Design, Simulation, and Experimental Demonstration of Self-Assembled DNA Nanostructures and Motors

J. H. Reif [*], T. H. LaBean, S. Sahu, H. Yan and P. Yin

Abstract Self-assembly is the spontaneous self-ordering of substructures into superstructures driven by the selective affinity of the substructures. DNA provides a molecular scale material for programmable self-assembly, using the selective affinity of pairs of DNA strands to form DNA nanostructures. DNA self-assembly is the most advanced and versatile system that has been experimentally demonstrated for programmable construction of patterned systems on the molecular scale. The methodology of DNA self-assembly begins with the synthesis of single-strand DNA molecules that self-assemble into macromolecular building blocks called DNA tiles. These tiles have sticky ends that match the sticky ends of other DNA tiles, facilitating further assembly into larger structures known as DNA tiling lattices. In principle, DNA tiling assemblies can form any computable two or three-dimensional pattern, however complex, with the appropriate choice of the tiles' component DNA. Two-dimensional DNA tiling lattices composed of hundreds of thousands of tiles have been demonstrated experimentally. These assemblies can be used as scaffolding on which to position molecular electronics and robotics components with precision and specificity. This programmability renders the scaffolding have the patterning required for fabricating complex devices made of these components. We overview the evolution of DNA self-assembly techniques from pure theory, through simulation and design, and then to experimental practice. We will begin with an overview of theoretical models and algorithms for DNA lattice self-assembly. Then we describe our software for the simulation and design of DNA tiling assemblies and DNA nanomechanical devices. As an example, we discuss models and algorithms for the key problem of error control in DNA lattice self-assembly, as well as the computer simulation of these methods for error control. We will then briefly discuss our experimental laboratory demonstrations, including those using the designs derived by our software. These experimental demonstrations of DNA self-assemblies include the assembly of patterned objects at the molecular scale, the execution of molecular computations, and freely running autonomous DNA motors.

[*] Contact address: Department of Computer Science, Duke University, Box 90129, Durham, NC 27708-0129. E-mail: reif@cs.duke.edu.

1 Introduction

Self-assembly is the spontaneous self-ordering of substructures into superstructures driven by the selective affinity of the substructures. This paper focuses on a method for self-assembly known as *DNA self-assembly*, where DNA provides a molecular scale material for effecting this programmable self-assembly, using the selective affinity of pairs of DNA strands to form DNA nanostructures. Self-assembling nanostructures composed of DNA molecules offer great potential for bottom-up nanofabrication of materials and objects with smaller features than ever previously possible [13, 28, 33]. The methodology of DNA self-assembly begins with the synthesis of single-strand DNA molecules that self-assemble into macromolecular building blocks called DNA tiles. These tiles have sticky ends that match the sticky ends of other DNA tiles, facilitating further assembly into larger structures known as DNA tiling lattices. In principle, DNA tiling assemblies can be made to form any computable two- or three-dimensional pattern, however complex, with the appropriate choice of the tiles' component DNA.

DNA self-assembly is an emerging subfield of nanoscience with the development of its theoretical basis and a number of moderate to large-scale experimental demonstrations. Recent experimental results indicate that this technique is scalable. Periodic 2D DNA lattices have been successfully constructed with a variety of DNA tiles [14, 22, 43, 48]. These lattices are composed of up to hundreds of thousands of tiles. Molecular imaging devices such as atomic force microscopes and transmission electron microscopes allow visualization of these self-assembled two-dimensional DNA tiling lattices. These assemblies can be used as scaffolding on which to position molecular electronics and other components such as molecular sensors with precision and specificity. The programmability lets this scaffolding have the patterning required for fabricating complex devices made of these components. Potential applications of DNA self-assembly and scaffolding include nanoelectronics, biosensors, and programmable/autonomous molecular machines.

