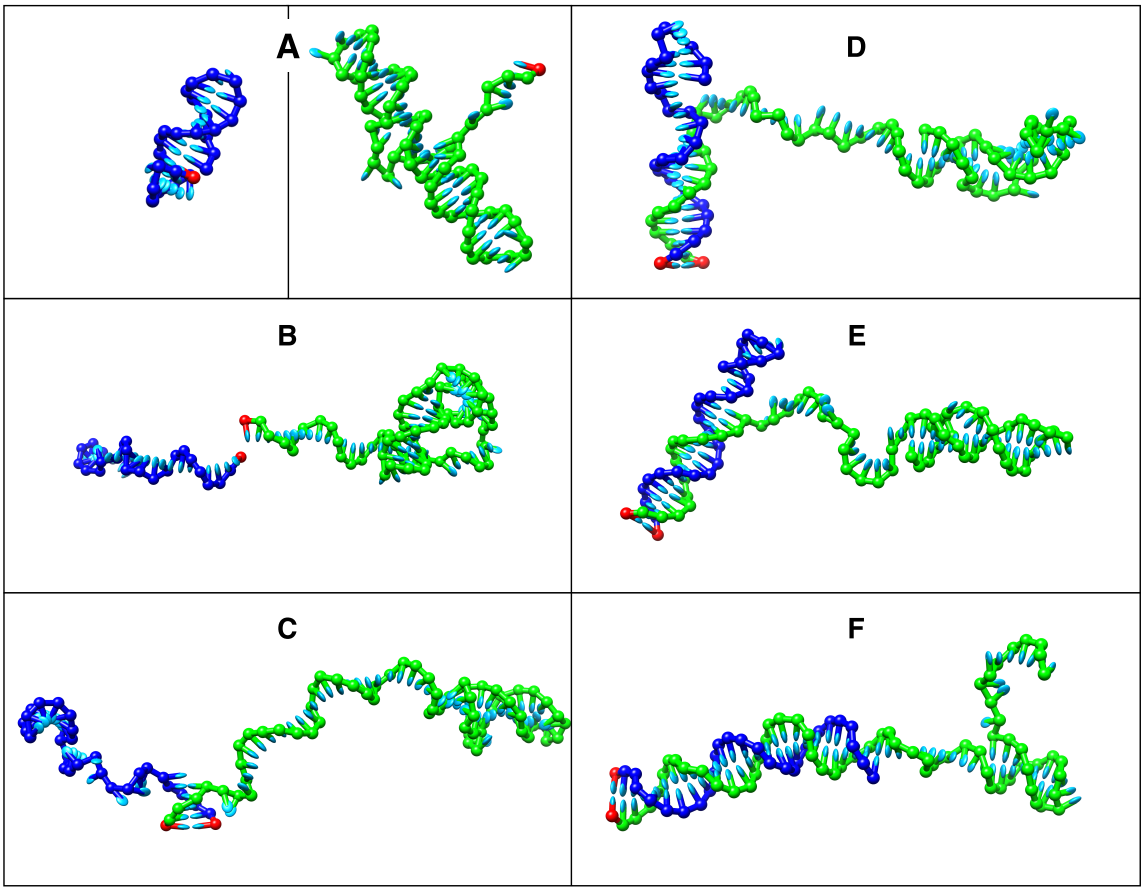


Figure 6.5: Formation of the start (blue) – push (green) complex (SP). The intermediate structures are shown in snapshots (A–F at 0, 3, 18, 24, 27, and 38 μs).

Here, we explore the interaction between the first two bricks of the DNA stack (Figure 6.5). First, two separated systems containing start and push bricks are joined into a single system (**A**). In order to avoid time-consuming simulation of the Brownian motion (which are not interesting in the given context), a mutual trap is added. A mutual trap will exert a constant force on two particles, thus pulling them together. Here the mutual trap is applied to the 3'-end of the start and 5'-end of the push brick (marked in red). As a consequence, the simulation is focused on the interaction between two bricks when they are in close proximity to each other (**B**). The hybridisation of the c and c' domains is initiated and the first five base pairs are formed (**C**). The c-to-c' duplex is completed (**D**); at this stage, the start brick hairpin is partially opened (**E**). This allows the formation of the final start-push complex (**F**). Throughout this process, the push brick hairpin (designed to bind on of the write bricks) remains in a stable configuration.

6.3.3 Longer Complex

Following the procedure described in the previous section, a formation of the longer SPXPX complex was analysed. First, a system with a stable SP complex was joined with a system containing the write$_X$ brick. The joined system was simulated until the hybridisation completed and equilibrium state was reached. In a similar way, additional push and write$_X$ were added. Interestingly, despite the correct binding, the final stack structure (Figure 6.6) varies from the predicted conformation (see Figure 5.5). Moreover, one can observe that the hybridisation of the write$_X$ and the following push brick did not run to completion (on the time scale of this simulation), Instead, two small hairpins were formed where a linear duplex was expected (marked with arrows). In consequence, the stack might not adopt a linear structure (as in Figure 5.5) but perhaps assemble into a zig-zag pattern.

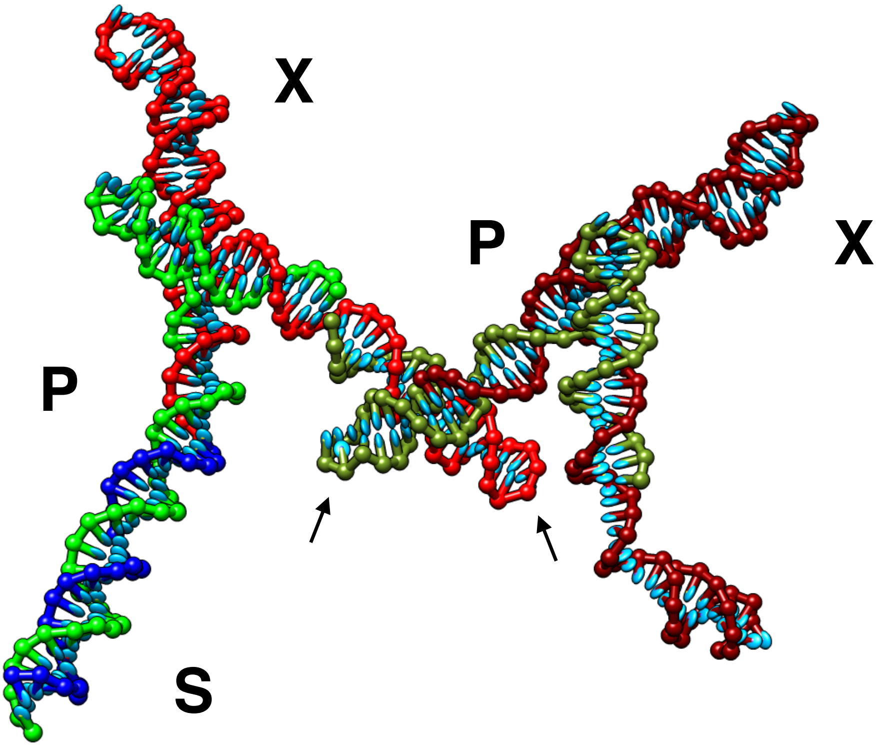


Figure 6.6: The SPXPX complex simulation. The individual bricks are added in order: start (blue), push (green), write$_X$ (red), push (dark green), and write$_X$ (dark red). The arrows indicate two small hairpins between the write$_X$ and push bricks.

6.3.4 Report Binding

During the design of the DNA stack, one aspect was particularly problematic. Namely, the binding of the report brick to the hairpin loop of the write brick. This issue is due to the fact that ViennaRNA lacks the ability to predict kissing stem-loops. The report brick is not a stem-loop *per se*, but its interaction with the write brick can be described as a kissing complex.

Here, a reporter binding is simulated with oxDNA. A reporter brick is joined with a system containing assembled SPX complex. Similarly to previous simulations, a mutual trap is set between the report brick and the corresponding domain of the write brick (Figure 6.7). Despite simulating the system for a relatively long time (i.e. 606 μs) the report brick failed to bind to the stack. Throughout the simulation, the hairpin loop of the write brick remains in a stable, closed conformation. This prevents the report brick to initiate hybridisation.

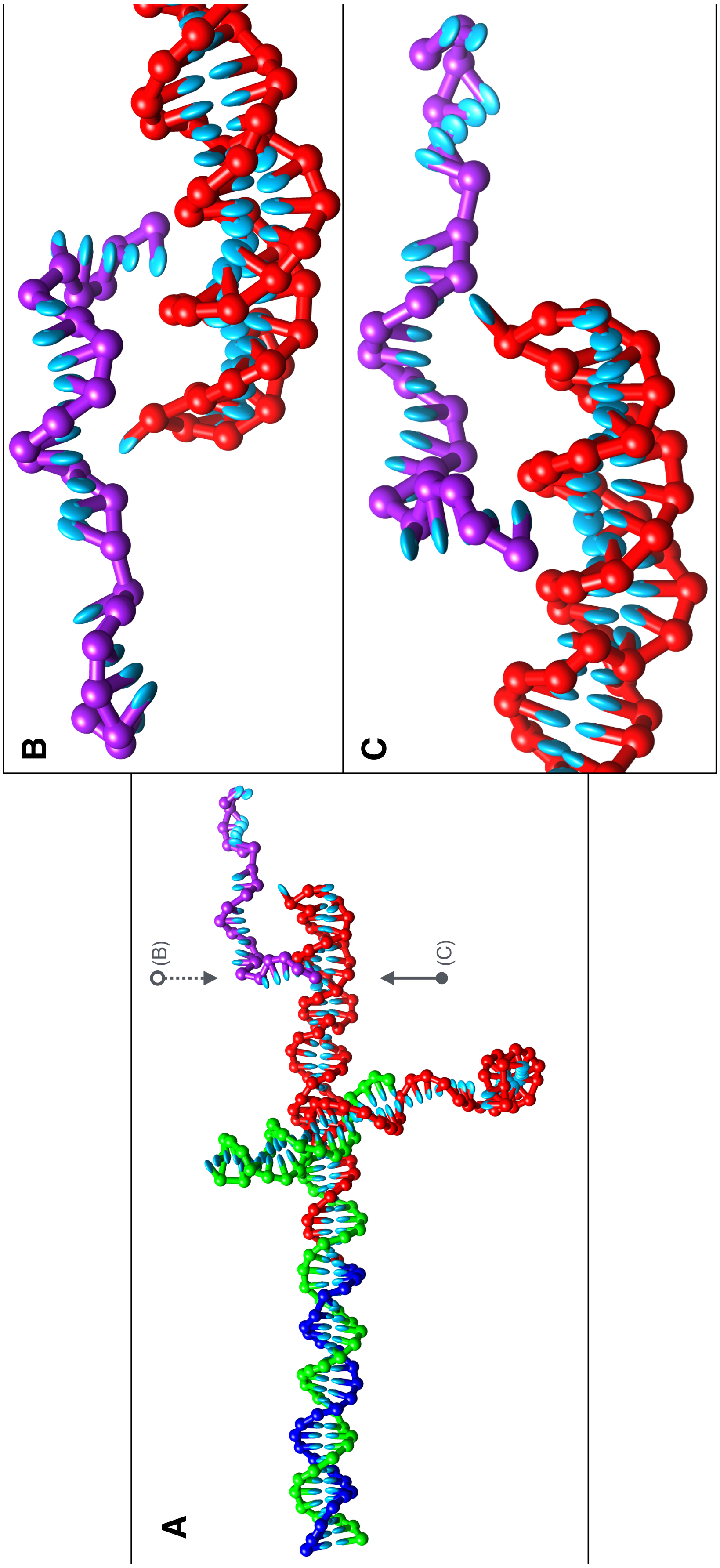


Figure 6.7: Attempt at binding the report (purple) to the hairpin loop of the write brick (red). Report strand is kept in close proximity to its target but the hairpin loop remain inaccessible.

6.4 Experimental Verification

This section presents experiments conducted with on-chip electrophoresis aimed at the characterization of the ssDNA bricks that compose the stack, as well as the validation of data recording and reading cycles. For additional confirmation of the data storage, the assembled nanodevice was imaged using transmission electron microscopy (TEM).

6.4.1 Bioanalyzer Results

We performed capillary electrophoresis measurements of all individual bricks in order to determine the response of the Agilent 2100 Bioanalyzer High Sensitivity DNA Assay for our non-standard DNAs. All bricks were provided in 200 nM concentration. Electropherograms always detected a single clear peak per brick. Table 6.1 summarises for each brick its known size, the measured migration time and fluorescence area under the peak, as well as the calculated size and molarity derived by the instrument software from comparison to the reference ladder. Averages and standard deviations have been calculated from at least three independent measurements.

		measured		**derived**	
brick	**size[nt]**	**time [s]**	**area [FU]**	**size [bp]**	**molarity [nM]]**
start (S)	27	45.22±0.92	94.6±61.23	51±7.6	34.80±15.92
push (P)	64	46.81±0.76	74.4±39.2	64±6.9	8.08±0.174
write-X (X)	98	53.27±0.34	55.93±39.65	128±3.78	5.961±0.473
write-Y (Y)	128	55.35±0.06	5.27±1.15	147±0.8	0.845±0.221
report-X (Rx)	22	44.81±0.81	248.5±60.57	47±6.4	78.25±16.81
report-Y (Ry)	22	45.18±1.02	241.3±84.49	47±11.3	86.44±12.77
read (R)	31	44.61±0.35	73.85±15.76	46±2.82	31.67±1.21
pop (Q)	64	47.89±0.28	28.13±25.4	74±3.4	6.602±6.78

Table 6.1: Calibration results (given as averages and standard deviation) for all individual strands provided in 200 nM concentrations.

The measurements successfully discriminate the migration times of almost all strands (disregarding *report* strands) with significant differences. Only *start* and *read* cannot be reliably differentiated.

Striking discrepancies between the known brick sizes and the sizes derived by the software from comparison to the ladder might be attributed to two reasons: firstly, short oligomers such as *start*, *read* and *report* are well below the detection limit of the high sensitivity kit, which can resolve dsDNA fragments between 50 – 7000 base pairs in length. Secondly, the reported deviations might lie in the fact that our bricks contain extensive secondary structures that might affect their motility in the gel matrix.

A similar discrepancy is observed in the derived molarity values. This is partly due to the fact that molarity calculation is based on the base pair estimation and will thus suffer from the issues described before, partly because our bricks contain extensive ssDNA regions which interact differently with the fluorescent dye than dsDNA.

Recording Process

To probe the performance of the data recording (push) cycle, we performed experiments in which we sequentially recorded five data tokens (X, X, X, Y, X) onto the growing stack. We ran five parallel experiments and stopped them at different steps in the protocol. Gel-like images of the Bioanalyzer output are shown in Fig. 6.8.

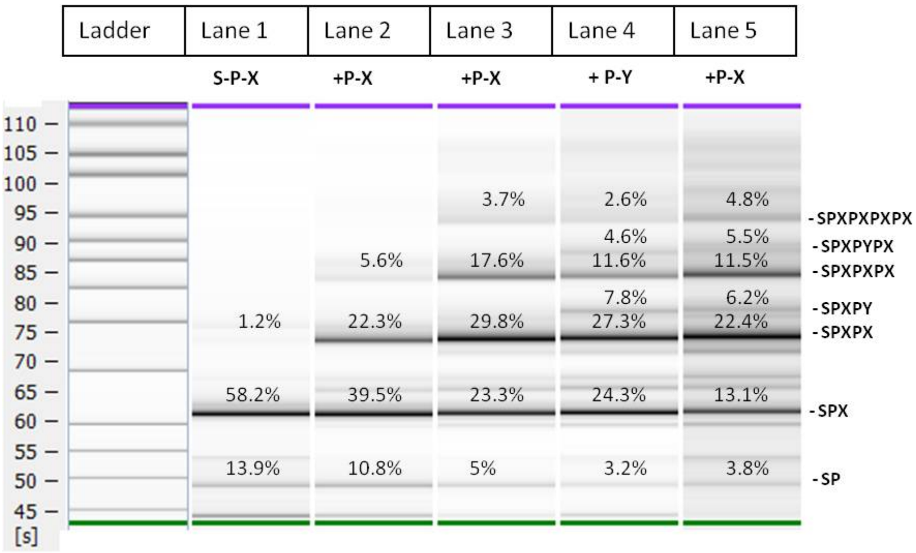


Figure 6.8: Capillary electrophoresis of the recording process. Lane 1=SPX; Lane2=SPXPX; Lane 3=SPXPXPX; Lane 4=SPXPXPXPY; Lane 5=SPXPXPX-PYPX. Data obtained from five parallel experiments.

For the first three recorded data tokens, the addition of each *write-X* brick is accompanied by the appearance of a new clear peak in the spectrum: after addition of the first *write-X* brick, this peak (*start-push-write-X* complex, or SPX) accounts for more than 58% of the total fluorescence. Lane 2 shows the appearance of a second peak (SPXPX) that corresponds to the two data tokens. However, this second peak accounts for only about 22% of the total fluorescence, whereas almost 40% still correspond to the single data token recorded (SPX). The situation repeats in the third lane, where the correct complex (SPXPXPX) accounts for slightly more than 17% of the fluorescence, the second peak (SPXPX) for about 30% and the first peak still for about 23%.

The addition of *write-Y* in lane 4 leads to the appearance of several new peaks, which we identify as SPY, SPXPY, and SPXPXPY. A very faint peak at about 98 s migration time might correspond to the desired SPXPXPPXPY, but the signal is too weak to be properly identified by the analysis software. Lane 5 essentially shows the same peaks as lane 4, with peak sizes changing as expected: peaks from complexes ending in a *write-Y* brick become smaller, whereas the corresponding complexes with added *write-X* become proportionally larger.

