

Research



Cite this article: Fu D, Reif J. 2024

A biomimetic branching signal-passing tile assembly model with dynamic growth and disassembly. *J. R. Soc. Interface* **21**: 20230755.
<https://doi.org/10.1098/rsif.2023.0755>

Received: 18 December 2023

Accepted: 1 May 2024

Subject Category:

Life Sciences—Mathematics interface

Subject Areas:

nanotechnology, biomimetics, biocomplexity

Keywords:

DNA tile assembly, self-assembly, branching random walks

Authors for correspondence:

Daniel Fu

e-mail: daniel.fu@duke.edu

John Reif

e-mail: reif@cs.duke.edu

Electronic supplementary material is available online at <https://doi.org/10.6084/m9.figshare.c.7232726>.

A biomimetic branching signal-passing tile assembly model with dynamic growth and disassembly

Daniel Fu and John Reif

Department of Computer Science, Duke University, Durham, NC, USA

DF, 0000-0001-5934-7314

Natural biological branching processes can form tree-like structures at all scales and, moreover, can perform various functions to achieve specific goals; these include receiving stimuli, performing two-way communication along their branches, and dynamically reforming (extending or retracting branches). They underlie many biological systems with considerable diversity, frequency, and geometric complexity; these include networks of neurons, organ tissue, mycorrhizal fungal networks, plant growth, foraging networks, etc. This paper presents a biomimetic DNA tile assembly model (Y-STAM) to implement dynamic branching processes. The Y-STAM is a relatively compact mathematical model providing a design space where complex, biomimetic branch-like growth and behaviour can emerge from the appropriate parametrization of the model. We also introduce a class of augmented models (Y-STAM⁺) that provide time- and space-dependent modulations of tile glue strengths, which enable further diverse behaviours that are not possible in the Y-STAM; these additional behaviours include refinement of network assemblies, obstacle avoidance, and programmable growth patterns. We perform and discuss extensive simulations of the Y-STAM and the Y-STAM⁺. We envision that these models could be applied at the mesoscale and the molecular scale to dynamically assemble branching DNA nanostructures and offer insights into complex biological self-assembly processes.

1. Introduction

Nanoscience has, across various materials and for numerous purposes, sought to construct increasingly complex structures by assembling nanoscale materials and devices of greater efficiency, functionality, and revolutionary applicability [1]. However, these are almost always accomplished by following direct, deterministic design methodologies. In stark contrast, nature often contends with novel, ever-changing environments, and hence natural biological self-assemblies need to use dynamic, non-deterministic self-assembly processes to adapt, grow, and survive.

Branching processes are one such example that underlies the fundamental understanding of a diverse range of natural biological phenomena, appearing in plants [2–5], fungi and slime moulds [6–10], ant trails [11–15], branching morphogenesis [16–23], and many more. The branching processes observed in these instances can be surprisingly sophisticated; they exhibit dynamic growth, sensitivity to the environment and chemical signalling, the degradation of branches that are not useful, and often feature two-way communication ability that verifies the discovery of intended targets which may trigger changes in the assembled branch, such as enforcing or disintegrating the branch.

For instance, the internal regulation of auxin controls branching in plants [3–5] according to factors such as light intake, nutrient uptake from soil, and the current size of the branch. Slime mould networks can use chemical trails to avoid obstacles and navigate toward new territory [8–10] while mycelium can optimize the geometry of its network for feeding [7]. Ants curate their pheromone trails according to various environmental conditions and to communicate the direction of food [11–15]. Organ tissue and neural tissue grow

and branch according to both intrinsic rules [19,20] and signals received from an extracellular matrix [24,25] to fulfil a desired density, shape, or function. Branching morphogenesis has also been considered a computational process that makes limited use of genetic programming; it is a generalized process that can be repurposed using different signals without major changes to the genetic programming that implements the branching process itself [26–30].

There have been extensive efforts to analytically understand these patterns of growth. Various computational and mathematical models include kinetic models [31,32], lattice gas models [33], generative models [34–36], branching random walks [37–39], Lindenmayer systems [40,41], and reaction–diffusion systems [42]. There has been a sense that coarse-grained models, such as cellular automata, could simulate natural processes [43–45], and there have been such instances specifically tailored to fungal growth [46,47] or morphogenesis [48–50].

Here, we use DNA nanotechnology and tile assembly models, which together form the subfield of DNA tile-based self-assembly [51,52], which is closely related to cellular automata [53–55]. DNA nanotechnology uses DNA molecules for either computational or structural purposes, instead of genetic. Nanostructures and logic circuits can be constructed with only DNA as a substrate using the intentional design of hybridizing domains. These factors combine to make DNA nanotechnology uniquely suitable for implementing tile assembly systems, which are algorithmic problems concerning the coverage of a plane using a small pool of repeated units (tiles) that have defined rules for their edge-to-edge adjacency with each other.

The field of DNA nanoscience thus blends chemistry, the life sciences, and mathematical modelling; it has a unique capacity to model or mimic natural processes, such as demonstrated by its significant progress in implementing chemical reaction networks [56]. Branching processes provide a thought-provoking paradigm from which to investigate the fundamental components and operations of a class of natural processes that are obscured behind complex behaviours. DNA nanotechnology offers an opportunity to physically manifest these insights *in vitro*, as opposed to only *in silico*.

In most cases, these branching processes are non-deterministic yet reproducibly exhibit behaviours or produce geometries that local constraints and conditions can dynamically influence. The mechanisms behind these processes have become increasingly understood owing to research progress, and the responsible macromolecules and chemical signalling pathways have often been identified and consolidated. However, it remains a challenge to develop a succinct model of these very complex biological self-assembly systems that would mimic and reflect their behaviour and generalize to engineering self-assembled structures at both large and nanometre scales. Current empirical strategies employed to confirm theory and computational models [19,20] mostly involve imaging tissue, either during early developmental stages [57–59] or when affected by growth factor mutations [60–63]. That is, in effect, a top-down approach where the insights of simulated models cannot be evaluated *de novo*.

This paper will formalize, simulate, and propose implementations for a biomimetic model based on DNA tile-based self-assembly to serve as a bottom-up model to understand and dynamically produce goal- and function-oriented structures that resemble natural examples, such as the growth of mycorrhizal fungal networks, slime mould networks, neural networks, and organs, which are each crafted by evolutionarily tailored branching self-assembly processes. In this model, DNA tiles can be used as a fundamental unit to implement, observe, and manipulate various branching behaviours that emerge from specifically composed elementary functions and signals. Moreover, the stochastic self-assembly strategies gleaned from natural branching processes may also lend new perspectives toward achieving the goals of much of DNA nanostructure research, which has been to self-assemble a nanostructure of some predetermined shape; many of the most complex examples of which have been accomplished by direct or repeated explicit programming of shapes [64], but others have also been described algorithmically by prior tile assembly models, such as the abstract Tile Assembly Model (aTAM) and its many derivatives [52,65–69].

The model defined in this paper, the Branching Signal-passing Tile Assembly Model (Y-STAM; ‘Y’ is used here as a symbolic abbreviation for ‘branching’) is derived from elements of the kinetic Tile Assembly Model (kTAM) [70] and the Signal-passing Tile Assembly Model (STAM) [71–73]. The STAM has previously been alluded to as a potential model of biological processes, but to the best of our knowledge, none have yet to be formally and thoroughly described and, for that matter, demonstrated. Mohammed *et al.* [74] previously experimentally demonstrated DNA nanotube assemblies that formed a connection between two molecular landmarks, but this occurred without branching. An implementation of the STAM has also been experimentally demonstrated, but also for only a one-dimensional growth [75].

The Y-STAM aims to distinguish several key characteristics of branching self-assembly processes, like those that drive morphogenesis, such as tip bifurcation (branching), elongation (growth), and termination, which seem to be widespread primitives of branching processes, as well as additional concepts like degradation and two-way communication. We later introduce time and space-inhomogeneous signals that can regulate each of the aforementioned factors. These resemble the presence of external influences such as growth-enabling nutrients or the limitations thereof, environmental stressors, and signalling from an extracellular matrix. We introduce the Augmented Branching Signal-passing Assembly Model (Y-STAM⁺) for the family of expanded models that accommodate such external influences outside the Y-STAM design space.

Using simulations restricted to the rules of our tile assembly models, we speculate upon and delineate the existence of several input parameters that configure and influence branching processes and evaluate their influence on the final shape and function of dynamic branching assemblies. Our work differs from prior formal models of branching processes, which have not been generalized across fields of study, have not been as comprehensive, and have generally not included two-way communication nor time- and space-dependent behaviours [19,20,23,76]. Our work is unique in its attempt to simultaneously accommodate multiple parameters of branching processes with greater generality than prior, field-specific models while maintaining attention to the broad, bottom-up applicability of the model. The results of the paper show that external influences on a statically programmed branching process can influence the behaviour and success of an assembly in numerous settings and functions, such as food-finding, resource competition, metabolic and ecological constraints, and obstacle navigation.

Our paper is organized as follows. Section 2 informally introduces both the Y-STAM and the Y-STAM⁺ and the simulations performed for each model. Section 3 formally defines the Y-STAM and the features present on each DNA tile to implement

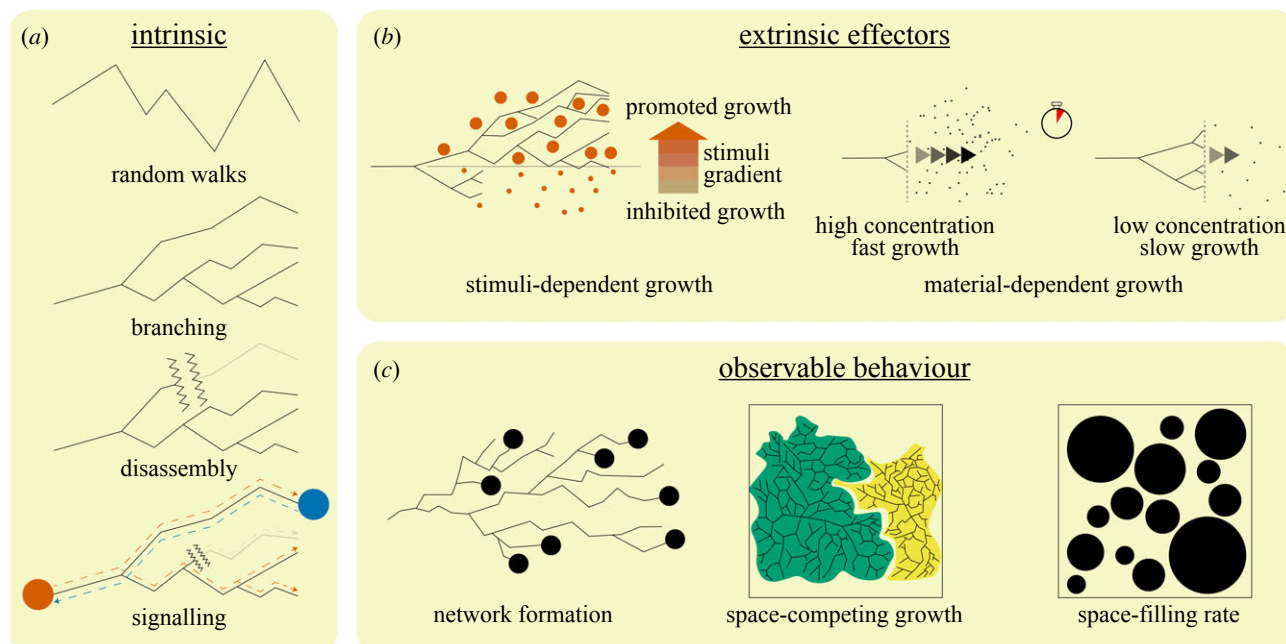


Figure 1. Overview of the branching tile assembly model. (a) Intrinsic properties of the model describe fundamental processes present across all instances of branching. Bolder lines indicate preferred or strengthened paths, whereas paths shown as thinner lines are more likely to degrade or are already degraded. (b) Extrinsic modulators broadly refer to chemical signalling and external processes that can affect intrinsic properties. (c) Observable behaviours result from the tuned interaction of the intrinsic properties and extrinsic effectors and fulfil the desired shape or function of the branching assembly process. For example, we can evaluate space-filling behaviours such as competitive growth for absorbing nutrients or the relationship between branching parameters and space-filling rates.

growth, branching, termination, degradation, and response to positive feedback (figure 1a). Section 4 discusses simulations for a restricted unidirectional version of the Y-STAM, where growth proceeds strictly from left to right, as a starting point to make intuitive observations about branching processes from more easily interpretable results. Section 5 adds formal definitions and simulations for the Y-STAM⁺, which enables the model to be sensitive to environmental factors (figure 1b), and we use this to study emerging behaviours that are the result of interactions between functions programmed directly into the tiles and functions that are programmed externally. Then, in §6, we remove the restrictions of unidirectional growth and perform simulations for the Y-STAM⁺ in two dimensions (figure 1c). Finally, in §7, we discuss potential strategies and challenges for implementing the model, discrepancies when compared to real-world examples, and speculate upon extensions and applicability of the model.