In addition to manufacturing DNA lattices, DNA has also been demonstrated to be a useful material for molecular computing systems [1, 3, 6, 19, 21] and mechanical devices [17, 20, 36, 50]. In particular, the self-assembly of DNA tiles can also be used as a powerful computational mechanism [15, 27, 39, 42], which in theory holds universal computing power [40]. See [25] for a more detailed survey of current experimental work in self-assembled DNA nanostructures. Also, see [26] and [28] for comprehensive surveys of the larger field of DNA computation (also known as biomolecular computation).

In this paper, we overview the evolution of DNA self-assembly techniques from pure theory, through simulation and design, and then to experimental practice. The rest of the paper is organized as follows. In Section 2, we overview the theoretical work in self-assembly. In Section 3, we describe software for the simulation and design of DNA nanostructures and motors. As a concrete example, in Section 4 we discuss error control, which we feel is a major theoretical and practical challenge remaining in the area of DNA self-assembly. Finally, in Section 5 we give a brief discussion of experimental practice in DNA nanostructures.

2 The Theory of Self-Assembly

This section overviews the emerging theory of self-assembly.

Domino Tiling Problems. The theoretical basis for self-assembly has its roots in *Domino Tiling Problems* (also known as Wang tilings) as defined by Wang [37]. For comprehensive text, see [9]. The input is a finite set of unit size square tiles. The sides of each square are labeled with symbols over a finite alphabet. Additional restrictions may include the initial placement of a subset of the these tiles, and the dimensions of the region where tiles must be placed. Assuming an arbitrarily large supply of each tile, the problem is to place the tiles, without rotation (a criterion that cannot apply to physical tiles), to completely fill the given region so that each pair of abutting tiles have identical symbols on their contacting sides.

Turing-universal and $\mathbb{NP}$-Complete Self-assemblies. Domino tiling problems over an infinite domain with only a constant number of tiles were first proved by Berger to be undecidable [7]. This and subsequent proofs [7, 29] rely on constructions where tiling patterns simulate single-tape Turing machines or cellular arrays [40]. Winfree later showed that computation by self-assembly is Turing-universal [40] and so tiling self-assemblies can theoretically provide arbitrarily complex assemblies even with a constant number of distinct tile types. Winfree also demonstrated various families of assemblies which can be viewed as computing languages from families of the Chomsky hierarchy [39]. It has been proved that Domino tiling problems over polynomial-size regions are $\mathbf{NP}$-complete [16]. Subsequently, [39], [10, 11], and [15] proposed the use of self-assembly processes (in the context of DNA tiling and nanostructures) to solve $\mathbf{NP}$-complete combinatorial search problems such as SAT and graph coloring.

Program-size Complexity of Tiling Self-assemblies. The programming of tiling assemblies is determined simply by the set of tiles, their pads, and sometimes the choice of the initial seed tile (a special tile from which the growth of the assembly starts). A basic issue is the number of distinct tile types required to produce a specified tile assembly. The *program size complexity* of a specified tiling is the number of distinct tiles (with replacement) to produce it. Rothemund and Winfree showed that the assembly of an $n \times n$ size square can be done using $\Theta(\log n / \log \log n)$ distinct tiles and that the largest square uniquely produced by a tiling of a given number of distinct tiles grows faster than any computable function [30]. Adleman recently gave program size complexity bounds for tree shaped assemblies [3].

Massively Parallel Computation by Tiling. Parallelism reveals itself in many ways in computation by self-assembly. Each superstructure may contain information representing a different calculation (*global parallelism*). Due to the extremely small size of DNA strands, as many as 10^{18} DNA tiling assemblies may be made simultaneously in a small test tube. Growth on each individual superstructure may also occur at many locations simultaneously via *local parallelism*. The *depth* of a tiling superstructure is the maximum number of self-assembly reactions experienced by any substructure (the depth of the graph of reaction events), and the *size* of a superstructure is the number of tiles it contains. Likewise we can define the number of layers for a superstructure. For example, a superstructure consisting of an array of $n \times m$ tiles, where $n > m$ has m layers. Tiling systems with low depth, small size, and few layers are considered desirable, motivating the search for efficient computations performed by such systems. Reif was the first to consider the parallel depth complexity of tiling assemblies and gave DNA self-assemblies of linear size and logarithmic depth for a number of fundamental problems (e.g., prefix computation, finite state automata simulation, and string fingerprinting, etc.) that form the basis for the design of many parallel algorithms [27]. Furthermore, [27] showed that these elementary operations can be combined to perform more complex computations, such as bitonic sorting and general circuit evaluation with polylog depth assemblies.