In all lanes, faint higher peaks indicate that there is a very small potential for run-away processes to create complexes with more data tokens than the provided ones. Yet, in all cases, the fluorescence of all these longer bands combined does not exceed 10% of the total.

Read Out Process

Lanes 1 through 3 reconfirm the working of the recording cycle with the same observations than for the experiment of the last section: each added *write* brick generates a new peak in the spectrum with very little evidence for run-away processes and persistence of peaks that indicate intermediate complexes.

Lane 4 shows the response of the device after provision of 200 nM *read* and *pop*, which is supposed to trigger one readout cycle: newly created *push-pop* as well as *read-write* complexes result in the appearance of three new peaks at around 47.42 (QP), 52.22 (RX), and 57.39 (RY) seconds. The *push-pop* complexes account for 38% of the fluorescence, whereas *start-write-X* and *start-write-Y* account for 2.8 and 12% respectively. Peaks associated with the different stack states SPXPYPY, SPYPY, SPXPY, and SPY decrease accordingly. The situation repeats in Lane 5 where the

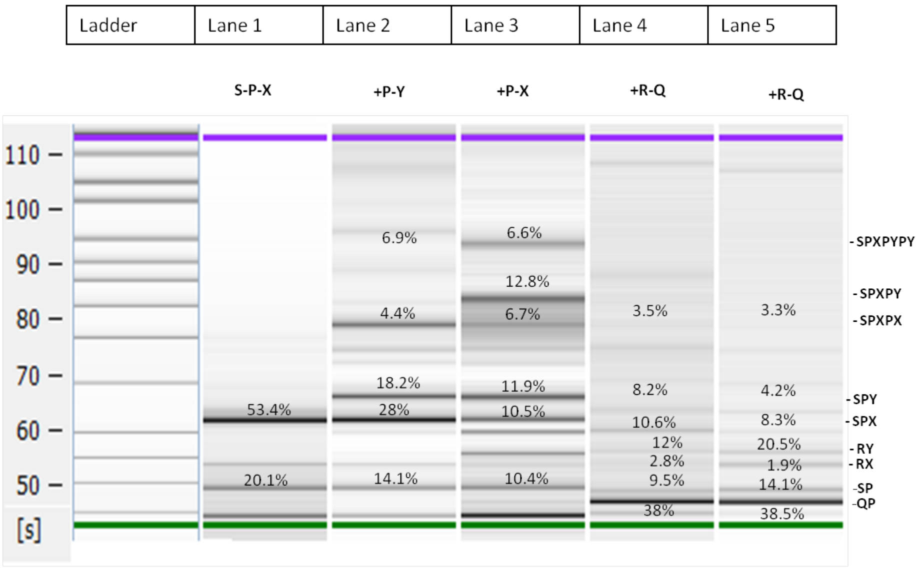


Figure 6.9: Capillary electrophoresis of the recording and reading of three data tokens. Recording: Lane 1=SPX; Lane2=SPXPY; Lane 3=SPXPYPY. Reading: Lane 4=SPXPYPY+RQ; Lane 5=SPXPYPY+RQRQ.

second readout cycle further increases *push-pop* and *read-write* peaks and simultaneously reduces intensities of the corresponding stack complexes. Noteworthily, after reading out the two recorded data tokens, 14.1% of the fluorescence results from the *start-push* complex whereas peaks of stacks that still contain recorded information only register with 8, 4.2, 4.8 and 3.3%.

To sumarise, the measurements successfully discriminate the migration times of almost all individual strands with significant differences. However, because we employ non-standard DNA strands, the electrophoresis analysis software does not correctly detect molecular concentrations, which prevents us to gain a precise quantitative picture of the involved processes. That problem is caused by the molarity calculation which is based on the base pair estimation and will thus suffer from the fact that the assembled stack contains extensive ssDNA regions which interact differently with the fluorescent dye than dsDNA (for which the analysis kit has been designed). Nonetheless, capillary electrophoresis indicates that the nanodevice is able to store at least three consecutive data tokens and does not suffer from problematic runaway processes.

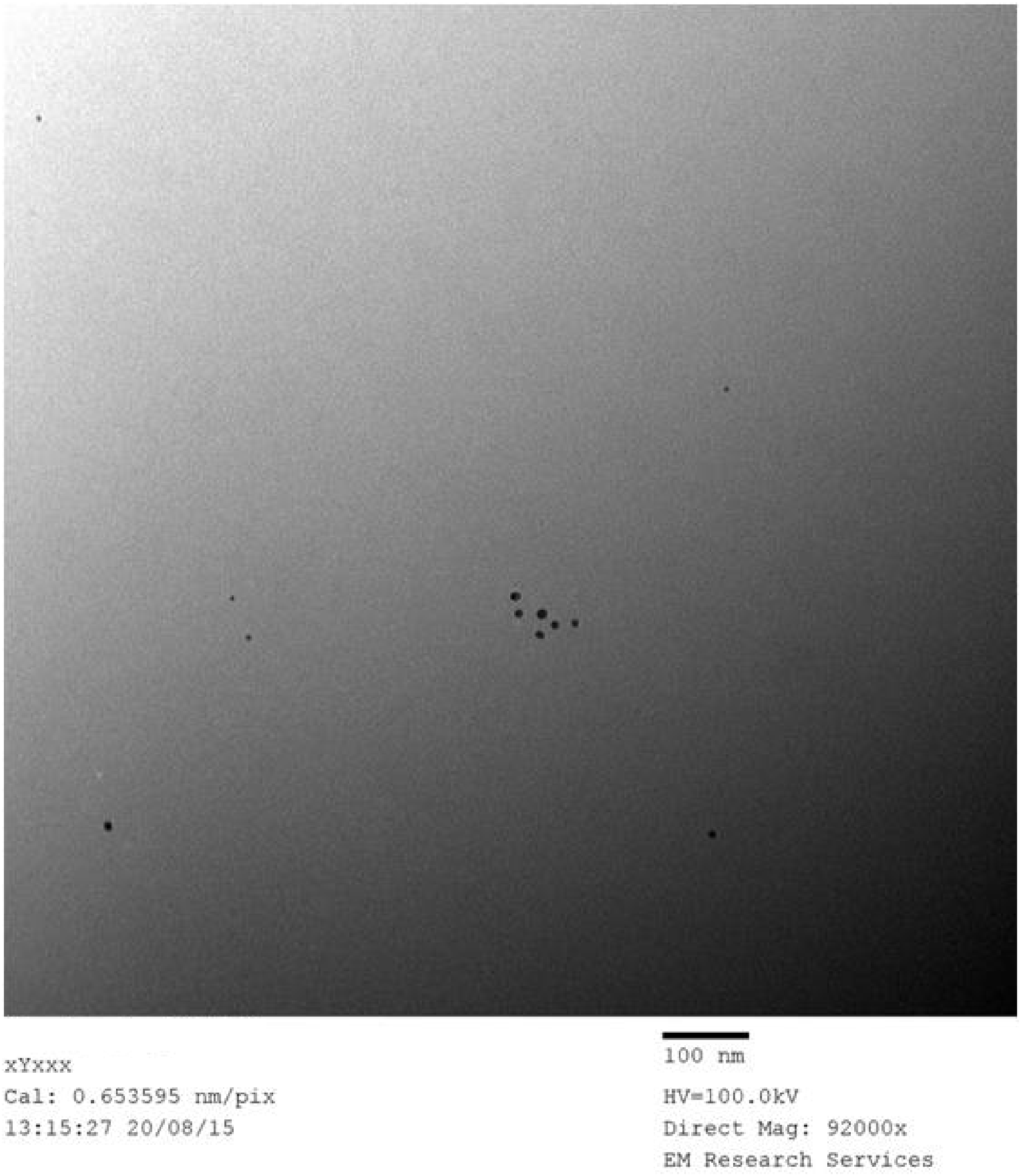


Figure 6.10: Representative TEM image of the DNA stack. The encoded information consists of XXYXXX (gold nanoparticles: 5 nm for X, 10 nm for Y).

6.4.2 TEM Imaging

For additional confirmation of the recording, we imaged the assembled nanodevice using TEM. For this purpose, assembled stacks were mixed with *report* strands that, in turn, are decorated with 5 and 10 nm gold nanoparticles. *Report* bricks associate with their respective *write* bricks at any position in the assembled stack. Nanoparticles appear in TEM images as black dots that can be easily distinguished and classified.

Figure 6.10 shows TEM results from an experiment where five data tokens (X, Y, X, X, X) have been recorded. The image show a stack with just one extra *write-X* on the left side of the recorder, resulting in a stack with six data tokens (X, X, Y, X, X, X). The image shows a separation of 15-20 nm between the nanoparticles with a zig-zag configuration predicted by the simulations (see Section 6.3.3).

6.5 Summary

In this chapter we confirmed the correct self-assembly of the DNA stack through secondary structure prediction and coarse-grained DNA model simulations. Through oxDNA simulations, we showed that upon initial collision, DNA bricks form desired structures. However, it is worth noting that the simulated structure slightly diverges from designed one in the arrangement of the backbone (zig-zag pattern rahter than rod-like). Moreover, using simulations, we identified a potential problem with the report binding (i.e. write brick hairpin loop is closed and stable).

The capillary electrophoresis indicates that the nanodevice is able to store at least three consecutive data tokens and does not suffer from problematic runaway processes. After recording several data tokens, electrophoresis analysis indicates that the device is not only present in the desired final state, but also in several intermediate recording states. Because of the limits of experimental quantification, we can currently not offer a satisfying explanation for these intermediate peaks. This currently impacts the readout cycle, as the pop operation interacts with all present stacks and thus returns a superposition of recorded data. While this is contrary to the intended working, we point out that such a superposition might also have advantages, as it might allow one to reverse engineer the composition *and order* of recorded information from a single electrophoresis read out.

TEM experiments shown an example of correctly arranged gold nanoparticles, however the observable yield is low. This may point to a bottleneck involving the binding of reporter strands to the assembled chain (as explained in Section 6.3.4).

Chapter 7

Conclusions and Discussion

The construction of DNA-based nanostructures has opened the door to a new realm in which we are able not only to construct but also to program synthetic nanodevices. This section contains the summary of what was achieved in this thesis and the main limitations; also a number of extensions to the studied systems of a somewhat less complete nature, but which present some exciting avenues down which this work is being pursued.

DNA Origami with Synthetic Scaffolds

Although the initial application of the scaffolded DNA origami folding was very successful, the approach does not scale to larger, more complex structures; it is also infeasible in certain biological systems, such as *E. coli* cells. The main limiting factors have been the reliance on the viral sequences and lack of attention to the biological interface and its side-effects.

In this thesis, we have proposed the use of a synthetic scaffold in the DNA origami to eliminate issues with the undesirable binding of staples; and careful optimisation of the synthetic sequence to minimise the side-effects caused by the biological interface (Chapter 3). We introduced an algorithm to design the DNA origami based on the combinatorial properties of the De Bruijn sequence (Chapter 3). We found that, contrary to what is now widely accepted in the scientific community, DNA origami are not uniquely addressable (Chapter 4). The repetitions in the natural scaffolds have an adverse impact on the staple specificity and cannot be neglected in long origami scaffolds (such as 7.2 kb-long M13mp18 and 49 kb-long λ-phage genomes), therefore the natural scaffolds are not scalable. On the other hand, the synthetic DBS scaffolds are uniquely addressable (on the sequence level) and bio-orthogonal by design (hypothesis H_1). Also, we have found that the sequence uniqueness improves the thermodynamic addressability of synthetic scaffolds (Chapter 4). In addition, we verified that they fold into DNA origami and RNA-DNA hybrid origami without alteration to the folding protocol (Chapter 4). Moreover, this new approach grants strict control over the interface between the DNA origami devices and various biomolecules through the insertion of biological sites in the otherwise bio-orthogonal scaffold. Finally, our approach provides a broad design space which allows tailoring the nanostructures for particular applications.

Limitations and Future Work

The computational tools for simulating the actual folding process of large DNA origami, based purely on scaffold and staple sequences, are currently lacking. Tools of that kind would be of great value while designing complex shapes before the cost and time-intensive sequence synthesis; they would also allow us to understand the folding process and help us define design better rules for the optimal folding. Also, it might be possible to track the folding process using advanced, high-speed AFM, which would provide the ultimate insight into the folding process.

The protocols for single-stranded DNA manipulation are not as advanced as those for double-stranded DNA. Despite numerous attempts, we were unable to establish a robust laboratory protocol for efficient production of DNA scaffolds. However, one possible workaround is to use transcribed RNA sequence as a scaffold. Our experimental results bear promise that this avenue is viable, especially suited for future *in vivo* applications. However, another issue that requires closer attention is efficient staple production *in vivo*; there are no ready solutions one can apply to achieve that yet. One interesting avenue to explore would be to include the staples in a high copy number plasmid and "cut" them out *in vivo* (using, for instance, CRISPR system [165, 166]).

The future efforts should concentrate on three goals: (1) the experimental confirmation of the superior folding for long scaffolds; (2) the investigation of potential strategies for scaffold and staple production *in vivo*; and (3) the development of experimental protocols allowing folding and visualisation of nanostructures *in vivo.*

DNA-based data structure

We have presented a working data storage device, implementing push and pop stack operations (Chapter 5). We used a genetic algorithm to optimise the DNA sequences in this nanodevice on constraints of the biological interface (Chapter 5). Through simulations (Chapter 6), we showed that synthetic DNA strands self-assemble into a functional data structure (hypothesis H_2) and in principle can store an unlimited number of data elements. Also, the coarse-grained model simulations provided insight into some of the nuances with the experimental detection of recorded data (i.e. report binding). Nonetheless, capillary electrophoresis and TEM imaging (Chapter 6) indicated that the data storage device was able to store at least three consecutive signals and did not suffer from problematic runaway processes.

Limitations and Future Work

After the recording of several signals, the data storage device was not only present in the desired final state, but also in several intermediate recording states. Because of the limits of the experimental quantification, we cannot currently offer a satisfying explanation for these intermediate states. As the pop operation interacts with all present stacks, it returns a superposition of recorded signals. While this is contrary to the design intentions, such a superposition might also have advantages. It might allow one to reverse engineer the composition and order of recorded information from a single electrophoresis read out.

Because non-standard DNA strands were used in the stack, the electrophoresis analysis is limited to qualitative analysis – it was not possible to gain a precise quantitative picture of the involved processes. This could be perhaps mitigated by the use of more advanced (and costly) techniques such as experiments with molecular beacons [167]. Better experimental quantification should also improve the calibration of computational models, and in turn, help to understand the fidelity of the storage device.

Here, we investigated DNA stack, which is an elementary abstract data type, however, one can envision molecular implementation of more complex data structures. It is likely that some data structures would be more suitable for certain operations than others in a cellular context. On the other hand, it might be possible that some data structures from computer science cannot be reliably realised as molecular nanodevices, and thus will require us to invent novel models of computation.

The future efforts should concentrate on three goals: (1) further optimisation of the design and experimental protocols (such as washing steps) for *in vitro* data recording and reading; (2) *in vivo* trials and subsequent linking of the stack to downstream processes (for control and monitoring purposes); and (3) investigation of alternative molecular data structures realised in DNA/RNA (e.g. list, heap, queue, tree, etc.).

Possible Extension: Multiple Stacks Construct

A multiple stack construct should implement all the same operations as a single stack. In addition, the multiple stacks should meet the following requirements:

1. Push, pop and read bricks are stack-specific (i.e. operations of one stack should not affect other stacks) while data tokens are orthogonal to the stack
2. It should be possible to move data from stack A to stack B (see Figure 7.1) using a new type of the move brick
3. The move$_{A-B}$ brick may be some combination of read$_A$ $-$ push$_B$ bricks (followed by pop$_A$)

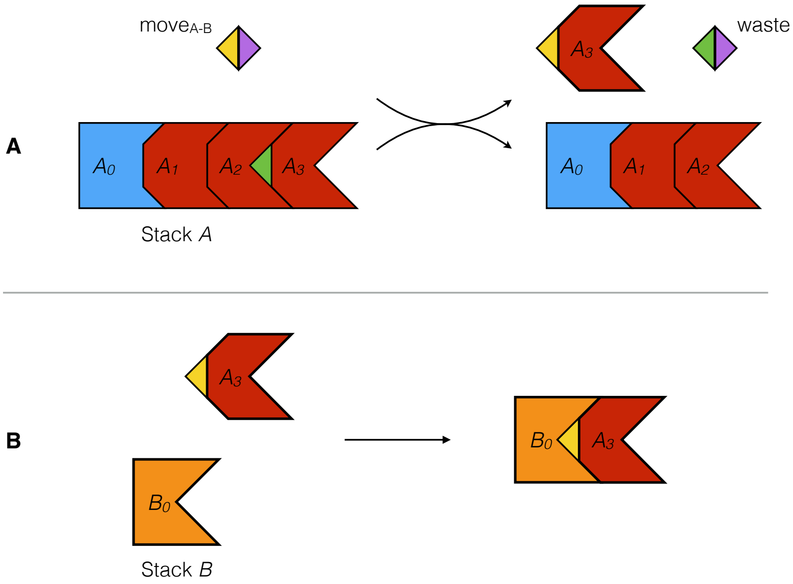


FIGURE 7.1: An abstract depiction of a multiple stack construct. Here, the data token (A_3) from one stack can be released using move operation (A). The released data token is then accepted by the target stack (B).

