

Simulating Self-replicating Patterns of DNA Tiles

Vinay K Gautam and Rajendra Prasath
Department of Computer and Information Science
The Norwegian University of Science and Technology
Sem Sælands Vei 7-9, NO - 7491, Trondheim, Norway
{vkgautam, rajendra.prasath}@idi.ntnu.no

ABSTRACT

This paper presents a simulation framework in which a pre-assembled rectangular pattern of DNA tiles can be put together with sets of other DNA tiles to autonomously assemble replicas of itself in a discrete two-dimensional grid. The simulator implements both abstract and chemical kinetics based modelling to simulate the tile pattern self-replication. While the abstract model uses only logical matching between the edges of tiles to guide the assembly process, the chemical kinetics model calculates stochastic preference for attachment and/or detachment of each tile during the self-replication. A comparison is made between pattern self-replication timing in the abstract model and cellular automata based models. Simulation of chemical kinetics behaviour shows that the physico-chemical parameters of tile self-assembly govern the tractability of self-replication process and reliability of replicating patterns. Observations are made about the limitations of the simulator, and a few suggestions for improvement and further studies are discussed.

CCS Concepts

•Computing methodologies → *Self-organization*;
Modeling and simulation;

Keywords

Algorithmically Programmable Pattern self-replication,
Simulation of self-replication, DNA self-assembly, DNA tile

1. INTRODUCTION

Self-replication is a fundamental mechanism in biology which has been used by nature to autonomously construct complex molecular systems inexpensively using evolution. Application of this inspiration to engineer artificial molecular systems has been a constant pursuit of nanosciences. Gunter von Kiedorowski [17] first introduced a minimal system of molecular self-replication, which

typically involves a three-step process. First, a template molecule assembles with few substrate molecules resulting in an intermediate complex formation. Second, the substrate molecules within the complex join together irreversibly by covalent binding, and thereby forming a replica of the template. Third, the complex molecule dissociates into two templates: the former template and the newly created replica. Each of these templates can reiterate the three-step process adding to the template population.

Template directed non-enzymatic self-replication has been used for the synthesis of nucleic acid sequences using only linear organization of short sequences of nucleic acids (primers). However, recent advances in structural DNA self-assembly have opened up perspectives for the non-enzymatic self-replication of two-dimensional (2-D) and three-dimensional (3-D) patterns of DNA [14, 18, 9, 2].

DNA tile self-assembly [19] is an emerging paradigm for nanostructure construction and molecular scale computation. DNA tiles [21], the building blocks of tile self-assembly, can be designed to interact with strength and specificity for the assembly of logically and/or algorithmically directed periodic and aperiodic 2-D intricate patterns. For a theoretical modelling of tile assembly, Erik Winfree first introduced an abstract Tile Assembly Model (aTAM) [19]. In the aTAM framework, the assembly starts from a single seed tile and the pattern grows in 2-D as more tiles adjoin one-by-one following a simple assembly rule — the total binding strength of an incumbent tile should be greater than or equal to a threshold value known as temperature parameter of assembly. However, DNA tile assembly is essentially a physico-chemical process, where local reaction temperature and tile concentration are the governing factors. Therefore, for a realistic modelling of tile assembly process, Winfree introduced kinetic Tile Assembly Model (kTAM) [19]. The kTAM considers each tile assembly step as a reversible process governed by the tile concentration, local reaction temperature and binding strengths of tiles. The model enables analysis of the assembly errors and growth rate for a given tile assembly system.

In this paper, we describe a simulation framework for minimal system of self-replicating patterns of DNA tiles, earlier studied in Gautam et al. [5]. The simulator produces replicas of 2-D target patterns in the tile self-assembly medium. There are two modes in which the simulator can be used: 1) abstract version of the simulator implements the aTAM [19], and 2) kinetic version

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
BICT 2017, March 15-16, Hoboken, United States

ISBN 978-1-63190-148-5

DOI: 10.4108/eai.22-3-2017.152414

Copyright © 2017 EAI

implements the kTAM [19].

The remainder of the article is structured as follows: background of DNA tile self-assembly and tile assembly models is described in Section 2. Section 3 describes the design of tile pattern self-replication simulator. In Section 4, we describe the details of underlying models used in the design of the self-replication simulator. Section 5 presents simulation results and observations based on these results. Section 6, concludes the article.

2. BACKGROUND

In this section we discuss briefly the background of the main concepts used in this article. This includes: a brief introduction to the self-assembly mechanism of DNA tile patterns, the abstract and kinetic modelling of tile assembly.

2.1 Self-assembly of Programmable DNA tile Patterns

Erik Winfree [19] introduced self-assembly of DNA molecular structures (DNA tiles [21]). DNA tiles serve as building blocks for programmable self-assembly of 2-D physical patterns of tiles. DNA tiles consist of four (≈ 50 nucleotide) ss-DNA molecules, synthesized for a given DNA tile design. Figure 1 illustrates the construction of a Double Crossover (DX) molecular DNA tile with four DNA strands. As shown in (a), each ss-DNA consists of a sequence of nucleotides (A, T, G, C). The tiles self-assemble through the bonding of these ss-DNAs at room temperature. The bonding process occurs when two complimentary strands meet and their base pairs: A-T and G-C, bind. Any left-over bases from each of the bonded strands form a sticky end(s) — as shown in (b). As the term implies, this end is available to "stick" or bond to another strand. DX molecular DNA tiles are square shaped structures where sticky-ends are represented by their respective square edges — as illustrated in (c).