2. Informal review of prior tile assembly models and introduction to the Y-STAM and the Y-STAM⁺

2.1. Informal review of the aTAM and the kTAM

We begin with an informal review of key definitions from prior models [52,70,77] and later omit to repeat their formal definitions. Wang tiling [51] uses square tiles, intending to fill or pattern the integer plane (or grid) $\mathbb{Z}^2$, where $\mathbb{Z}$ is the set of integers. DNA tile-based self-assembly models, like the aTAM [52] and the kTAM [70], are based on Wang tiling and are interesting models of biological computation owing to being intrinsically universal [78].

The aTAM is the precursor model of DNA tile-based self-assembly. It defines a set of square, non-rotating tiles with *glues* on each side. Each glue has a *colour* (later more commonly known as its *label*), and the glues, by matching colours, dictate the attachment of tiles to each other. Tile assemblies begin from a *seed*, which is a set of tiles that initiates the assembly and is also used as a fixed reference point. As the tile assembly fills the plane, the glues can be chosen such that the tiles, observed by a label for the tile itself, form specific, algorithmic patterns, such as the Sierpinski triangle [66]. Furthermore, glues are each defined with some *strength*. The strength of a glue–glue bond is compared to the *temperature*, a global variable of the assembly, which determines whether a glue–glue bond is valid within the assembly.

The kTAM is closely related to the aTAM, except that the kTAM further resembles realistic processes by interpreting the strength of glues as a binding coefficient between two glues. The kTAM adds greater thermodynamic context into tile assembly models, defining an entropic cost of *association* (addition of a tile into a vacant spot of the assembly), G_{mc} , and a free energy cost of *disassociation* (breaking a bond to remove a tile from the assembly), G_{se} . The temperature is then the ratio $\tau = G_{mc}/G_{se}$, and this is translated into forward (r_f) and backward (r_b) rates for the assembly. The kTAM thus provides a probabilistic model that more closely models realistic biochemical reactions.

2.2. Informal review of the STAM

The STAM adds power to tile assembly models by the definition of glue states and a transition function that can alter the state of a glue depending on binding events occurring to other glues on the tile. Glue states (*on*, *latent* and *off*) control whether a glue can bind or not. In DNA strand displacement terms, when two glues bind, a localized strand is released that activates a subsequent

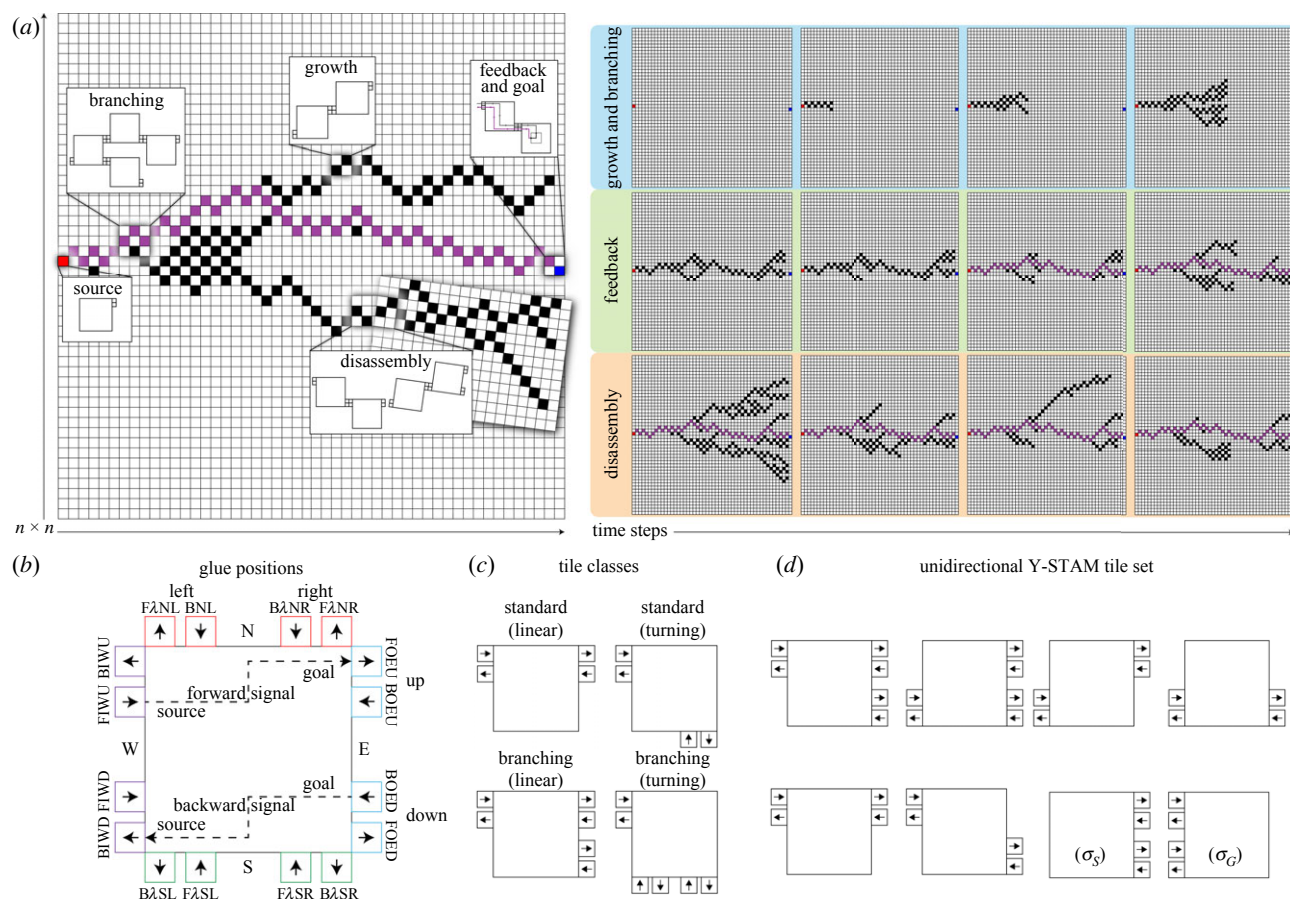


Figure 2. Schematic overview of basic branching assembly processes. (a) Growth, branching, disassembly, feedback, and source and goal tiles are labelled as they appear in our simulations. New tiles binding to a pathway are always diagonally adjacent to the existing tile in the pathway, but having separate pathways that may each grow tiles into diagonally adjacent positions does not mean those tiles are also connected. (b) A tile that shows the position of all glue types T . Glues of each channel exist pairwise in each direction of the tile. Traversal directions are included in the shorthand labelling only where they are obvious from additional information. Otherwise, they are excluded as they cannot be indicated from glue position alone. The dotted forward and backward signalling lines are examples of signal propagation through a tile. (c) Tile types are classified according to whether their addition into the assembly causes the pathway to branch (standard or branching) and/or turn (turning or unidirectional). (d) A unidirectional implementation uses only six distinct tile types plus two additional source and goal tile types for seeding the assembly.

glue on the tile. Thus, a glue ‘passes a signal’ to other glues on the same tile. This was shown to be a powerful modification to previous tile assembly models, as more complex assemblies could be formed with comparatively simpler tile definitions. In our paper, we will uphold the same assumptions as are present in previous work and impose minimal limitations on the speed of signals.

2.3. Informal definitions of the Y-STAM and the Y-STAM⁺

We next informally introduce the Y-STAM (see §3 for a formal definition). We note that, in contrast to most tile assembly models, the Y-STAM is less a model of computation and more an analytical tool, based on the construction of tile assembly systems, that has been applied to understand vividly biological processes with greater visual intuition. As such, we may forgo some of the mathematical compactness of typical tile assembly models in the interest of interpretability in the context of their biological resemblance and do not consider the computational expressivity of the model. Some behaviours of the model, such as evaluating the influence of environmental parameters, may be bulky to discuss without the transformations provided in the Y-STAM. (See electronic supplementary material, figures S1 and S2, for an extended discussion on these transformations.)

The stochastic branching process of the Y-STAM forms a tree-like assembly within a two-dimensional space using DNA tiles. As in previous work, tiles in this model bind to each other by glues. In our stochastic model, the forward and backward rates (or sometimes also referred to as the growth and disassembly rates) are the likelihood that the bond formed between two glues remains bound, and these are trivially derived from the glue strength in the Y-STAM, but have several additional nonlinear contributing factors to consider in the Y-STAM⁺. Tile association causes the assembly to *grow*, while disassociation of tiles is referred to as *disassembly* or *degradation*. Elongation occurs when a tile binds a single new tile. *Branching* (or bifurcation) occurs when a tile binds two new tiles, which can each continue to grow in separate directions (figure 2a).

An *assembly* begins from a *source* tile as a nucleation site. The source tile is a unique tile, as it can potentially bind as many tiles as it may have space for (up to four), thus branching the assembly in similarly many directions. Each continuous series of tiles that can be traced from the source tile is called a *pathway*. A non-negative number of *goal* tiles can also be placed during the initialization of the assembly. Termination can occur when a pathway binds to a goal tile, and no new tiles will bind to that pathway.

Instead of tiles associating horizontally or vertically adjacent to each other, tiles bind diagonally. When the assembly branches, the two new tiles are placed such that a line between them is parallel to a Cartesian axis. The branching axis, which bisects the bifurcation site, is then also parallel to a Cartesian axis, and the growth of the assembly can be more intuitively assessed in cardinal directions as opposed to ordinal directions (electronic supplementary material, figure S1).

Each tile can hold a finite state and passes a forward and backward signal. Emitting from a source, a forward signal weakly sent along all branches enforces the *connectivity* (likelihood of the tiles remaining in the assembly instead of disassociating) of the assembled tiles, and tiles that were once bound but then lose this connectivity cannot associate to the assembly again and are considered to be waste. If the goal tile in the environment is found by a tile that has grown at the tip of a branching pathway originating from the source tile, the goal tile produces a backward signal sent along the branch of the tile that discovered it, and this signal strengthens the connectivity of all the tiles on that branch. These forward and backward signals reinforce the connectivity by multiplying the effect of the baseline glue strength of the tiles and we refer to this as *feedback*.

The Y-STAM⁺ (see §5 for formal definitions of the augmented features) then introduces the potential to modulate the forward and backward rates further using programmable adjustments by effectors separate from the tile assembly. Here, we differentiate the intrinsic properties of the branching process (*intrinsic*), which are the static properties directly programmed into the tiles (figure 1a), and extrinsic influences (*extrinsic*) that are programmed externally, which can alter assemblies without altering the programming of tiles (figure 1b). They are considered external because other than the tiles having fixed receptors to receive the outputs of these modulators, the characteristics of the assembly that change in response to these modulators do not depend on changes to the tile design. These are unattached to the tiles but proximal to affixed, discrete positions that each tile can bind at, and they can increase or decrease the local forward and backward rates, proportionally to the glue strength, and have a localized effect on the growth or disassembly of the assembly at that precise location. We consolidate and describe the activity of intrinsic and extrinsic as being in separate *layers* of the model.

We seek to show that the wide dynamism of branching processes can be largely achieved with time- and space-based changes to the growth and disassembly rates of glues, which cannot be fully captured by the Y-STAM and in the tiles themselves, thus necessitating the augmented model. We note that while only a single additional layer is demonstrated in the current work, we will not formally limit the number of additional layers that can be used as extrinsic modifiers to the assembly that occurs in the tiling layer (electronic supplementary material, figure S3). The augmented model more generally encompasses the family of models that define tile-independent functions that may interact with the tile assembly. The simulations of this model help to speculate upon an algorithmic perspective of branching processes, such as morphogenesis. We show that an organizational hierarchy of extrinsic applied to universal mechanisms, the intrinsic, can produce controlled and complex geometries. From the bottom up, we postulate potential foundations for why such processes can be ubiquitous across nature.

We present several simulations to demonstrate the influences of introduced parameters on the dynamism of the model. We first simplify the model to a unidirectional model that grows strictly from left to right to understand several basic properties. First of all, the amount of space reached and covered by branches depends on how frequently branches can form. Rates of disassembly are then evaluated for their effects on the previous distributions. Next, backward signalling occurs when the assembly reaches a goal location and this counters disassembly for a selected portion of the assembly.

The next section of simulations adds the extrinsic to fulfil the role of the external environment, which could be chemical signalling, an extracellular matrix, or the presence of growth-enabling nutrients or other gradients, to the branching self-assembly process. Simulations evaluate whether the growing tile assembly can grow along an externally programmed path, responsively assemble or disassemble at time- and space-inhomogeneous rates across the entire assembly, and grow dependently on available resources. The electronic supplementary material also investigates the ability to react to obstacles and observes trends in growth rates and functional success of the assembly depending on branching frequency and the concentration of tiles (electronic supplementary material, figures S4–S6).

Finally, simulations are performed on the two-dimensional expansion of the model where the assembly can grow in any direction from a central seed. These show macroscale behaviours, such as a pair of assemblies contending for nutrients that stabilize larger assemblies or the space-filling rate of densely nucleated assemblies.