Note that in the actual design, some of the operation (such as write) may have to be implemented with multiple DNA strands to ensure orthogonality of operations on different stacks. Also, a design of the move bricks has to prevent the run-away process. These features would allow the implementation of (Turing-universal) stack machines as in References [24] and [158].

Bibliography

[1] Ingo Brigandt. Systems biology and the integration of mechanistic explanation and mathematical explanation. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 44(4, Part A):477 – 492, 2013. ISSN 1369-8486. doi: http://dx.doi.org/10.1016/j.shpsc.2013.06.002. URL http://www.sciencedirect.com/science/article/pii/S1369848613000903.

[2] D. Ewen Cameron, Caleb J. Bashor, and James J. Collins. A brief history of synthetic biology. *Nat Rev Micro*, 12(5):381–390, May 2014. ISSN 1740-1526. URL http://dx.doi.org/10.1038/nrmicro3239.

[3] Steven A. Benner and A. Michael Sismour. Synthetic biology. *Nat Rev Genet*, 6 (7):533–543, July 2005. ISSN 1471-0056. URL http://dx.doi.org/10.1038/nrg1637.

[4] P. W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006. URL http://www.nature.com/nature/journal/v440/n7082/abs/nature04586.html.

[5] Shawn M. Douglas, Ido Bachelet, and George M. Church. A logic-gated nanorobot for targeted transport of molecular payloads. *Science*, 335(6070): 831–834, 2012. doi: 10.1126/science.1214081. URL http://www.sciencemag.org/content/335/6070/831.abstract.

[6] D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proc. Nat. Acad. Sci. USA*, 107(12):5393–5398, 2010. doi: /10.1073/pnas.0909380107.

[7] Yaniv Amir, Eldad Ben-Ishay, Daniel Levner, Shmulik Ittah, Almogit Abu-Horowitz, and Ido Bachelet. Universal computing by DNA origami robots in a living animal. *Nat Nano*, 9(5):353–357, May 2014. ISSN 1748-3387. URL http://dx.doi.org/10.1038/nnano.2014.58.

[8] Benjamin Groves, Yuan-Jyue Chen, Chiara Zurla, Sergii Pochekailov, Jonathan L. Kirschman, Philip J. Santangelo, and Georg Seelig. Computing in mammalian cells with nucleic acid strand exchange. *Nat Nano*, advance online publication:–, December 2015. ISSN 1748-3395. URL http://dx.doi.org/10.1038/nnano.2015.278.

[9] Qian Mei, Xixi Wei, Fengyu Su, Yan Liu, Cody Youngbull, Roger Johnson, Stuart Lindsay, Hao Yan, and Deirdre Meldrum. Stability of DNA origami nanoarrays in cell lysate. *Nano Lett.*, 11(4):1477–1482, April 2011. ISSN 1530-6984. doi: 10.1021/nl1040836. URL http://dx.doi.org/10.1021/nl1040836.

[10] Ralf Jungmann, Stephan Renner, and Friedrich C Simmel. From DNA nanotechnology to synthetic biology. *HFSP Journal*, 2(2):99–109, February 2008. ISSN 1955-205X. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2645571/.

[11] Nadrian C. Seeman. From genes to machines: DNA nanomechanical devices. *Trends in Biochemical Sciences*, 30(3):119–125, 2005. ISSN 0968-0004. doi: 10.1016/j.tibs.2005.01.007. URL http://dx.doi.org/10.1016/j.tibs.2005.01.007.

[12] N. C. Seeman. DNA in a material world. *Nature*, 421(6921):427–431, 2003. ISSN 0028-0836. doi: 10.1038/nature01406. URL http://www.nature.com/nature/journal/v421/n6921/abs/nature01406.html.

[13] Ray Kurzweil. *The Singularity Is Near: When Humans Transcend Biology.* Penguin (Non-Classics), 2006. ISBN 0143037889.

[14] B. C Crandall. *Nanotechnology: molecular speculations on global abundance.* Cambridge, Mass.: MIT Press, 1996.

[15] Pengcheng Fu. Grand challenges in synthetic biology to be accomplished. *Frontiers in Bioengineering and Biotechnology*, 1(2), 2013. ISSN 2296-4185. doi: 10.3389/fbioe.2013.00002. URL http://www.frontiersin.org/synthetic_biology/10.3389/fbioe.2013.00002/full.

[16] Andrs Moya, Natalio Krasnogor, Juli Peret, and Amparo Latorre. Goethe's dream. challenges and opportunities for synthetic biology. *EMBO Reports*, 10 (Suppl 1):S28–S32, August 2009. ISSN 1469-3178. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2726005/.

[17] Thomas Trring and Kurt V Gothelf. DNA nanotechnology: a curiosity or a promising technology? *F1000Prime Reports*, 5:14–, May 2013. ISSN 2051-7599. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3643079/.

[18] Chuan Zhang and Chengde Mao. Dna nanotechnology: Bacteria as factories. *Nat Nano*, 3(12):707–708, December 2008. ISSN 1748-3387. URL http://dx.doi.org/10.1038/nnano.2008.358.

[19] Michael R Green and Joseph Sambrook. *Molecular Cloning: A Laboratory Manual*. Cold Spring Harbor Laboratory Press, 2012.

[20] Andre V. Pinheiro, Dongran Han, William M. Shih, and Hao Yan. Challenges and opportunities for structural DNA nanotechnology. *Nat Nano*, 6(12):763–772, December 2011. ISSN 1748-3387. URL http://dx.doi.org/10.1038/nnano.2011.187.

[21] Arturo Casini, Georgia Christodoulou, Paul S. Freemont, Geoff S. Baldwin, Tom Ellis, and James T. MacDonald. R2oDNA designer: Computational design of biologically neutral synthetic DNA sequences. *ACS Synthetic Biology*, 3(8): 525–528, 2014. doi: 10.1021/sb4001323. URL http://dx.doi.org/10.1021/sb4001323. PMID: 24933158.

[22] Milan N. Stojanovi and Darko Stefanovi. Deoxyribozyme-Based Half-Adder. *J. Am. Chem. Soc.*, 125(22):6673–6676, 2003. ISSN 0002-7863, 1520-5126. doi: 10.1021/ja0296632. URL http://pubs.acs.org/doi/abs/10.1021/ja0296632.

[23] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. Enzyme-Free Nucleic Acid Logic Circuits. *Science*, 314(5805):1585–1588, 2006. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1132493. URL `http://www.sciencemag.org/cgi/doi/10.1126/science.1132493`.

[24] Lulu Qian, David Soloveichik, and Erik Winfree. Efficient Turing-Universal Computation with DNA Polymers. In Yasubumi Sakakibara and Yongli Mi, editors, *DNA Computing and Molecular Programming*, number 6518 in Lect. Notes Comput. Sci., pages 123–140. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-18304-1 978-3-642-18305-8. URL `http://link.springer.com/chapter/10.1007/978-3-642-18305-8_12`.

[25] George A Khoury, James Smadbeck, Chris A Kieslich, and Christodoulos A Floudas. Protein folding and de novo protein design for biotechnological applications. *Trends in biotechnology*, 32(2):99–109, November 2013. ISSN 1879-3096. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3922204/`.

[26] Harold Fellermann, Annunziata Lopiccolo, Jerzy Kozyra, and Natalio Krasnogor. In vitro implementation of a stack data structure based on DNA strand displacement. In *Unconventional Computation and Natural Computation*, pages 87–98. Springer Nature, 2016. doi: 10.1007/978-3-319-41312-9_8. URL `https://doi.org/10.1007%2F978-3-319-41312-9_8`.

[27] Harold S. Bernhardt. The RNA world hypothesis: the worst theory of the early evolution of life (except for all the others)(a). *Biology Direct*, 7(1):1–10, 2012. ISSN 1745-6150. doi: 10.1186/1745-6150-7-23. URL `http://dx.doi.org/10.1186/1745-6150-7-23`.

[28] Clyde A. Hutchison, Ray-Yuan Chuang, Vladimir N. Noskov, Nacyra Assad-Garcia, Thomas J. Deerinck, Mark H. Ellisman, John Gill, Krishna Kannan, Bogumil J. Karas, Li Ma, James F. Pelletier, Zhi-Qing Qi, R. Alexander Richter, Elizabeth A. Strychalski, Lijie Sun, Yo Suzuki, Billyana Tsvetanova, Kim S. Wise, Hamilton O. Smith, John I. Glass, Chuck Merryman, Daniel G. Gibson, and J. Craig Venter. Design and synthesis of a minimal bacterial genome.

Science, 351(6280), 2016. ISSN 0036-8075. doi: 10.1126/science.aad6253. URL `http://science.sciencemag.org/content/351/6280/aad6253`.

[29] Andy Coghlan. Interview with Richard Kitney. Small is beautiful: Why a synthetic minimal genome is a big deal. New Scientist, [online] Available from: https://www.newscientist.com/article/2082313-small-is-beautiful-why-a-synthetic-minimal-genome-is-a-big-deal/ (Accessed 28 March 2016), 2016.

[30] Richard Wheeler. A-DNA, B-DNA and Z-DNA. Creative Commons Attribution-ShareAlike 3.0 license, 2007. URL `https://en.wikipedia.org/wiki/File:A-DNA,_B-DNA_and_Z-DNA.png`.

[31] J. D. Watson and F. H. C Crick. A structure for deoxyribose nucleic acid. *Nature*, 171:737738, 1953.

[32] Sponk. Difference DNA RNA. Creative Commons Attribution-ShareAlike 3.0 Unported license, 2010. URL `https://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-EN.svg`.

[33] John SantaLucia. A unified view of polymer, dumbbell, and oligonucleotide dna nearest-neighborthermodynamics. *Proceedings of the National Academy of Sciences*, 95(4):1460–1465, 1998. URL `http://www.pnas.org/content/95/4/1460.abstract`.

[34] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, April 2003. ISSN 1362-4962. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC169194/`.

[35] Joseph N. Zadeh, Conrad D. Steenberg, Justin S. Bois, Brian R. Wolfe, Marshall B. Pierce, Asif R. Khan, Robert M. Dirks, and Niles A. Pierce. Nupack: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173, 2011. ISSN 1096-987X. doi: 10.1002/jcc.21596. URL `http://dx.doi.org/10.1002/jcc.21596`.

[36] Ronny Lorenz, Stephan H Bernhart, Christian Hner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. ViennaRNA

package 2.0. *Algorithms for Molecular Biology : AMB*, 6:26–26, November 2011. ISSN 1748-7188. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3319429/.

[37] Jonathan P. K. Doye, Thomas E. Ouldridge, Ard A. Louis, Flavio Romano, Petr Sulc, Christian Matek, Benedict E. K. Snodin, Lorenzo Rovigatti, John S. Schreck, Ryan M. Harrison, and William P. J. Smith. Coarse-graining DNA for simulations of DNA nanotechnology. *Phys. Chem. Chem. Phys.*, 15:20395–20414, 2013. doi: 10.1039/C3CP53545B. URL http://dx.doi.org/10.1039/C3CP53545B.

[38] M Zuker and P Stiegler. Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, January 1981. ISSN 1362-4962. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC326673/.

[39] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990. ISSN 1097-0282. doi: 10.1002/bip.360290621. URL http://dx.doi.org/10.1002/bip.360290621.

[40] Stefan Wuchty, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–165, 1999. ISSN 1097-0282. doi: 10.1002/(SICI)1097-0282(199902)49:2⟨145::AID-BIP4⟩3.0.CO;2-G. URL http://dx.doi.org/10.1002/(SICI)1097-0282(199902)49:2<145::AID-BIP4>3.0.CO;2-G.

[41] Thomas E. Ouldridge, Ard A. Louis, and Jonathan P. K. Doye. DNA nanotweezers studied with a coarse-grained model of DNA. *Physical Review Letters*, 104(17), April 2010. doi: 10.1103/physrevlett.104.178101. URL http://dx.doi.org/10.1103/PhysRevLett.104.178101.

[42] T. E. Ouldridge, P. Sulc, F. Romano, J. P. K. Doye, and A. A. Louis. DNA hybridization kinetics: zippering, internal displacement and sequence

dependence. *Nucleic Acids Research*, 41(19):8886–8895, August 2013. doi: 10.1093/nar/gkt687. URL `http://dx.doi.org/10.1093/nar/gkt687`.

[43] Benedict E. K. Snodin, Ferdinando Randisi, Majid Mosayebi, Petr ulc, John S. Schreck, Flavio Romano, Thomas E. Ouldridge, Roman Tsukanov, Eyal Nir, Ard A. Louis, and Jonathan P. K. Doye. Introducing improved structural properties and salt dependence into a coarse-grained model of DNA. *The Journal of Chemical Physics*, 142(23):234901, 2015. doi: 10.1063/1.4921957. URL `http://dx.doi.org/10.1063/1.4921957`.

[44] Nancy C. Stellwagen. Electrophoresis of DNA in agarose gels, polyacrylamide gels and in free solution. *ELECTROPHORESIS*, 30(S1):S188–S195, June 2009. doi: 10.1002/elps.200900052. URL `https://doi.org/10.1002%2Felps.200900052`.

[45] I. Nachamkin, N. J. Panaro, M. Li, H. Ung, P. K. Yuen, L. J. Kricka, and P. Wilding. Agilent 2100 Bioanalyzer for restriction fragment length polymorphism analysis of the campylobacter jejuni flagellin gene. *Journal of Clinical Microbiology*, 39(2):754–757, February 2001. doi: 10.1128/jcm.39.2.754-757.2001. URL `https://doi.org/10.1128%2Fjcm.39.2.754-757.2001`.

[46] Richard P. Feynman. *Miniaturization*, chapter Theres plenty of room at the bottom, page 282296. Reinhold, 1961.

[47] Carlos Ernesto Castro, Fabian Kilchherr, Do-Nyun Kim, Enrique Lin Shiao, Tobias Wauer, Philipp Wortmann, Mark Bathe, and Hendrik Dietz. A primer to scaffolded DNA origami. *Nat Meth*, 8(3):221–229, March 2011. ISSN 1548-7091. URL `http://dx.doi.org/10.1038/nmeth.1570`.

[48] N. C. Seeman. Nucleic-acid junctions and lattices. *J. Theor. Biol.*, 99:237–247, 1982. URL `http://dx.doi.org/10.1016/0022-5193(82)90002-9`.

[49] Erik Winfree, Furong Liu, Lisa A. Wenzler, and Nadrian C. Seeman. Design and self-assembly of two-dimensional dna crystals. *Nature*, 394(6693):539–544, August 1998. ISSN 0028-0836. URL `http://dx.doi.org/10.1038/28998`.

[50] Hao Yan, Sung Ha Park, Gleb Finkelstein, John H. Reif, and Thomas H. LaBean. Dna-templated self-assembly of protein arrays and highly conductive nanowires. *Science*, 301(5641):1882–1884, 2003. ISSN 0036-8075. doi: 10.1126/science.1089389. URL `http://science.sciencemag.org/content/301/5641/1882`.

[51] Hanying Li, Joshua D. Carter, and Thomas H. LaBean. Nanofabrication by DNA self-assembly. *Materials Today*, 12(5):24 – 32, 2009. ISSN 1369-7021. doi: http://dx.doi.org/10.1016/S1369-7021(09)70157-9. URL `http://www.sciencedirect.com/science/article/pii/S1369702109701579`.

[52] Junghuei Chen and Nadrian C. Seeman. Synthesis from DNA of a molecule with the connectivity of a cube. *Nature*, 350(6319):631–633, April 1991. URL `http://dx.doi.org/10.1038/350631a0`.

[53] William M. Shih, Joel D. Quispe, and Gerald F. Joyce. A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron. *Nature*, 427(6975):618–621, February 2004. ISSN 0028-0836. URL `http://dx.doi.org/10.1038/nature02307`.

[54] R. P. Goodman, I. A. T. Schaap, C. F. Tardin, C. M. Erben, R. M. Berry, C. F. Schmidt, and A. J. Turberfield. Rapid chiral assembly of rigid DNA building blocks for molecular nanofabrication. *Science*, 310(5754):1661–1665, 2005. ISSN 0036-8075. doi: 10.1126/science.1120367. URL `http://science.sciencemag.org/content/310/5754/1661`.

[55] Luc Jaeger and Arkadiusz Chworos. The architectonics of programmable RNA and DNA nanostructures. *Current Opinion in Structural Biology*, 16 (4):531 – 543, 2006. ISSN 0959-440X. doi: http://dx.doi.org/10.1016/j.sbi.2006.07.001. URL `http://www.sciencedirect.com/science/article/pii/S0959440X06001187`. Membranes / Engineering and design.

[56] Yonggang Ke, Jaswinder Sharma, Minghui Liu, Kasper Jahn, Yan Liu, and Hao Yan. Scaffolded DNA origami of a DNA tetrahedron molecular container. *Nano*

Letters, 9(6):2445–2447, 2009. doi: 10.1021/nl901165f. URL http://dx.doi.org/10.1021/nl901165f. PMID: 19419184.

[57] Ebbe S. Andersen, Mingdong Dong, Morten M. Nielsen, Kasper Jahn, Ramesh Subramani, Wael Mamdouh, Monika M. Golas, Bjoern Sander, Holger Stark, Cristiano L. P. Oliveira, Jan Skov Pedersen, Victoria Birkedal, Flemming Besenbacher, Kurt V. Gothelf, and Jorgen Kjems. Self-assembly of a nanoscale DNA box with a controllable lid. *Nature*, 459(7243):73–76, May 2009. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature07971.

