Computational modeling of the kinetic Tile Assembly Model using a rule-based approach

Abdulmelik Mohammed\textsuperscript{1}, Elena Czeizler\textsuperscript{2}, Eugen Czeizler\textsuperscript{3}

\textsuperscript{1}Dept. of Computer Science, Aalto University, Espoo, Finland
\textsuperscript{2}Citrus Solutions Oy, Helsinki, Finland
\textsuperscript{3}The National Institute for Research and Development in Biological Sciences, Bucharest, Romania
\textsuperscript{4}Dept. of Computer Science, Åbo Akademi University, Turku, Finland

Abstract

The Tile Assembly Model (TAM), is a mathematical paradigm for the study and algorithmic design of DNA self-assembly systems. It employs the use of so-called DNA-tiles, which are abstractions of experimentally achievable complexes with similar inter-matching behaviour. To this day, there are more than 5 different experimental implementations of DNA tiles and their subsequent algorithmic assembly into larger complexes. In order to provide further insight into the assembly process, the abstract model has been extended to a kinetic counterpart (kTAM). Although there is a wide abundance of different variants of the abstract model, e.g., stage, step, hierarchical, temperature-k, signal-passing, etc., numerical simulations of the kinetic counterpart have been performed only for a few types of systems. This might be due to the fact that the numerical models and simulations of kTAM were almost exclusively implemented using classical stochastic simulation algorithms frameworks. In this paper we introduce an agent- and rule-based modeling approach for the kTAM. We show not only how the modelling of kTAM can be implemented, but we also explore the advantages of this modelling framework for kinetic simulations of kTAM and the easy way such models can be updated and modified. We present numerical comparisons both with classical numerical simulations of kTAM, as well as comparison in between four different kinetic variant of the TAM model, all implemented within stand-alone rule-based models.

1. Introduction

Recent advances in DNA-based nano-technology have opened the way towards the systematic engineering of inexpensive, nucleic-acid based nano-scale devices for a multitude of purposes [17, 19, 20, 21]. The field evolved dramatically in the past 10-15 years from a state where careful manual design and intimate knowledge of DNA’s atomic structure were needed for the design of
simple structures [25], to the current algorithmic approaches employing the use of universally engineered elementary building blocks that are further functionalyzed and driven to self-assemble into the desired shapes, see [13, 29, 2, 1].

The Tile Assembly Model (TAM), which is one of the theoretical design platforms exploiting such a modular assembly scheme, employs the use of so-called DNA-tiles, which can be seen as unit square blocks with active glues on their four edges (North, East, South and West). These active glues, implemented using single-stranded DNA sticky-ends, are driving the self-assembly process and determine the controlled aggregation of the tiles into the desired structures.

Since its introduction [22], the TAM formalism has been used successfully both in designing complex assembly nano-structures and in providing predictions regarding the possible experimental outcomes of certain designs. To this extent, the Tile Assembly Model has been expanded to a kinetic counterpart, kTAM [28]. kTAM incorporates two types of reactions: association of tiles to an assembly (forward reaction) and dissociation (reverse reaction), see e.g. Figure 2a). In a forward reaction, any tile can attach to the assembly at any position, even if only a weak bond is formed; the rate of this reaction, \( r_f \), is proportional to the concentration of free tiles in the solution. In the second reaction, any tile can detach from the assembly with rate \( r_r \), which depends exponentially on the total number \( b \) of the bonds between the tile and the assembly, as well as the strength of these bonds. Thus, tiles which are connected to the assembly by fewer or weaker bonds are more prone to dissociation than those which are strongly connected. Previous computational modeling of kTAM has been performed almost exclusively using a special tailored version of the Gillespie’s stochastic simulation algorithm [3, 11, 30]. The assembly starts at \( t = 0 \) from a seed structure. Then, in discrete time steps, tiles are added or detached from the assembly, according to the corresponding association and dissociation rates.

In our work, we propose a rule-based modelling approach for predicting the time evolution of kTAM systems and their variants. Rule-based modelling is a discrete modeling technique [9] in which molecules are represented as agents with a finite number of free sites. The sites allow for agent-agent binding, thus generating molecular complexes. Rules are defined based on local patterns rather than by the full specification of the reactants, and thus provide a compact representation on how agents interact. Thus, rather than handling explicitly a large number of model variables, within this framework we only have a small number of local interaction rules. This makes the rule-based paradigm well suited in handling the problem of the combinatorial explosion of the state space, as is the case of modelling self-assembly or polymerization systems, see e.g., [27].

In the next section we briefly introduce the abstract Tile Assembly Model, aTAM, and its kinetic counterpart, kTAM. In Section 3 we describe the concept of agent- and rule-based modeling, some general notions about this framework, as well as some of the advantages of agent- and rule-based computational modeling. In Section 4 we give a rule-based model for the kTAM and compare its predictions to other available documented simulations of the kTAM. Also, we detail on the advantages of this model, both from a computational point of view, and from the designer point of view. Inspired by the general principle of kinetic
modeling of chemical reactions, in Section 5 we introduce some modifications in
the kinetic modeling level in the kTAM model. Then, using the new rule-based
computational model, we analyze how this impacts the time evolution of the
assembly in terms of assembly growth and error-fraction. Finally, in Section 6
we conclude our investigation.