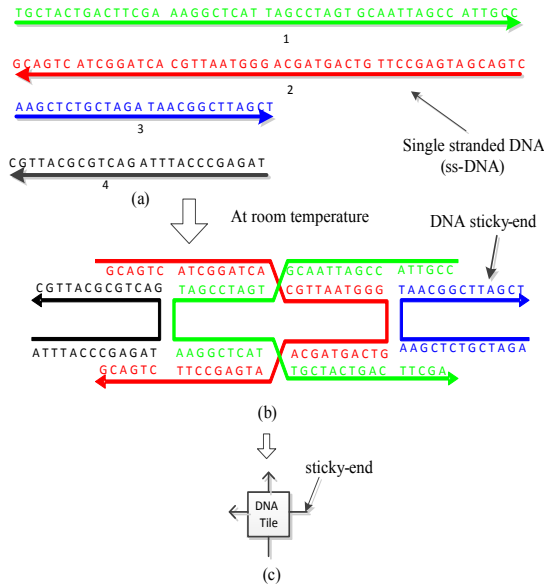


Figure 1: DX DNA Tile Structure(a) four ss-DNA (b) assembled DNA tile (c) abstract representation of DNA tile.

2.2 Tile Self-assembly Models

The physical implementation of tile self-assembly in a wet-lab is often time-consuming, expensive and challenging with respect to reproducibility of results. Computer simulations of tile self-assembly enable a faster way to design tile systems, study their performance and gain important insights without wet laboratory experiments. There are two simulation models of tile self-assembly, developed by Winfree [13, 19]: 1) The abstract Tile Assembly Model (aTAM), and 2) The kinetic Tile Assembly Model (kTAM).

Abstract Tile Assembly Model (aTAM)

In abstract model, a DNA-tile, t , is a quadruple $(\sigma_S(t), \sigma_W(t), \sigma_N(t), \sigma_E(t))$, where $\sigma \in \Sigma$ is glue type associated with the four sides (North (N), South (S), West (W), East (E)) of a rotationally asymmetric unit square. The glue type, Σ , is a finite set, which is used to derive a glue strength function $(s : \Sigma \times \Sigma \rightarrow \mathbb{N})$ for a legitimate tile association between two glues of tiles. The glue strength function is symmetric, i.e., $s(\sigma_1, \sigma_2) = s(\sigma_2, \sigma_1) \forall \sigma_1, \sigma_2 \in \Sigma$.

A tile pattern assembly system (TPAS) $\mathcal{T} = (T, S, s, \tau)$ consists of a finite set T of tile types, an assembly S termed as seed assembly, a glue strength function s and a temperature parameter $\tau \in \mathbb{Z}^+$. A tile assembly system has a temperature ' τ ' if any larger structure of tiles cannot be dissociated into smaller assemblies without breaking bonds of total strength at least ' τ '. Alternatively, a tile can join the assembly as long as the sum of the strengths of the bonds that it makes with tiles already in the assembly is at least τ .

Figure 2 illustrates an example of tile self-assembly the Sierpinski pattern [20, 12] at temperature 2 ($\tau = 2$). The tile set in (a) comprises a seed tile, two boundary tiles and four rule tiles. Tile edges are marked by non-negative integers illustrating their respective glue strengths. The South and West glues of the tiles are designed as inputs and the North and East glues are outputs.

Tile pattern assembly in the aTAM starts from a given seed structure that nucleates the pattern formation which further grows into a finite or infinite pattern as more tiles join, as shown in Figure 2(b). Tiles join by forming bonds with strength at least of τ (e.g., a $\tau = 2$ assembly requires tiles to bind with total strength at least 2). For a given TPAS, a pattern assembly P is said to be terminal, if no tile can be added further that satisfies the τ - stability criteria.

The aTAM has given insights to important theoretical aspects of the tile assembly systems, e.g, 1) what can or can't be self-assembled?, and 2) if something can be assembled, how efficient it could be?

The Kinetic Tile Assembly Model (kTAM)

The kTAM captures physico-chemical reality of the tile self-assembly process in its modelling, which was ignored in the aTAM. The kTAM considers each tile assembly step as a reversible process, governed by the tile concentration, local reaction temperature and the length of the tile's sticky ends. The rate of tile attachment at a binding site of a tile aggregate is directly proportional to the tile concentration. The concentration of each type of tile (except the seed tile) can be given by $e^{-G_{mc}}$, where G_{mc} is the decrease in entropy when a tile binds at a vacant site.

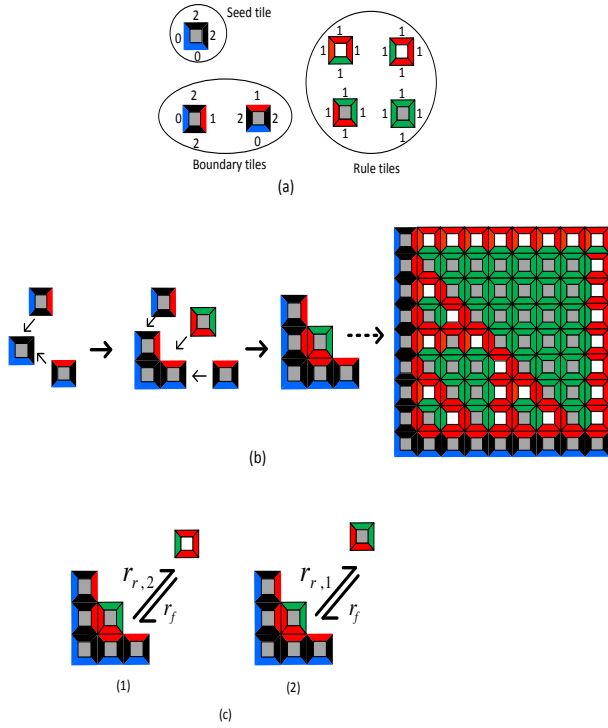


Figure 2: Sierpinski pattern self-assembly.(a) Sierpinski tile set (XOR tile set). (b) Steps of self-assembly of Sierpinski Pattern of size 9x9. (c) Kinetics of tile assembly in $kTAM$.