[58] Shawn M. Douglas, Hendrik Dietz, Tim Liedl, Bjorn Hogberg, Franziska Graf, and William M. Shih. Self-assembly of DNA into nanoscale three-dimensional shapes. *Nature*, 459(7245):414–418, May 2009. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature08016.

[59] Hendrik Dietz, Shawn M. Douglas, and William M. Shih. Folding DNA into twisted and curved nanoscale shapes. *Science*, 325(5941):725–730, 2009. doi: 10.1126/science.1174251. URL http://www.sciencemag.org/content/325/5941/725.abstract.

[60] Tim Liedl, Bjrn Hgberg, Jessica Tytell, Donald E Ingber, and William M Shih. Self-assembly of 3D prestressed tensegrity structures from DNA. *Nature nanotechnology*, 5(7):520–524, June 2010. ISSN 1748-3395. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2898913/.

[61] Zhao Zhao, Hao Yan, and Yan Liu. A route to scale up DNA origami using DNA tiles as folding staples. *Angewandte Chemie International Edition*, 49(8):1414–1417, 2010. ISSN 1521-3773. URL http://dx.doi.org/10.1002/anie.200906225.

[62] Enzo Kopperger, Tobias Pirzer, and Friedrich C. Simmel. Diffusive transport of molecular cargo tethered to a DNA origami platform. *Nano Letters*, 15(4):2693–2699, 2015. doi: 10.1021/acs.nanolett.5b00351. URL http://dx.doi.org/10.1021/acs.nanolett.5b00351. PMID: 25739805.

[63] Alexander E. Marras, Lifeng Zhou, Hai-Jun Su, and Carlos E. Castro. Programmable motion of DNA origami mechanisms. *Proceedings of the National Academy of Sciences*, 112(3):713–718, 2015. doi: 10.1073/pnas.1408869112. URL `http://www.pnas.org/content/112/3/713.abstract`.

[64] Chengde Mao, Weiqiong Sun, Zhiyong Shen, and Nadrian C. Seeman. A nanomechanical device based on the B-Z transition of DNA. *Nature*, 397(6715): 144–146, January 1999. ISSN 0028-0836. URL `http://dx.doi.org/10.1038/16437`.

[65] Bernard Yurke, Andrew J. Turberfield, Allen P. Mills, Friedrich C. Simmel, and Jennifer L. Neumann. A DNA-fuelled molecular machine made of DNA. *Nature*, 406(6796):605–608, August 2000. ISSN 0028-0836. URL `http://dx.doi.org/10.1038/35020524`.

[66] William B. Sherman, , and Nadrian C. Seeman*. A precisely controlled DNA biped walking device. *Nano Letters*, 4(7):1203–1207, 2004. doi: 10.1021/nl049527q. URL `http://dx.doi.org/10.1021/nl049527q`.

[67] Jonathan Bath, Simon J. Green, and Andrew J. Turberfield. A free-running DNA motor powered by a nicking enzyme. *Angewandte Chemie International Edition*, 44(28):4358–4361, 2005. ISSN 1521-3773. doi: 10.1002/anie.200501262. URL `http://dx.doi.org/10.1002/anie.200501262`.

[68] Carlos E. Castro, Hai-Jun Su, Alexander E. Marras, Lifeng Zhou, and Joshua Johnson. Mechanical design of DNA nanostructures. *Nanoscale*, 7(14):5913–5921, 2015. doi: 10.1039/c4nr07153k. URL `https://doi.org/10.1039%2Fc4nr07153k`.

[69] T. Gerling, K. F. Wagenbauer, A. M. Neuner, and H. Dietz. Dynamic DNA devices and assemblies formed by shape-complementary, non-base pairing 3D components. *Science*, 347(6229):1446–1452, March 2015. doi: 10.1126/science.aaa5372. URL `https://doi.org/10.1126%2Fscience.aaa5372`.

[70] Kyle Lund, Anthony J. Manzo, Nadine Dabby, Nicole Michelotti, Alexander Johnson-Buck, Jeanette Nangreave, Steven Taylor, Renjun Pei, Milan N. Stojanovic, Nils G. Walter, Erik Winfree, and Hao Yan. Molecular robots guided by prescriptive landscapes. *Nature*, 465(7295):206–210, May 2010. ISSN 0028-0836. URL `http://dx.doi.org/10.1038/nature09012`.

[71] Masayuki Endo, Yousuke Katsuda, Kumi Hidaka, and Hiroshi Sugiyama. A versatile DNA nanochip for direct analysis of DNA base-excision repair. *Angewandte Chemie International Edition*, 49(49):9412–9416, 2010. ISSN 1521-3773. doi: 10.1002/anie.201003604. URL `http://dx.doi.org/10.1002/anie.201003604`.

[72] Niels V. Voigt, Thomas Torring, Alexandru Rotaru, Mikkel F. Jacobsen, Jens B. Ravnsbaek, Ramesh Subramani, Wael Mamdouh, Jorgen Kjems, Andriy Mokhir, Flemming Besenbacher, and Kurt Vesterager Gothelf. Single-molecule chemical reactions on DNA origami. *Nat Nano*, 5(3):200–203, March 2010. ISSN 1748-3387. URL `http://dx.doi.org/10.1038/nnano.2010.5`.

[73] Barbara Sacc, Rebecca Meyer, Michael Erkelenz, Kathrin Kiko, Andreas Arndt, Hendrik Schroeder, Kersten S. Rabe, and Christof M. Niemeyer. Orthogonal protein decoration of DNA origami. *Angewandte Chemie International Edition*, 49(49):9378–9383, 2010. ISSN 1521-3773. doi: 10.1002/anie.201005931. URL `http://dx.doi.org/10.1002/anie.201005931`.

[74] Martyn Amos. DNA computing. In Robert A. Meyers, editor, *Computational Complexity: Theory, Techniques and Applications*, pages 882–896. Springer, New York, NY, 2012.

[75] A. Gibbons, M. Amos, and D. Hodgson. DNA computing. *Current Opinion in Biotechnology*, 8(1):103–106, 1997.

[76] LM Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994. ISSN 0036-8075. doi: 10.1126/science.7973651. URL `http://science.sciencemag.org/content/266/5187/1021`.

[77] M. R. Garey Michael R. Garey, David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, chapter A1.3, pages 199–200. W. H. Freeman And Company, 2011. ISBN 0716710455. URL http://www.ebook.de/de/product/3637119/michael_r_garey_david_s_johnson_m_r_garey_computers_and_intractability_a_guide_to_the_theory_of_np_completeness.html.

[78] Martyn Amos, Gheorghe Pun, Grzegorz Rozenberg, and Arto Salomaa. Topics in the theory of DNA computing. *Theoretical Computer Science*, 287(1):3 – 38, 2002. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/S0304-3975(02)00134-2. URL http://www.sciencedirect.com/science/article/pii/S0304397502001342.

[79] M. Ogihara and A. Ray. Simulating boolean circuits on a DNA computer. *Algorithmica*, 25(2-3):239–250, June 1999. doi: 10.1007/pl00008276. URL http://dx.doi.org/10.1007/PL00008276.

[80] Yaakov Benenson, Binyamin Gil, Uri Ben-Dor, Rivka Adar, and Ehud Shapiro. An autonomous molecular computer for logical control of gene expression. *Nature*, 429(6990):423–429, May 2004. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature02551.

[81] Yaakov Benenson, Tamar Paz-Elizur, Rivka Adar, Ehud Keinan, Zvi Livneh, and Ehud Shapiro. Programmable and autonomous computing machine made of biomolecules. *Nature*, 414(6862):430–434, November 2001. ISSN 0028-0836. URL http://dx.doi.org/10.1038/35106533.

[82] J. Bonnet, P. Yin, M. E. Ortiz, P. Subsoontorn, and D. Endy. Amplifying genetic logic gates. *Science*, 340(6132):599–603, March 2013. doi: 10.1126/science.1232758. URL http://dx.doi.org/10.1126/science.1232758.

[83] L. Qian and E. Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–201, 2011. doi: 10.1126/science.1200520.

[84] Y. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig. Programmable chemical controllers made from DNA. *Nat. Nano.*, 8 (10):755–762, 2013. ISSN 1748-3387. doi: 10.1038/nnano.2013.189. URL http://www.nature.com/nnano/journal/v8/n10/abs/nnano.2013.189.html.

[85] David Yu Zhang and Georg Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nat Chem*, 3(2):103–113, February 2011. ISSN 1755-4330. URL http://dx.doi.org/10.1038/nchem.957.

[86] W Szybalski and A Skalka. Nobel prizes and restriction enzymes. *Gene*, 4(3):181 – 182, 1978. ISSN 0378-1119. doi: http://dx.doi.org/10.1016/0378-1119(78)90016-1. URL http://www.sciencedirect.com/science/article/pii/0378111978900161.

[87] Priscilla E. M. Purnick and Ron Weiss. The second wave of synthetic biology: from modules to systems. *Nat Rev Mol Cell Biol*, 10(6):410–422, June 2009. ISSN 1471-0072. URL http://dx.doi.org/10.1038/nrm2698.

[88] Joachim Boldt. Synthetic biology: Origin, scope, and ethics. *Minding Nature*, 3(1), April 2010.

[89] Peter Dabrock. Playing god? synthetic biology as a theological and ethical challenge. *Systems and Synthetic Biology*, 3(1-4):47–54, October 2009. doi: 10.1007/s11693-009-9028-5. URL http://dx.doi.org/10.1007/s11693-009-9028-5.

[90] Richard Dawkins. *Climbing Mount Improbable*, pages 3–4. Penguin Books Ltd (UK), 2006. ISBN 0141026170. URL http://www.ebook.de/de/product/5211794/richard_dawkins_climbing_mount_improbable.html.

[91] Richard Kelwick, James T MacDonald, Alexander J Webb, and Paul Freemont. Developments in the tools and methodologies of synthetic biology. *Frontiers in Bioengineering and Biotechnology*, 2:60–, November 2014. ISSN 2296-4185. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4244866/.

[92] Manuel Porcar, Antoine Danchin, Victor Lorenzo, Vitor A. dos Santos, Natalio Krasnogor, Steen Rasmussen, and Andrés Moya. The ten grand challenges

of synthetic life. *Systems and Synthetic Biology*, 5(1):1–9, 2011. ISSN 1872-5333. doi: 10.1007/s11693-011-9084-5. URL `http://dx.doi.org/10.1007/s11693-011-9084-5`.

[93] Martyn Amos. Population-based microbial computing: a third wave of synthetic biology? *International Journal of General Systems*, 43(7):770–782, May 2014. doi: 10.1080/03081079.2014.921001. URL `https://doi.org/10.1080%2F03081079.2014.921001`.

[94] Christopher A Voigt. Genetic parts to program bacteria. *Current Opinion in Biotechnology*, 17(5):548–557, October 2006. doi: 10.1016/j.copbio.2006.09.001. URL `https://doi.org/10.1016%2Fj.copbio.2006.09.001`.

[95] Adam P. Arkin and Drew Endy. A standard parts list for biological circuitry. Technical report, DARPA white paper, 1999. URL `http://dspace.mit.edu/handle/1721.1/29794`.

[96] M. A. Brasch. ORFeome cloning and systems biology: Standardized mass production of the parts from the parts-list. *Genome Research*, 14(10b):2001–2009, October 2004. doi: 10.1101/gr.2769804. URL `https://doi.org/10.1101%2Fgr.2769804`.

[97] Reshma P Shetty, Drew Endy, and Thomas F Knight. Engineering BioBrick vectors from BioBrick parts. *Journal of Biological Engineering*, 2(1):5, 2008. doi: 10.1186/1754-1611-2-5. URL `https://doi.org/10.1186%2F1754-1611-2-5`.

[98] Jonathan Bath and Andrew J. Turberfield. DNA nanomachines. *Nat Nano*, 2 (5):275–284, May 2007. ISSN 1748-3387. URL `http://dx.doi.org/10.1038/nnano.2007.104`.

[99] Víctor de Lorenzo. Beware of metaphors: Chasses and orthogonality in synthetic biology. *Bioengineered Bugs*, 2(1):3–7, January 2011. doi: 10.4161/bbug.2.1.13388. URL `http://dx.doi.org/10.4161/bbug.2.1.13388`.

[100] Benjamin Kick, Florian Praetorius, Hendrik Dietz, and Dirk Weuster-Botz. Efficient production of single-stranded phage DNA as scaffolds for DNA origami.

Nano Lett., pages –, June 2015. ISSN 1530-6984. doi: 10.1021/acs.nanolett.5b01461. URL http://dx.doi.org/10.1021/acs.nanolett.5b01461.

[101] Cody Geary, Paul W. K. Rothemund, and Ebbe S. Andersen. A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Science*, 345(6198):799–804, 2014. doi: 10.1126/science.1253920. URL http://www.sciencemag.org/content/345/6198/799.abstract.

[102] Daan Frenkel. Order through entropy. *Nature Materials*, 14(1):9–12, December 2014. doi: 10.1038/nmat4178. URL https://doi.org/10.1038%2Fnmat4178.

[103] Paul W. K. Rothemund and Erik Winfree. The program-size complexity of self-assembled squares (extended abstract). In *Proceedings of the thirty-second annual ACM symposium on Theory of computing - STOC 2000*. Association for Computing Machinery (ACM), 2000. doi: 10.1145/335305.335358. URL https://doi.org/10.1145%2F335305.335358.

[104] William M. Jacobs and Daan Frenkel. Self-assembly of structures with addressable complexity. *Journal of the American Chemical Society*, 138(8):2457–2467, March 2016. doi: 10.1021/jacs.5b11918. URL http://dx.doi.org/10.1021/jacs.5b11918.

[105] Bryan Wei, Mingjie Dai, and Peng Yin. Complex shapes self-assembled from single-stranded DNA tiles. *Nature*, 485(7400):623–626, May 2012. doi: 10.1038/nature11075. URL https://doi.org/10.1038%2Fnature11075.

[106] Y. Ke, L. L. Ong, W. M. Shih, and P. Yin. Three-dimensional structures self-assembled from DNA bricks. *Science*, 338(6111):1177–1183, November 2012. doi: 10.1126/science.1227268. URL https://doi.org/10.1126%2Fscience.1227268.

[107] Cameron Myhrvold, Mingjie Dai, Pamela A. Silver, and Peng Yin. Isothermal self-assembly of complex DNA structures under diverse and biocompatible conditions. *Nano Letters*, 13(9):4242–4248, September 2013. doi: 10.1021/nl4019512. URL https://doi.org/10.1021%2Fnl4019512.

[108] Bryan Wei, Mingjie Dai, Cameron Myhrvold, Yonggang Ke, Ralf Jungmann, and Peng Yin. Design space for complex DNA structures. *Journal of the American Chemical Society*, 135(48):18080–18088, December 2013. doi: 10.1021/ja4062294. URL https://doi.org/10.1021%2Fja4062294.

[109] Wen Wang, Tong Lin, Suoyu Zhang, Tanxi Bai, Yongli Mi, and Bryan Wei. Self-assembly of fully addressable DNA nanostructures from double crossover tiles. *Nucleic Acids Research*, 44(16):7989–7996, August 2016. doi: 10.1093/nar/gkw670. URL http://dx.doi.org/10.1093/nar/gkw670.

[110] Ludovico Cademartiri and Kyle J. M. Bishop. Programmable self-assembly. *Nature Materials*, 14(1):2–9, December 2014. doi: 10.1038/nmat4184. URL https://doi.org/10.1038%2Fnmat4184.

[111] Yanming Fu, Dongdong Zeng, Jie Chao, Yanqiu Jin, Zhao Zhang, Huajie Liu, Di Li, Hongwei Ma, Qing Huang, Kurt V. Gothelf, and Chunhai Fan. Single-step rapid assembly of DNA origami nanostructures for addressable nanoscale bioreactors. *Journal of the American Chemical Society*, 135(2):696–702, January 2013. doi: 10.1021/ja3076692. URL http://dx.doi.org/10.1021/ja3076692.

[112] Katherine E. Dunn, Frits Dannenberg, Thomas E. Ouldridge, Marta Kwiatkowska, Andrew J. Turberfield, and Jonathan Bath. Guiding the folding pathway of DNA origami. *Nature*, advance online publication:–, August 2015. ISSN 1476-4687. URL http://dx.doi.org/10.1038/nature14860.

[113] Jean-Philippe J. Sobczak, Thomas G. Martin, Thomas Gerling, and Hendrik Dietz. Rapid folding of DNA into nanoscale shapes at constant temperature. *Science*, 338(6113):1458–1461, 2012. doi: 10.1126/science.1229919. URL http://www.sciencemag.org/content/338/6113/1458.abstract.

[114] William M. Jacobs, Aleks Reinhardt, and Daan Frenkel. Rational design of self-assembly pathways for complex multicomponent structures. *Proceedings of the National Academy of Sciences*, 112(20):6313–6318, May 2015. doi: 10.1073/pnas.1502210112. URL http://dx.doi.org/10.1073/pnas.1502210112.

[115] N. Srinivas, T. E. Ouldridge, P. Sulc, J. M. Schaeffer, B. Yurke, A. A. Louis, J. P. K. Doye, and E. Winfree. On the biophysics and kinetics of toehold-mediated DNA strand displacement. *Nucleic Acids Research*, 41(22):10641–10658, September 2013. doi: 10.1093/nar/gkt801. URL `https://doi.org/10.1093%2Fnar%2Fgkt801`.

[116] N. de Bruijn. A combinatorial problem. *Proc. Nederl. Akad. Wetensch.*, 49: 758–764, 1946.