3. Formal definitions and constructions

We now present a formal description of the Y-STAM, which has a set of tile types that (i) can hold state, (ii) have capabilities for branching of assemblies, and (iii) have capabilities for sending signals forward and backward along branches. In a later section, we will also define a family of models described by the Y-STAM⁺ that describe (iv) capabilities for dynamic strengthening and weakening of branches of assemblies, which are introduced as additional layers which interact with the tiling layer of the basic Y-STAM.

The base model (Y-STAM) covers the definitions of the tiles, glues, signal-passing, assembly, and the growth and degradation of the assembly. We generally discuss definitions in two dimensions to make the model more intuitive and not overburden the definitions with too many restrictions, as the semantics are quite versatile to fall back down to unidirectional modes later or even be expanded upwards into three dimensions.

We define the environment on the discrete, two-dimensional integer space $\mathbb{Z}^2$.

Let $\mathbf{U} = \{(1, 1), (1, -1), (-1, 1), (-1, -1)\}$ be a set of *growth vectors*. That is, tiles grow diagonally.

Let t be a unit square tile that cannot be rotated. We denote each side of a tile using the cardinal compass directions $D_+ = \{N, S, E, W\}$. It follows that the growth vectors match the ordinal directions and denote the corners of the tile $D_\times = \{NE, NW, SE, SW\}$. While it may seem natural to use four glues on the tile corresponding to the corners, for the sake of later interpreting the simulations using only the cardinal directions, we instead simulate the corners using eight positions.

To do so, we next define *orientation* $O = O_V \cup O_H$ to specifically address the proximity to either corner of an edge, where $O_V = \{L, R\}$ indicates left (L) or right (R) corners for the N and S edges and $O_H = \{U, D\}$ indicates top (U) or bottom (D) corners for the E and W edges. It follows that $O_V \times \{E, W\} = \emptyset$ and $O_H \times \{N, S\} = \emptyset$.

There are thus eight total glue positions, $D_+ \times O = \{NR, NL, EU, ED, SR, SL, WU, WD\}$, where each position still corresponds to the growth vector of its nearest corner.

Let T be a finite set of tile types, σ is the set of seed tiles that initializes the assembly, and K is a tile distribution vector. A Y-STAM assembly is $\mathcal{T} = (T, \sigma, K)$ and forms a symmetric, directed tree graph $G = (V, E)$, where each vertex $v \in V$ is a tile and each edge $e \in E$ is a glue–glue bond. Each tile has exactly one parent node and, at most, two child nodes, except for seed tiles $\sigma = \{\sigma_S, \sigma_G\}$. The root node is the *source tile*, σ_S ; it has no parents and is not limited to two child nodes. σ_G is a goal tile; it has no child nodes.

Any assembly must be initialized with at least one source tile, whereas goal tiles can be optional. All seed tile placements are also permanent. That is, they cannot be added after initialization, and disassembly processes that would remove other tile types from the assembly do not remove seed tiles from the assembly.

A *valid configuration* of the tile assembly model is any graph G with a non-empty set of σ_S and every tile is reachable from exactly one source tile.

We also define *Position* : $V \rightarrow \mathbb{Z}^2$, which maps a tile to its location in the two-dimensional integer plane, which is not particularly necessary for the Y-STAM, but becomes relevant later in the Y-STAM⁺ to coordinate the signals in other layers to the tiling layer.

Define the *traversal direction* $\lambda = \{I, O\}$, which refers to a tile's *input* (I), which is the edge pointing to its parent node, and the tile's *output* (O), which are the edges pointing to its child nodes.

We then define the signal propagation *channels* $C = \{F, B\}$, which will serve to implement feedback.

Now, we can define a glue on tile t with a string label $\gamma \in \Gamma$, strength $s \in [0, 1]$, and state $q \in Q$. These values determine if a tile t can form a glue–glue bond with another glue on another tile t' .

We use the channel (C), traversal direction (λ), compass direction (D), and orientation (O) to build the glue label string, e.g. FOWD, for a forward-channel, output glue placed on the west edge at the bottom corner (figure 2b). All possible glue label strings form the alphabet Γ .

The glue strength s is a fixed, intrinsic property of the tile. We separately introduce the disassembly rate $r_b = 1 - s$ as a parameter that can be modified (dynamically, after the introduction of the Y-STAM⁺).

Finally, we describe the states $Q = \{\text{latent}, \text{on}, \text{off}\}$, where a glue in the *on* state can bind or is currently bound and can be switched to the *off* state. In the *latent* state, it cannot bind but can be switched to the *on* or *off* states. In the *off* state, a glue cannot bind or be switched to any other state. In general, each glue has initial states such that inputs are *on* and outputs are *latent*.

Let M be a mapping function of glue label strings to their complement. Only glues with complementary strings can bind. Let γ be a glue on tile t and γ' be a glue on tile t' . γ binds γ' if and only if the labels are of the same channel, along the same compass axis (e.g. N and S , E and W , or as shown on the square tile, glues on the same compass axis exist on opposing edges of the square), have opposing orientations (e.g. U and D , L and R), and opposing traversal directions. Building the label strings in such a way allows us to, later, in simulations, easily filter glue types by single characters.

For example, the pairs of glues FOWD and FIEU, or BONL and BISR may bind, whereas FINL and FISL, or FOWD and BIEU, may not bind. Both glues must also be *on* and will not bind if either glue is in other states. These rules also exclude certain tiles from being included among the tile types, such as those with inputs and outputs toward the same tile position, e.g. a tile with a FIWD glue cannot also have a FOSL glue as the growth direction of the tile will conflict with prior tiles in the assembly. Any tile will also not simultaneously have both FOWD and FOSL glues, as the new tiles they would bind would also overlap.

Next, we describe the *signal-passing* process as follows. When the input of a tile is bound, that signal is *passed*, and that same tile's outputs of the corresponding channel are converted to the *on* state. We also introduce the notion of *signal-enforcing*: where if a tile that is initially bound later becomes unbound (loses the signal), all of its glues are converted to the *off* state, and the corresponding tiles are also removed, as later described by *disassembly*.

We can now generalize several categories across all tile types to facilitate discussion. All tiles used in the model will have only one input. A tile with two outputs that will each bind a tile in a distinct direction is a *branching tile*, T_B . Branching only occurs along the forward channel. A tile with only one output, regardless of position, is a *standard tile*, T_S . A *turning tile*, T_T , is one where the input and output glues do not exist on the same compass axis, and a *unidirectional tile*, T_U , is a tile where the glues do exist on the same compass axis. Turning and unidirectional tiles are separate categorizations independent from standard and branching tiles (figure 2c). In total, the tile types are a set $T = \{T_B \cup T_S \cup T_T \cup T_U\}$.

To define the non-deterministic growth of the system, K is a tuple $K = (k_B, k_S, k_T, k_U)$ where each element is an integer weight factor corresponding to each subset of tile types T . The association of a new tile is thus a probability distribution function that samples T , where K weights the probability of selecting a tile from each classification.

Finally, we describe *feedback*, which occurs when a pathway meets a goal tile. σ_G is a unique tile that can transduce a forward-channel input signal to convert a backward-channel glue from *latent* to *on*. By our assumption that signals propagate asynchronously, the backward-channel signal will convert all backward-channel glues of tiles along the pathway, tracing back to σ_S , to *on*. We assume that each newly converted glue immediately forms a glue–glue bond between their adjacent tiles owing to proximity. This feedback process increases the overall glue binding strength between tiles along that particular pathway from $1s$ to $2s$, where s is the glue strength, and can be thought of as opening another pathway of connectivity between the source and the goal. That is, at any disassembly phase, two glue–glue bonds would have to be broken simultaneously to remove a tile from an assembly. Otherwise, if only one is broken, it is assumed the broken bond rebinds immediately on the

next step. This pathway resists degradation more than the rest of the assembly and is more likely to persist while the rest of the assembly continues to undergo growth and disassembly.

An *assembly* is a valid configuration of tiles that can be created from an initial set of seed tiles, by following the dynamic growth/disassembly rules. Let $\mathcal{A}(\mathcal{T}, \tau)$ be the set of all possible assemblies that can be created from the tile assembly system $\mathcal{T}$ within the finite maximum allotted time $\tau \in \mathbb{N}$ for the assembly to grow. Let i be any time-step where $i \leq \tau$. Let α_i be an instance of the assembly at time-step i , and it follows that $\forall i, \alpha_i \in \mathcal{A}(\mathcal{T}, \tau)$. Only the seed tiles are present at α_0 , which is the initialization of the assembly. Any assembly α_i then proceeds forward to α_{i+1} by first evaluating all tiles for *disassembly*, whereby any glue–glue bond previously attaching a tile to the assembly may be broken, according to some probability. *Signal-enforcing* also occurs at this stage. Then, new tiles can *associate* into the assembly, whereby each valid, vacant location in the assembly can bind a randomly chosen tile with validly binding glues.

More specifically, for *disassembly*, any glue–glue bond has an independent probability of unbinding according to the disassembly rate $r_b = 1 - s$. The value r_b is used here instead of s such that it can later accommodate other factors determining the disassembly rate that are not intrinsic to the tile design, and these will be further discussed later. If a glue–glue bond breaks, then the tile containing the input glue of the bond is removed from the assembly. Owing to the signal-enforcing property, as the signal has been interrupted, if there are subsequently connected tiles in the pathway, those tiles will also be removed from the assembly. In doing so, every tile in the assembly must be connected to the source; the goal tile cannot nucleate assemblies, and no supertiles form (supertiles are sub-assemblies that can nucleate apart from the main assembly yet still attach as a single piece to the main assembly).

For association, a tile with output glues in the *on* state may attempt to bind a tile from the distribution of tile classes, given by k_B . We may also more commonly refer to k_B as the *branching factor*, β . Here, we also define the forward rate $r_f = [0, 1]$, which determines the probability that a tile will bind to an available binding site during the growth phase. However, until stated otherwise, $r_f = 1$. Any assembly is thus a non-deterministic self-assembly that forms branching pathways nucleated from σ_S , continuously growing and disassembling while attempting to connect from σ_S to σ_G .

Later (particularly in our simulations), we will use a limited variation of this model, which we term the *unidirectional Y-STAM*, that only permits the assembly to grow unidirectionally. A source tile is placed to the west, relative to a goal tile, and the overall assembly proceeds eastwards. This model does not require its own formal definition and is straightforwardly obtained from the model defined thus far. Simply, glues with $D = \{N, S\}$ are removed, while *input* ($R = \{I\}$) traversal directions are permitted only on $D = \{W\}$ glues and *output* ($R = \{O\}$) traversal directions are permitted only on $D = \{E\}$ glues. This equates to removing turning tiles from the pool and aligning the traversal direction of all remaining unidirectional tiles, yielding the simple tile set as shown in figure 2d.

This formally describes a randomized growth and degradation process that can potentially yield multiple branching pathways nucleated from a source. The self-assembled tilings continuously grow and disassemble until a pathway discovers a goal location, upon which that single pathway receives a signal that propagates along itself towards the source, which decreases the likelihood of that pathway disassembling. Before discussing the augmentation of this model, we first use simulations of the unidirectional implementation of the model to quantify the properties of only the Y-STAM, which include its assembly, disassembly, and feedback activity.

4. Simulations of the unidirectional Y-STAM

We restrict our simulations of the Y-STAM and, later, the Y-STAM⁺ to an n -by- n subset of the two-dimensional lattice. In the context of biological morphogenesis, it is more likely that multiple, seeded assemblies proceed adjacently and concurrently. As such, each instance may have limited space available to it. These space bounds are vital to evaluate whether certain properties provide sufficient morphogenic behaviours within their confined area. To facilitate the programming of the simulator, we add discrete time $i \in \mathbb{N}$ and evaluate association and disassociation once per time step for each tile in the assembly.

We evaluated the dynamic characteristics of the Y-STAM assemblies using stochastic simulations programmed in MATLAB. To test and interpret some behaviours of the model more easily, we began with the unidirectional variant of the model. This corresponds to left-to-right growth in our illustrations. That is, the source tile is placed at the leftmost (corresponding to west) side of the simulation space, while a goal tile is placed at the rightmost (corresponding to east) side of the simulation space. These simulations offer some initial intuition on observed trends of assembly and disassembly rates and the feedback mechanism to basic parameters such as the branching factor β , glue strength s , and the size of the space n .

In all the following simulations, the possible positions of σ_G are constrained. Owing to the discretization of the grid and the diagonal growth direction of glues, new tiles alternate between even and odd y -positions. Depending on n and the location of σ_S , whether σ_G is placed at an even or odd y -coordinate position determines whether it is reachable. Thus, for any n and σ_S , there are only $n/2$ acceptable placements for σ_G where the assembly can behave as intended.

We use the following metrics to assess the mesoscale activities of any Y-STAM assembly.

- (i) *Hit*. A Boolean which is 1 if the path has found a goal tile, and otherwise 0.
- (ii) *Connected time*. The number of discrete time-steps during which a path is connected between the source tile to the goal tile.
- (iii) *Assembly size*. The number of tiles in the assembly at a time-step.
- (iv) *Duration*. The number of discrete time-steps the simulation runs for.