2. The abstract Tile Assembly Model, and its kinetic counterpart

The abstract Tile Assembly Model (aTAM) [22, 28] is a generalization of
Wang tile systems, customarily designed for the study of self-assembly systems.
The basic components of the aTAM are non-rotatable unit square tiles, uniquely
defined by the sets of four glues placed on top of their edges. The glues are
part of a finite alphabet and each pair of glues is associated a strength value,
determining the stability of a link between two tiles having these glues on the
abutting edges. In most cases, it is assumed that the strength of two distinct
glues is zero, while a pair of matching glues has strength either 1 or 2.

Let \( \Sigma \) be the alphabet of glues. A tile type \( t \) is uniquely determined by the
ordered set \((g_1, g_2, g_3, g_4) \in \Sigma^4\) of glues placed on its North, East, South, and
West edges, respectively. A tile system \( T \subseteq \Sigma^4 \) is a finite collection of different
tile types. Let \( \rho : \Sigma^2 \rightarrow \mathbb{N} \) be the strength function. Unless otherwise specified,
we assume that for all \( g, g_1, g_2 \in \Sigma, g_1 \neq g_2 \), \( \rho(g_1, g_2) = 0 \), while \( \rho(g, g) \in \{1, 2\} \).

Given a tile system \( T \), an assembly \( A \) is a partial mapping \( A : \mathbb{Z}^2 \rightarrow T \)
assigning tiles to various elements from the two dimensional space. For each tile
in an assembly, the strength of its binding is given by the sum of all strength
values of the pairs of glues placed along the boundary between the tile and
the assembly. A Tile Assembly System (TAS) is a structure \((T, S, \rho, \tau)\), where
\( T \) is a tile system, \( S \) is an assembly called the seed structure, \( \rho \) is a strength
function, and \( \tau \) is the temperature threshold for the assembly. Given an existing
assembly \( A \), such as the seed structure \( S \), a tile can adjoin the assembly if its
total strength of binding surpasses the temperature threshold \( \tau \). In Figure 1
we present a TAS with 21 tile types and temperature \( \tau = 2 \) which, starting
from the seed tile, assembles a continuously growing structure corresponding
to the evolution of the elementary cellular automaton “rule 30” starting from
the single-1 initial configuration, i.e., the configuration \( ...0001000... \). Out of
the 21 tile types in Figure 1, one can distinguish the tile used as seed (point d.
in the figure), 4 tile types which assemble the zero boundary of the structure
(point c. in the figure), and 16 rule-tile types, which fill the area in between
the \( V \) shaped boundary (point d. in the figure). In a similar manner, one can
consider any radius-half cellular automaton with \( n \) states and construct a TAS
with \( O(n^2) \) tile types whose assembly simulates the evolution of the cellular
automaton starting from a finite configuration (i.e., a configuration with finite
non-zero entries). In particular, rule 110 elementary cellular automaton, which
is Turing universal [4], can also be assembled using a TAS with only 16 tile types
(plus seed tile, plus tile types forming the boundary V-shape of the structure).

The aTAM is a good formalization for crystal growth, incorporating the
algorithmic assembly principles of these structures. Such theoretical framework
is essential for the algorithmic design of assembling strategies for patterns and structures. The problem of finding minimal tile system for the unique assembly of a given pattern, PATS [15, 6], or the task of assessing the possible complexity of patterns assembled using specific tile types, [22], are only some of the important tasks which are most suitable to be addressed at this level. However, when analyzing, simulating, or assessing the experimental assembly of nano-structures, using e.g. DNA double-crossover tiles [12, 29] or DNA origami tiles [5, 31, 33], the aTAM is not suitable any more. This is because of the simplistic way in which this model reproduces the assembly system, not taking into consideration the strength of the bonds, the various erroneous assemblies of tiles due to partial matching of the "sticky end" strands, the reversibility of the reactions, i.e., detaching of tiles, etc. For dealing with all these aspects, the kinetic Tile Assembly Model (kTAM) has been proposed, [28], as a kinetic counterpart of the aTAM. Several variants of the kTAM exist, see e.g., [24, 14], however the main elements are similar. We present in the following the initial kinetic model introduced in [28], while other variations of this model are discussed later in the paper.

Within kTAM there are two types of reactions, each involving a tile and an assembly: i) addition of tiles (to the assembly), and ii) detaching of tiles (from the assembly), see e.g. Figure 2 a). Interactions between two tiles forming a new assembly, as well as interactions between two assemblies, are not taken into
consideration in this kinetic model. In the first type of reactions, any tile can attach to the assembly on any position (up to the assumption that the alignment of the tiles is preserved), even if only a week bond is formed. The rate of this reaction is proportional to the concentration of the free-floating tiles in the solution (assuming that all tile types are provided in similar concentrations). In the second type of reactions, any tile can detach from the assembly, with a rate which is exponentially correlated with the total strength of the bond between the tile and the assembly. Thus, tiles which are connected to the assembly by weaker or fewer bonds, a more prone to detaching than those which are strongly connected by several bonds.

Given a tile \( t \in \Sigma^4 \), the association (forward) rate constant \( r_f \) corresponding to the addition reaction of \( t \) to an existing assembly is

\[
r_f = k_f [t] / \text{sec},
\]

where \([t]\) is the in-solution concentration of tile-type \( t \), and \( k_f \) is a temperature dependent parameter. In case of DNA double-crossover (DX) tiles, this parameter is given by

\[
k_f = A_f e^{-E_f/RT},
\]

where \( A_f = 5 \cdot 10^8 \text{ /M/sec} \), \( E_f = 4000 \text{ cal/mol} \), \( R = 2 \text{ cal/mol/K} \), and \( T \) is the temperature (in K), [28].