Therefore, the forward reaction rate (r_f) can be given by $r_f = k_f e^{-G_{mc}}$, where $k_f = 6 \times 10^5$ /M/sec is the reaction rate constant. Similarly, the tile detachment process is controlled by the energy required to break any single tile-aggregate bond and denoted by G_{se} . The value of G_{se} depends on the sticky end length (s) and the temperature (T), where $G_{se} \approx (4000/T - 11)s$. The tile reverse reaction rate involving b tile bonds is given by $r_{r,b} = k_f e^{-bG_{se}}$.

A larger value of G_{mc} thus implies a lower tile concentration and consequently a slower forward reaction rate (or vice versa). Similarly, a larger value of G_{se} results in a slower detachment rate. The optimum growth rate with low error rates happens near thermodynamic equilibrium ($G_{mc} \approx 2G_{se}$) [20], and may be given by $r^* \approx r_f - r_{r,2}$ and $\varepsilon \approx e^{-G_{se}}$, respectively. Therefore, a relation between optimum growth rate and minimum error rate may be given by $r^* \approx \beta \varepsilon^2$ where, $\beta = 0.75 \times 10^6$ /M/sec. Thus, any effort to reduce the error rate (ε) by tuning physical parameters (G_{mc} and G_{se}) would result in a quadratic reduction of the growth rate. Error reduction without significant fall off in assembly growth rate could be achieved using redundant tile sets [20, 7] or by protecting tile's inputs and outputs [4, 10, 6].

3. TILE PATTERN SELF-REPLICATION SIMULATOR

This section describes the design of Tile Pattern Self-replication Simulator (TPSS) based on its previously proposed theoretical model [5]. The simulator takes algorithmically programmable rectangular pattern of tiles

as an input and produces replicas of the pattern. At least one pre-assembled copy of a target pattern is required to start the self-replication process.

A target pattern is initialized in two steps: First, an L-shaped seed structure (West and South edges) of rectangular pattern is created using an XOR tile set. A corner tile is first chosen and then the two arms the L-shaped structure self-assemble in parallel as more tiles matching by single side join the growing seed structure. Second, using the seed structure, full pattern is formed as more tiles matching by two sides (or decided by the kinetics of the tile binding) assemble to the seed structure.

The simulator first assembles a mold of the target pattern using mold forming tile sets, and thus forming a combined Pattern-Mold structure (P-M). The P-M structure is dissociated into the mold and the pattern at the end of the cycle. In the subsequent replication cycles, the dissociated pattern and mold copies serve as seeds to drive the two pathways of the cross-coupled system, as illustrated schematically in Figure 3.

In the start of the pattern self-replication, cycle C0 shown in Figure 3, at least one copy of the pre-assembled pattern, P_0^0 , is supplied. The self-replicator produces a complex P-M, which is subsequently dissociated into new copies of P_1^0 and M_1^0 at the end of the cycle C1. In the subsequent replication cycle, C_i , 2^{i-1} copies of P and M are produced. In the C_i^{th} cycle, pattern copies are presented by P_i^j and mold copies are represented by M_i^j , where j varies from 0 to $(2^{i-1} - 1)$.

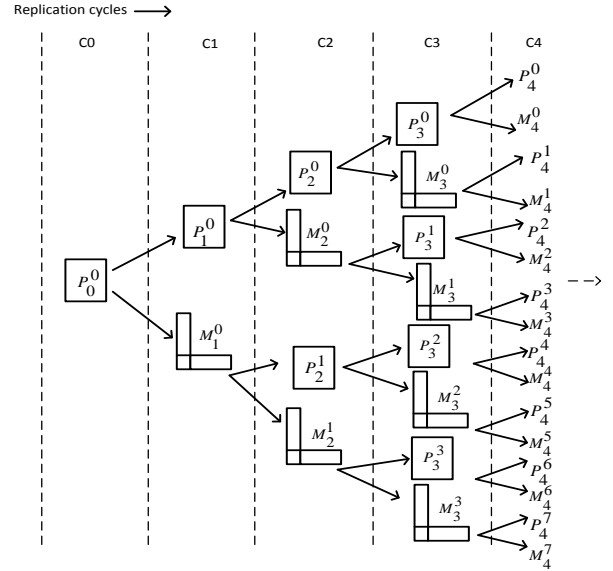


Figure 3: Tile pattern self-replication: unrolled cycles of cross-coupled system of tile pattern self-replication.

In the following, we describe: 1) pattern forming tile set; 2) mold forming tile set; 3) pattern-to-mold formation process ; 4) mold-to-pattern formation process.

3.1 Pattern Forming Tile Set

To assemble a pattern in the tile self-assembly framework, a tile set and an initial configuration of tiles (seed) is required. In the absence of seed, tiles do not assemble to produce large aggregates. However, the

presence of seed acts as a nucleation point where tiles join to grow the pattern. In the following, a pattern forming tile set, seed structure and pattern initialization are illustrated in brief.

A pattern forming XOR tile set is illustrated in Figure 4. A square tile unit having four edges (N, S, E, W) is shown in Figure 4(a). Input(ip)/output(op) edges and XORing logic operating implemented between them is shown in Figure 4(b) and Figure 4(c), respectively. The four tiles of XOR tile set used as Pattern forming tiles are shown in Figure 4(d).