Simulations were terminated owing to either (i) duration beyond a prescribed time limit, (ii) upon reaching the goal or (iii) reaching the rightmost column of the allowed assembly space. We will discuss the simulation results of the model in an incremental fashion, beginning with a most basic understanding of the model that is only the growth of a single pathway akin to a random walk, then

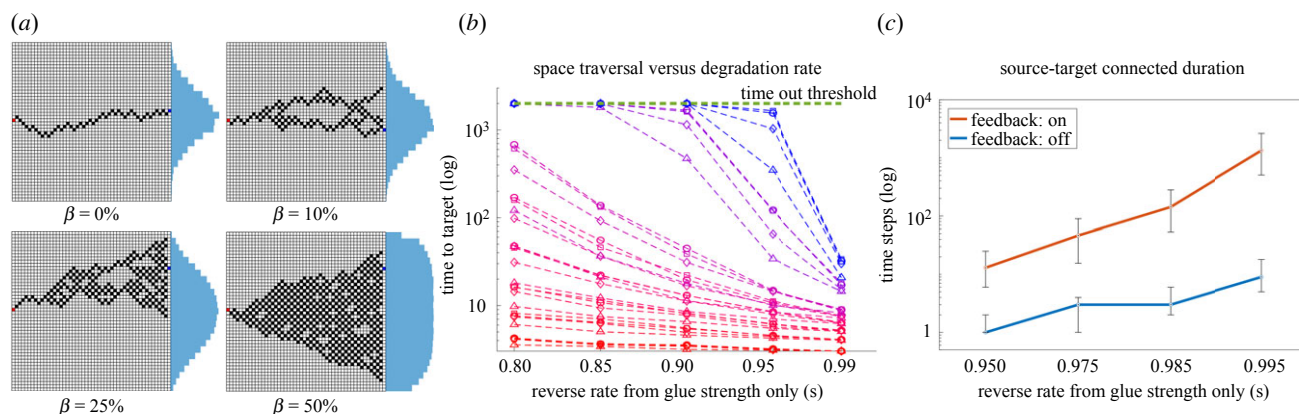


Figure 3. (a) Simulation of the Y-STAM for $\beta = 0\%$, 10% , 25% , 50% . Histograms show the tail-end distribution of the pathway ($N = 5000$). Each mean is centred on the position of the source, but higher branching can expand coverage to a more uniform bandwidth. (b) Effect of binding strength s on the mean duration to reach a target distance ($N = 100$). Colour gradient from red to blue shows increments of $n = \{5, 6, 7, 8, 9, 10, 15, 20\}$ from smallest to largest. In the same listed order, circle, square, diamond, and triangle markers indicate $\beta = \{0, 0.01, 0.1, 0.5\}$. To limit runtimes, simulations time out after 2000 time-steps. Note that time to target is reported in log scale. (c) For $n = 25$, after the pathway has activated feedback, the number of time-steps before the feedback-activated pathway degrades is recorded ($N = 100$). In situations where the pathway has discovered the goal tile, a feedback signal significantly improves the persistence of that path by approximately one magnitude for glue strengths $s > 0.95$. Differences are less distinct at lower values.

comprehensively add and discuss branching, degradation, and feedback, in that order. In general, we see that adding more diverse parametrization to the model increases the upper bounds of likelihood for successfully locating the goal from the source location. However, higher parametrization also increases resource expenditure, in this case, tiles, as well as the time spent in the growth phase prior to finding the goal.

4.1. A random walk model for growth of the Y-STAM tile-based self-assembly

For the following simulations, let the position of σ_s be at a fixed location across all simulation instances, and let $\mathcal{A}(\mathcal{T})$ denote possible resulting assemblies. Without branching, all possible assemblies $\mathcal{A}(\mathcal{T})$ are a random walk that begin at Position $(\sigma_s) = (1, \lceil n/2 \rceil)$, with σ_G randomly placed at any valid location in the rightmost column of the grid. This simplistic model is biased towards finding a goal tile directly horizontally across from the source tile position. The success of this process is highly dependent on the size of the space n and the relative location of σ_G to σ_s . In this case, σ_s requires *a priori* knowledge (that the goal tile will be located near the mean) to maximize its likelihood of finding σ_G , but this is not the behaviour we desire to achieve.

These chances improve if the branching factor $\beta > 0$. The branching factor sets the probability by which a branching tile is placed at a growth front, whereupon at the next time-step, the branching tile will bind one additional tile to each of its two output glues, thus branching the pathway. We enforce some rules to avoid potentially uninterpretable results. Growth is limited to remain within the $n \times n$ grid within $\mathbb{Z}^2$. If a tile would grow out of the boundaries of the simulation, it cannot bind. In some rare cases, this terminates the growth of the assembly. This is also done for collisions when new tiles are added to two different sub-paths and attempt to occupy the same position. While all growth fronts are evaluated per time-step, it is still done sequentially. Thus, no priority is defined other than giving precedence to the tile already added to the assembly.

We quantify the differences in the distribution of the tail-end (rightmost) tile of each branched pathway in the assembly and repeat these trials for $N = 5000$ distinct independent executions. Each trial is terminated upon first reaching the last column. Since $r_f = 1$, all assemblies grow at the same speed, and all branching pathways reach the rightmost column of the space simultaneously. Comparing different values of β for $n = 50$, branching causes the coverage of target locations to become an almost uniform distribution (figure 3a). Intuitively, when branching tiles are added into the assembly at a higher frequency, it also increases the chances of the assembly saturating all possible tile positions, and the total assembly more closely resembles a uniform, conical growth than a stochastic process. However, excessive branching can be a wasteful use of material if the reward space is small, so we look towards different methods for the system to circumvent failure.

4.2. Degradation of branches of a Y-STAM tile-based self-assembly

In the previous simulation, the pathway was programmed to locate the target within a fixed time corresponding to the size of the space n . Biological processes often use degradation as a penalty or an opportunity to retrain and reattempt a process. In our model, degradation is implemented as a weaker glue strength ($s < 1$) corresponding to a reverse rate $r_b = 1 - s > 0$, and a glue breakage (separation of adjacent tiles) can occur if $\Pr(X < r_b)$, for $X \sim \mathcal{U}[0, 1]$. When glue breakage occurs, it may also cause an entire sub-assembly to break off with it (figure 2a (disassembly)). Since breakage is evaluated independently for every glue, this effect becomes exponential with respect to the length of a pathway, and there will be a length where the rate of growth that is linear to the number of branches is unlikely to overcome the exponentially scaling degradation likelihood. Thus, there is a soft limit to the farthest distance that can be travelled by any branch that could also be considered the maximum size of the assembly.

Our simulations tested fixed glue strengths $s = \{0.8, 0.85, 0.9, 0.95, 0.99\}$ across varying n and β (figure 3b). The simulation was set to terminate once any path fully traversed the x -axis space, regardless of whether it finds the target. Our simulations showed that as the space becomes larger, more successive failures owing to having more particles along which any branch could completely fail impedes the ability of the total pathway to grow to the maximum size of the space. In each case of s , this is slightly offset by the branching factor, as having more paths is akin to having redundancy against failure. However, this does not overcome the cost of distance.

4.3. Feedback-responsive strengthening along a branch of a Y-STAM tile-based self-assembly

Next, we simulated feedback (figure 2a (feedback)). Intuitively, the assembly reacting to a moiety at the goal location triggers a signal along the direct pathway back to the source tile, indicating that preserving that pathway should be prioritized over other branches. Many organisms respond to stimuli, such as chemical (chemotropism) or physical (thigmotropism). If any path of the tile assembly does reach the target tile, we evaluate the efficacy of a squared improvement (instead of one glue, tiles are then connected by two glues of equal strength) to the binding strength against the amount of time a pathway can stay connected. A signal is propagated through the backward-channel glues, switching their states similarly to the forward-channel propagation of the signal. In our stochastic model, a tile is considered disassembled if both glues are disconnected within a single time-step. If only one glue is disconnected and both glues are on, then the disconnected glue can be evaluated for reconnection on the next time-step based on the existing r_f of the system. For $r_f = 1$, the reconnection is immediate. Thus, for any tile to break is the product of the independent probabilities of each glue to break, or $Pr(X < r_b)^2$.

We may liken this to the biological process of finding ‘nutrients’, which in this case, is the goal tile. If a pathway successfully finds nutrients, it becomes favourable and ideally preserved. We evaluate this using the *connected time*, which is the amount of time any path in the network connects σ_S to σ_G . Since the likelihood of path breakage is exponential with respect to the path length, while the feedback benefits are quadratic, we expect that the path preservation of feedback is most effective when r_b is low. We measure this by connecting σ_S and σ_G and counting the time-steps until the connection breaks. At $n = 25$ and low r_b (figure 3c), connected times between having feedback on and off differ by about one magnitude. These differences become less noticeable at higher environment sizes and higher degradation rates.

The above properties have defined the fundamental processes of branching, known as the intrinsic properties. We next describe the influence provided by extrinsic effectors. These factors can optimize the operation of the branching program for its specific purpose and environment while the program itself (the design of the tiles) has not changed.

5. Definitions of the Y-STAM⁺ and simulations of resulting behaviours

We now define a family of various augmentations of the basic Y-STAM to address its interactions with external factors. We term this family of model augmentations Y-STAM⁺. These are presented as further formalizations and simulations to introduce time- and space-inhomogeneous external modulators (extrinsics) into the model, which can change the behaviour of the tile assembly system without changing any properties that are intrinsic to the tile set or assembly itself, such as the branching factor β or glue strengths s . This additional complexity can enable systems to fulfil more complex activities, such as using memory, following paths, time-dependent growth, or complex network construction.

The Y-STAM exhibits sufficient functionality to be responsive to bidirectional signalling and has methods to prune itself, but otherwise still has few strategies to prioritize and reinforce positive behaviours. In coming up with external signalling factors, we focused on basic and intuitive properties of time-based and space-based changes, such as decay, occurring with respect to time and state, and gradients of temperature or chemicals, affecting growth with respect to space. We note that these are speculative and broad factors that have not been correlated to empirical observations. In particular, we define *memory*, which alters the reverse rate according to time elapsed and can be localized to position. Pre-programmed memory can also define spatial gradients, such as those caused by physical, chemical, or temperature factors. We also introduce the influence of *tile concentration* in this part of the model, which correlates the reverse rate to the state of the assembly with additional complexity to the feedback mechanism. We now present formal definitions that supplement the existing definitions in §2.

We first distinguish the previous construction of the assembly $\mathcal{A}(\mathcal{T})$ as the *tiling layer*. We also now formally define *tile presence* at time i and position $(\hat{x}, \hat{y})$, which is the position of a tile in $\mathbb{Z}^2$, as $\delta^{(i)}(\hat{x}, \hat{y})$, such that $\delta^{(i)}(\hat{x}, \hat{y}) = 1$ when a tile is attached and 0 otherwise.

We next define an example augmented layer as a similar two-dimensional grid of the same size that we term the *memory layer*, $\mathcal{M}$. Each position of this grid is a *memory signal* $m_{(\hat{x}, \hat{y})} = (\hat{m}^{(i)}, \Phi(\hat{m}^{(i)}, \Psi(\delta^{(i)}(\hat{x}, \hat{y})), \kappa(s, \hat{m}^{(i)})))$, where $\hat{m}^{(i)} \in \mathbb{R}^+$ is the magnitude of the memory signal, or *memory strength*, at time-step i , $\hat{m}^{(i+1)} = \Phi(\hat{m}^{(i)})$ is a time-responsive update rule that defines how the memory strength changes with respect to time, $\hat{m}^{(i+1)} = \Psi(\delta^{(i)}(\hat{x}, \hat{y}))$ is an event-based update rule that defines the change in the memory strength with respect to tile attachment, and $r_b = \kappa(s, \hat{m}^{(i)})$ maps the memory strength and glue strength to the disassembly rate of the tile and its glue at the corresponding position. The $(\hat{x}, \hat{y})$ subscript indicates the position of the memory signal on the grid, which corresponds to the position of a tile within the tiling layer given previously by $Position(\hat{t}) = (\hat{x}, \hat{y})$, where $\hat{t}$ is a single, specific tile. For succinctness, we henceforth refer to Ψ , Φ , and κ without their arguments.

We first elaborate on the time-responsive update rule of Φ . Let $\omega, \omega^* \in \mathbb{R}$ be *memory rates*, and ω is fixed while the value of ω^* is the *tile-based modifier* of the memory rates which will be either 1 or a fixed positive or negative real number value depending on whether there is a tile present at the corresponding coordinates $(\hat{x}, \hat{y})$ of the memory signal. In cases where $\omega^* > 0$, it modulates the rate of memory strength deterioration, and in cases where $\omega^* < 0$, it modulates the rate of memory strength accrual. We define these two separate variables to distinguish different interactions between the assembly and the environment for when, for example, an

organism is present compared to when it is not. In our simulations, we chose a simple first-order, half-life rate of degradation to attenuate the memory signal and we define $\Phi(\hat{m}^{(i)}) = \hat{m}^{(i)}/2^{\omega\omega^*}$. This represents a half-life reaction of growing or degrading the memory strength, proportional to the magnitude and sign of the product $\omega\omega^*$.