Linear Self-Assemblies. Tiling systems that produce only superstructures with k layers, for some constant k, are said to use *linear self-assembly*. [27] gave some simple linear tiling self-assemblies for integer addition as well as related operations (e.g., prefix XOR summing of n Boolean bits). Seeman's group demonstrated the first example of DNA computation using DNA tiling self-assembly [21], as described in Section 5. These linear tilings were refined in [44] to a class of String tilings that have been the basis for further DNA tiling experiments in [46] described in Section 5.

Kinetic Models of Tiling Self-assembly Processes. Domino tiling problems do not presume or require a specific process for tiling. Winfree first observed that self-assembly processes can be used for computation via the construction of DNA tiling lattices [38]. The sides of the tiles are assumed to have some methodology for selective affinity, which we call *pads*. Pads function as programmable binding domains, which hold together the tiles. Each pair of pads have specified binding strengths. The self-assembly process is initiated by a singleton tile (the *seed tile*) and proceeds by tiles binding together at their pads to form aggregates known as *tiling assemblies*. The preferential matching of tile pads facilitates the further assembly into tiling assemblies. Using the kinetic modeling techniques of physical chemistry, Winfree developed a kinetic model for the self-assembly of DNA tiles [41]. Following the classical literature of models for crystal assembly processes, Winfree considers assembly processes where the tiling assembly is only augmented by single tiles (known in crystallography as *monomers*) which bind to the assembly at their tile pads [38]. The likelihood of a particular tile binding at (or dissociating from) a particular site of the assembly is assumed to be a fixed probability dependent on that tile's concentration, the respective pad's binding affinity, and a temperature parameter. In addition, Adleman developed stochastic differential equation models for self-assembly of tiles and determined equilibrium probability distributions and convergence rates for some 1-dimensional self-assemblies [2, 4]. His model allowed for binding between subassemblies and assumed a fixed probability for tile binding events independent of the size of tile assemblies. Since the movement of tile assemblies may depend on their size (and thus mass), this model might in the future be refined to make the probability for tile binding events dependent on the size of tile assemblies.

Optimization of Tiling Assembly Processes. There are various techniques that may promote assembly processes in practice. One important technique is the tuning of the parameters (tile concentration, temperature, etc.) governing the kinetics of the process. Adleman considers the problem of determining tile concentrations for given assemblies and conjectures this problem is $\sharp\mathbf{P}$-complete [3]. Various other techniques may improve convergence rates to the intended assembly. A blockage of tiling assembly process can occur if an incorrect tile binds in an unintended location of the assembly. While such a tile may be dislodged by the kinetics of subsequent time steps, it still may slow down the convergence rate of the tiling assembly process to the intended final assembly. To reduce the possibility of blockages of tiling assembly processes, Reif proposed the use of distinct tile pads for distinct time steps during the assembly [27]. [27] also described the use of self-assembled tiling *nano-frames* to constrain the region of the tiling assemblies.

3 Simulation and Design Software

Software for Kinetic Simulation of Tiling Assembly Processes. Winfree developed software for discrete time simulation of the tiling assembly processes, using approximate probabilities for the insertion or removal of individual tiles from the assembly [41]. These simulations gave an approximation to the kinetics of self-assembly chemistry and provided some validation of the feasibility of tiling self-assembly processes. Using this software as a basis, our group developed an improved simulation software package (sped up by use of an improved method for computing on/off likelihood suggested by Winfree) with a Java interface for a number of example tilings, such as string tilings for integer addition and XOR computations. In spite of an extensive literature on the kinetics of the assembly of regular crystalline lattices, the fundamental thermodynamic and kinetic aspects of self-assembly of tiling assemblies are still not yet well understood. For example, the effect of distinct tile concentrations and different relative numbers of tiles is not yet known; probably it will require an application of Le Chatelier's principle.