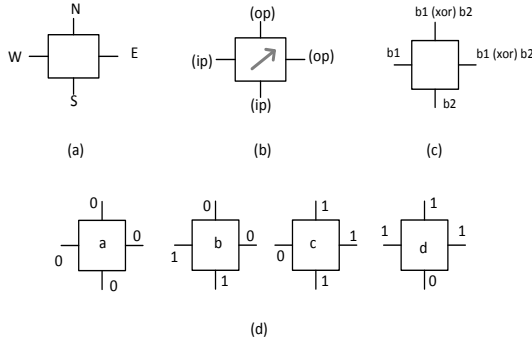


Figure 4: Pattern forming XOR tile set. (a) Square unit of tile. (b) Input(ip) and output(op) edges of an oriented tile unit. (c) XOR operation between inputs to derive output logic. (d) Four XOR tiles of the pattern forming tile set.

Programmable rectangular patterns of tiles can be produced using the pattern forming XOR tile set illustrated in Figure 4. A target rectangular pattern can be introduced by initializing an L-shaped seed from which the entire pattern is assembled. A number of L-shaped seed structures, as shown in Figure 5, can be initialized.

To initialize an L-shaped seed structure, first a random tile is picked and placed at the corner of the structure, and subsequently more tiles are added to form the two arms of the structure. To form the horizontal arm of the L-shaped structure, tiles are selected such that logic value on the West edge of each added tile matches with the logic on the East edge of the previous tile. Similarly, to form the vertical arm, tiles are selected such that logic value on the South edge of each added tile matches with the North edge of the previous tile.

Using an L-shaped seed structure corresponding tile pattern can be formed further using the tile assembly mechanism [19]. The pattern forms by filling tiles to the seed structure, where tiles are added by matching the logic at their input pins (left and bottom) with the available sites in the growing pattern.

3.2 Mold Forming Tile Set

We describe herein the sets of mold forming tiles. Based on the bottom and left borders of the pattern, the mold forming tiles are further divided into two sets of tiles: the horizontal mold forming tile set and vertical mold forming tile set, respectively. Mold forming tile sets implement the XOR logic, but the orientation of the tiles is different from the pattern forming tile set illustrated earlier in Figure 4. The horizontal mold forming tile set, Figure 6, uses tiles with

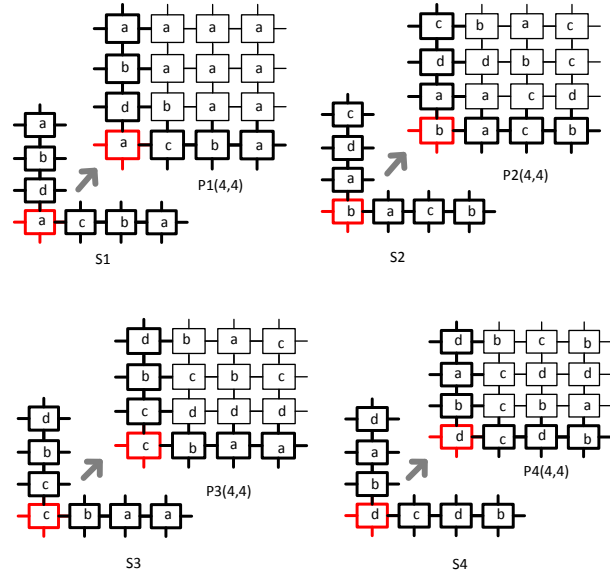


Figure 5: Initialization of L-shaped seed structures and tile patterns. Four different L-shaped seed structures (S1, S2, S3, S4) and corresponding tile patterns (P1(4,4), P2(4,4), P3(4,4), P4(4,4)) are shown.

top and right edges as inputs, and right and bottom edges as outputs. The vertical mold forming tile set, Figure 7, uses tiles with right and bottom edges as inputs, and left and top edges as outputs. The two tile sets form the horizontal and vertical arms of the mold, respectively, as shown in Figure 8.

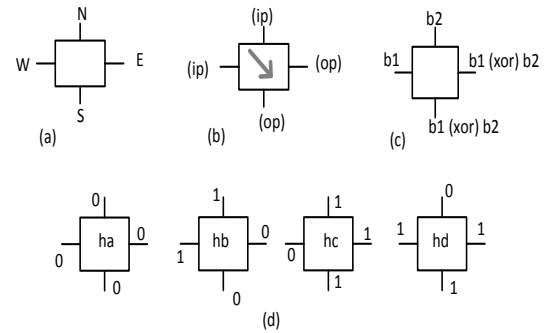


Figure 6: Horizontal mold forming tile set. (a) Square unit of tile. (b) Input(ip) and output(op) edges of an oriented tile unit. (c) XOR operation between inputs to derive output logic. (d) Four XOR tiles of pattern forming tile set.