Next, the event-based update rule is $\Psi(\hat{m}^{(i)}) = \max(m_{ref}, \hat{m}^{(i)} + \psi m_{ref})$, where $\psi \in [0, 1]$ is the *deposit rate* and $m_{ref} \in \mathbb{R}_{\neq 0}$ is a reference value. Both values are fixed across the memory layer. Ψ is only applied if, at time i , a tile is added in the assembly at position $(\hat{x}, \hat{y})$ that was not previously there at time-step $i-1$. This serves to describe an interaction with the environment when the assembly first arrives, which may displace or consume existing resources or deposit its own. When a new tile is placed, the memory strength at that position is immediately offset, approaching a maximum set by the reference value m_{ref} at the rate ψ . Ψ takes precedence over Φ , that is, the memory strength for time-step $i+1$ following the attachment of a tile at time i will first be increased by Ψ , then immediately attenuated for the value of one time-step by Φ .

Finally, κ defines the disassembly rate of the tile at the corresponding position $(\hat{x}, \hat{y})$ with respect to the memory strength and the glue strength. κ completes the loop to allow the memory layer to feed back its values and affect the tiling layer. It is based on the sigmoid function and transformed for the range $S(x) = [-1, 1]$ such that the resulting reverse rate $r_b = [0, 1]$. This provides a simple function to handle negative and positive memory strength symmetrically while tapering excess memory to have diminishing returns. Its piecewise definitions are shown in equations (5.1)–(5.6):

$$\hat{m}^{(i)} < 0 : r_b = \kappa(s, \hat{m}^{(i)}) \quad (5.1)$$

$$= 1 - \left[s + s \left(\frac{2}{1 + e^{-\hat{m}^{(i)}}} - 1 \right) \right], \quad (5.2)$$

$$\hat{m}^{(i)} > 0 : r_b = \kappa(s, \hat{m}^{(i)}) \quad (5.3)$$

$$= 1 - \left[s + (1-s) \left(\frac{2}{1 + e^{-\hat{m}^{(i)}}} - 1 \right) \right] \quad (5.4)$$

$$\text{and} \quad \hat{m}^{(i)} = 0 : r_b = \kappa(s, \hat{m}^{(i)}) \quad (5.5)$$

$$= 1 - s. \quad (5.6)$$

In our simulations, we make observations of various behaviours of the tile growth/disassembly activity that occur within 2000 time steps. To achieve this, simulation settings were chosen to: (i) avoid committing to spurious growths. If memory signals are set to increase too rapidly, the assembly generally behaves no differently than if glue strength $s = 1$, and the assembly rapidly completes, without any exploration. (ii) If a pathway is repeatedly preferred, then a corresponding memory signal reaches the threshold for exponential growth after multiple, repeated, previously incomplete attempts to grow in that direction, thereby reinforcing preferred behaviours. In this context, *incompleteness* means that the pathway may have grown along that direction but later decayed before discovering the goal tile. However, since it encountered no obstacles, it is inclined to try again in that direction. This behaviour can also be modulated to cause the inverse to happen. That is, long, unsuccessful paths are less likely to be retread, such as when encountering an obstacle (electronic supplementary material, figure S3). These interactions and relationships are also summarized in figure 4a.

Next, we define the effects of adding a tile concentration influence on the Y-STAM. Let $[t]^{(i)}$ denote the concentration of tile type t at time-step i . Previously, the forward rate was $r_f = 1$ by default. Here, we follow Michaelis–Menten kinetics to define a new forward binding rate $r_f = [t]^{(i)} / (K_d + [t]^{(i)})$. First, a tile type is selected from the tile set by building a probability distribution function from the concentrations of all tile types at time-step i . The random variable $X \sim U[0, 1]$ first selects a tile type from the distribution, then random variable $Y \sim U[0, 1]$ determines whether the tile will associate, $Pr(\text{associate}) = Pr(Y < r_f)$. Let $c_t^{(i)}$ be any number of instances of a single tile type added to the assembly at a time-step i , then $[t]^{(i+1)} = [t]^{(i)} - c_t^{(i)}$. For simulations in this paper, $K_d = 1$.

5.1. Externalized memory in an overlaid memory layer

In simulations, we implemented the memory layer as a secondary grid that is updated after each growth step according to the update rules Ψ and Φ . For glue strengths $s = [0, 1]$, the effect of memory signals becomes asymptotically equivalent close to $\hat{m}^{(i)} = 5$ (figure 4b), so we generally used a maximum reference memory strength of $m_{ref} = 5$. In all instances, combining the memory signal and feedback further lowers the reverse rate.

First, we evaluated only the effects of κ on the disassembly rate using a pre-set and fixed memory layer. Figure 4c shows a V-shaped track set to the reference memory strength $m_{ref} = 5$. Without influencing tile placement and only the disassociation of tiles, the overall direction and geometry of the assembly could be influenced. The memory layer allows us to lower glue strengths (e.g. $s = 0.5$), which will promote a more noticeable expression of the effects of the memory layer. Akin to signalling from an extracellular matrix, the memory layer can now dictate how the assembly grows. In this example, the growth of the pathway followed the border defined by the track, whereupon a high degradation rate severely impeded any growth outside of the track.

We then simulated the addition of the event-based rule (Ψ) and degradation of the memory layer as specified by the time-responsive update rule (Φ) into the memory layer updates. We initialized a memory layer where all memory strengths were zero. Then, for example, we could set the following parameters: $\psi = 0.25$ causes the placement of a new tile into the assembly to deposit a quarter of $m_{ref} = 5$ into the corresponding memory layer position. $\omega = 0.2$ defines a disassembly rate such that the magnitude of each memory signal is reduced per time-step with a half-life of five time-steps. $\omega^* = -0.1$ would cause all memory signal locations with a tile present in the corresponding location in the tiling layer to increase the magnitude of the memory signal with a doubling rate of every 50 time-steps. Pathways that degrade will leave a slowly attenuating signal that dampens the disassembly rate for any later growth (figure 4d).

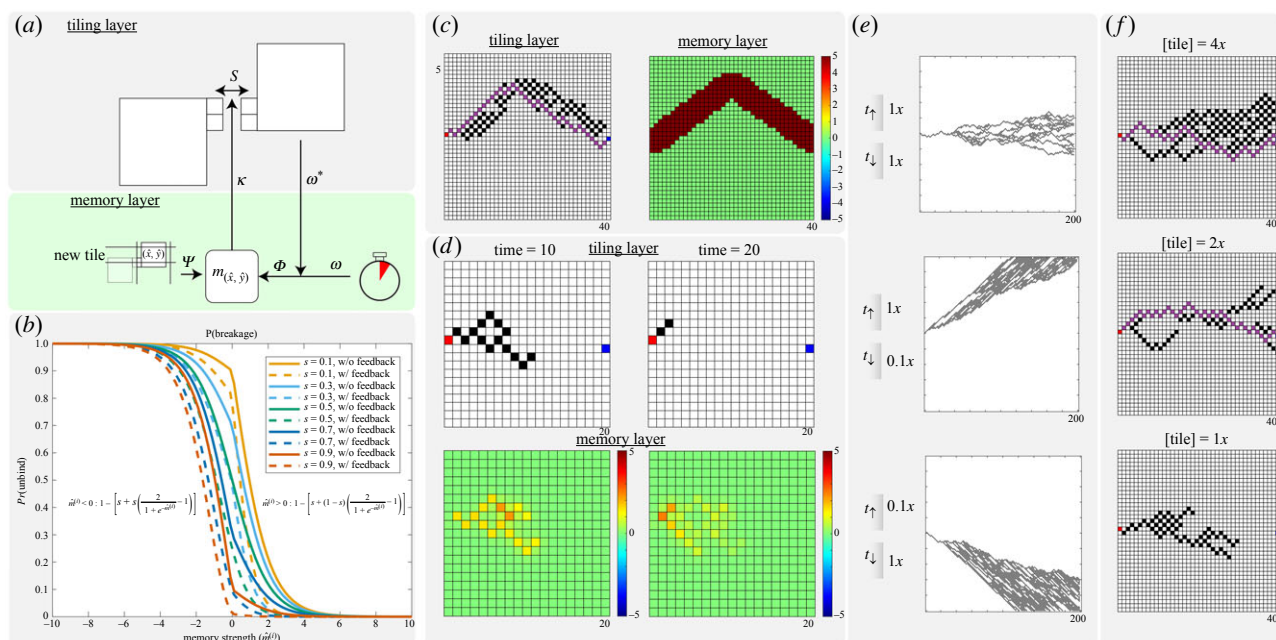


Figure 4. Evaluating behaviours enabled in the augmented Y-STAM. (a) The memory layer is an overlaid layer that remembers magnitudes per time-step $\hat{m}^{(i)}$ that can increase or decrease the disassembly rate r_b relative to the glue strength s per discrete location in the space with some desired relationship κ . The memory strength at each position can also be made time-changing and responsive to tile presence. (b) Our simulations implement a sigmoidal relationship between $\hat{m}^{(i)}$ and s defined piecewise for position and negative $\hat{m}^{(i)}$ to ensure that $s, r_b \in [0, 1]$. Feedback and memory strength can cooperatively strengthen pathways. (c) Pre-set memory can restrict the growing pathway to follow a specific shape. The shape and memory weights are shown in the memory layer (right), while the resulting tile assembly is shown in the tiling layer (left). (d) Values in the memory layer can also be updated with respect to tile position and time. Even after an assembly may have degraded, values in the memory layer decay slower and dampen the immediate effect of reverse binding rates when new tiles grow in those positions again. (e) Non-uniform tile concentrations can change the overall shape of the assembly to grow in biased directions. Grid lines are removed here for visibility. (f) Tile concentrations programmed with respect to the grid size (e.g. for $n = 40$, $1x = 40$). Greater concentration can control the time-dependent behaviour of the assembly. As tile concentrations are sufficiently high, the assembly grows across the space to find the goal tile and then gradually prunes excess branches. However, when tile concentrations are initially low, the assembly cannot grow far enough before being overcome by degradation effects. Snapshots are at $i = 100$.

Moreover, we can also duplicate another layer of memory signals where $m_{ref} < 0$. In certain cases, like collisions with an obstacle, a separate layer accommodates a different set of stimuli for activating Ψ . While one memory layer continues to reinforce pathways, the second memory layer can be programmed to promote disassociation around the obstacle, which can help the pathway circumnavigate the obstacle (electronic supplementary material, figure S6).

The presence of degradation and enhancement in the memory layer is crucial such that memory signals are not simply a baseline increase of the glue strengths of tiles that exist on the path. If a feedback pathway exists for a sufficiently long enough time, the effect of the memory layer should increase with respect to time. Conversely, repeated but failed pathways should not compound memory for easier growth if no goal is actually achieved. Ideally, the variables introduced by the memory layer for their own moderation and their effect on the tiling layer are each set to retain promising tile growth while degrading old paths, although we have not evaluated what the optimal values may be.

5.2. Using localized tile concentration to modulate growth rates to model variation of ‘nutrition’ over two dimensions

For high β , the assembly nearly saturates all possible growth locations. Furthermore, even after a pathway has located the goal tile, other branches may continue propagating from branching tiles along the feedback-activated pathway, which is a wasteful expenditure when the assembly is already in an accepting state. Intuitively, we can relate this to the concept of having finite ‘nutrition’ that can be metabolized to support growth. A tile concentration parameter can simulate the consumption of finite growth material, thus limiting the maximum size of the assembly and enforcing an active duration, after which no further observable activity occurs.

To evaluate the effect of tile concentration, we set the tile concentration as a function of the size of the environment n , where $1x = n$, and evaluated the assembly at the following uniform initial tile concentrations: $4x, 3.5x, 3x, 2.5x, 2x, 1.75x, 1.5x, 1.25x, 1x, 0.875x, 0.75x, 0.675x$. It should be noted that other than discerning trends, this has no bearing on actual concentrations.

We observed a time-based dynamism in the behaviour of the branching assembly. While tile concentrations were high, growth proceeded as usual for any β , including highly saturating growth. However, as the assembly continued to grow and consume tiles, a reduction in r_f caused some growth fronts to fail to bind a tile at each time-step, and the overall growth rate of the assembly began to slow down. If any pathway had already found the goal tile and triggered a feedback signal, then that single pathway was most likely preserved while other branching pathways began to grow slower and eventually disappear. Eventually, the tile concentration fell far enough such that the forward rate of binding new tiles was low enough that the assembly appeared relatively static. Alternatively, without feedback or finding the goal tile, the assembly would eventually fully degrade, leaving only the source and goal tiles and no other activity.