In the case of dissociation type (reverse) reactions, for a tile \( t \) which is connected to the assembly by a summed bond strength \( b \), the rate constant \( r_{r,b} \) is given by

\[
r_{r,b} = k_f e^{\Delta G^0_b/RT},
\]

where \( \Delta G^0_b \) is the standard free energy needed in order to break \( b \) bonds. For DX tiles whose sticky-end glues are 5 base-long single-stranded DNA molecules, \( \Delta G^0_b \) can be estimated to

\[
\Delta G^0_b = e^{5b(11-\frac{4000}{T})} + 3 \text{ cal/mol},
\]
by using the nearest-neighbor model [23], [28]. The integer-base $b$ parameter is ranging from 0 to 4, corresponding to cases ranging from a totally erroneous placement of the tile (no bonds connect it to the assembly) to fully integration of the tile into the assembly (all four sticky-ends are correctly matched).

In order to easily represent and scale the system, the free parameters involved in the formulas of $r_f$ and $r_{r,b}$ rate constants are re-distributed into just two parameters, $G_{mc}$ and $G_{se}$, which are dimensionless parameters, and their values range in similar scale intervals:

$$r_f = \hat{k}_f e^{-G_{mc}},$$
$$r_{r,b} = \hat{k}_f e^{-bG_{se}},$$

where, in case of DX tiles, $\hat{k}_f = e^3k_f$ is adjusted in order to take into consideration possible entropic factors, such as orientation or location of the tiles.

The previous parameter re-distribution is made possible for the kinetic model introduced in [28] due to the assumption that all tile types are provided into the solution in similar concentrations, and that the consumption of the free monomers is negligible compared to the initial concentration. Because of this, the (free) tile concentration remains constant in time. Thus, the free parameters of kTAM are: $G_{mc}$, which is concentration dependent, and $G_{se}$, which is temperature dependent.

The ratio $\frac{G_{mc}}{G_{se}}$ plays in some sense the role of the temperature $\tau$ from the aTAM. Let $\frac{G_{mc}}{G_{se}} = b - \epsilon$ for some bond strength $b \in \{1, 2, 3, 4\}$ and small $0 < \epsilon < 1$. Then, for a tile $t$ attached to the assembly by bond strength $b$ (or similarly any bond larger than $b$), since $\frac{r_f}{r_{r,b}} = e^{bG_{se}-G_{mc}} = e^{G_{se}} > 1$, the (forward) tile addition reaction on a neighborhood of $t$ is favored, in comparison to the tile $t$ detaching (reverse) reaction. Similarly, for a bond $b' \in \{0, 1, 2, 3, 4\}$, $b' < b$, we have $\frac{r_f}{r_{r,b}} = e^{b'G_{se}-G_{mc}} = e^{(b'-b+\epsilon)G_{se}} < 1$, and thus, the tile $t$ detaching reaction is favored in comparison to the addition reaction of new tiles abutting $t$. In conclusion, the growth of the assembly is favored only in the vicinity of those tiles which are attached to the crystal by a summed bond strength larger than or equal to $b$.

A major difference between aTAM and kTAM is that while in the first case, attachment of tiles is not allowed if the total bond strength is below temperature $\tau$, in the second case such erroneous attachments can happen. Indeed, although it may be unfavorable, a tile $t$ may be locked in place inside an assembly even if its initial attachment to the crystal was done using a total bond strength below $\frac{G_{mc}}{G_{se}}$. This may happen if after attaching $t$ a new tile $t'$ is attached in such a way that the stability, i.e., total bond strength, of both $t$ and $t'$ is increased above the $\frac{G_{mc}}{G_{se}}$ value. Thus, erroneous assemblies are possible in kTAM.

2.1. Computational modeling of the kTAM

Computational modeling of kTAM has been performed almost exclusively using a special tailored version of the Gillespie’s Stochastic Simulation Algorithm (SSA), see e.g. [3, 11, 30]. The assembly starts at $t = 0$ from a seed
structure. Then, in discrete time steps, tiles are added or detached from the assembly, according to the corresponding association and dissociation rates. The structure of the assembly as well as the types of the tiles within are stored using a 2D array.

Let us assume that at some moment of time, the assembly consists of \( n \) tiles and there are a total of \( m \) empty sites around the crystal. During the next time step, there are two possible reaction types: a tile addition, or dissociation. The total tile association (on) rate is given by \( \sum_m r_f \), while the total tile dissociation (off) rate is given by \( \sum_n r_{r,b} \) where, for each of the \( n \) tiles, \( r_{r,b} \) depends on the total bond strength \( b \) of that particular tile. Thus, the overall probability that the next event is an association reaction is \( \Pr(\text{on}) = \frac{\sum_m r_f}{\sum_m r_f + \sum_n r_{r,b}} \). The exact position, out of the \( m \) possible ones, at which a tile will be added, as well as the type of the tile added there, are chosen randomly, with equal probability. Regarding the dissociation reactions, the probability the next event is a tile removal reaction is \( \Pr(\text{off}) = \frac{\sum_n r_{r,b}}{\sum_m r_f + \sum_n r_{r,b}} \). Then, the tile to be removed from the assembly is chosen according to the dissociation rate distribution, namely, for a tile \( t \) with total bond strength \( b' \), the dissociation probability is \( \frac{r_{r,b'}}{\sum_n r_{r,b}} \).