3.3 Pattern-to-Mold Formation

Pattern-to-mold formation pathway in the cross-coupled system of pattern self-replicator assembles mold around the south-west boundary of the pattern. Mold formation, as shown in Figure 8, starts at the corner tile of the pattern, where mold forming tiles first bind, and subsequently the horizontal and vertical arms of the mold structure grow along the south and west edges of the pattern, respectively. Each of the horizontal and vertical edges of the mold are

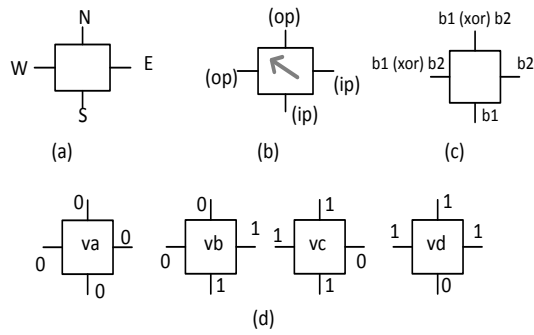


Figure 7: Vertical mold forming tile set. (a) Square unit of tile. (b) Input(ip) and output(op) edges of an oriented tile unit. (c) XOR operation between inputs to derive output logic. (d) Four XOR tiles of pattern forming tile set.

formed by tiles from separate tile sets: horizontal mold forming tile set and vertical mold forming tile set, as shown in Figure 6 and Figure 7, respectively. A tile in the horizontal arm of the mold attaches if its north and west edges have matching with the south edge of a tile in the pattern and east edge of the previous tile in the mold, respectively. Similarly, a tile in the vertical arm of the mold attaches if its east and south edges have matching with the west edge of a tile in the pattern and north edge of the previous tile, respectively.

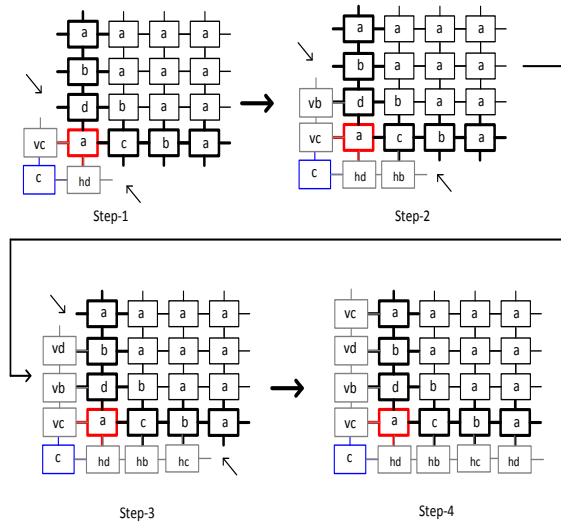


Figure 8: Pattern to mold formation using mold forming tile sets.

3.4 Mold-to-Pattern Formation

In each cycle, assembled pattern-mold complexes dissociate in mold and pattern structures. The dissociated mold structure serves as a seed in the mold-to-pattern formation process of the cross-coupled system of self-replication. Figure 9 illustrates a few steps of the mold to pattern formation process. Glues on the inner side of the mold structure provide information to self-assemble the

pattern using the pattern forming tiles. A tile attaches at a vacant site of the growing pattern if its south and west edges match with the north and east edges of the tiles in the vacant site, respectively. Once the filing of the rectangular pattern is complete, incoming tiles can no more attach by more than one bond, thus the pattern can not grow further: terminally assembled pattern.

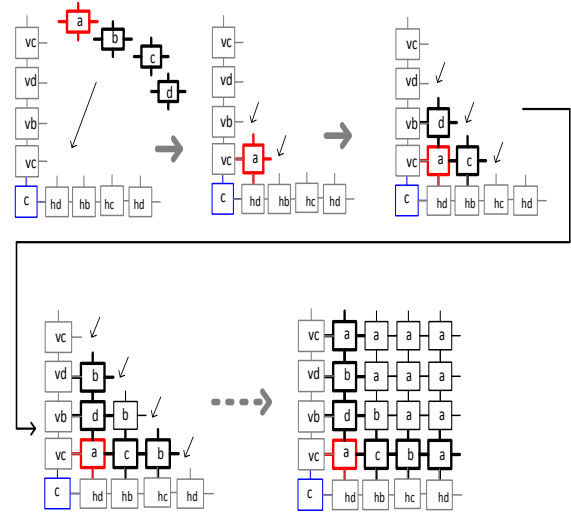


Figure 9: Formation of Pattern from Mold.

4. TILE PATTERN SELF-REPLICATION MODELS

A two dimensional rectangular grid of size $m \times n$, as shown in Figure 10, is used as a workbench of the simulator. Each cell in the grid is identified by coordinates (i, j) , where $0 \leq i \leq n$ and $0 \leq j \leq m$. The conventions used to represent each of the L-shaped seed, pattern, and mold are as follows. 1) for a $m \times n$ size pattern, cell $(m, 0)$ in the grid, is used for the corner tile of the L-shaped seed of the pattern. The L-shaped seed of the pattern occupies the left most and the bottom most cells of the grid. 2) mold of a $m \times n$ pattern occupies the left most and the bottom most cells of the grid of size $(m + 1) \times (n + 1)$. 3) we used data structure to hold dynamic data corresponding to patterns and molds of self-replication cycles (shown in Figure 3). The data structure initializes a separate grid of size $(m + 1) \times (n + 1)$ to perform processes — *seed* → *mold* formation and *mold* → *seed* formation — for every copy of seed and mold in each cycle.

The TPSS comes in two versions: 1) abstract Tile Pattern Self-replication Model (aTPSM) based on the the aTAM; kinetic Tile Pattern Self-replication Model (kTPSM) based on the kTAM. The two models used in the simulator design are described below.

4.1 Abstract Model

The aTPSM implements the pattern self-replication process in four steps: 1) initialization of two-dimensional rectangular pattern that is to be replicated; 2) pattern to mold formation; 3) mold to pattern formation; 4)

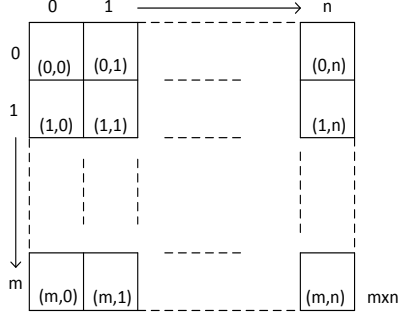


Figure 10: Two-dimensional grid of $m \times n$ dimension.

self-replication cycles: running the steps 2 and step 3 for each copy of the pattern and the mold in parallel. In the aTPSM, the criteria for a stable attachment of a tile is the same as in the aTAM, that is, each tile must have at least τ total binding strength as it attaches with a lattice site. Tiles attaching by a total binding strength $< \tau$ are rejected; there is no stochastic preference involved in the attachment and/or detachment of such tiles.