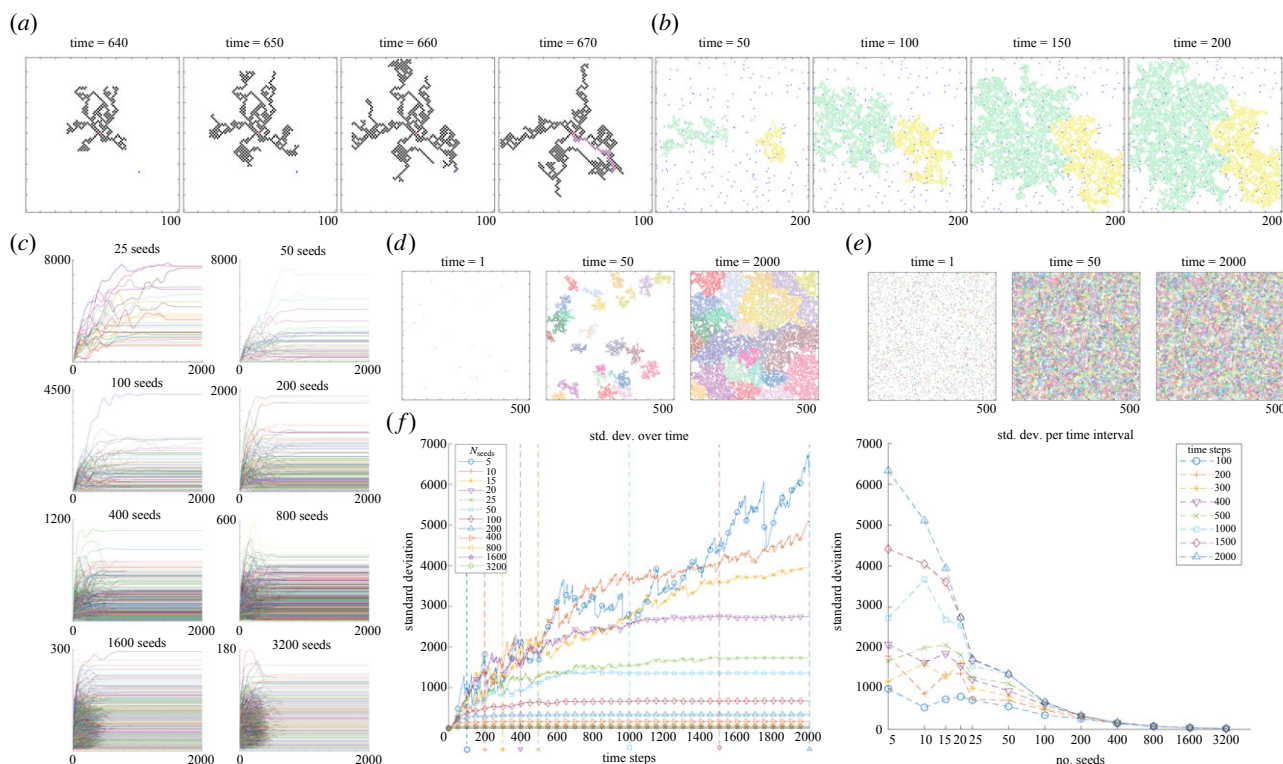


Figure 5. Demonstrations of behaviours of the Y-STAM in a full two-dimensional space. (a) Feedback operates similarly to its one-dimensional growth counterpart but takes longer owing to the quadratically larger space. (b) Competitive growth illustrates the analogy of goal tiles to nutrition, which strengthens the number of supported pathways, thus allowing the structure to grow faster. Tiling behaviour shows how many assemblies of varied sizes self-organize to fill the lattice space competitively. (c) Graphs show the growth of each seeded assembly; line colours match tile colours. Note the decreasing scale of the y -axis. (d) Snapshots of the grid at $i = 1, 50, 2000$ for 25 seeds. (e) Snapshots of the grid at $i = 1, 50, 2000$ for 3200 seeds. (f) Standard deviation of each seed quantity over time. (g) A cross-section of the previous chart shows the convergence of an assembly from initialization to its steady state.

Figure 4f compares snapshots of assemblies at $i = 100$ and $n = 40$ for $[t] = 4x, 2x$ and $1x$. At $[t] = 4x$, while the assembly had found the goal tile, it also exhibited a large amount of excess growth. At $[t] = 2x$, there was already sufficient time to find the goal tile, but by then, there was also a low concentration of remaining tiles that could grow excess pathways. However, at $[t] = 1x$, the snapshot at $i = 100$ captures when the assembly does not have a high enough tile concentration to reliably grow far enough to reach or find the goal tile. Electronic supplementary material, figure S4, looks at overall trends across $N = 100$ repeated trials of each tile concentration value, and we note that there could be fairly precise ranges whereupon growth of a branching system is optimized between its function and its available resources.

6. Simulation of the two-dimensional Y-STAM⁺

In the above sections, simulations and discussions proceeded based on the unidirectional implementation of the Y-STAM. We now discuss simulations that support two-dimensional growth across the grid by including turning tiles within the tile set. We do not consider assemblies without branching, as the results are somewhat trivial. We use feedback, varying support matrix settings, and randomized placement of numerous source and goal tiles to demonstrate emergent behaviours that include a majority system between two competing growths and the self-organization of a high number of nucleated assemblies. Owing to the high computational costs of simulating numerous trials of the two-dimensional implementation compared to the unidirectional implementation, we discuss these results by observing single cases. As the growth of pathways can halt on collisions with themselves, the concentration of turning tiles is also lower than other tile types to avoid frequently forming looped pathways.

6.1. Feedback-supported growth

First, we reproduced the ability to receive feedback from a goal tile within the two-dimensional implementation of the Y-STAM⁺ (figure 5a). Unlike the unidirectional implementation where the goal tile was placed at the rightmost column of the lattice, thus greatly constraining the search space, for simulations in two dimensions, the goal tile can be placed anywhere in the space, albeit, still on the corresponding $\text{mod } 2$ position such that the diagonally growing pathway can discover it. As before, β must be adjusted so that the area is searched efficiently. Similar values appeared to be effective ($\beta = 25\%$), as there were still enough branched pathways to mostly saturate the surrounding space of the seed tile. Furthermore, we allowed the source tile to nucleate assemblies from each of its four corners. That is, it had all possible output glues. Owing to the limited data to make quantitative observations, we could not conclusively say whether this two-dimensional implementation took longer to find the goal tile than the unidirectional implementation, but we expect that it does. Its subsequent behaviour to retain the feedback-strengthened pathway remained the same.

We used feedback behaviour to demonstrate the competitive growth of two separately seeded structures. Two source tiles were placed in central positions at opposite halves of the grid ($n = 200$), while 200 goal tiles were randomly placed throughout the grid. For this simulation, we liken each structure created through a branching self-assembly process to an 'organism' and the goal tiles to 'nutrients'. As the organism grew, it discovered nearby nutrients that sent a feedback signal along its pathway back to the source, thus strengthening its growth in that direction. Along this pathway, branching tiles that propagated alternative subpaths could be formed. If those should degrade, the pathway did not need to restart from the source tile but could regrow from the existing branching tile on the feedback-reinforced pathway. Thus, the rate at which each organism grew was correlated to the amount of nutrients it could more rapidly access.

As this process continued, the two organisms eventually collided, thereby disabling growth along the fronts that they did. Along non-colliding fronts, they continued to grow. It follows that it can be implicitly measured which assembly is more well-nourished after some time has passed, and it was visually obvious which assembly was larger (figure 5b). Next, suppose we denote the larger organism *A* and the smaller organism *B*. As the simulation continued to run, *A* underwent drastic degradation by chance, thus allowing *B* to grow into some of the previously occupied space. However, the same could also happen to *B*. Over time, *A* was still the larger organism more frequently. This process represents a random distribution of goal tiles into a self-assembled physical form, showing which of our organisms had easier access to goal tiles to support its growth, essentially measuring clustering.

6.2. Space-filling tiling behaviour

In the next simulation, we evaluated the effect of crowding and the free space available for each assembly to grow. The space was initialized with no goal tiles to avoid randomly biasing any growth. Fifty randomly placed source tiles were added. Each assembly grew at the same rate until it was large enough to collide with neighboring assemblies. We used a negative ω^* to cause assemblies to strengthen over time (the presence of a tile in that position increases the magnitude of the memory signal after every time-step) and eventually only rarely degrade. Thus, instead of oscillating trade-offs in occupied space, we eventually reached a steady state.

We simulated twelve increments of randomly placed seeds at roughly exponential benchmarks (the number of seeds, $N_{\text{seeds}} = \{5, 10, 15, 20, 25, 50, 100, 200, 400, 800, 1600, 3200\}$), with grid size $n = 500$, and a time limit of $i_{\text{max}} = 2000$ steps. We also prevented source tiles from being initialized next to each other (which could immediately cause a collision).

We tracked the number of tiles in each assembly as its *population*, and the population of an assembly at time i is henceforth denoted as ρ_i . Figure 5c shows the populations of several assemblies for a single run at each seed quantity, and figure 5d,e shows examples of the assembly at 25 seeds and 3200 seeds at initialization $i = 1$, then later at $i = 50$, and $i = 2000$. Each assembly is colour-coded with its population chart (see electronic supplementary material, figures S2–S9, for more time-step instances of each seed quantity). We used a moving average filter with a window size of 100 time-steps to reduce noise when viewing each population chart. Generally, it seems consistent across all seed quantities that the number of assemblies in the lower half of the population sizes is typically two to three times more than those in the upper half.

To observe the volatility of growth, we calculated the standard deviation of the population of assemblies at time-step i to see how quickly each seed quantity reached a steady state (figure 5f). In a steady state, assemblies have stopped growing owing to colliding with neighbours but are also not degrading owing to a strong memory signal. At low seed density ($N_{\text{seeds}} = 5$), there is growth activity all the way up until and likely past the time limit. As the number of seeds increases, the available space to grow per assembly is consumed much faster, and the entire environment reaches a steady state much sooner. Moreover, we take a cross-section of the trends for the standard deviation of each seed quantity over time to observe the scale between their initial state and steady state. For high seed quantities ($N_{\text{seeds}} > 200$), individual assemblies in the grid have barely grown before already colliding with another assembly.

Crowding, as well as the distributions shown in figure 5c, could hinder the total environment from achieving enough assemblies of the target size. Furthermore, these results suggest that for randomly growing branching structures with the purpose of saturating a space, there are diminishing returns to initializing more independent growths. When varying the branching factor and tracking assembly sizes (electronic supplementary material, figure S5), the growth trends also corroborate with established growth patterns, such as defined by the Gompertz function, and similarly shown for crystal growth [79–82]. If, in a general sense, growth costs energy, then in situations where those resources are limited, such as either time or nutrition, there are perhaps optimal distributions of growth loci and parameters to saturate a space as efficiently as possible. Or, making assumptions about the functional size of an organism or the cost of nucleating new growths, the simulation results support the evaluation of optimal points to maximize the efficiency of branching growths, their interfacial area, and so on.

7. Future work

7.1. Implementing tiles from the Y-STAM as active DNA nanostructures

Previously suggested implementations of STAM [71,75] used localized DNA strand-displacement reactions to mediate signal-passing state changes between glues. Here, we describe in greater detail using a DNA domain-based description of a signal-passing model that could provide an implementation of branching processes of the Y-STAM.

Figure 6 follows several transitions of a tile that has bound and then later becomes unbound from an assembly, with only forward-signalling channels shown. Backward-signalling channels are implemented by reversing the design orientation whilst relabelling domains to limit cross-reactivity as needed but are otherwise not shown to simplify the illustration. The illustration shows the cycle of a free-floating tile in solution that, beginning in its initial state, binds a tile already on the assembly, propagates

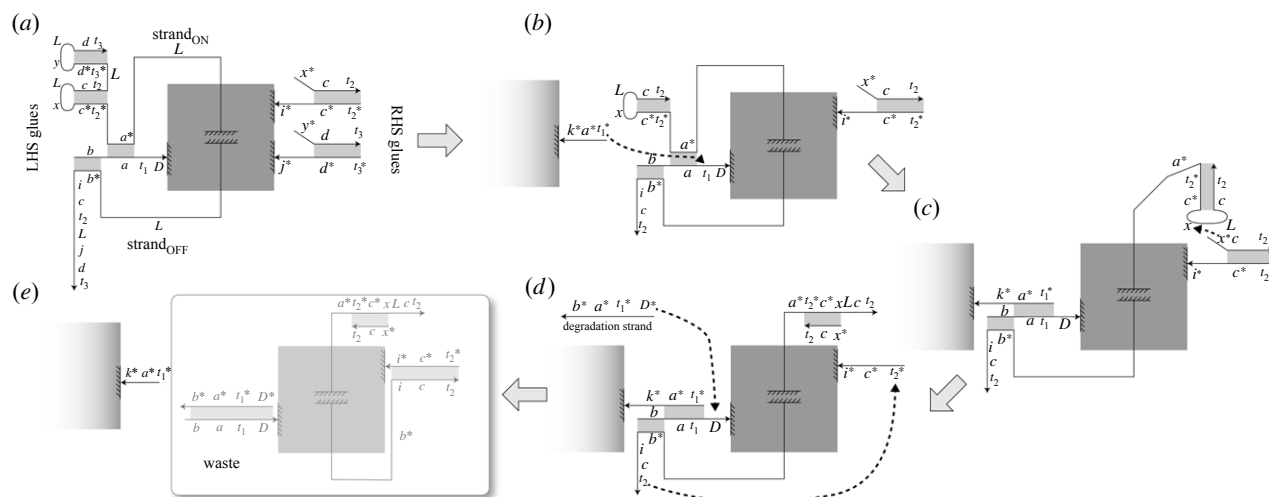


Figure 6. Domain-based description of potential branching tile implementation. Here, we follow an example progression of states as a tile propagates and then loses a forward-channel signal. Only forward-channel components are shown. Anchors indicate that a strand is attached to the tile, and arrows represent the 3' end of strands. (a) The initial state of a branching tile. Each tile has four main components: inputs (LHS glues), outputs (RHS glues) and localized signal-passing (Strand_{ON}) and signal-enforcing (Strand_{OFF}) strands. For simplicity, subsequent steps show standard tiles simply by removing the corresponding signal-passing and signal-enforcing domains following the removal of one output. L is a spacer domain that is also further omitted. (b) An output glue of a tile already on the pathway binds the t_1 toehold on the input of a free-floating tile. This also displaces Strand_{ON} to propagate the forward signal to turn the output glue on. (c) Strand_{ON} displaces the protected strand x^*cb via the internal hairpin loop toehold x , activating the output glue to on. (d) A glue–glue bond is reversible or can be made irreversible by degradation strands present in the solution. The degradation strand displaces Strand_{OFF}, which was physically separated from the output glue via localization but can now fully bind and switch the output glue to off. (e) The degradation strand deactivates the input, and the signal-enforcing strand deactivates the output, resulting in a waste tile. The tile still attached to the assembly may now bind another tile.

the forward-channel signal, then assuming a glue–glue bond breakage, shuts off its output glue to enforce a loss of signal. Since glues cannot be converted from off to any other state, the tile must be fully deactivated such that its deactivated output glue does not act as a terminus for this assembly if it were to bind to the assembly again. Thus, a corresponding degradation strand also deactivates its input glue, and the tile becomes waste.