For determining the increment \( \Delta t \) of the time interval we use the probability distribution \( \Pr(\Delta t) = \left( \frac{\sum_m r_f + \sum_n r_{r,b}}{\sum_m r_f + \sum_n r_{r,b}} \right)^{\sum_m r_f + \sum_n r_{r,b}} \). After an event is chosen and executed, the time is incremented with \( \Delta t \) and, for the next time step, the array is updated as well as all the reaction rates. (In practice, only part of the reaction rates are modified, while most of them remain unchanged.)

In the following section, we present a new computational model for the kTAM, based on a very different generic modeling framework using agent- and rule-based representations. As we show next, this modeling framework possesses significant advantages, particularly for modeling systems with potentially infinite number of species, such as kTAM.

### 3. Modeling by agent- and rule-based representations

Agent- and rule-based modeling is a recent and promising discrete modeling approach [18], which has been used successfully in simulating biological signalling pathways [8, 7]. In this modeling paradigm molecules are represented as agents with a number of free sites. The sites, which may have several internal states, allow for agent-agent bonding, thus generating molecular complexes. Rules are defined based on local patterns rather than full specification of the reactants, and thus provide a compact representation on how agents interact. In this way, rather than handling explicitly a high number of model variables, we only have a (often small) number of local interaction rules. This makes the agent- and rule-based paradigm well suited for handling the problem of the combinatorial explosion of the state space. The applicability of this approach for modeling protein self-assembly systems has also been investigated, see e.g. [27, 32] as in such systems, the number of different chemical species is potentially infinite.

From an algorithmic point of view, most of the rule-based modeling languages, as well as the corresponding computational modeling tools, represent
proteins (and protein complexes) as graphs, while the reaction rules are implemented as graph rewriting rules. Thus, computational modeling of these systems is making use of well-established and highly efficient graph algorithms.

BioNetGen [10] is one of the dedicated rule-based languages and computational model simulators. Several reaction simulators are based on the BioNetGen language (BNGL), e.g., Rule Bender [26] and NFsim [27]. In this study, the computational modeling of kTAM using agent- and rule-based representations has been performed using the NFsim modeling platform. This software suite augments the BNGL with several new features which proved to be useful for the simulation of the kTAM.

3.1. Computational modeling using BNGL and NFsim

In order to use the NFsim simulator, a model description has to be provided using the BNGL format. A BNGL file for NFsim contains six separate blocks: parameters, molecule types, species, observables, functions, and reaction rules.

For a more detailed description of these blocks and of the syntax of the NFsim simulator we recommend [10] and [27]. The parameters section is used for introducing the constants used in the model. In the molecule types block, one defines the different agents used in the model, the number and identification name of its sites, as well as the different states each site can be placed in. For example, in the case of kTAM, besides a generator and a trash agent, we use only one generic agent call Tile containing several sites. Depending on the functionalization of this Tile agent, i.e., the particular state initialization of the various sites, it will represent a tile of one particular type or another. The species section introduces the initial particle numbers of various fully-instantiated agents, i.e., each site is in exactly one of its possible states, thus representing the initial species population.

In order to interrogate the current state of the system, we introduce observables. These are patterns of single or site-interacting multi-agents, with partial specifications of their site’s states or connectivity to other agents. During simulation, for each unit of time, the software reports the exact count of instances fitting each of the observables, in the current state of the system. For example, in the case of our Tile agents, we can define a site “in” with states 1 or 0, depending on whether that tile is in an assembly, or it is free-floating. Then, the observable Molecules TilesInAssembly Tile(in∼1) will report (inside the observable TilesInAssembly) the number of tiles which are part of an assembled complex (i.e., not free-floating) at each unit of time.

The reaction rules section describes the agent interaction rules based on which the system evolves in time. For a detailed description of the syntax of rules in NFsim see [27]. As in the case of observables, the reaction rules are defined based on patterns rather than full specifications of the reactant’s site’s states or connectivity. For example, in the modelling of DNA tile systems, the primary agent, Tile, may have a number of sites, including Nedge, Eedge, Sedge, and Wedge, each with internal states corresponding to possible glues of these tiles. Then, both tile-association and tile-dissociation reactions can be implemented by appropriate local rules. For example, the addition of a tile to
a free North site, as depicted in Figure 2 b), can be implemented by the local rule,

\[
\text{Tile}(\text{Nedge}, \text{in} \sim 1) + \text{Tile}(\text{Sedge}, \text{in} \sim 0) \rightarrow \text{Tile}(\text{Nedge}!1, \text{in} \sim 1) \cdot \text{Tile}(\text{Sedge}!1, \text{in} \sim 1), \ k_{\text{on}}
\]

where as before, the site \text{in} indicates whether the tile is in an assembly (\text{in} \sim 1) or it is free-floating (\text{in} \sim 0). The above rule can be interpreted as follows: a Tile with an unbounded \text{Nedge} site placed inside the assembly (i.e., \text{in} \sim 1) interacts with a free Tile (i.e., \text{in} \sim 0) with an unbounded \text{Sedge} site, and the two become bonded on the sites \text{Nedge} and \text{Sedge}; the reaction has a kinetic rate constant \(k_{\text{on}}\). Tile dissociation reactions, which are dependent on the total bond strength of the tiles, can be implemented in a similar manner.