4.2 Kinetic Model

The kTPSM is based on the kTAM that was used in the tile assembly simulator “Xgrow” [1]. While the main steps of pattern self-replication in the kTPSM are the same as aforementioned, tiles assemble and disassemble stochastically in $mold \rightarrow pattern$ and $pattern \rightarrow mold$ formation processes. At each site (i, j) within the cluster of cells in the grid that would be occupied by tiles when aggregate formation is complete, the rates of association and dissociation of tiles are calculated using stochastic rules described below.

A 2-D array is used to store the arrangement of tiles in the current assembly. Initially (at time $t = 0$) the array contains all zeros to indicate empty sites, except for the cluster of sites (i, j) , which are occupied by the seed. At every simulation step ($t = \Delta t, 2\Delta t, \dots$), one of the following two events is stochastically chosen to simulate the reaction between an associating tile and an assembled aggregate: an ON-event, where a new tile is added to the array, and an OFF-event, where a tile in the assembly is removed.

ON-event rate: For every time step, all m empty sites adjacent to the aggregate are counted to calculate the net on rate (r_{on})

$$r_{on} = mk_f e^{-G_{se}} \quad (1)$$

OFF-event rate: For every time step, the off rate (r_{off}) is calculated as the following.

For all occupied sites (i, j) within the aggregate (except for the initial aggregate of the seed), the total binding strength (b_{ij}) of the tile in the occupied site is calculated. The net off rate (r_{off}) is calculated by summing up the dissociation rates of all the tiles occupying the aggregate. The net off rate is

$$r_{off} = \sum_b r_{off,b} \quad (2)$$

, where $r_{off,b} = n_b k_f e^{-bG_{se}}$ and n_b is the number of sites (i, j) having tiles attached by binding strength b . The kinetic rates of tile association, dissociation and the parameters, k_f , G_{mc} , G_{se} are described in Section 2.2.

The total rate for events of any kind is $r_{total} = r_{on} + r_{off}$, which is used to calculate both simulation time step and the type of the next event.

Simulation Time Step: time until the next event occurs, simulation time step (Δt), is chosen based on the Boltzmann distribution $P_r(\Delta t) = r_{total} e^{r_{total} \Delta t}$. Thus, Δt can be given by $\Delta t = -\ln \frac{(0,0, 1)}{r_{total}}$.

In each simulation step, on-event or off-event is chosen based on the following stochastic criteria. An on-event is chosen with the probability, $P_r(on) = r_{on}/r_{total}$, in this case, a tile is attached to a vacant site (i, j) adjacent to the aggregate. All the adjacent sites and all tile types are equally likely to be chosen. If on-event was not chosen, an off-event occurs, and the probability that some site (i, j) within the aggregate dissociates the tile attached to it. The probability that a site with b bonds dissociates is $r_{off,b}/r_{off}$, and again all such sites are equally likely. Once the event is chosen and the array is updated, all rates are calculated again to decide the next event.

To simulate the $mold \rightarrow pattern$ process, the L-shaped mold serves as seed and a 2-D array is used to store the status of pattern formation. For $pattern \rightarrow mold$ process, the seed consists of the entire pattern and the corner tile of the mold. The array used to store the status of mold formation consists of cluster of cells lying on the south most and the west most boundary of the 2-D grid.

5. SIMULATION RESULTS

The abstract model of self-replication produces 2^{i-1} copies of the target tile pattern, where i is the number of self-replication cycles. For a given target pattern of size $m \times n$, producing 2^{i-1} copies requires $t_s = \max(m, n) + (i-1) \times (m+n) + i$ time-steps. The abstract model produces time stamps for each tile addition step and total time to reproduce the 2^{i-1} copies of the target tile pattern. The self-replicated copies produced using the abstract model are perfect (containing no erroneous tiles) by design.

In the context of abstract tile pattern self-replication model, it is tempting to draw a comparison between the CA-based self-replication loops [15]. In particular, the parameters, such as loop sizes and replication periods of different CA based self-replicating loops resemble with the terms, pattern size and replication time of the abstract tile pattern self-replication model.

The closest self-replication loops to the tile pattern self-replication are Byl’s loop [3] and Chou-Reggia loop [11]. The Byl’s loop consists of 12 cells, and reproduces the loop in 25 time-steps. The cells in Byl’s loop can be in one of the 6 possible states and there are total 43 transition rules. The Chou-Reggia loop further reduces the loops size to 5 cells, each cell can be in one of the 8 possible states, and reproduces the loop in merely 15 time-steps. Considering the abstract model of tile pattern self-replication, which uses 12 types of tiles (considering each tile to be equivalent to a state in CA concept) and four transition rules ($00 \rightarrow 00, 11 \rightarrow 00, 01 \rightarrow 11, 10 \rightarrow 11$). Tile patterns of size 4×3 and 3×2 that are similar to the

sizes of Byl's loop and Chou-Reggia loop, can be reproduced in 11 and 8 time-steps, respectively. This margin in self-replication time of tile pattern self-replicator comes from the parallelism and asynchronism inherent in the tile self-assembly mechanism.