The concentration of degradation strands can bias the reverse binding rate separately from the intrinsic binding strength of glues. Higher concentration increases the degradation rate by more frequently removing tiles from the assembly but should not be so high as to prevent the growth of the assembly altogether owing to the continually exposed degradation strand toehold D . However, this design may allow flexible modulation of r_b rather than using only fixed sequence- or temperature-dependent rates. The degradation strand has a toehold D shared across all instances of the degradation strand and all tile types, but it can be further addressed by t_i and the other glue–glue binding domains (e.g. a , b) that can be made to be unique per tile type. Thus, its concentration should be adjusted relative to its corresponding tile type.

Implementing the extrinsic modulators part of the augmented model likely requires a spatially aligned layer of strand displacement activity at addressable locations, with this layer affixed relative to the locations of the source and goal tiles. Tiling and memory layers may be implemented by anchoring tiles onto a two-dimensional substrate, such as a lipid bilayer or a sufficiently large DNA array. Positive memory would downregulate the presence of the degradation strand by competitively binding its toehold domains, while negative memory would upregulate the degradation strand, perhaps by acting as a catalytic amplifier to duplex species carrying additional copies of the corresponding degradation strand. However, we can currently only conceptually articulate this concept, as discussing a full design is likely too complex to be within the scope of this paper. Owing to the reliance on localization, tiles may already need to be quite large to keep reactive strands physically separate. In addition, given the number of tile types and reactions that must occur locally, each tile type's unique domains and variations may already need to be very high to reduce erroneous crosstalk. Accommodating positional addressability for communication between the tiling and memory layers would only further increase this complexity.

7.2. Future challenges

An implementation of the Y-STAM, even without augmentations, is most likely a very complex ordeal owing to the number of interacting strands that must be localized to a single substrate. This does not necessarily have to be DNA, but nevertheless, DNA nanostructures are still one of the most accurate physically addressable biomaterials. Moreover, components conducive to constructing branched assemblies have already been demonstrated in DNA nanotechnology, such as branched tiles with multi-arm junctions, although the assemblies they have produced are most commonly regular lattices that lack the dynamism resembling of the assemblies shown in this paper [83]. While we believe careful design and operation could realize the suggested implementation of a Y-STAM tile set, an implementation accommodating the modulating layers in the Y-STAM⁺ may not be impossible but most likely require a complex system of both confined reactions and physical supports, resembling of very complex natural biological environments.

Furthermore, as we have delineated several functions of branching processes here, it remains to be seen how they can be set to align more closely with natural signalling pathways. It is not the mere replication of branching behaviours that is important in this

model, but rather, it is the quantifiable parametrization of biochemical reactions that it enables. The model may provide insights on which chemical controls may be necessary to invoke which corresponding behaviours, but that does not provide any framework beyond continuing to empirically search for what the specific biological counterparts may be.

As we have defined, in the augmented Y-STAM⁺, an extensible framework for adding different definitions to modulate the growth of branching assemblies, it also means that the examples that we have sufficed to demonstrate may be only a small portion of possible modulators. Moreover, it is likely in biological counterparts that modulators themselves may interact between themselves combinatorially before affecting the tiles themselves. We have not accommodated that possibility here, but we do not limit ourselves from doing so by recalling that the augmented model is allowed to add layers and is not limited to the single memory layer we have shown here.

While we have defined the Y-STAM and its augmentations in two dimensions, this does not accurately reflect many of the natural biological examples of branching assembly that grow in three dimensions. However, assuming the necessary direction placements of glues are made to accommodate a voxel geometry, we expect the semantics of our branching model to remain essentially the same. Experimentally, though, few examples of three-dimensional tile-based self-assembly that preserve the algorithmic dynamism of tile-based assembly have been demonstrated [84–86]. Yet, owing to the stochastic construction and sparse, network-like structure of assemblies produced using a Y-STAM foundation, it may be more adaptable to being supported in other mediums, such as DNA hydrogels [87,88].

The following is an enumeration of some possible future modifications to our Y-STAM and Y-STAM⁺:

- (i) It is clearly evident how much even a single memory layer enables the tile assembly system to achieve complex behaviours. As mentioned, we do not restrict the model to a single layer of extrinsic modulators. Furthermore, complex systems often have many interwoven, interacting assemblies whose influences consolidate into creating a consequent, functional branching network. Future work will seek to simulate similar effects, such as combinatorial interactions between layers of the model.
- (ii) We also note that the model and implementations do not rule out adding more glues than the two shown in these demonstrations, which, in reference to our simulations, could improve the effectiveness of feedback beyond only a squared improvement against the reverse binding rate, but at the cost of greater implementation complexity.
- (iii) While we have defined a mode of interaction that modulates glue strengths, additional modes of interaction from additionally defined layers would most likely enable further leaps in complexity. In particular, paths are not sensitive to their own length, and disassembly closer to the root can be overly punishing. Allometric theory [89,90] suggests the presence of factors that strike an equilibrium with respect to size. Attenuating signals from the source tile or being able to mark tiles at specific intervals along the path (e.g. tip, middle, root) for disassembly could still be within the DNA nanotechnology toolbox to implement [91].
- (iv) Current signal-passing modes only have single tile resolution. Additional operations that activate specific single glues on each tile could assist in improving the complexity of geometries relative to the scale of the assembly, elapsed duration, and propagated signals by allowing, for instance, standard tiles to transition to branching tiles.
- (v) Current growth parameters do not specify any direction of growth. An additional layer that records a local history of turn angles of each tile along the pathway will assist in controlling the overall direction of specific paths in the assembly and facilitate the creation of more patterned geometries.
- (vi) The scalability of the model towards novel functions may include catalytic activity (changes in tile concentration based on attachment events or consumption of other tiles) and orthogonal moieties (two assemblies that may interact, but whose tiles and assembly happen in parallel). However, the computational costs of the simulation scale with the simulated size of the assembly.

We note a few challenges and issues facing experimental demonstration of our Y-STAM and Y-STAM⁺:

- (i) The local reactivity of strands may make it such that each tile may not be able to be synthesized in one-pot assemblies and instead may need to be synthesized piecemeal, separating reaction strands until they have been appropriately protected.
- (ii) The concentration of the degradation strand relative to the tiles must also be carefully set so as not to disable tiles faster than the assembly can grow owing to the always exposed *D* domain, although, as previously mentioned, this can be somewhat ameliorated by uniquely addressing input glue domains.
- (iii) The design of each DNA tile requires an optimization of the unique hybridization domains (i.e. their lengths and sequence content) of each component DNA sequence of the tile, and the optimization of the DNA sequences implementing glue–glue bonds between tiles. Furthermore, while the STAM has been previously demonstrated [75], branching and backwards signal propagation adds a significant degree of complexity to the tile design. It is reasonable not to be confined to a single tile and consider simulating various strand displacements that must occur for a single growth or disassembly step with a staged assembly, using multiple tiles to simulate one. However, the full details of such an implementation are out of the scope of this paper.

8. Conclusion

In this paper, we presented two novel formal models for stochastic branching processes that have capabilities (as found in many natural biological branching processes) for receiving stimuli, performing two-way communication along their branches, and dynamically reforming to extend or retract branches. We have presented a formal definition of the Y-STAM, as well as random walk

analysis and software simulations. We also introduced the Y-STAM⁺ and simulated external influences that can affect the Y-STAM assembly processes. We further presented domain-level DNA-based designs. In upcoming work, we intend to experimentally demonstrate the Y-STAM to observe similarities to specific natural biological branching systems, such as networks of neurons and capillaries. We also intend to simulate more complex instances of the model. Looking further outward, we suggest that these reward-based self-assembly models could potentially assist with understanding and engineering biomimetic, self-assembling technologies, especially those that increasingly exist in biological mediums or integrate biocompatible materials.

While there has been a growing mathematical, computational, and empirical understanding of branching self-assembly processes, we believe the models presented in this paper are a first ever foray into attempting to physically engineer branching self-assembly processes. DNA-based self-assembly has proven to be a versatile substrate to implement deterministic computation and structural self-assembly with thus far unmatched precision at the nanoscale. However, that itself has also been encroaching upon the limits of implementing complex functions in synthesized nanodevices when using only explicit design methodologies. Engineering branching self-assembly processes offers an opportunity to exploit ‘intelligent’, goal-oriented self-assembly that more closely resembles natural systems. In a sense, replicating natural models from the perspective of bottom-up self-assembly serves as a bridge between artificial systems created by nanoscale engineering and the manipulation of natural systems in synthetic biology.

We anticipate that the potential to physically implement the Y-STAM using DNA-based tiles can provide greater insight into branching self-assembly systems. Synthesizing DNA-based systems is a compromise between computational studies and speculative environments whose resolution is limited by computational power and empirical studies in excised tissue with potentially many unknown or uncontrollable variables. On a conceptual level, the Y-STAM can be decoupled with the shape restriction of tiles and, more generally, illustrates an understanding of how structures can form via branching that is controlled simply by the binding affinities and availability of its individual units. Gradients, deterioration, and catalysts are straightforwardly captured by concepts already common to DNA nanotechnology. The Y-STAM introduces another perspective upon these preexisting concepts to envision novel modes of natural, non-deterministic self-assembly.

Ethics. This work did not require ethical approval from a human subject or animal welfare committee.

Data accessibility. MATLAB source code for the Y-STAM simulator can be found at <https://github.com/dfu99/Y-STAM> [92]. The version used in the current work (v0.9-beta) is also archived by Zenodo and can be downloaded at <https://doi.org/10.5281/zenodo.10398468> [93].

The data are provided in electronic supplementary material [94].

Authors' contributions. D.F.: conceptualization, data curation, formal analysis, investigation, methodology, project administration, resources, software, validation, visualization, writing—original draft, writing—review and editing; J.R.: conceptualization, funding acquisition, project administration, resources, supervision, writing—review and editing.

Both authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. This work was supported by NSF Award no. CCF-2113941.