3.2. General advantages of rule-based modelling in simulating Tile Assembly Systems

There are a number of advantages in using a rule-based modelling approach for kTAM. Since this is a coarse-grain modelling framework, it allows the examination of a very diverse family of observables. Thus, the system can be analyzed extensively and both final and intermediate states can be inquired with detailed precision. Also, due to the current availability of appropriate software frameworks, numerical simulations of rule-based models are easy to run. Such simulations can be written in pseudocode, using e.g. BNGL [9] or \(\kappa\) [16]. Thus, the emphasis is placed on describing the system's reaction rules, and not in dealing with the numerical simulation algorithm. Hence, custom simulations are easy to create, update, and modify.

Another advantage strongly related to the current availability of numerical simulator softwares is that within these numerical simulators one is not restricted to have a pre-initialized number of different species in the system. Namely, new species (i.e., complexes of linked agents) are created and simulated on the go in these numerical simulators. This is extremely important when simulating self-assembly systems, as the assembly grows in an un-ordered and non-deterministic fashion. Thus, there is potentially an infinite number of different species that the system will encompass during its evolution, each corresponding to a different possible state of the assembly.

A major drawback of previous modelling approaches was that they could express only those systems where there exists a single growing assembly, and all the reactions (addition and dissociation) were between this unique assembly and the free floating tiles. This situation however, does not cover the case in which two partial assemblies, each consisting of more than one tile, are interacting. However, using the rule-based modelling framework one can implement such reactions too. This opens the possibility of modelling various variants of TAM (which are closer to experimental implementations), accounting for both assembly–tile interactions and assembly–assembly interactions, e.g., staged or hierarchical tile assembly models.
4. Rule based modeling of kTAM

In this section we will provide further details on our rule-based implementation of kTAM, as well as a comparison between its numerical simulation and the numerical simulation of a similar assembly process modeled in Xgrow [30]. Also, we will consider two small alterations of the model and analyze how these modifications in the modeling paradigm affect the assembly growth and the frequency of errors emerging in the assembly process.

4.1. Model description

The rule-based model introduced here is lightly customized for the case when we need only 4 glues, namely, \{0, 1, 2, 3\}, on the tile edges, including the null glue; a more general model for \( n \) glues can be similarly created, even using an automated approach in case of large values of \( n \). The model will be used to simulate a tile assembly system generating the Sierpinski pattern according to the kinetics described in [28]. The Sierpinski tile system consists of a seed tile, two boundary tiles and four rule tiles. The seed tile has two strength-2 glues (north and east edges) and two strength-0 glues (south and west edges). The aggregate first grows along the direction of the strength-2 edges by the addition of boundary tiles. Moreover, the boundary tiles have each a strength-0 edge that restricts the growth of the aggregate to other directions. The rule tiles on the other hand encode the XOR operation that underlies the Sierpinski triangle pattern. The tile types and aggregate growth of the Sierpinski tile system are depicted in Figure 3.

![Figure 3: A 7-tile TAS assembling the Sierpinski triangle (i.e., simulating the XOR binary operation on the input ...000111...). a) Some initial part of the assembled structure. b)-d) the 7 tile system containing: b. 4 rule-tiles c. 2 boundary tiles, and d. one seed tile. The dark-circle glue has strengths 2, while the triangle glues have strength 1; the strength 0 glues on the seed and boundary tiles are not depicted.](image)

In our BNGL model of kTAM we used a generic tile agent whose site’s states will determine the desired tile type. The specification of a tile agent is given by,
where sites are separated by commas (,) and possible states of a certain site are separated by tildes (~). The first group of 4 sites, $N_l$, $E_l$, $S_l$, and $W_l$, encode the binding edges of a tile agent. For a square tile, these correspond to the northern, eastern, southern and western links of a tile. All reaction rules in the BNGL model where a bond is either established or broken utilize these sites. All other sites of a tile agent provide only the context where such bond formation occurs. The second group of sites, $N_g$, $E_g$, $S_g$, and $W_g$, encode the glues of the four edges. Each of these sites has four possible states corresponding to the four edge labels defined in the Sierpinski tile system. The encoding is as follows: 0 for the strength-0 glue, 1 for the strength-2 glue, 2 for the strength-1 black glue and 3 for the strength-1 white glue. The third set of sites, $N_m$, $E_m$, $S_m$, and $W_m$, is used to indicate whether a tile has matching glues with its neighbors. If such a site is in state 1, the tile has a matching glue with its neighbor on that edge. If it is in state 0, there is a mismatch on that edge.

In addition, a tile has three more sites unrelated to the four edges. The first of these, the site $in$, differentiates between the tiles within the assembly, i.e., $in \sim 1$, and the free floating ones, i.e., $in \sim 0$. The second site, $bnd$, determines the rate of dissociation of a tile. The states of this site encode the total strength by which the tile is bound to the assembly. Lastly, the site $act$ signals the next action to be implemented on this tile.

For the reaction rules, we can distinguish several types of rules, with different purposes. Naturally, there are groups of reaction rules which model tile association and dissociation. However, there are also complementary rules which check glue matching, link a newly added tile to its neighbors, or update the bond strength of a tile, etc.

In Appendix I we provide detailed descriptions of these rules and their mechanistic behaviours.