Software for Kinetic Simulation of Nanomechanical Devices. We have developed a software to simulate autonomous nanomechanical DNA devices driven by ligase and restriction enzymes in a solution system. This software does discrete time simulation of the ligation and restriction events on the DNA duplex fragments of the nanomechanical device. The approximate probabilities of ligation is calculated based on the concentrations of individual DNA fragments present in the solution system. These simulations can provide insight to the kinetics of such nanomechanical systems. We have used this software to simulate a DNA walker and a universal DNA Turing machine.

Software for Design of DNA Lattices and Nanomechanical Devices. A major computational challenge in constructing DNA objects is to optimize the selection of DNA sequences so that the DNA strands can correctly assemble into desired DNA secondary structures. A commonly used software package, *Sequin*, was developed by Seeman, which uses the symmetry minimization algorithm. Sequin, though very useful, only provides a text-line interface and generally requires the user to step through the entire sequence selection process. Our lab recently developed a software package, *TileSoft*, which exploits an evolution algorithm and fully automates the sequence selection process. TileSoft also provides the user with a graphical user interface, on which DNA secondary structure and accompanying design constraints can be directly specified and the optimized sequence information can be pictorially displayed. TileSoft is initially designed to solve optimization problem for a set of multiple tiles, but can also be used to design individual DNA objects, such as DNA nanomechanical devices.

4 Error Control in DNA Tiling Assemblies

A chief challenge in DNA tiling self-assemblies is the control of assembly errors. This is particularly relavant to computational self-assemblies, which, with complex patterning at the molecular scale, are prone to a quite high rate of error, ranging from approximately between 0.5% to 5%, and the key barrier to large-scale experimental implementation of 2D computational DNA tilings exhibiting patterning is this significant error rate in the self-assembly process. The limitation and/or elimination of these errors in self-assembly is perhaps the single most important major challenge to nanostructure self-assembly.

There are a number of possible methods to decrease errors in DNA tilings:

(a) Annealing Temperature Optimization. This is a well known technique used in hybridization and also crystallization experiments. It can be used to decrease the defect rates at the expense of increased overall annealing time duration. In the context of DNA tiling lattices, the parameters for the temperature variation that minimize defects have not yet been determined.

(b) Error Control by Step-wise Assembly. Reif suggested the use of serial self-assembly to decrease errors in self-assembly [26].

(c) Error Control by Redundancy. There are a number of ways to introduce redundancy into a computational tiling assembly. In [24] we describe a simple method that can be developed for linear tiling assemblies: we replace each tile with a stack of three tiles executing the same function, and then add additional tiles that essentially 'vote' on the pad associations associated with these redundant tiles. This results in a tiling of increased complexity but still linear size. This error resistant design can easily be applied to the integer addition linear tiling described above, and similar redundancy methods may be applied to higher dimension tilings.

Work in 2003 by Winfree provided a method to decrease tiling self-assembly errors without decreasing the intrinsic error rate of assembling a single tile, however, his technique resulted in a final assembled structure that is four times the size of the original one [45].

Recently we have developed improved methods for compact error-resilient self-assembly of DNA tiling assemblies and analyzed them by probabilistic analysis, kinetic analysis and computer simulation; and plan to demonstrate these error-resilient self-assembly methods by a series of laboratory experiments. Our compact error-resilient tiling methods do not increase the size of the tiling assembly. They use 2-way overlay redundancy such that a single pad mismatch between a tile and its immediate neighbor forces at least one further pad mismatch between a pair of adjacent tiles in the neighborhood of this tile. Theoretical probabilistic analysis and empirical studies of the computer simulation of Sierpinsky Triangle tilings have been used to validate these error-resilient 2-way overlay redundancy tiling results; the analysis shows that the error rate is considerably reduced.