Another interesting attribute of the tile pattern self-replication is the programmability of target patterns: using the same set of tiles a larger pattern can be self-replicated by providing at least one copy of the pattern to start with. In contrast to the programmable nature of tile pattern self-replication, CA-based self-replicating loops are usually non-programmable and often implement a single dedicated functionality. Although Tempesti's loop [16] and Perrier's loop [8] introduce programmability in the self-replicating loops, large number of states (e.g., 63 states in the Perrier's loop) and increased complexity make them difficult for realization. A typical programmable self-replicating Tempesti's or Perrier's loop of size 150 cells requires ≈ 300 time-steps to reproduce itself, which is too large in comparison to the 41 time-steps needed to self-replicate a tile pattern of the similar size.

We performed simulations of pattern self-replication using the kTPSM for a range of G_{mc}, G_{se} values for both fixed size patterns and patterns of varying sizes. Figure 11 and Figure 12 report observations obtained from these simulations.

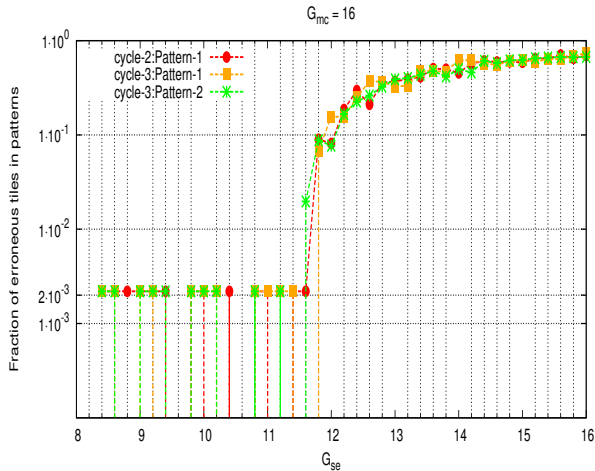


Figure 11: Assembly errors in self-replicating patterns produced by simulation using the kTPSM. Simulation statistics of assembly errors are collected by running the TPSS for 3 replication cycles for a 20x25 size target pattern. In these simulations, G_{mc} is fixed at 16 and G_{se} varies from 8.4 to 16 with an interval of 0.2.

In particular, the values of these parameters were chosen to cover the self-replication dynamics near melting ($G_{mc} \approx 2G_{se}$) transition. For a fixed G_{mc} ($G_{mc} = 16$), the self-replicator produces patterns with very few errors (1 tile out of 1000 may be faulty) for $8.4 \leq G_{se} \leq 11.4$, as can be observed from Figure 11. For $11.4 < G_{se} \leq 16$, errors are introduced in the self-replicating patterns. The errors grow swiftly as G_{se} approaches the G_{mc} . Simulation of self-replicating patterns of different sizes shows no

significant variation in the errors near melting transition, but near $G_{mc} \approx 1.40G_{se}$ fraction of erroneous tiles in a self-replicated pattern increases with its size, as shown in Figure 12. Errors in the patterns become very large, as G_{se} approaches the G_{mc} .

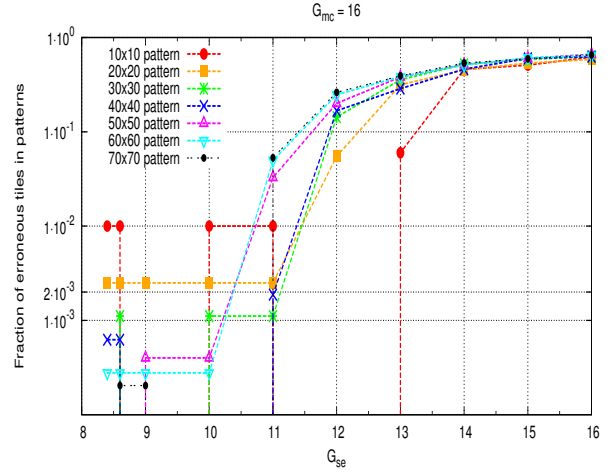


Figure 12: Simulation results of assembly errors in self-replicating patterns of different sizes. For each pattern size, the simulation parameters are: number of replication cycles = 2; $G_{mc} = 16$; G_{se} is varied from 8.4 to 16.

Time required and accuracy of the self-replication task in the kinetic model depends on the physico-chemical parameters (G_{mc} and G_{se}) of self-assembly. A phase diagram of the self-replication dynamics observed from the simulations using the kTPSM is illustrated in Figure 13. Pattern self-replicator works reliably in the region ($1.45G_{se} \leq G_{mc} \leq 1.9G_{se}$), shown in blue color. In the region $G_{mc} \leq 1.45G_{se}$, assembly errors get introduced in the self-replicating patterns (with G_{se} approaching G_{mc} , errors increase swiftly). For $G_{mc} \geq 1.90G_{se}$, the pattern replication becomes intractable, and therefore no patterns are produced.

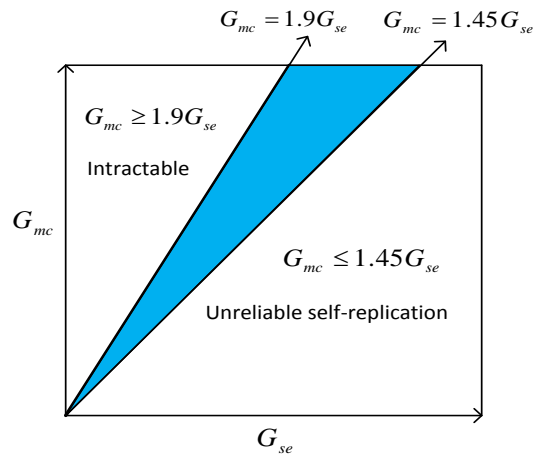


Figure 13: Phase diagram of tile pattern self-replicator dynamics using kinetic model.