4.2. Simulation results

To test the correctness of our rule-based implementation, we compared the simulated assembly growth of the Sierpinski TAS in both NFsim and Xgrow. Figure 4 a) shows the resulting assembly sizes (from both numerical simulation frameworks) after 100 simulated seconds for various values of the parameter $G_{se}$ and a fixed value of 16 for the parameter $G_{mc}$. Each plotted point is an average over 30 simulation runs.

[1] Although in theory a site can have various states and be used for connecting with other sites, from our implementation in NFsim we saw that such cases may generate undesired numerical errors in the simulation. Thus, we use sites $N_l$, $E_l$, $S_l$, and $W_l$ for implementing the physical bond, and sites $N_g$, $E_g$, $S_g$, and $W_g$ for capturing the glue state of the corresponding edge.
Fig. 4: Comparison between the numerical predictions of a) aggregate growths and b) error fraction performed with NFsim, for rule-based implementation, and Xgrow, a dedicated simulator for kTAM. Both numerical simulators are implementing the same assembly system forming the Sierpinski pattern; the data is collected after simulating 100 seconds of model time and averaged over 30 runs. The $G_{mc}$ parameter is set to 16 in both comparisons.

As it can be seen from the figure, the NFsim simulation results closely follow the Xgrow simulations, thus supporting the validity of the rule-based implementation. In both simulations, there is almost zero growth when the temperature is greater than two (i.e. $G_{se} < 8$), as expected in the TAS, since the border tiles can only bind to the seed with strength two. When the temperature decreases larger assemblies are formed, as a lower level of cooperation for the incoming tile becomes sufficient for growth.

Next, we study how well the rule-based model captures assembly errors by comparing the error fractions of NFsim simulations of the Sierpinski TAS with its Xgrow simulations. Figure 4 b) shows the error fraction, measured as the number of mismatched bonds per the assembly size, for the same parameter values as in Figure 4 a). Once again, the curve of the error fraction obtained from the rule-based model closely matches its counterpart from Xgrow, as both NFsim and Xgrow simulations confirm that larger aggregates with more mismatched bonds form at lower temperatures. This is explained by the fact that as the temperature is decreased (by increasing the value of $G_{se}$), incoming tiles which are matching only one of their glues with some un-covered edge in the assembly are more likely to survive within the aggregate, because of the exponentially less likelihood of the dissociation reactions.

5. Variations of the kTAM model

As previously mentioned, one of the advantages of the rule-based modelling methodology is the ease by which one can modify a model and re-run the simulations. To demonstrate this we consider two possible (and well justified) modifications of the kTAM model, which will thus generate four variant models, depending on whether we implement one, the other, both, or neither of these modifications.

The first modification refers to the kTAM assumption that each tile can
Alternatives | Slot vs Site
---|---
Un-Matching vs Matching | S1 S2
Matching | S3 S4

Table 1: Four possible scenarios based on two modelling alternatives: i) Slot vs Site interactions: Tiles can associate to the assembly on a free position (slot) basis, or on a free edge (site) basis; ii) Reactivity of Un-Matching vs Matching glues: Tiles can attach to the assembly in any circumstance or if at least one matching glue exists.

attach to an assembly on each free position, with the argument that if no common glues would keep this tile in place, the detachment reaction has a very high reaction rate. In the alternative we consider here, a certain tile can attach to an assembly on some position only if there exists at least one common glue on two opposite edges of a tile in the assembly and the free-floating tile. We will denote this alternative choice as the “Un-Matching vs Matching” alternative.

The above considerations lead also to a second possible modification within the kTAM model. In the classical assumption, a tile attachment reaction assumes one free-floating tile is inserted on an empty position (or slot) of the assembly, independent on whether it is matching one, two, or zero glues. An alternative to this is that each individual edge of a free-floating tile can interact with a corresponding (i.e., with cardinal opposite orientation) free edge within the assembly. This modification has the potential to generate different behaviours, as the number of free edges is different than the number of empty positions. Moreover, in the framework of the first modification above, if a free-floating tile is matching the glues of both edges fitting a free position in the assembly, it has twice the chance of attaching to that position in comparison with a tile which is matching only one glue from the two neighboring edges. We will denote this second alternative choice as the “Slot vs Site” alternative.

Based on the above two possible modifications we constructed and explored the dynamics of four possible scenarios, for the cases when each modification is either implemented or not, see Table 1 for the description of these scenarios. Note that according to the partitioning in Table 1, scenario S1 stands for the classical kTAM model. We implemented rule-based models for all 4 scenarios, and run parallel simulations for 100 (model-time) seconds, using the parameter $G_{mc} = 16$ and $G_{se} = 8.2$. The comparative time evolution of the aggregate size and of the error fraction, computed as the ratio between the number of mismatched bonds over the total number of tiles in the aggregate, are presented in Figure 5.

Based on the numerical simulations we can conclude that while the “Un-Matching vs Matching” alternative had little effect over aggregate growth, i.e. scenarios S1, S3 and respectively S2, S4 present very similar growth behaviours, see Figure 5 a), it had considerable impact over the accumulated errors. Indeed, the numerical simulations show that allowing only matching interactions (i.e., scenarios S3 and S4) significantly reduces the error fraction, see Figure 5 b), albeit these error fractions are relatively small for this particular choice of parameters $G_{mc}$ and $G_{se}$ for all 4 scenarios. An opposite dynamics behaviour
Figure 5: a) Aggregate growth for scenarios S1-S4. Larger aggregates form with site interactions (scenarios S2, S4) as there are more possible association reactions that can be triggered. On the other hand, aggregate growth is less affected by the distinction based on matching glues. b) Error fraction of the assembly in scenarios S1-S4, calculated as the number of mismatched bonds over the number of tiles in the assembly. Allowing only matching interactions (scenarios S3, S4) significantly reduces the error fraction, albeit low overall error rates. For each scenario, the data is collected from simulations covering 100 seconds of model time, and averaged over 30 runs; we use parameters $G_{mc} = 16$ and $G_{se} = 8.2$.

can be seen for the “Slot vs Site” alternative, which has very little influence on the error fraction, but highly impacts the aggregate growth.