5 Experimental Progress

Self-assembled DNA Tiling Lattices. Seeman first pioneered DNA structure nanofabrication in the 1980s by assembling a multitude of DNA nanostructures (such as rings, cubes, and octahedrons) using DNA branched junctions [18, 31, 32]. However, these early DNA nanostructures were not very rigid. Later, rigid and stable DNA nanostructures (known as *tiles*) were developed. Typically they contain multiple DNA anti-parallel crossovers. Individual DNA tiles interact by annealing with other specific tiles via their ssDNA pads to self-assemble into desired superstructures known as *DNA tiling lattices*. These lattices can be either: *non-computational,* containing a fairly small number of distinct tile types in a periodic pattern; or *computational,* containing a larger number of tile types with more complicated association rules which perform computation during lattice assembly.

Periodic 2D DNA lattices have been successfully constructed with a variety of DNA tiles, for example, double-crossover (DX) DNA tiles [43], rhombus tiles [22], and triple-crossover (TX) tiles [14]. Our lab recently developed a “waffle”-like DNA lattice composed of a novel type of DNA tiles [48]. In addition, we have recently developed a new method for the assembly of aperiodic patterns [47]. This *directed nucleation assembly technique* uses a synthetic input DNA strand that encodes the required pattern, and specified tiles assemble around this input DNA strand, forming the required 1D or 2D lattice pattern.

The self-assembly of DNA tiles can also be used as a powerful computational mechanism [15, 27, 39, 42], and Winfree showed that two dimensional DNA tiling lattices can in theory be used to perform universal computation [39]. In collaboration with Seeman's lab, we have experimentally demonstrated a one-dimensional algorithmic self-assembly of triple-crossover DNA molecules (TX tiles), which performs a 4-step cumulative XOR computation [21]. In addition, we have recently demonstrated the first parallel computation with DNA tile self-assembly [46], exploiting the "string tile" model proposed by Winfree [44]. In our experimental implementation, the DX tiles, each encoding a whole entry of a bit wise XOR operation, associated with each other randomly in a parallel fashion, generating a molecular look-up table.

DNA Robotics. Existing DNA nanomechanical devices can exhibit motions such as open/close [34, 35, 50], extension/contraction [5, 8, 17], and rotation [20, 36]. These motions are mediated by external environmental changes such as the addition and removal of DNA fuel strands [5, 8, 17, 34, 35, 36, 50] or the change of ionic strength of the solution [20]. Our lab has recently constructed a robust sequence-dependent DNA nanomechanical actuator and have incorporated it into a 2D parallelogram DNA lattice [22]. The actuator can be switched reversibly between two states, mediated by the addition and removal of fuel DNA strands. In addition, we have achieved in our lab the construction of a unidirectional DNA walker that moves autonomously along a linear DNA track [49].

6 Conclusion

The self-assembly of DNA is a promising emerging method for molecular scale constructions and computations. We have overviewed the area of DNA tiling self-assemblies and noted a number of open problems. We have discussed the potential approaches for error-control in self-assembly techniques for DNA computation; particularly the use of error-resilient modified tiling methods. We have identified some technological impacts of DNA assemblies, such as using them as platform for constructing molecular electronic and robotic devices. Important future work includes further investigating potential broader technological impacts of DNA lattices. Many applications of DNA lattices rely on the development of appropriate attachment methods between DNA lattice and other nanoparticles, which itself is a key challenge in

DNA based nanoscience.

Acknowledgement

This work was supported by DARPA/AFSOR Contract F30602-01-2-0561, NSF ITR Grant EIA-0086015, DARPA/NSF Grant CCR-9725021, NSF QuBIC Grant EIA-0218376, and NSF QuBIC Grant EIA-0218359

References

[1] L. Adleman. *Science*, 266:1021, 1994.

[2] L. Adleman. Technical Report 00-722, USC, 2000.

[3] L. Adleman *et al.* In *34th ACM STOC*, page 23. ACM Press, 2002.

[4] L. Adleman *et al.* Linear self-assemblies: equilibria, entropy, and convergence rates. manuscript.

[5] P. Alberti and J. Mergny. *PNAS*, 100:1569, 2003.