[117] Bonnie Berger, Jian Peng, and Mona Singh. Computational solutions for omics data. *Nat Rev Genet*, 14(5):333–346, May 2013. ISSN 1471-0056. URL `http://dx.doi.org/10.1038/nrg3433`.

[118] Phillip E C Compeau, Pavel A Pevzner, and Glenn Tesler. How to apply de bruijn graphs to genome assembly. *Nat Biotech*, 29(11):987–991, November 2011. ISSN 1087-0156. URL `http://dx.doi.org/10.1038/nbt.2023`.

[119] Daniel R Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–829, March 2008. ISSN 1549-5477. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2336801/`.

[120] Pavel A Pevzner, Haixu Tang, and Michael S Waterman. An eulerian path approach to dna fragment assembly. *Proceedings of the National Academy of Sciences of the United States of America*, 98(17):9748–9753, June 2001. ISSN 1091-6490. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC55524/`.

[121] V.R Chechetkin. Block structure and stability of the genetic code. *Journal of Theoretical Biology*, 222(2):177 – 188, 2003. ISSN 0022-5193. doi: http://dx.doi.org/10.1016/S0022-5193(03)00025-0. URL `http://www.sciencedirect.com/science/article/pii/S0022519303000250`.

[122] Klaus F. Wagenbauer, Christian H. Wachauf, and Hendrik Dietz. Quantifying quality in DNA self-assembly. *Nat Commun*, 5:–, April 2014. URL `http://dx.doi.org/10.1038/ncomms4691`.

[123] Joe Sawada, Aaron Williams, and Dennis Wong. A surprisingly simple de bruijn sequence construction. *Discrete Mathematics*, 339(1):127 – 131, 2016. ISSN 0012-365X. doi: http://dx.doi.org/10.1016/j.disc.2015.08.002. URL `http://www.sciencedirect.com/science/article/pii/S0012365X15002873`.

[124] Solomon W. Golomb. *Shift Register Sequences*. Aegean Park Press, Laguna Hills, CA, USA, 1981. ISBN 0894120484.

[125] Harold Fredricksen and James Maiorana. Necklaces of beads in k colors and k-ary de bruijn sequences. *Discrete Mathematics*, 23(3):207 – 210, 1978. ISSN 0012-365X. doi: http://dx.doi.org/10.1016/0012-365X(78)90002-X. URL `http://www.sciencedirect.com/science/article/pii/0012365X7890002X`.

[126] Harold Fredricksen. Generation of the ford sequence of length 2n, n large. *Journal of Combinatorial Theory, Series A*, 12(1):153 – 154, 1972. ISSN 0097-3165. doi: http://dx.doi.org/10.1016/0097-3165(72)90091-X. URL `http://www.sciencedirect.com/science/article/pii/009731657290091X`.

[127] A Ralston. A new memoryless algorithm for de bruijn sequences. *Journal of Algorithms*, 2(1):50 – 62, 1981. ISSN 0196-6774. doi: http://dx.doi.org/10.1016/0196-6774(81)90007-9. URL `http://www.sciencedirect.com/science/article/pii/0196677481900079`.

[128] Harold Fredricksen and Irving Kessler. Lexicographic compositions and debruijn sequences. *Journal of Combinatorial Theory, Series A*, 22(1):17 – 30, 1977. ISSN 0097-3165. doi: http://dx.doi.org/10.1016/0097-3165(77)90059-0. URL `http://www.sciencedirect.com/science/article/pii/0097316577900590`.

[129] Harold Fredricksen. A class of nonlinear de bruijn cycles. *Journal of Combinatorial Theory, Series A*, 19(2):192 – 199, 1975. ISSN 0097-3165. doi: http://dx.doi.org/10.1016/S0097-3165(75)80007-0. URL `http://www.sciencedirect.com/science/article/pii/S0097316575800070`.

[130] T. Etzion and A. Lempel. Algorithms for the generation of full-length shift-register sequences. *IEEE Transactions on Information Theory*, 30(3):480–484,

May 1984. doi: 10.1109/tit.1984.1056919. URL http://dx.doi.org/10.1109/TIT.1984.1056919.

[131] Yuejiang Huang. A new algorithm for the generation of binary de bruijn sequences. *J. Algorithms*, 11(1):44–51, February 1990. ISSN 0196-6774. doi: 10.1016/0196-6774(90)90028-D. URL http://dx.doi.org/10.1016/0196-6774(90)90028-D.

[132] Joe Sawada, Brett Stevens, and Aaron Williams. *De Bruijn Sequences for the Binary Strings with Maximum Density*, pages 182–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-19094-0. doi: 10.1007/978-3-642-19094-0_19. URL http://dx.doi.org/10.1007/978-3-642-19094-0_19.

[133] Joe Sawada, Aaron Williams, and Dennis Wong. *Universal Cycles for Weight-Range Binary Strings*, pages 388–401. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-45278-9. doi: 10.1007/978-3-642-45278-9_33. URL http://dx.doi.org/10.1007/978-3-642-45278-9_33.

[134] M. H. Martin. A problem in arrangements. *Bulletin of the American Mathematical Society*, 40(12):859–865, December 1934. doi: 10.1090/s0002-9904-1934-05988-3. URL http://dx.doi.org/10.1090/S0002-9904-1934-05988-3.

[135] Abbas M. Alhakim. A simple combinatorial algorithm for de bruijn sequences. *The American Mathematical Monthly*, 117(8):728, 2010. doi: 10.4169/000298910x515794. URL http://dx.doi.org/10.4169/000298910X515794.

[136] Andreas Klein. *Stream Ciphers*, page 59. Springer London, 2013. URL http://www.ebook.de/de/product/25034618/andreas_klein_stream_ciphers.html.

[137] A. Travers. *DNA-Protein Interactions*. Springer, 1993. ISBN 0412259907.

[138] C. O. Pabo and R. T. Sauer. Protein-DNA recognition. *Annual Review of Biochemistry*, 53(1):293–321, June 1984. doi: 10.1146/annurev.bi.53.070184.001453. URL `https://doi.org/10.1146%2Fannurev.bi.53.070184.001453`.

[139] Douglas H Turner and David H Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, 38(Database issue):D280–D282, October 2009. ISSN 1362-4962. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2808915/`.

[140] Shawn M Douglas, Adam H Marblestone, Surat Teerapittayanon, Alejandro Vazquez, George M Church, and William M Shih. Rapid prototyping of 3D DNA-origami shapes with caDNAno. *Nucleic Acids Research*, 37(15):5001–5006, May 2009. ISSN 1362-4962. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2731887/`.

[141] Pengfei Wang, Seung Hyeon Ko, Cheng Tian, Chenhui Hao, and Chengde Mao. RNA-DNA hybrid origami: folding of a long RNA single strand into complex nanostructures using short dna helper strands. *Chem. Commun.*, 49:5462–5464, 2013. doi: 10.1039/C3CC41707G. URL `http://dx.doi.org/10.1039/C3CC41707G`.

[142] Richard Mnch, Karsten Hiller, Heiko Barg, Dana Heldt, Simone Linz, Edgar Wingender, and Dieter Jahn. PRODORIC: prokaryotic database of gene regulation. *Nucleic Acids Research*, 31(1):266–269, 2003. doi: 10.1093/nar/gkg037. URL `http://nar.oxfordjournals.org/content/31/1/266.abstract`.

[143] Robert D. Finn, Alex Bateman, Jody Clements, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, Erik L. L. Sonnhammer, John Tate, and Marco Punta. Pfam: the protein families database. *Nucleic Acids Research*, 42(D1):D222–D230, 2014. doi: 10.1093/nar/gkt1223. URL `http://nar.oxfordjournals.org/content/42/D1/D222.abstract`.

[144] I. Sillitoe, T. E. Lewis, A. Cuff, S. Das, P. Ashford, N. L. Dawson, N. Furnham, R. A. Laskowski, D. Lee, J. G. Lees, S. Lehtinen, R. A. Studer, J. Thornton, and

C. A. Orengo. CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Research*, 43(D1):D376–D381, October 2014. doi: 10.1093/nar/gku947. URL `https://doi.org/10.1093%2Fnar%2Fgku947`.

[145] Todd M. Lowe and Sean R. Eddy. tRNAscan-SE: A program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Research*, 25(5):0955–964, 1997. doi: 10.1093/nar/25.5.0955. URL `http://nar.oxfordjournals.org/content/25/5/0955.abstract`.

[146] A. L. Delcher, K. A. Bratke, E. C. Powers, and S. L. Salzberg. Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*, 23(6): 673–679, January 2007. doi: 10.1093/bioinformatics/btm009. URL `https://doi.org/10.1093%2Fbioinformatics%2Fbtm009`.

[147] Anders Krogh, Bjrn Larsson, Gunnar von Heijne, and Erik L.L Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *Journal of Molecular Biology*, 305(3):567–580, January 2001. doi: 10.1006/jmbi.2000.4315. URL `https://doi.org/10.1006%2Fjmbi.2000.4315`.

[148] S. Griffiths-Jones. The microRNA registry. *Nucleic Acids Research*, 32(90001): 109D–111, January 2004. doi: 10.1093/nar/gkh023. URL `https://doi.org/10.1093%2Fnar%2Fgkh023`.

[149] S. Griffiths-Jones. miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic Acids Research*, 34(90001):D140–D144, January 2006. doi: 10.1093/nar/gkj112. URL `https://doi.org/10.1093%2Fnar%2Fgkj112`.

[150] Dan Gusfield. Linear-time construction of suffix trees. In *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

[151] Alexandria N. Marchi, Ishtiaq Saaem, Briana N. Vogen, Stanley Brown, and Thomas H. LaBean. Toward larger DNA origami. *Nano Letters*, 14(10): 5740–5747, 2014. doi: 10.1021/nl502626s. URL `http://dx.doi.org/10.1021/nl502626s`. PMID: 25179827.

[152] Elisabeth Pound, Jeffrey R. Ashton, Hctor A. Becerril, and Adam T. Woolley. Polymerase chain reaction based scaffold preparation for the production of thin, branched DNA origami nanostructures of arbitrary sizes. *Nano Letters*, 9(12): 4302–4305, 2009. doi: 10.1021/nl902535q. URL `http://dx.doi.org/10.1021/nl902535q`. PMID: 19995086.

[153] Donald Ervin Knuth. *The Art of Computer Programming 1. Fundamental Algorithms*, chapter 2.2.1: Stacks, Queues, and Deques, pages 238–242. Addison Wesley, 1997. ISBN 0201896834. URL `http://www.ebook.de/de/product/3236571/donald_ervin_knuth_the_art_of_computer_programming_1_fundamental_algorithms.html`.

[154] Richard A. J. Woolley, Julian Stirling, Adrian Radocea, Natalio Krasnogor, and Philip Moriarty. Automated probe microscopy via evolutionary optimization at the atomic scale. *Applied Physics Letters*, 98(25):253104, 2011. doi: http://dx.doi.org/10.1063/1.3600662. URL `http://scitation.aip.org/content/aip/journal/apl/98/25/10.1063/1.3600662`.

[155] Germán Terrazas and Natalio Krasnogor. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2011)*, chapter Genotype-Fitness Correlation Analysis for Evolutionary Design of Self-assembly Wang Tiles, pages 73–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-24094-2. doi: 10.1007/978-3-642-24094-2_5. URL `http://dx.doi.org/10.1007/978-3-642-24094-2_5`.

[156] G. Terrazas, M. Gheorghe, G. Kendall, and N. Krasnogor. Evolving tiles for automated self-assembly design. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 2001–2008, 2007.

[157] Peter Siepmann, Christopher P. Martin, Ioan Vancea, Philip J. Moriarty, and Natalio Krasnogor. A genetic algorithm approach to probing the evolution of self-organized nanostructured systems. *Nano Letters*, 7(7):1985–1990, 2007. doi: 10.1021/nl070773m. URL `http://dx.doi.org/10.1021/nl070773m`. PMID: 17552572.

[158] M. R. Lakin and A. Phillips. Modelling, simulating and verifying Turing-powerful strand displacement systems. *Proceedings of the 17th International Conference on DNA Computing and Molecular Programming*, 6937:130–144, 2011.

[159] Rui Sousa and Srabani Mukherjee. T7 RNA polymerase. volume Volume 73, pages 1–41. Academic Press, 2003. URL `http://www.sciencedirect.com/science/article/pii/S0079660303010018`.

[160] E D Jorgensen, R K Durbin, S S Risman, and W T McAllister. Specific contacts between the bacteriophage T3, T7, and SP6 RNA polymerases and their promoters. *Journal of Biological Chemistry*, 266(1):645–651, 1991. URL `http://www.jbc.org/content/266/1/645.abstract`.

[161] J E Brown, J F Klement, and W T McAllister. Sequences of three promoters for the bacteriophage SP6 RNA polymerase. *Nucleic Acids Research*, 14(8):3521–3526, April 1986. ISSN 1362-4962. URL `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC339790/`.

[162] Rajiv P. Bandwar, Yiping Jia, Natalie M. Stano, and Smita S. Patel. Kinetic and thermodynamic basis of promoter strength: multiple steps of transcription initiation by T7 RNA polymerase are modulated by the promoter sequence. *Biochemistry*, 41(11):3586–3595, March 2002. doi: 10.1021/bi0158472. URL `http://dx.doi.org/10.1021/bi0158472`.

[163] Anton Kuzyk, Robert Schreiber, Zhiyuan Fan, Gunther Pardatscher, Eva-Maria Roller, Alexander Hogele, Friedrich C. Simmel, Alexander O. Govorov, and Tim Liedl. DNA-based self-assembly of chiral plasmonic nanostructures with tailored optical response. *Nature*, 483(7389):311–314, March 2012. ISSN 0028-0836. URL `http://dx.doi.org/10.1038/nature10889`.

[164] Melanie Mitchell. *An Introduction to Genetic Algorithms.* MIT PR, 1998. ISBN 0262631857. URL `http://www.ebook.de/de/product/3240124/melanie_mitchell_an_introduction_to_genetic_algorithms.html`.

[165] Melody Redman, Andrew King, Caroline Watson, and David King. What is CRISPR/cas9? *Archives of disease in childhood - Education & practice edition*, 101(4):213–215, April 2016. doi: 10.1136/archdischild-2016-310459. URL `https://doi.org/10.1136%2Farchdischild-2016-310459`.

[166] P. Horvath and R. Barrangou. CRISPR/Cas, the immune system of bacteria and archaea. *Science*, 327(5962):167–170, January 2010. doi: 10.1126/science.1179555. URL `https://doi.org/10.1126%2Fscience.1179555`.

[167] Sanjay Tyagi and Fred Russell Kramer. Molecular beacons: Probes that fluoresce upon hybridization. *Nature Biotechnology*, 14(3):303–308, March 1996. doi: 10.1038/nbt0396-303. URL `https://doi.org/10.1038%2Fnbt0396-303`.

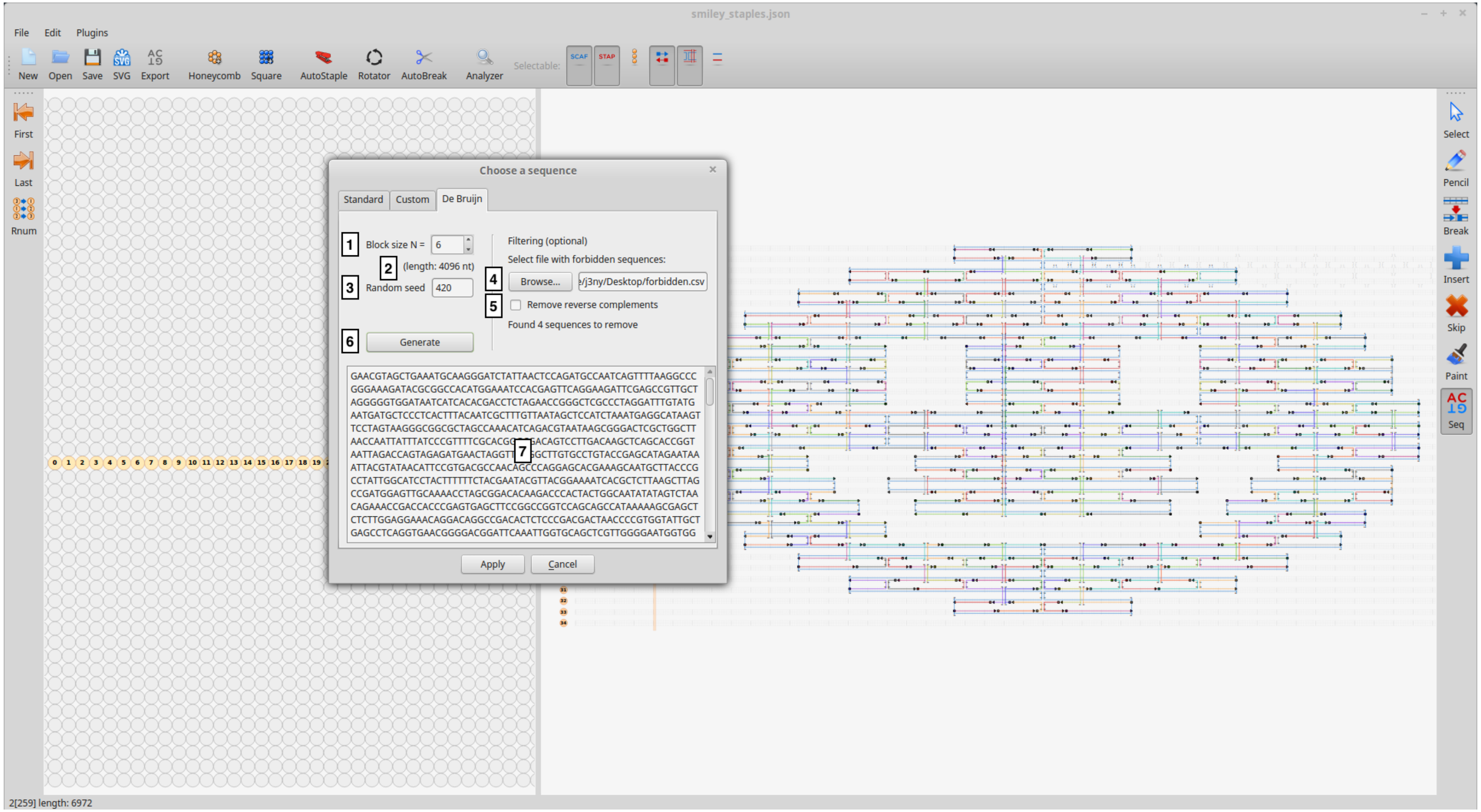
Appendix A

Appendix A: DNA origami

A.1 Software Tools

The sequence tool, shown on Figure A.1, has following options:

1. Block size: a number specifying the order of underlying De Bruijn graph, the larger the number the longer the DBS that can be created
2. Maximum length available (before filtering)
3. Random seed: used by a random number generator during the graph traversal
4. Browse: allows to select a .csv file with taboo sequences
5. Remove reverse complements: if ticked will also remove the sequences that are reverse complements of the ones provided
6. Generate button: starts the algorithm
7. A textfield with resulting DBS sequence



Choose a sequence

Standard | Custom | De Bruijn

1 Block size N = 6

2 (length: 4096 nt)

3 Random seed 420

6 Generate

Filtering (optional)

Select file with forbidden sequences:

4 Browse... e/j3ny/Desktop/forbidden.csv

5 □ Remove reverse complements

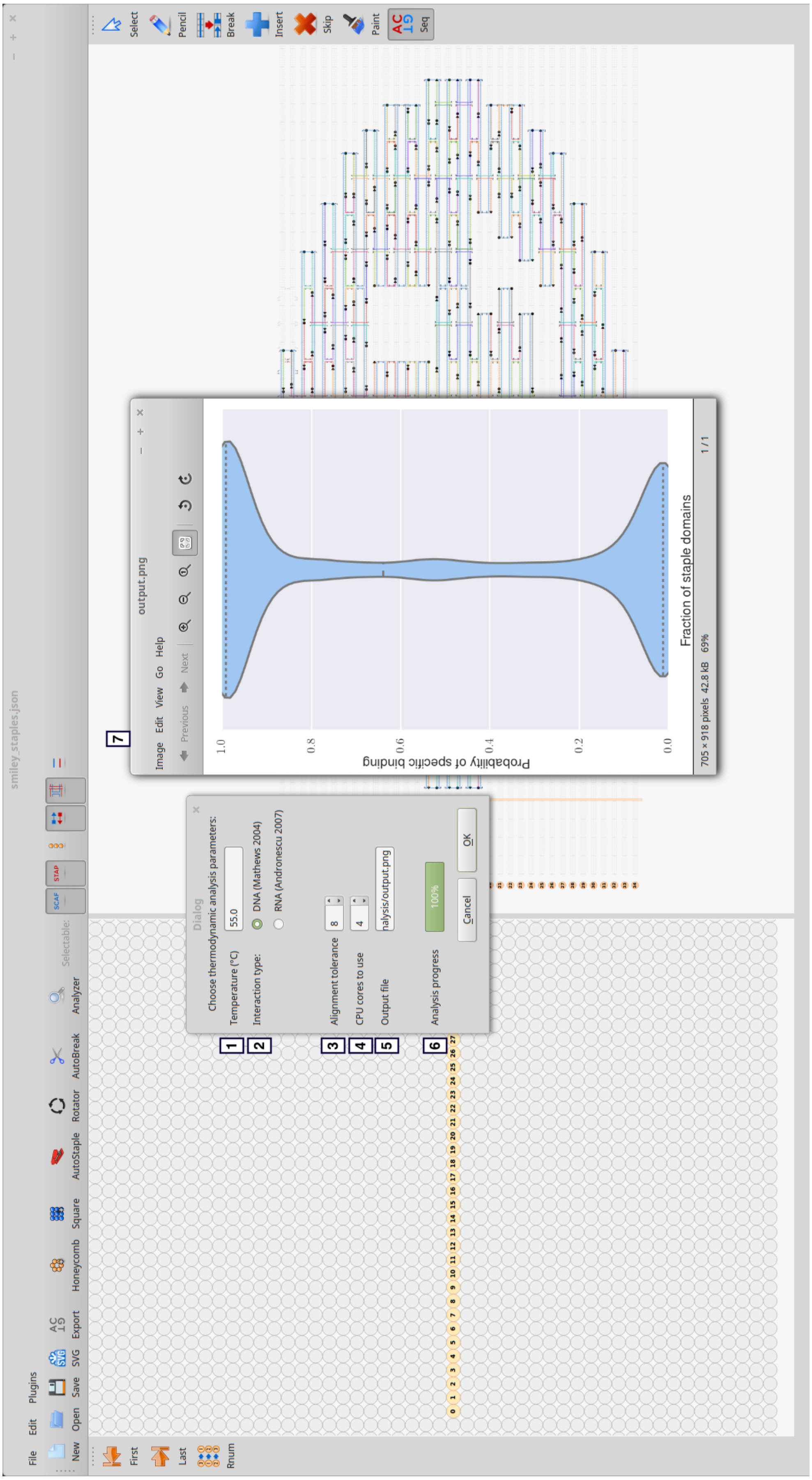
Found 4 sequences to remove

7

```
GAACGTAGCTGAAATGCAAGGGATCTATTAACTCCAGATGCCAATCAGTTTTAAGGCCC
GGGAAAGATACGCGGCCACATGGAAATCCACGAGTTCAGGAAGATTCGAGCCGTTGCT
AGGGGGTGGATAATCATCACACGACCTCTAGAACCGGGCTCGCCCTAGGATTTGTATG
AATGATGCTCCCTCACTTTACAATCGCTTTGTTAATAGCTCCATCTAAATGAGGCATAAGT
TCCTAGTAAGGGCGGCGCTAGCCAAACATCAGACGTAATAAGCGGGACTCGCTGGCTT
AACCAATTATTTATCCCGTTTTCGCACGG GACAGTCCTTGACAAGCTCAGCACCGGT
AATTAGACCAGTAGAGATGAACTAGGTT GGCTTGTGCCTGTACCGAGCATAGAATAA
ATTACGTATAACATTCCGTGACGCCAACAGCCCAGGAGCACGAAAGCAATGCTTACCCG
CCTATTGGCATCCTACTTTTTTCTACGAATACGTTACGGAAAATCACGCTCTTAAGCTTAG
CCGATGGAGTTGCAAAACCTAGCGGACACAAGACCCACTACTGGCAATATATAGTCTAA
CAGAAACCGACCACCCGAGTGAGCTTCCGGCCGGTCCAGCAGCCATAAAAAGCGAGCT
CTCTTGGAGGAAACAGGACAGGCCGACACTCTCCCGACGACTAACCCCGTGGTATTGCT
GAGCCTCAGGTGAACGGGGACGGATTCAAATTGGTGCAGCTCGTTGGGGAATGGTGG
```

Apply | Cancel

FIGURE A.1: A screenshot of sequence tool available as caDNAno plugin.



Dialog

Choose thermodynamic analysis parameters:

1 Temperature (°C) 55.0

2 Interaction type: ● DNA (Mathews 2004) ○ RNA (Andronescu 2007)

3 Alignment tolerance 8

4 CPU cores to use 4

5 Output file nalysis/output.png

6 Analysis progress 100%

Cancel OK

Figure A.2: A screenshot of addressability analyser available as caDNAno plugin.

A.2 Preparation of Samples

DNA Origami with pUC19

Briefly, the pUC19 plasmid was treated with Nt.BspQI nicking enzyme (NEB, UK) at 50 for 90 minutes. The mix solution was then incubated for 20 hours at 37 with T7 exonuclease and Lambda exonuclease to remove the complementary strand, leaving the scaffold intact After adding ethylenediaminetetraacetic acid (EDTA; 10 mM final concentration) for 30 minutes at room temperature to inactivate the enzyme, the ssDNA scaffold was ethanol precipitated, air dried and then dissolved in Tris-EDTA (TE) buffer.

To generat a linear pUC19 scaffold, a short oligo (GCCACCTGACGTCTAAGAAA) which contains restriction enzyme site (underlined), ZraI, was designed and synthesized. The circular single-strand pUC19 was then incubated with this oligo and treated with ZraI at 37 for 45 min. After heat inactivation, the digested DNA was then purified and concentrated by ethanol precipitation and re-suspended in TE buffer as linear single-strand scaffold.

DNA Origami with DBS

The 2.4 kb DBS was synthesized and cloned into a plasmid commercially (Life Technologies, UK). To generate linear ssDNA, a PCR based method with 5' phosphorylated forward primer and biotinylated reverse 3' primer was used as in previous study [152]. The biotinlyated PCR product was captured by streptavidin coated magnetic beads. After the treatment with 0.2M NaOH, the (ssDNA) scaffold strand was released and subsequently neutralized by NH_4(OAc), ethanol precipitated and resuspended in TE buffer.

To generate a circular DBS scaffold, the linear ssDNA was ligated with Circligase (Epicentre, US) following manufacturer's instruction. The remaining linear ssDNA substrate and linear single-stranded adenylated intermediate was removed by treatment with E. coli exonuclease I and exonuclease III (NEB, UK). The circularized DBS ssDNA was then purified and eluted in TE buffer (Figure 4.7c).

For the assembly reaction, 20 nM ssDNA scaffold and 200 nM each staples oligos were mixed in a folding buffer containing 5 mM Tris, 5 mM NaCl, 1 mM EDTA (pH of 8) and 8 mM $MgCl_2$ (pH of 8). The reaction was heated to 95 for 30 second and cooled to 25 at the rate of 100 sec per degree in a thermal cycler.

Electrophoresis of the folded DNA was carried out in 2% agarose gel containing 0.5 μg/ml ethidium bromide and 0.5x TBE/Mg buffer (44.5 mM Tris base, 44.5 mM borix acid, 1mM EDTA, 11mM $MgCl_2$). The electrophoresis gels were run in 0.5x TBE/Mg buffer for 2 hours at 70 V in an ice/water-cooled tray. The DNA bands in gels were visualized using ultraviolet light and desired band was excised by scalpels. The DNA in excised gels was then extracted using Bio-Rad freezensqueeze column according to manufacturers instruction. The recovered material was then prepared for imaging.

DNA/RNA Hybrid Origami with DBS

The 1.1 kb DBS preceded by a T7 promoter was synthesised and cloned in a 14AA575P plasmid commercially (Life Technologies, UK). The DNA template-scaffold was obtained by PCR amplification with Phusion Hi-fidelity DNA Polymerase (NEB, UK). The RNA scaffold was synthesised using Ampliscribe T7-Flash Transcription kit (Epicentre) on the DNA scaffold template at 42 °C for 100 minutes. The scaffold was subsequently purified through a phenol-chloroform-isoamyl (125:24:1 Sigma Aldrich) and chloroform (Sigma Aldrich) precipitation. The concentration of the nucleic acids was evaluated by Nanodrop analysis (Thermo scientific).

Folding of the origami was carried out in TAE buffer (40 mM Tris, 4 mM Acetate, 1 mM EDTA) enriched with Magnesium acetate 12.5 mM. The reaction was performed using a concentration of 10 nM of RNA scaffold and 100 nM of DNA staples oligos. The folding solution was incubated for 10 minutes at 65 °C followed by a temperature ramp of 0.01 °C/s to 25 °C and maintained at that temperature for 5 minutes. The solution was then held at constant temperature of 4 °C to stop the reaction. The origami structures were purified through Amicon ultra filters 100 KDa to remove the excess of free staples and to concentrate the samples.

A.3 Scaffold and Staple Sequences

DBS scaffold for square DNA origami of length 2.4 knt.

```
TTTCTATGTCTGAGCCTGAAAATGCGATATATCGTCAGTCTCTGCGGCTGCCGAAACGCGCTCAACCTCTACCGTGGAAC
CACGGGAAAGGCAACCGAACCCTTTAAGGCTAATCGCGAGCCGGGTCCCTAAGACAGCGGGATTACCCGGGCCGCGTCAC
GCAGTCCTGTCTACTAAGCCTACAGTGTAAAGAGAGCCAAGAGGTCTCGTGTCATGGTCGCACGCCTGGTTGAGTCAGGC
TTAGACTCTTGCATCCCCAGCAATAAGTACATTGACGTGCCGTTCACGTACGTTTCCTGGACGCATGTGTGCGTAAGGTC
ATAGAAGCCGATCTCACCAAGCGCTTACAGAAGAGCTGGCGACACGGATGGCGGTATACCGATACCCCCATATAAAGTTC
GTATAAGGGCAGGAGTTACCTCGCGGTTCGGTGGCCGACGGCTCACTGGATGTATAGTCCCACTTCCTCAGATGCACATC
CTCGAAGACTTCTGTTCGCATTTTAGAAAACTAACAGCTCTCCAGCCGCCCAAGTTAAAACGACCCTGTTTGGTCAATGA
AGGTGGGAGTGCTTGCCGCAGGTAGCGAGGTACACTTACGCCGGACCAAATCTTTGGCCCGTGTATGGATCATCCATAGC
GCGAAGTGACACACTGCCCCACCTCATCTGACTACGGTAAGTGCGGATTCGGCATGGGGAACAAAGCTCATTGGATAGCT
GAATAGCCATACTGAGGATAAACACTAGGAATCGGGGGATATCCGTGAAGTTGACCATTACGGGCGCTACCATGACCGAG
GGATGACGAGATTTAGGCACGTTGTCCTACTTAACCCCTTGCGGTCGGACTTTCGCGTGCTCTAATGACTCGATTTGGGA
TCGTGGCGTTGGTGTAGAGCGTATTGGCACTGTTGCAATGTGAAATCGAACATGGAGACGTTAGATGAGTGTGATCCACG
TGAGCTTTGCAGACAAAACAATGGTGATACTTCGTTGCTCAGGTGAGGCATAAGATGGTACTTGCTTATCGCAGCCTTAA
AGCAGGGTCAGAGTCGGCTTCAGACCGGAAAAATTCAAAAGCGACTGTCGGTTATTCGCTCGCAATTATCTCGCTTTCAC
CTGTACCCAACAACGTATCTTCCCCGATTCACTTTAGCCGTGCGACGCTTGTCGATAACGCTATCCTGCACTTGAGAAAT
TAAACCAGCGAATCTATACTACTCGTAGCAGATTGCTGCGTTCGATCCCGGTGACCTAACGGAGCTACATCTAAGGAAGC
GTCCTTTTGGACTGACGGAATTAGCTATGACAATAGTAACCGGCTATTACACGATAGTGGTTAAGAGTGAACACGCGACC
GCGCCCGAGTGGAGTACCAGGCGCGGCGATTAAGTCTATTTATGGTTTCGACTATGCTCGGCCCTTAGGACTAGCATCTC
TCTTATTTTGCTAAATACAAGGGAGATCAGTGAGTTGCCTCTTCATAAATCACGAAGGGGCATTGCCCGGCACACAGCAT
TAGGTCCAGGACGACAAGAATCAGAATTGCGTCTAAAAGTAAGCACGGCGGGTGTCGCTAACCTGACATCGTTTTCTGCT
GAGTAGAATACTCAGTATATACATAATGGCAAATGAGCATATGGGAAGGATGCGGGGTAGTCACTAAACTTCACACCTAC
GCAAAGATCGACATGTTCAGTTATGCGTGTGTCCCGTCGCGCGCATCGAGTTTGCCAGGGAATAATCTGTCAGCGTTTGT
GTACGCGTTAACTATAGGTTCAATTTCCGTCTGTAAGAAACAGATAAGCGGTGCAAGACCTGGCTTGGCTACGAGTAATC
ATGAAAGTCGTAATGTCAAATAGAGTTCCTAGGGACTCATGCCTAGCCTCCCTGCGAGACTAATACGATTGTGACGCGGG
CTCGTCGGGTTAGCGGCCAACTTGGAAGTAGTTGTGGCATCAGGGCCACAAATTGAGCGATCGGTAGGAGCAAGGAGAAC
TTTGTCTCAGCTAAGTTTCAGGATTTTCCCTTCCGAGAGACACCCTCGGTCACCGACTTATACGCTGTCCGGTTTGAATG
TACTCTGAACGTCTCCTTCGCCAAAATCCGAAGCAAAAACCGCAAGTGTCTTCGGATACACATACGTGTTTCTTGTTTTG
TTACTATTCTCAAAGTGGCTGACCCACACGTCATCGGCGTCGTGCATTCCAAGGTTACGAACTAGAACAGTCGCCTATGG
CTCTGGAATGCAACAGGAAACTCACGGTTGGGGCGGCAGAGGCCCATGTCCAAAGGGTGAATTTTTAATCCCTCACATTC
TTCTTTCTCTAGGTAATAGGCTGGGTCGAAAAGGTATGCAGTAGGTGTGGATTGGTTCTGGCAGTTTTATAGACATTTGC
GAACGCCCCCTGGGCTGTGAGACCCGCGATGGGCAATCGTACCTATAACAAGCCAGAAAGAAGGCGGACATAGTTAGGGC
GAAA
```