6. Conclusions

There are a number of significant advantages in using a rule-based modelling approach for kTAM. Since this is a coarse-grain modelling framework, it allows the examination of a very diverse family of observables. Thus, the system can be analyzed extensively and both final and intermediate states can be inquired with detailed precision. Also, due to the current availability of appropriate software frameworks, numerical simulations of rule-based models are simple to run. Such simulations can be written in pseudocode, using e.g. BNGL [9] or $\kappa$ [16]. Thus, the emphasis is placed on describing the system’s reaction rules, and not in dealing with the numerical simulation algorithm. Hence, custom simulation are easy to create, update, and modify.

A major drawback of previous modelling approaches was that they could be used only for those systems in which there exists a single growing assembly, and all the reactions (addition and dissociation) were between this unique assembly and the free floating tiles. This situation however, does not cover the case where two partial assemblies, each consisting of more than one tile, are interacting. Using the rule-based modelling framework allows the implementation of such reactions too. This opens the possibility of modelling different variants of TAM (which are closer to experimental implementations), accounting for both assembly–tile interactions and assembly–assembly interactions, e.g., staged or hierarchical tile assembly models.
In our study, we first created a rule-based model of kTAM. Using the same kinetic parameters as in previous numerical implementations of kTAM, we compared the prediction of those models (regarding time-growths and error-fraction of the assembly) with that of our model. This comparison shows that the rule based modeling framework is at least as expressive as the previous state-of-the-art modelling methodologies for kTAM. Then, we introduced some small variations into the model and studied how this affects the overall behaviour of the system. In particular, while the “Un-Matching vs Matching” alternative had little effect over aggregate growth, it had considerable impact over the accumulated errors. At the same time, the “Slot vs Site” alternative exhibits an opposite behaviour, that is, it has very little influence on the error fraction, but highly impacts the aggregate growth.

Future studies will concentrate on the implementation and analysis of various other types of TAM formalisms, such as stage hierarchical, signal-passing, etc. A milestone for this approach will be overcome when one would succeed in capturing both the dynamic behaviour of a general DNA assembly process and its 3D structural details, e.g. capturing the space- and time-dynamics of a DNA Origami assembly process.

References


Appendix I

*Detailed description of a rule-based implementation of the kTAM model*

The rule-based model introduced here is lightly customized for the case when we need only 4 glues, namely, \{0, 1, 2, 3\}, on the tile edges, including the null glue; a more general model for \(n\) glues can be similarly created, even using an automated approach in case of large values of \(n\). The model will be used to simulate a tile assembly system generating the Sierpinski pattern according to the kinetics described in [28]. The Sierpinski tile system consists of a seed tile, two boundary tiles and four rule tiles. The seed tile has two strength-2 glues (north and east edges) and two strength-0 glues (south and west edges). The aggregate first grows along the direction of the strength-2 edges by the addition of the boundary tiles. Moreover, the boundary tiles each have a strength-0 edge that restrict the growth of the aggregate to two directions. The rule tiles on the other hand encode the XOR operation that underlies the Sierpinski triangle pattern. The tile types and aggregate growth of the Sierpinski tile system are depicted in Figure 3.

In our BNGL model of the Sierpinski tile system, we used a generic tile agent whose site’s states determine the desired tile type. The specification of a tile agent is given by,

\[
\text{Tile}(Nl,El,Sl,Wl,Ng\sim0\sim1\sim2\sim3,Eg\sim0\sim1\sim2\sim3,Sg\sim0\sim1\sim2\sim3,Wg
\sim0\sim1\sim2\sim3,Nm\sim0\sim1,Em\sim0\sim1,Sm\sim0\sim1,Wm\sim0\sim1,in\sim0\sim1,bnd
\sim0\sim1\sim2\sim3\sim4\sim5,act\simnone\simck\glue\simdel\siminc\simdec\siminc\simdec\simdec)
\]

where sites are separated by commas (,) and possible states of a certain site are separated by tildes(~). The first set of sites, \((Nl, El, Sl, Wl)\), encode the binding edges of a tile agent. For a square tile, these correspond to the northern, eastern, southern and western links of a tile. All reaction rules in the BNGL model where a bond is either established or broken utilize these sites. All other sites of a tile agent provide only the context where such bond formation occurs. The second set of sites, \((Ng, Eg, Sg, Wg)\), encode the edge glues of the four edges. Each of these sites have four possible states corresponding to the four edge labels defined in the Sierpinski tile system. The encoding is as follows; 0 to strength-0 glue, 1 to strength-2 glue, 2 to strength-1 black glue and 3 to strength-1 white glue. The third set of sites, \((Nm, Em, Sm, Wm)\), are used to

\footnote{Although in theory a site can have both various states and be used for connecting with other sites, from our implementation in NFsim we saw that such cases may generate errors in the numerical simulation. Thus, we use sites \(Nl, El, Sl,\) and \(Wl\) for implementing the physical bond, and sites \(Ng, Eg, Sg,\) and \(Wg\) for capturing the glue state of the corresponding edge.}
indicate whether a tile has matching glues with its neighbors. If such a site is in state 1, the tile has a matching glue with its neighbor on that edge. If it is in state 0, there is a mismatch on that edge.