[6] Y. Benenson. *et al. Nature*, 414:430, 2001.

[7] R. Berger. *Memoirs of the American Mathematical Society*, 66, 1966.

[8] L. Feng *et al. Angew. Int. Ed.*, 42:4342, 2003.

[9] S. Grunbaum *et al. Tilings and Patterns* H Freeman and Company.

[10] N. Jonoska *et al.* In *Proceedings of IEEE ICEC'97*, page 261, 1997.

[11] N. Jonoska *et al. Computing with Bio-Molecules, theory and experiments*, page 93, 1998.

[12] T. H. LaBean *et al.* In *DNA Based Computers 5*, page 123, 2000.

[13] T. H. LaBean. *Introduction to Self-Assembling DNA Nanostructures for Computation and Nanofabrication*. World Scientific, 2002.

[14] T. H. LaBean *et al. JACS*, 122:1848, 2000.

[15] M. G. Lagoudakis *et al.* In *DNA Based Computers 5*, page 141, 2000.

[16] H. Lewis *et al. Elements of the Theory of Computation*. Prentice-Hall.

[17] J. Li *et al. Nanoletter*, 2:315, 2002.

[18] F. Liu, *et al. Biochemistry*, 38:2832, 1999.

[19] Q. Liu *et al. Nature*, 403:175, 2000.

[20] C. Mao *et al. Nature*, 397:144, 1999.

[21] C. Mao *et al. Nature*, 407:493, 2000.

[22] C. Mao *et al.* *JACS*, 121:5437, 1999.

[23] J. H. Reif. *Biomol. Computing, New Generation Computing*, 2002.

[24] J. H. Reif. In *29th ICALP, Mlaga, Spain*, page 1, 2002.

[25] J. H. Reif *et al.* *Lecture Notes in Computer Science*, 2054:173, 2001.

[26] J. H. Reif. In *First International Conference on Unconventional Models of Computation, Auckland, New Zealand*, page 72, 1998.

[27] J. H. Reif. In *DNA Based Computers 3*, 1997.

[28] J. H. Reif. *special issue on Bio-Computation, Computer and Scientific Engineering Magazine, IEEE Computer Society*, page 32, 2002.

[29] R. Robinson. *Inventiones Mathematicae*, 12:177, 1971.

[30] P. W. K. Rothemund *et al.* In *the 32nd ACM STOC*, page 459, 2000.

[31] N. Seeman *et al.* *J. Vac. Sci. Technol.*, 12:4:1895, 1994.

[32] N. C. Seeman. *Angew. Chem. Int. Edn Engl.*, 37:3220, 1998.

[33] N. C. Seeman. *Nature*, 421:427, 2003.

[34] F. Simmel *et al.* *Physical Review E*, 63:041913, 2001.

[35] F. Simmel *et al.* *Applied Physics Letters*, 80:883, 2002.

[36] H. Y. *et al.* *Nature*, 415:62, 2002.

[37] H. Wang. *Bell Systems Technical Journal*, 40:1, 1961.

[38] E. Winfree. In *DNA Based Computers 1*, page 187, 1995.

[39] E. Winfree. In *DNA Based Computers 1*, page 199, 1995.

[40] E. Winfree, *et al.* In *DNA Based Computers 2*, page 191, 1996.

[41] E. Winfree. PhD thesis, California Institute of Technology, 1998.

[42] E. Winfree. Technical Report 1988.22, Caltech, 1998.

[43] E. Winfree *et al.* *Nature*, 394:539, 1998.

[44] E. Winfree *et al.* In *DNA Based Computers 6*, page63, 2000.

[45] E. Winfree *et al.* In *DNA Based Computers 9*, 2003.

[46] H. Yan *et al.* *JACS*, 125:47, 2003.

[47] H. Yan *et al.* *PNAS*, 100:8103, 2003.

[48] H. Yan *et al.* *Science*, 301:1882, 2003.

[49] P. Yin *et al.* 2004. In preparation.

[50] B. Yurke *et al.* *Nature*, 406:605, 2000.