Staple sequence	**Length**
TAGGAACTGAGGATGTGCATCTGAGCGAACAG	32
CTCAATTTGCAGGGAGGCTAGGCACTTTCATG	32
AAAACGATTCCCCCGATTCCTAGTGTCAACTT	32
CACGGATAGTCAGGTTAGCGACACCTCAGCAG	32
CTGGTACTCCACTCGGGCGCGTCCGTTAGGTCACCGGGATCG	42
GCAGAGACTATAGGTACGATTGCCCATCGCGGGTCTCACAGCCCAGGG	48
GTATCGGTATCCAGTGAGCCGTCGGGGCGGCT	32
TCCTGCCCTTATACGAACTTTGGAAACGTACGTGAACGGCAC	42
CTCCATGTCCTGAGCAACGAAGTACGCTTTTG	32
TGGGGCAGGCTATTCAGCTATCCAGTCATCCC	32
CCCGCATCCTTCCCATATGCTTCTTGTCGTCCTGGACCTAAT	42
TTAGTCTCGTGGCCCTGATGCCACCCGAGGGT	32
GCCACTTTCAGAGCCATAGGCGACAAGAAGAA	32
ACGTATGTAGACCTCTTGGCTCTCGTGCGACC	32
TTCACTCTGCATAGTCGAAACCATCTTCGTGA	32
GTAGGCTTGGTTCGGTTGCCTTTCCGGCAGCC	32
AAAGGACGGTTTAATTTCTCAAGTGCAGGATAGCGTTATCGACAAGCG	48
AGGACAACGTGCCTAAATCTCATGAGCTTTGTTCCCCATGCC	42
TGCGAGCGAATAACCGACAGTTCACCATTGTTTTGTCTGCAA	42
AACGCAGCAATCTGCTACGAGTAGTATAG	29
ATTACTCGAACGCTGACAGATTATCACGCATA	32
AAGTCTTCCTATTTGACATTACGATGAGTCCC	32
GGGCCTCTGAGGTTGAGCGCGTTTCCGTGGTT	32
AAACTCGACGGGCCAAAGATTTGGATGGATGA	32
GTCTCTCGATTTTGGCGAAGGAGAGTGGGTCA	32
CTTCTGTAAGTTCTCCTTGCTCCTAACTTAGC	32
GCTGTGTGCCGGGCAATGCCCAAATAGACTTAATCGCCGCGC	42
AATACGCTAAAATAAGAGAGATGCCTCCCTTG	32
CGTACACATAGCCAAGCCAGGTCTCAATCGTA	32
TGCTTCGGGAAGGGAAAATCCTGAACCGATCG	32
AGGGCCGATAACCACTATCGTGTATCAGTCCA	32
AGCTCACGTGGATCACACTCATCCGACCGCAAGGGGTTAAGT	42
ATGCCTCATCGATTTCACATTGCAATCCCAAA	32
AATTTTTCAAGATACGTTGTTGGGTACAGGTGAAAGCGAGATAAT	45
GCTGTCTTAGGGACCCGGCTCTTTCAGGCTCAGACATAGAAA	42

TGGCTTGTTGACGATATATCGCATGCGATTAG	32
TTACTATTGCTGCGATAAGCAAGTCTGACCCT	32
CCACGGTAGCCGCCCCAACCGTGATTGGACAT	32
AATGCACGACGCCGATGACGTCGTTCAGAGTACATTCAAACC	42
ATTCGCTGCTTCCTTAGATGTAGCGTCGCGTG	32
CAAACAGGGTCGTTTTAACTTGCCACCGAACCGCGAGGTAAC	42
CTTACTTTGAGGCAACTCACTGATTAGTCCTA	32
AAGTGTACGTTAGTTTTCTAAAATGGAAGTGG	32
TCAGTATGTGTGTCACTTCGCGCTTCCGGCGT	32
TCGGTCATTTAGAGCACGCGAAAGTCTAACGT	32
TCCATACATGCGCGCGACGGGACATCCCTGGC	32
TAACCCGACGAGCCCGCGTCATGCACCGCTTATCTGTTTCTT	42
TTTATGAATAGACGCAATTCTGATCATTTGCC	32
ACCCAGCCTATTACCTAGAGATGTTCTAGTTCGTAACCTTGG	42
GGACAGCGTATAAGTCGGTGAAACTACTTCCAAGTTGGCCGC	42
ATTATGTATGTCGATCTTTGCGTAAGTTAACG	32
TCGCACGGCTAAAGTGAATCGGGGCGGTCTGAAGCCGACTACCATCTT	48
ACAGACGGAAATTGAACCTATGGTGTGAAGTTTAGTGACTAC	42
TATTTAGCCTACACCAACGCCACGACAGTGCC	32
TGTGAGGGTGCCAGAACCAATCCACACCTACTGCATACCTTTTCG	45
GGCGTTCGCAAATGTCTATAAAACATTAAAAATTCACCCTGTTTCCTG	48
AGTCTAAGCGCACACATGCGTCCAATATGGGG	32
GACTATACATACCGCCATCCGTGTGGCTTCTA	32
TTTCGCCCTAACTATGTCCGCCTTCTTTC	29
ATGACACGGTATCCGAAGACACTTCAAGAAAC	32
TGAGACAAAGCGCTTGGTGAGATCCGCCAGCT	32
GGAGAGCTCTCGCTACCTGCGGCAAGATGAGG	32
TCGAGTCAGGTAGCGCCCGTAATGGTTTATCC	32
GAATCCGCACTTACCGTAGTCAGCACTCCCACCTTCATTGAC	42
TGACCTTACCTGACTCAACCAGGCTTTACACT	32
GTCAATGTACTTATTGCTGGGTGACGCGGCCCGGGTAATCCC	42
ACTGAACATATACTGAGTATTCTACCGCCGTG	32
GCTTTAAGGTCATAGCTAATTCCGATAGCCGG	32
TTGCATTCGAGAATAGTAACAAAAGCGGTTTT	32
CCTTAAAGAGTAGACAGGACTGCGGATGCAAG	32

Table A.1: Staple sequences for square DNA origami based on DBS scaffold.

pUC19 scaffold region (2.4 knt) used for folding the square DNA origami.

```
GGATCTCAACAGCGGTAAGATCCTTGAGAGTTTTCGCCCCGAAGAACGTTTTCCAATGATGAGCACTTTTAAAGTTCTGC
TATGTGGCGCGGTATTATCCCGTATTGACGCCGGGCAAGAGCAACTCGGTCGCCGCATACACTATTCTCAGAATGACTTG
GTTGAGTACTCACCAGTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCAGTGCTGCCATAACCAT
GAGTGATAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGAGCTAACCGCTTTTTTGCACAACATGG
GGGATCATGTAACTCGCCTTGATCGTTGGGAACCGGAGCTGAATGAAGCCATACCAAACGACGAGCGTGACACCACGATG
CCTGTAGCAATGGCAACAACGTTGCGCAAACTATTAACTGGCGAACTACTTACTCTAGCTTCCCGGCAACAATTAATAGA
CTGGATGGAGGCGGATAAAGTTGCAGGACCACTTCTGCGCTCGGCCCTTCCGGCTGGCTGGTTTATTGCTGATAAATCTG
GAGCCGGTGAGCGTGGGTCTCGCGGTATCATTGCAGCACTGGGGCCAGATGGTAAGCCCTCCCGTATCGTAGTTATCTAC
ACGACGGGGAGTCAGGCAACTATGGATGAACGAAATAGACAGATCGCTGAGATAGGTGCCTCACTGATTAAGCATTGGTA
ACTGTCAGACCAAGTTTACTCATATATACTTTAGATTGATTTAAAACTTCATTTTTAATTTAAAAGGATCTAGGTGAAGA
TCCTTTTTGATAATCTCATGACCAAAATCCCTTAACGTGAGTTTTCGTTCCACTGAGCGTCAGACCCCGTAGAAAAGATC
AAAGGATCTTCTTGAGATCCTTTTTTTCTGCGCGTAATCTGCTGCTTGCAAACAAAAAAACCACCGCTACCAGCGGTGGT
TTGTTTGCCGGATCAAGAGCTACCAACTCTTTTTCCGAAGGTAACTGGCTTCAGCAGAGCGCAGATACCAAATACTGTTC
TTCTAGTGTAGCCGTAGTTAGGCCACCACTTCAAGAACTCTGTAGCACCGCCTACATACCTCGCTCTGCTAATCCTGTTA
CCAGTGGCTGCTGCCAGTGGCGATAAGTCGTGTCTTACCGGGTTGGACTCAAGACGATAGTTACCGGATAAGGCGCAGCG
GTCGGGCTGAACGGGGGGTTCGTGCACACAGCCCAGCTTGGAGCGAACGACCTACACCGAACTGAGATACCTACAGCGTG
AGCTATGAGAAAGCGCCACGCTTCCCGAAGGGAGAAAGGCGGACAGGTATCCGGTAAGCGGCAGGGTCGGAACAGGAGAG
CGCACGAGGGAGCTTCCAGGGGGAAACGCCTGGTATCTTTATAGTCCTGTCGGGTTTCGCCACCTCTGACTTGAGCGTCG
ATTTTTGTGATGCTCGTCAGGGGGGCGGAGCCTATGGAAAAACGCCAGCAACGCGGCCTTTTTACGGTTCCTGGCCTTTT
GCTGGCCTTTTGCTCACATGTTCTTTCCTGCGTTATCCCCTGATTCTGTGGATAACCGTATTACCGCCTTTGAGTGAGCT
GATACCGCTCGCCGCAGCCGAACGACCGAGCGCAGCGAGTCAGTGAGCGAGGAAGCGGAAGAGCGCCCAATACGCAAACC
GCCTCTCCCCGCGCGTTGGCCGATTCATTAATGCAGCTGGCACGACAGGTTTCCCGACTGGAAAGCGGGCAGTGAGCGCA
ACGCAATTAATGTGAGTTAGCTCACTCATTAGGCACCCCAGGCTTTACACTTTATGCTTCCGGCTCGTATGTTGTGTGGA
ATTGTGAGCGGATAACAATTTCACACAGGAAACAGCTATGACCATGATTACGCCAAGCTTGCATGCCTGCAGGTCGACTC
TAGAGGATCCCCGGGTACCGAGCTCGAATTCACTGGCCGTCGTTTTACAACGTCGTGACTGGGAAAACCCTGGCGTTACC
CAACTTAATCGCCTTGCAGCACATCCCCCTTTCGCCAGCTGGCGTAATAGCGAAGAGGCCCGCACCGATCGCCCTTCCCA
ACAGTTGCGCAGCCTGAATGGCGAATGGCGCCTGATGCGGTATTTTCTCCTTACGCATCTGTGCGGTATTTCACACCGCA
TATGGTGCACTCTCAGTACAATCTGCTCTGATGCCGCATAGTTAAGCCAGCCCCGACACCCGCCAACACCCGCTGACGCG
CCCTGACGGGCTTGTCTGCTCCCGGCATCCGCTTACAGACAAGCTGTGACCGTCTCCGGGAGCTGCATGTGTCAGAGGTT
TTCACCGTCATCACCGAAACGCGCGAGAGTATTCAACATTTCCGTGTCGCCCTTATTCCCTTTTTTGCGGCATTTTGCCT
TCCTGTTTTTGCTCACCCAGAAACGCTGGTGAAAGTAAAAGATGCTGAAGATCAGTTGGGTGCACGAGTGGGTTACATCG
AACT
```

Staple sequence	**Length**

CACGACGTTGGCGTAATCATGGTCTCACAATT	32
TAAGACACGACTTATCGGTGGCCTAACTACGGCTCTGCTG	40
AAGCCCGTAGTGCACCATATGCGGATACCGCA	32
ACCATCTGGATAACTACGATACGGTCAATCTA	32
CGAGCGCAGATACCGCGAGACCCATCATCCAT	32
TCAGCAATAAACCAGCCAGCCAATAGTTTGCGCAACGTTGTT	42
CTTCACCTGTCTGACGCTCAGTGGGGTGGTTT	32
GATGCCGGACGGTCACAGCTTGTCTGAGCAAAAACAGGAA	40
AAGCCAGTGCAAGCAGCAGATTACTGATCTTT	32
GCTGCAATGAAGTGGTCCTGCAACGCCGGGAA	32
GGCAAAATGCCGCAAACATGCAGCTCCCGGAGGAGCAGAC	40
AAACAAACCACCGCTGGTAGCAACGAAAACTCACGTTAAGGG	42
GCTGCATTCCAGTCGGGAAACCTGTGAAATTG	32
AAGGCCAGGAACCGTAAAAAGTAAAGATACCAGGCGTTTCCC	42
CTGATCTTGCGAAAACTCTCAAGGTGCCCGGC	32
ACTTGGTCGACTCCCCGTCGTGTAGCCCCAGT	32
ATCCGTAAGCATAATTCTCTTACTCGATCAAG	32
TCGACGCTCCCCCCTGACGAGCATGCGGTAAT	32
GATCGGTGCGGGCCTCTTCGCGTGAATTCGAGCTCGGTACCC	42
CTCGTCGTAAGTAGTTCGCCAGTTGGAAGGGC	32
GCTTCCTCGCTCACTGACTCGATGTGAGCAAAAGGCCAGCAA	42
TCAGGGGATTTTCCATAGGCTCCGCAAGTCAG	32
AAGTATATAAAATGAAGTTTTAAATCTCAAGA	32
ATGCGGCGACCGAGTTGCTCTATCTTACCGCTGTTGAGATCC	42
GCTAGAGTTTGGTATGGCTTCATTCATGATCC	32
TGCGTTGCCATACGAGCCGGAAGCTGCAGGCA	32
TCCAAGCTGGGCTGTGTGCACGAACCCCC	29
TTATCCGCATAGCTGTTTCCTGTGGGGTAACG	32
CATCATTGACATAGCAGAACTTTAGTCATGCC	32
GGTCGTTCCGCGGGGAGAGGCGGTACATTAAT	32
GCGAGGTATGTAGGCGGTGCTGTTGGTAGCTCTTGATCCGGC	42
TCGCGCGTTTCGGTGATGACGGTGTTGGCGGGTGTCGGGGCT	42
TCTGTGACGGGATAATACCGCGCCGAAAACGT	32
TCAGCGATCTGTCTATTTCGTCGCTCACCGGCTCCAGATTTA	42
ACGGTTATATCAGCTCACTCAAAGTCGTGCCA	32
TATCTGCGCTACACTAGAAGAACACTTGAGTCCAACCCGG	40

GGGGATCCTCTAGAGTCGACCATAAAGTGTAAAGCCTGGGGT	42
GCCTAATGAGTGAGCTAACTCTTGCGTATTGGGCGCTCTTCC	42
CCAGGGTTGCAAGGCGATTAAGTTGATGCGTA	32
CAGCTGGCCATTCGCCATTCAGGCAGCAGATT	32
TGCAAGCTTGTAAAACGACGGCCATATTACGC	32
GTACTGAGCAGGGCGCGTCAGCGGGTGAAAAC	32
AGATCCTTGCGCAGAAAAAAAGGAGTATTTGG	32
AGTTGCCTTGACAGTTACCAATGCAAAAGGAT	32
AGTTCGATGTAACCCACTCGTGCACCCAA	29
TCTACGGGAGATCCTTTTAAATTAATGAGTAA	32
TCAGGCGCGAAAGGGGGATGTGCTTTCCCAGT	32
CCCATGTTTGGTTATGGCAGCACTGATGCTTT	32
TTTTGTTTTACCTTCGGAAAAAGAACAGAGTT	32
TCTTCGGGCAGCATCTTTTACTTTCACCAGCGTTTCTGGGAAAGTGCT	48
TAATTGTTTTTATCCGCCTCCATCGAGGGCTT	32
AAGCGTGGCCTGTCCGCCTTTCTCCACAAAAA	32
CTTGAAGTGCCACTGGCAGCAGCCACTGGTAACAGGATTAGCAGA	45
ATTTTGGTCATGAGATTATCATTAATCAGTGAGGCACCTATC	42
CCACACAAGCTCACTGCCCGCTTTAATGAATC	32
GCCATTGCTACAGGCATCGTGAGCTCCTTCGGTCCTCCGATC	42
GGCTTAACTATGCGGCATCAGTGCGCAACTGTTGGGAAGGGC	42
TCCGACCCAAACCCGACAGGACTAGCCGCGTT	32
CGTTCAGCACGCTGTAGGTATCTCTCTCCTGT	32
ATCACTCAGTGCAAAAAAGCGGTTGTGTCACG	32
CATAGCTCCCGACCGCTGCGCCTTATCCGGTAACTATCGTCCTTCGGG	48
GGCCAACGGGCTGCGGCGAGCGGTCCACAGAA	32
GTTGTCAGAAGTAAGTTGGCCAAGTCATTCTGAGAATAGTGT	42
GCGAGTTACAGCTCCGGTTCCCAACAGTCTAT	32
AGGAGAAATGTGAAATACCGCACATGTAAGCG	32
AGGTGGCGTGCCGCTTACCGGATACGCTTTCT	32
GCTGGCGTTAACGCAGGAAAGAACCTGCGCTC	32
CTCTGACAAAAGGGAATAAGGGCGACACGGAAATGTTGAATACTC	45
CCTGGAAGCTCCCTCGTGCGCAGTTCGGTGTAGGTCGTTCGC	42
GTCAATACTGGTGAGTACTCAACCGCAGTGTT	32

Table A.2: Staple sequences for square DNA origami based on pUC19 scaffold.