In addition, a tile has three more sites unrelated to the four edges. The first of these, the site \textit{in}, differentiates the tiles which are part of the assembly from the free floating ones. A tile has a state 1 in its \textit{in} site if it is in the assembly, or a state 0 otherwise. The second site, the site \textit{bnd}, determines the rate of dissociation of a tile. The states of the site \textit{bnd} represent the total strength by which the tile is hold within the assembly. Finally, the site \textit{act} signals the next action to undertake on its tile.

For the reaction rules, we can distinguish several types of rules, with different purposes. Naturally, there are groups of reaction rules which model tile association and dissociation. However, there are also complementary rules which check glue matching, which link a newly added tile to its neighbors, which update the bond strength of a tile, etc.

In a tile association reaction, a free floating tile attaches to a tile in the assembly. The attachment can be between any two tile types in kTAM, as long as one is free and the other is in the assembly. Thus, in a reaction rule of tile association, one tile agent has state 1 in \textit{in}, while the other has state 0. For instance, a tile-association reaction for the addition of a tile on the east of the assembly was implemented by the local rule,

\[
\text{Tile}(E_l, \text{in}~1, \text{act}~\text{none}) + \text{Tile}(W_l, \text{in}~0, \text{act}~\text{none}) \rightarrow \text{Tile}(E_l!1, \text{in}~1, \text{act}~\text{ckglue}).\text{Tile}(W_l!1, \text{in}~1, \text{act}~\text{ckglue}) \ k\text{on}.
\]

This rule specifies that a free tile binds with a tile with an unbound eastern link inside the assembly at its western edge. Furthermore, the \textit{in} site of the free floating tile has been updated to 1 to indicate it is now part of the assembly. Since tiles are added to the assembly at an equal rate independent of the tile type, the reaction proceeds with a reaction rate \textit{kon}, a constant.

Naturally, we expect three more analogous rules for tile associations on free west, north and south edges. However, such a specification for north-south interactions would allow two different tiles to fill one concave corner slot. Thus, only east-west bindings are of the above form. North-south bindings are conditioned on the case that there are no tiles in the east and west sides of the slot. Moreover, a reaction rule ensures that a tile placed on a corner slot binds with its other neighbor (that is, if the tile was bound east/west, this rule binds it to north/south neighbour; and vice versa). In a sense, this binding establishes the slot solely for the newly added tile. The new binding may be between matching glues - requiring a strength update of the interacting tiles. Thus, the \textit{act} site of these tiles change their state to \textit{ckglue}.

In the association reaction rule and slot establishing reaction, the \textit{act} site was changed from state \textit{none} to \textit{ckglue}. When this state change occurs, a set of reactions which check whether attached tiles have matching glues become activated. For instance, the reaction rule,

\[
\text{Tile}(E_l!1, E_m~1, E_m~0, \text{act}~\text{ckglue}).\text{Tile}(W_l!1, W_m~1, W_m~0, \text{act}~\text{ckglue})
\]
ckglue) -> Tile(El!1,Eg~1,Em~1,act~inc2).Tile(Wl!1,Wg ~1,Wm~1,act~inc2) kmax,
declares that if the linked tiles both have a strength-2 glue (Eg~1 and Wg~1),
the summed strength of their matching glues needs to be incremented by two
(act~inc2). For matching glues of strength 1, act site will be updated to state inc1. The actual incrementation is done via a rule,

Tile(bnd-0,act~inc2) -> Tile(bnd-2,act~none) kmax,

for the case where the tile initially had no bond strength and the increment
size is 2. Similarly, rules are defined for checking strength-1 matching glues
and for size one increments. The matching-glue checking reactions also updated
the state of the Em/Wm sites to 1 enabling the identification of correct bonds.
Finally, both reactions proceed at the maximum rate, occurring instantaneously
so as not to affect the dynamics of kTAM, until all tiles which have participated
in a binding (either due to initial tile attachment or due to slot establishment)
have their binding strength incremented.

On the other hand, tile dissociation was achieved by a combination of a set
of reactions. First, tiles in the assembly are marked to be deleted by rules of
the form,

Tile(in~1,bnd~2,corner~0,act~none) -> Tile(in~1,bnd~2,
corner~0,act~del) kTonReal/exp(2*Gse)
at a rate determined by their total bond strength. Next, the bonds of the
marked tiles are broken by rules of the form,

Tile(Nl!1,in~1,act~none).Tile(Sl!1,in~1,act~del) -> Tile(
Nl,in~1,act~none) + Tile(Sl,in~1,act~del) kmax

When the tiles have detached, their former neighbors may need to update
their bond strength if their glue was labeled as matching. Hence, local rules of
the form,

Tile(El,Eg~1,Em~1,act~none) -> Tile(El,Eg~1,Em~0,act~dec2
) kmax.

signal the need to decrement the total bond strength of the tile. Finally, the
bond strength is decremented using rules of the form,

Tile(bnd-2,act~dec2) -> Tile(bnd-0,act~none) kmax