6. CONCLUSION

We described a simulation framework of a minimal self-replication system of DNA-tile patterns. The self-replicator is based on a cross-coupled model, where both pattern copies and assembled mold copies serve as seeds to further assemble copies of each other. There are total three sets of tiles used in the design of the pattern self-replicator: one set of tiles self-assemble the target patterns using mold as a seed, and other two sets of tiles self-assemble molds of the patterns. Each set consists of four types of tiles, where each tile implements an XOR logic between its designated inputs and outputs.

The simulator is programmable, i.e. a user can initialize a target pattern for self-replication out of a variety of different patterns of the same size. A further programmability in the simulator can be added by extending it to include the self-replication of tile patterns that are assembled using a variety of other types of tiles, e.g. OR, AND, NAND, and XNOR tile sets. Currently, the simulator does not consider any chemical kinetics for the dissociation of pattern-mold complexes. Future work is expected to include this aspect into the self-replicator so as to further analyze its role in the dynamics of the pattern self-replication.

In addition to serving as a model for the self-replication of programmable DNA tile patterns in the DNA self-assembly medium, the simulator may pave ways to study the fundamental principles behind the evolution of biological life in which self-replication is a key concept. Such understanding may help us to design nature-inspired computation systems, overcoming the limitations posed to the existing bio-inspired techniques which presently abstract away from natural evolution.

Simulator source code:

A Java source code for the simulator is available at the link: <https://github.com/VKGAUTAM/TPSS>

7. REFERENCES

- [1] XGROW. <http://www.dna.caltech.edu/Xgrow/>.
- [2] Z. Abel, N. Benbernou, M. Damian, E. D. Demaine, M. L. Demaine, R. Y. Flatland, S. D. Kominers, and R. T. Schweller. Shape replication through self-assembly and rnaase enzymes. In *SODA*, pages 1045–1064. SIAM, 2010.
- [3] J. Byl. Self-reproduction in small cellular automata. *Physica D: Nonlinear Phenomena*, 34(1):295 – 299, 1989.
- [4] K. Fujibayashi, D. Y. Zhang, E. Winfree, and S. Murata. Error suppression mechanisms for DNA tile self-assembly and their simulation. *Natural Computing*, 8(3):589–612, 2009.
- [5] V. K. Gautam, E. Czeizler, P. C. Haddow, and M. Kuiper. Design of a minimal system for self-replication of rectangular patterns of DNA tiles. *Lecture Notes in Computer Science*, pages 119–133. Springer International Publishing, 2014.
- [6] V. K. Gautam, P. C. Haddow, and M. Kuiper. Reliable self-assembly by self-triggered activation of enveloped DNA tiles. volume 8273 of *Lecture Notes in Computer Science*, pages 68–79. Springer Berlin Heidelberg, 2013.
- [7] C. Ho-Lin and A. Goel. Error free self-assembly using error prone tiles. In *In DNA Computing 10, volume 3384 of LNCS*, pages 1–11, 2005.
- [8] J. Z. Jean-Yves Perrier, Moshe Sipper. Toward a viable, self-reproducing universal computer. *Physica D: Nonlinear Phenomena*, 97(4):335 – 352, 1996.
- [9] A. Keenan, R. T. Schweller, and X. Zhong. Exponential replication of patterns in the signal tile assembly model. In *DNA19*, volume 8141 of *Lecture Notes in Computer Science*, pages 118–132. Springer, 2013.
- [10] U. Majumder, T. H. Labean, and J. H. Reif. Activatable tiles: Compact, robust programmable assembly and other applications. In *in DNA Computing: DNA13*, pages 15–25. LNCS, 2007.
- [11] J. A. Reggia, S. L. Armentrout, H.-H. Chou, and Y. Peng. Simple systems that exhibit self-directed replication. *Science*, 259(5099):1282–1287, 1993.
- [12] P. W. Rothmund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA sierpinski triangles. *PLoS Biol*, 2(12):e424, 2004.
- [13] P. W. K. Rothmund and E. Winfree. The program-size complexity of self-assembled squares. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC’00, pages 459–468. ACM, 2000.
- [14] R. Schulman. Self-replication and evolution of DNA crystals. In *Advances in Artificial Life: 8th European Conference (ECAL)*, volume *LNCS 3630*, pages 734–743. Springer-Verlag, 2005.
- [15] M. Sipper. Fifty years of research on self-replication: An overview. *Artif. Life*, 4(3):237–257, 1998.
- [16] G. Tempesti. A new self-reproducing cellular automaton capable of construction and computation. In *in ECAL95: Proceedings of the Third European Conference on Artificial*, pages 555–563. Springer-Verlag, 1995.
- [17] G. von Kiedrowski. A self-replicating hexadeoxynucleotide. *Angewandte Chemie International Edition in English*, 25(10):932–935, 1986.
- [18] T. Wang, R. Sha, R. Dreyfus, M. E. Leunissen, C. Maass, D. J. Pine, P. M. Chaikin, and N. C. Seeman. Self-replication of information-bearing nanoscale patterns. *Nature*, 478(7368):225–228, 2011.
- [19] E. Winfree. *Algorithmic Self-Assembly of DNA*. PhD thesis, California Institute of Technology Pasadena, California, USA, 1998.
- [20] E. Winfree and R. Bekbolatov. Proofreading tile sets: Error correction for algorithmic self-assembly. In *In DNA Based Computers 9, volume 2943 of LNCS*, pages 126–144, 2004.
- [21] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394(6693):539–44, 1998.