DBS scaffold for triangle RNA-DNA hybrid origami of length 1.1 knt.

```
GGCGCACGGTTCTGTGATCGTGGCGGTCCAGCTAGCAGGTTTGCGGCTCAGAAGAGCTGTTGTGTTTGTTTTCGACTACC
AGAACGGAGTCTCTAGCGTGAGATAAGTAAGATTAGGCTCGGAGAGTGTGAGGCTTCGTAATCGTACCACACACCAGGCG
TAACCGCACTTAGACGCACAGGGTACAAGTGATAGGTAAAGTTACGGCAGGACGCCCAAAAGTCTGGAGCACAAACGGGG
CCCCGCTAGGGAAAACGCCGGGGTAACTATTGTTATAATTCAAGAATTAGAACTAAAAGGTAGTAGCACCACTCGGTGGG
TTAAACTAGCTAAAGACACCGCTCCAACAGCCGAAAGTGTACGCTGAATCACAGTCAAATTATACGGTGTTCGAGATCGC
GAGTTTTGTGGGATTTGCACTCCAGATACCGATTCGGTAGCTTTATCGTTCACTGTGTCACGCGCAGCGCCACCAAAGCT
GAGACGTTCTCGAAATTCTAATTTCTACGATTAAGTCCAAACAGAAAGCAATCTATTACACTGGAAGTCAGTAAAACAAA
GGGATACAGATCCCGTGACGGCTAGTGCTGTGGTGTCCGAAGTTGACTGTCAGAGAACAATCGCACCGGACAGTTCGTTG
AGTTCCAGTTGCAATTGCGATAGTAATAATAGATAGAGGCCGTGGAACCCCGTACTTCAGCGAGAAGTGGTCTTGGACTT
GTACTGGGGCGAGCGGTGCGGGAACTCGTGTTGCCCGCAAGCACTGCAACACAGCGGAAGGATAGCAACGATCACTCTTG
CTTTGTCGGACTCAGTCTAGGAGCCGCCGAGCCAGTCCCGCGCGTTCCCACGTTTCCGTAAACGTCCGCTTGGCCCGTCC
ACTGATATAGTTGGATCGGGAGAAATCGAAGCTCACGAACAGGAACGTAAGGCTGCTTGTTCTTTCACGGATCTCGGGCA
GAATCTCAAACTCAATTACTCGATTTAGGTCGTCGCAGTACAGCTTCCACGGGCTTGAAATAGCTC
```

Staple sequence	**Length**
ACTTGTACCACAGAACCGTGTGCTTGCGGGCAACACCCTTCCGC	44
GCGTCCTGCCGTAACTTGTGCTCCAGACTTTTGGCTGCGACGACCTAAATCGAGTA	56
ACTTATCTCAGCTAGTTTAACCCACCGAGTGGTGCTACTACCCTCACACT	50
CTCCGAGCCTGGTAGTCGAAAACAAACACAACAGCTCTTCTTGGTACGA	49
TTACGAAGCTTTTAGTTCTAATTCTTGAATTATAACAATAGTGGTTACGC	50
CTGGTGTGGAGCCGCAAACCTGCTAGCTGGACCGCCACGATCCTGTGCG	49
TCTAAGTGCTACCCCGGCGTTTTCCCTAGCGGGGCCCCGTTTTACCTATC	50
TGTATCCCCCACAGCACTAGCCGTCACGCTAGAGACTCCGTTCTAATCTT	50
GCGTACACTTTCGGCTGTTGGAGCGGTGTCTTT	33
CGTCTCAGCGCGTGACACAGTGAACGATAAAGCTACCGAATCTCGTAGAA	50
ATTAGAATCGCAATTGCAACTGGAACTCAACGAACTGTCCGTTTCTGTT	49
TGGACTTAAGGTATCTGGAGTGCAAATCCCACAAAACTCGCGCAGTGTAA	50
TAGATTGCGTGCGATTGTTCTCTGACAGTCAACTTCGGACATTTGTTTT	49
ACTGACTTCATCTCGAACACCGTATAATTTGACTGTGATTCAACGGGATC	50
AAACGTGGGGCCTCTATCTATTATTACTATTTCGAGAA	38
GCGCTGCTTTGGTGAGCGGACCGGGCCA	28
TGTGTTGCAATTGAGTTTGAGATTCTGCCCGAGATCCGTGAAGAGTGATC	50
GTTGCTATGAGTTCCCGCACCGCTCGCCCCAGTACAAGTCCTGAGTCCG	49
ACAAAGCAAAGAACAAGCAGCCTTACGTTCCTGTTCGTGAGCTCGGCGGC	50
TCCTAGACAAGACCACTTCTCGCTGAAGTACGGGGTTCCACGAACGCGC	49
GGGACTGGCTTCGATTTCTCCCGATCCAACTATATCAGTGGAGTTTACGG	50

Table A.3: Staple sequences for triangle RNA-DNA hybrid origami based on DBS scaffold.

A.4 Secondary Structure of Scaffolds

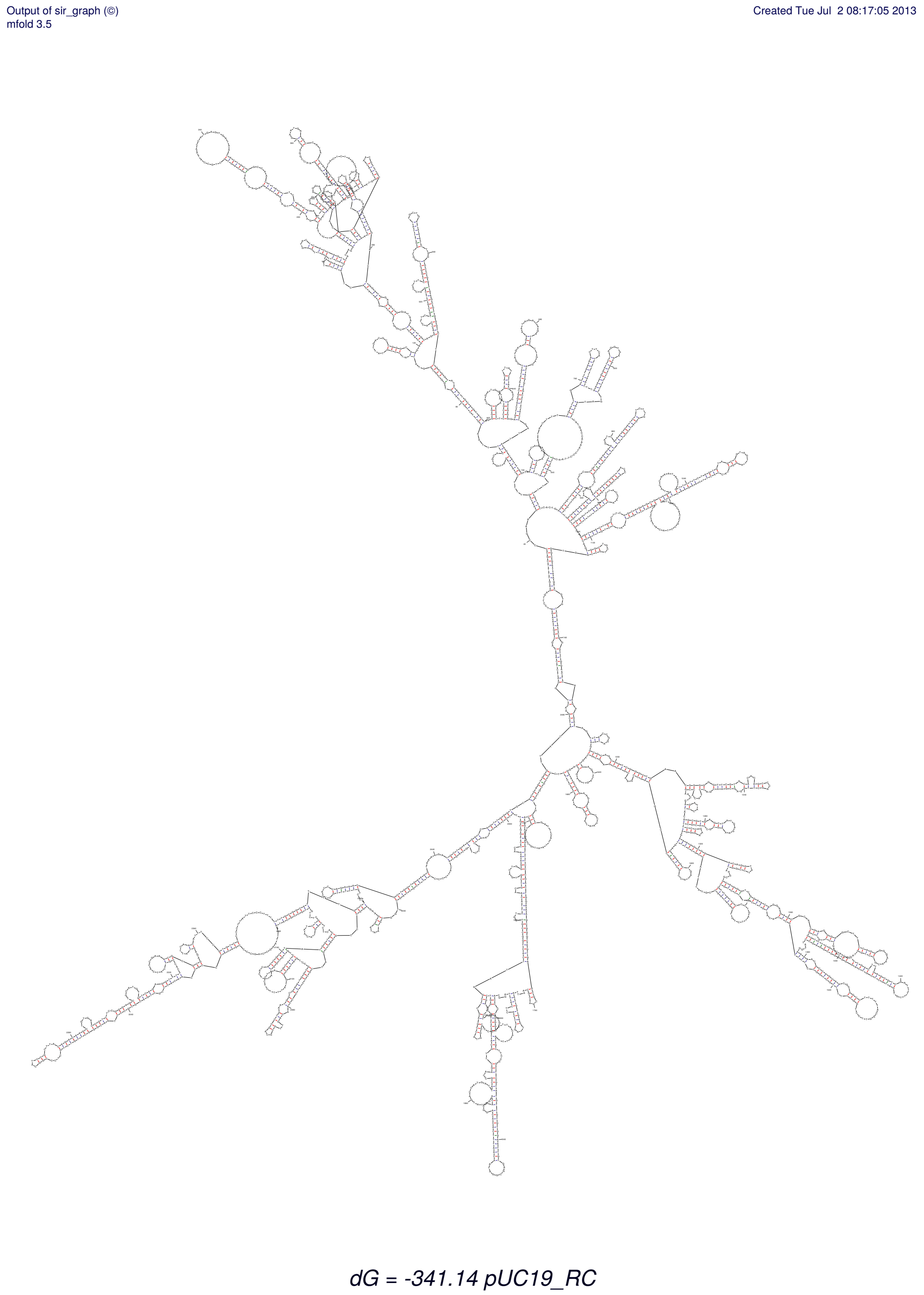


FIGURE A.3: Secondary structure of pUC19 scaffold (generated with UNAFold Web Server).

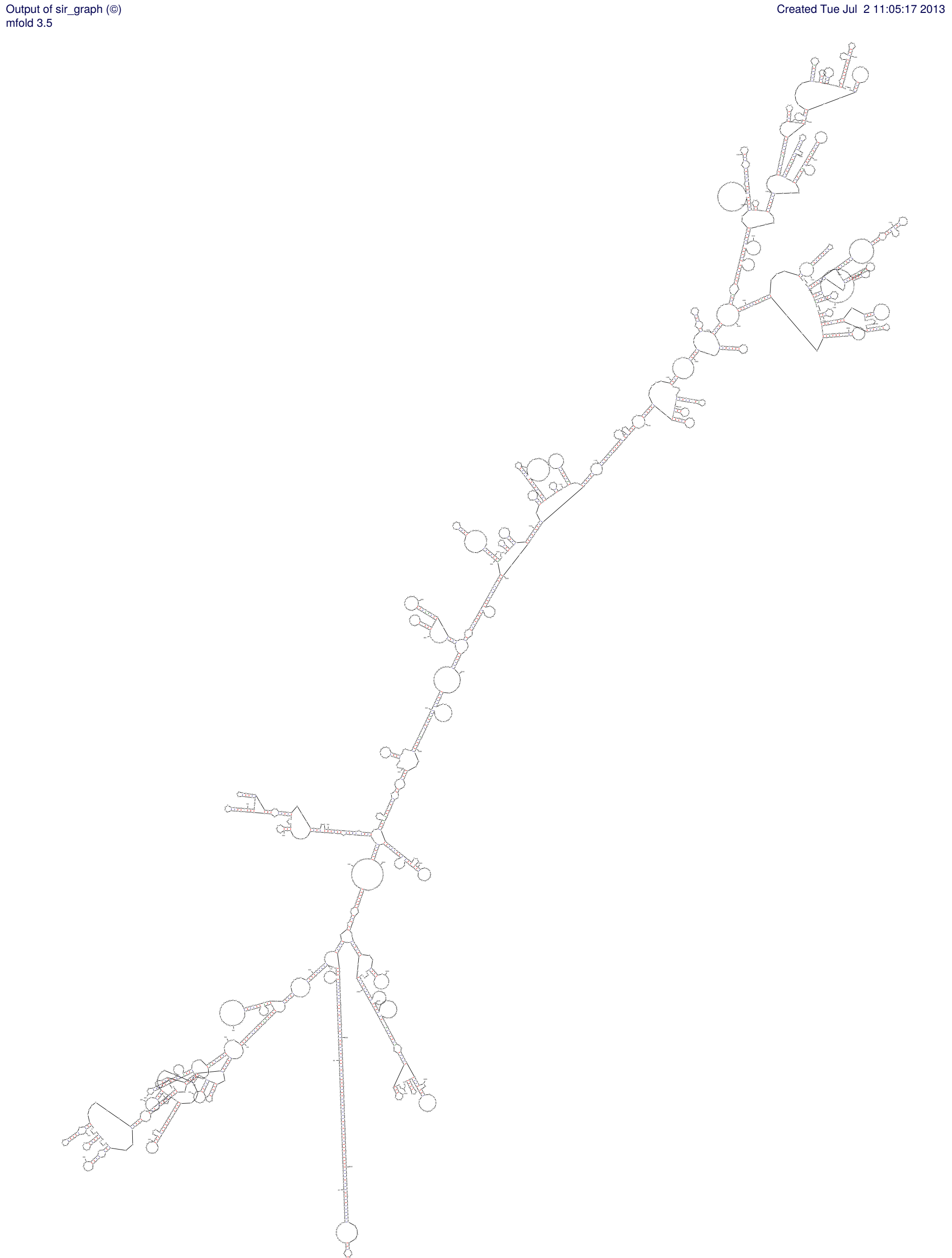


Figure A.4: Secondary structure of DBS scaffold (2.6 knt). Notice the self-complementary fragments forming the long stem near the beginning of the DBS scaffold. This fragment (making 5% of scaffold length) contributes roughly 25% to ΔG in the secondary structure (generated with UNAFold Web Server).

Appendix B

Appendix B: DNA stack

B.1 Full Set of Fitness Functions

function	**input** s	**function**	**input** s_1	**input** s_2
S_{hlf}	start	S_{ih}	start	push
S_{hlf}	push	S_{ih}	startpush	write$_x$
S_{hlf}	write$_x$	S_{ih}	startpush	write$_y$
S_{hlf}	write$_y$	S_{ih}	write$_x$	read
S_{hlf}	pop	S_{ih}	write$_y$	read

function	**input** s_1	**input** s_2
S_{eg+}	start	push
S_{eg+}	push	pop
S_{eg-}	start	start
S_{eg-}	push	push
S_{eg-}	read	read
S_{eg-}	pop	pop
S_{eg-}	write$_x$	write$_x$
S_{eg-}	write$_y$	write$_y$

function	**input** s_1	**input** s_2	**function**	**input** s_1	**input** s_2
S_{eg+}	startpush	write$_x$	S_{eg+}	startpush	write$_y$
S_{eg+}	write$_x$	push	S_{eg+}	write$_y$	push
S_{eg+}	write$_x$	read	S_{eg+}	write$_y$	read
S_{eg-}	write$_x$	report$_y$	S_{eg-}	write$_y$	report$_x$
S_{eg-}	report$_x$	start	S_{eg-}	report$_y$	start
S_{eg-}	report$_x$	push	S_{eg-}	report$_y$	push
S_{eg-}	report$_x$	read	S_{eg-}	report$_y$	read
S_{eg-}	report$_x$	pop	S_{eg-}	report$_y$	pop

Table B.1: Complete set of evaluator functions used to calculate the fitness of an individual.

B.2 oxDNA Simulation Parameters

CPU simulations

```
##############################
####  PROGRAM PARAMETERS  ####
##############################
backend = CPU
backend_precision = double

####    SIM PARAMETERS    ####
sim_type = MC
ensemble = NVT
steps = 1e9
check_energy_every = 1e4
check_energy_threshold = 1.e-4
delta_translation = 0.10
delta_rotation = 0.2
T = 23C
verlet_skin = 0.20

####    INPUT / OUTPUT    ####
topology = joined.top
conf_file = joined.dat
trajectory_file = trajectory.dat
no_stdout_energy = 0
restart_step_counter = 1
energy_file = energy.dat
print_conf_interval = 1e5
print_energy_every = 1e4
time_scale = linear
```

```
## External forces
external_forces = 1
external_forces_file = external.conf
```

CUDA simulations

```
##############################
####  PROGRAM PARAMETERS  ####
##############################
backend = CUDA
backend_precision = mixed
CUDA_list = verlet
CUDA_sort_every = 0
use_edge = 1
edge_n_forces = 1
seed = 19382
debug = 1

####    SIM PARAMETERS    ####
steps = 1e11
dt = 0.001
thermostat = john
diff_coeff = 0.5
newtonian_steps = 53
T = 295K
verlet_skin = 0.05

####    INPUT / OUTPUT    ####
conf_file = joined.dat
topology = joined.top
trajectory_file = trajectory.dat
energy_file = energy.dat
restart_step_counter = 1
refresh_vel = 1
print_conf_interval = 1e8
print_energy_every = 50000
time_scale = linear
timings_filename = timings_1_0.001
print_timings = yes
```

B.3 Laboratory Protocols

DNA oligomers were provided by Eurogentec (Belgium) on a 100 μM synthesis scale, with a standard desalting procedure or a required denaturing polyacrylamide gel electrophoresis (PAGE) purification for oligomers longer than 50 nucleotides and/or any 3'/5' modification. Streptavidin coated gold nanoparticles of 5 and 10 nm diameter were supplied by Life Technologies (Alexa Fluor 488 streptavidin). Samples and stock solutions were stored at -20 °C.

The DNA recorder was prepared by sequentially adding 200 nM of each brick with 240 minutes waiting time between additions. DNA samples were dissolved in a total volume of 20 μL of nuclease free water and 50 mM potassium acetate, 20 mM tris-acetate, 10 mM magnesium acetate, pH 7.9 buffer at room temperature (25 °C) and incubated for ten minutes if not otherwise specified. The mixture was shaken at 300 revolutions per minute in an Eppendorf Thermomixer Comfort set at 25 °C.

Capillary electrophoresis has been performed using the Agilent Technology 2100 Bioanalyzer system with its DNA High Sensitivity Chip and adhered to manufacturer protocols.

Transmission electron microscopy (TEM) was performed with a Philips CM 100 Compustage (FEI) microscope and digital images were collected using an AMT CCD camera (Deben). A volume of 5μL sample was applied on glow discharge grids preliminary washed with 0.5 mM magnesium chloride to change the hydrophilic surface charge orientation.