References

- Kelkar AD, Herr DJ, Ryan JG. 2014 *Nanoscience and nanoengineering: advances and applications*. Boca Raton, FL: CRC Press.
- Fiorello I, Del Dottore E, Tramacere F, Mazzolai B. 2020 Taking inspiration from climbing plants: methodologies and benchmarks—a review. *Bioinspi. Biomim.* **15**, 031001. (doi:10.1088/1748-3190/ab7416)
- Müller D, Leyser O. 2011 Auxin, cytokinin and the control of shoot branching. *Ann. Bot.* **107**, 1203–1212. (doi:10.1093/aob/mcr069)
- Evers JB, van der Krol AR, Vos J, Struik PC. 2011 Understanding shoot branching by modelling form and function. *Trends Plant Sci.* **16**, 464–467. (doi:10.1016/j.tplants.2011.05.004)
- Domagalska MA, Leyser O. 2011 Signal integration in the control of shoot branching. *Nat. Rev. Mol. Cell Biol.* **12**, 211–221. (doi:10.1038/nrm3088)
- Boswell GP, Jacobs H, Ritz K, Gadd GM, Davidson FA. 2007 The development of fungal networks in complex environments. *Bull. Math. Biol.* **69**, 605–634. (doi:10.1007/s11538-005-9056-6)
- Boddy L, Hynes J, Bebbler DP, Fricker MD. 2009 Saprotrophic cord systems: dispersal mechanisms in space and time. *Mycoscience* **50**, 9–19. (doi:10.1007/S10267-008-0450-4)
- Reid CR, Latty T, Dussutour A, Beekman M. 2012 Slime mold uses an externalized spatial ‘memory’ to navigate in complex environments. *Proc. Natl Acad. Sci. USA* **109**, 17 490–17 494. (doi:10.1073/pnas.1215037109)
- Nakagaki T, Yamada H, Toth A. 2001 Path finding by tube morphogenesis in an amoeboid organism. *Biophys. Chem.* **92**, 47–52. (doi:10.1016/S0301-4622(01)00179-X)
- Nakagaki T, Yamada H, Hara M. 2004 Smart network solutions in an amoeboid organism. *Biophys. Chem.* **107**, 1–5. (doi:10.1016/S0301-4622(03)00189-3)
- Ma Q, Johansson A, Tero A, Nakagaki T, Sumpter DJ. 2013 Current-reinforced random walks for constructing transport networks. *J. R. Soc. Interface* **10**, 20120864. (doi:10.1098/rsif.2012.0864)
- Czaczkes TJ, Weichselgartner T, Bernadou A, Heinze J. 2016 The effect of trail pheromone and path confinement on learning of complex routes in the ant *Lasius niger*. *PLoS ONE* **11**, e0149720. (doi:10.1371/journal.pone.0149720)
- Van Oudenhove L, Billoir E, Boulay R, Bernstein C, Cerdá X. 2011 Temperature limits trail following behaviour through pheromone decay in ants. *Sci. Nat.* **98**, 1009–1017. (doi:10.1007/s00114-011-0852-6)
- Robinson E, Green K, Jenner E, Holcombe M, Ratnieks F. 2008 Decay rates of attractive and repellent pheromones in an ant foraging trail network. *Insectes Soc.* **55**, 246–251. (doi:10.1007/s00040-008-0994-5)
- Czaczkes TJ, Heinze J. 2015 Ants adjust their pheromone deposition to a changing environment and their probability of making errors. *Proc. R. Soc. B* **282**, 20150679. (doi:10.1098/rspb.2015.0679)
- Davis GE, Senger DR. 2005 Endothelial extracellular matrix: biosynthesis, remodeling, and functions during vascular morphogenesis and neovessel stabilization. *Circ. Res.* **97**, 1093–1107. (doi:10.1161/01.RES.0000191547.64391.e3)
- Clause KC, Barker TH. 2013 Extracellular matrix signaling in morphogenesis and repair. *Curr. Opin. Biotechnol.* **24**, 830–833. (doi:10.1016/j.copbio.2013.04.011)

18. Kleinman HK, Philp D, Hoffman MP. 2003 Role of the extracellular matrix in morphogenesis. *Curr. Opin. Biotechnol.* **14**, 526–532. (doi:10.1016/j.copbio.2003.08.002)
19. Hannezo E, Scheele CL, Moad M, Drogo N, Heer R, Sampogna RV, Van Rheenen J, Simons BD. 2017 A unifying theory of branching morphogenesis. *Cell* **171**, 242–255. (doi:10.1016/j.cell.2017.08.026)
20. Uçar MC, Kameneyev D, Sunadome K, Facht D, Lallemand F, Adameyko I, Hadjab S, Hannezo E. 2021 Theory of branching morphogenesis by local interactions and global guidance. *Nat. Commun.* **12**, 6830. (doi:10.1038/s41467-021-27135-5)
21. Long H, Ou Y, Rao Y, van Meyel DJ. 2009 Dendrite branching and self-avoidance are controlled by Turtle, a conserved IgSF protein in *Drosophila*. *Development* **136**, 3475–3484. (doi:10.1242/dev.040220)
22. Jan YN, Jan LY. 2010 Branching out: mechanisms of dendritic arborization. *Nat. Rev. Neurosci.* **11**, 316–328. (doi:10.1038/nrn2836)
23. Lang C, Conrad L, Michos O. 2018 Mathematical approaches of branching morphogenesis. *Front. Genet.* **9**, 673. (doi:10.3389/fgene.2018.00673)
24. Lu P, Takai K, Weaver VM, Werb Z. 2011 Extracellular matrix degradation and remodeling in development and disease. *Cold Spring Harb. Perspect. Biol.* **3**, a005058. (doi:10.1101/cshperspect.a005058)
25. Rozario T, DeSimone DW. 2010 The extracellular matrix in development and morphogenesis: a dynamic view. *Dev. Biol.* **341**, 126–140. (doi:10.1016/j.ydbio.2009.10.026)
26. Metzger RJ, Krasnow MA. 1999 Genetic control of branching morphogenesis. *Science* **284**, 1635–1639. (doi:10.1126/science.284.5420.1635)
27. Varner VD, Nelson CM. 2014 Cellular and physical mechanisms of branching morphogenesis. *Development* **141**, 2750–2759. (doi:10.1242/dev.104794)
28. Ochoa-Espinosa A, Affolter M. 2012 Branching morphogenesis: from cells to organs and back. *Cold Spring Harb. Perspect. Biol.* **4**, a008243. (doi:10.1101/cshperspect.a008243)
29. Iber D, Menshykau D. 2013 The control of branching morphogenesis. *Open Biol.* **3**, 130088. (doi:10.1098/rsob.130088)
30. Hu MC, Rosenblum ND. 2003 Genetic regulation of branching morphogenesis: lessons learned from loss-of-function phenotypes. *Pediatr. Res.* **54**, 433–438. (doi:10.1203/01.PDR.0000085170.44226.DB)
31. Kowalska A, Boruta T, Bizukojc M. 2019 Kinetic model to describe the morphological evolution of filamentous fungi during their early stages of growth in the standard submerged and microparticle-enhanced cultivations. *Eng. Life Sci.* **19**, 557–574. (doi:10.1002/elsc.201900013)
32. Prosser J. 1995 Mathematical modelling of fungal growth. In *The growing fungus* (eds NAR Gow, GM Gadd), pp. 319–335. Berlin, Germany: Springer.
33. Shinde B, Khan S, Muhuri S. 2020 Model for growth and morphology of fungal mycelium. *Phys. Rev. Res.* **2**, 023111. (doi:10.1103/PhysRevResearch.2.023111)
34. Stillman NR, Mayor R. 2023 Generative models of morphogenesis in developmental biology. *Semin. Cell Biol.* **147**, 83–90. (doi:10.1016/j.semcdb.2023.02.001)
35. Mordvintsev A, Randazzo E, Niklasson E, Levin M. 2020 Growing neural cellular automata. *Distill* **5**, e23. (doi:10.23915/distill.00023)
36. Sudhakaran S, Grbic D, Li S, Katona A, Najarro E, Glanois C, Risi S. 2021 Growing 3D artefacts and functional machines with neural cellular automata. In *Proc. 2021 Conf. on Artificial Life*, 18–22 July 2021. Cambridge, MA: MIT Press.
37. Codling EA, Plank MJ, Benhamou S. 2008 Random walk models in biology. *J. R. Soc. Interface* **5**, 813–834. (doi:10.1098/rsif.2008.0014)
38. Yamamura K, Moriya S, Tanaka K. 2003 Discrete random walk model to interpret the dispersal parameters of organisms. *Ecol. Modell.* **161**, 151–157. (doi:10.1016/S0304-3800(02)00345-9)
39. Bordeu I, Amarteifio S, Garcia-Millan R, Walter B, Wei N, Pruessner G. 2019 Volume explored by a branching random walk on general graphs. *Sci. Rep.* **9**, 15590. (doi:10.1038/s41598-019-51225-6)
40. Prusinkiewicz P, Lindenmayer A. 2012 *The algorithmic beauty of plants*. New York, NY: Springer Science & Business Media.
41. Rozenberg G, Salomaa A. 2012 *Lindenmayer systems: impacts on theoretical computer science, computer graphics, and developmental biology*. New York, NY: Springer Science & Business Media.
42. Okubo A, Levin SA. 2001 *Diffusion and ecological problems: modern perspectives*, vol. 14. Berlin, Germany: Springer.
43. Wolfram S. 1984 Cellular automata as models of complexity. *Nature* **311**, 419–424. (doi:10.1038/311419a0)
44. Johnston IG, Ahnert SE, Doye JP, Louis AA. 2011 Evolutionary dynamics in a simple model of self-assembly. *Phys. Rev. E* **83**, 066105. (doi:10.1103/PhysRevE.83.066105)
45. Ahnert SE, Johnston IG, Fink TM, Doye JP, Louis AA. 2010 Self-assembly, modularity, and physical complexity. *Phys. Rev. E* **82**, 026117. (doi:10.1103/PhysRevE.82.026117)
46. Laszlo JA, Silman RW. 1993 Cellular automata simulations of fungal growth on solid substrates. *Biotechnol. Adv.* **11**, 621–633. (doi:10.1016/0734-9750(93)90031-H)
47. Halley JM, Robinson CH, Comins HN, Dighton J. 1996 Predicting straw decomposition by a four-species fungal community: a cellular automaton model. *J. Appl. Ecol.* **33**, 493–507. (doi:10.2307/2404979)
48. Poorkhanalikhdeh A, Zimmermann KH. 2021 Cellular automaton for kidney branching morphogenesis. *WSEAS Trans. Biol.* **18**, 170–182. (doi:10.37394/23208.2021.18.20)
49. Markus M, Böhm D, Schmick M. 1999 Simulation of vessel morphogenesis using cellular automata. *Math. Biosci.* **156**, 191–206. (doi:10.1016/S0025-5564(98)10066-4)
50. Beros A, Chyba M, Fronville A, Mercier F. 2018 A morphogenetic cellular automaton. In *2018 Annual American Control Conf. (ACC), Milwaukee, WI, USA, 27–29 June 2018*, pp. 1987–1992. (doi:10.23919/ACC.2018.8431498)
51. Wang H. 1961 Proving theorems by pattern recognition—II. *Bell Syst. Tech. J.* **40**, 1–41. (doi:10.1002/j.1538-7305.1961.tb03975.x)
52. Winfree E. 1998 Algorithmic self-assembly of DNA. PhD thesis, California Institute of Technology, Pasadena, CA, USA.
53. Yin P, Sahu S, Turberfield AJ, Reif JH. 2006 Design of autonomous DNA cellular automata. In *DNA computing: DNA 2005* (eds A Carbone, NA Pierce), pp. 399–416. Berlin, Germany: Springer. (doi:10.1007/11753681_32)
54. Hendricks J, Patitz MJ. 2013 On the equivalence of cellular automata and the tile assembly model. (<http://arxiv.org/abs/1309.1273>)
55. Cantu AA, Luchsinger A, Schweller R, Wylie T. 2020 Signal passing self-assembly simulates tile automata. In *31st Int. Symp. on Algorithms and Computation*, pp. 53:1–53:17. (doi:10.4230/LIPIcs.ISAAC.2020.53)
56. Soloveichik D, Seelig G, Winfree E. 2010 DNA as a universal substrate for chemical kinetics. *Proc. Natl Acad. Sci. USA* **107**, 5393–5398. (doi:10.1073/pnas.0909380107)
57. Short KM, Combes AN, Lisnyak V, Lefevre JG, Jones LK, Little MH, Hamilton NA, Smyth IM. 2018 Branching morphogenesis in the developing kidney is not impacted by nephron formation or integration. *Elife* **7**, e38992. (doi:10.7554/eLife.38992)
58. Lefevre JG, Short KM, Lamberton TO, Michos O, Graf D, Smyth IM, Hamilton NA. 2017 Branching morphogenesis in the developing kidney is governed by rules that pattern the ureteric tree. *Development* **144**, 4377–4385. (doi:10.1242/dev.153874)
59. Combes AN. 2018 Haploinsufficiency for the Six2 gene increases nephron progenitor proliferation promoting branching and nephron number. *Kidney Int.* **93**, 589–598. (doi:10.1016/j.kint.2017.09.015)
60. Bellusci S, Grindley J, Emoto H, Itoh N, Hogan BL. 1997 Fibroblast growth factor 10 (FGF10) and branching morphogenesis in the embryonic mouse lung. *Development* **124**, 4867–4878. (doi:10.1242/dev.124.23.4867)
61. Pepicelli CV, Lewis PM, McMahon AP. 1998 Sonic hedgehog regulates branching morphogenesis in the mammalian lung. *Curr. Biol.* **8**, 1083–1086. (doi:10.1016/S0960-9822(98)70446-4)

62. Sekine K *et al.* 1999 Fgf10 is essential for limb and lung formation. *Nat. Genet.* **21**, 138–141. (doi:10.1038/5096)
63. Min H, Danilenko DM, Scully SA, Bolon B, Ring BD, Tarpley JE, DeRose M, Simonet WS. 1998 Fgf-10 is required for both limb and lung development and exhibits striking functional similarity to *Drosophila branchless*. *Genes Dev.* **12**, 3156–3161. (doi:10.1101/gad.12.20.3156)
64. Dey S *et al.* 2021 DNA origami. *Nat. Rev. Methods Primers* **1**, 13. (doi:10.1038/s43586-020-00009-8)
65. Evans CG, O'Brien J, Winfree E, Murugan A. 2022 Pattern recognition in the nucleation kinetics of non-equilibrium self-assembly. (<http://arxiv.org/abs/2207.06399>)
66. Rothmund PWK, Papadakis N, Winfree E. 2004 Algorithmic self-assembly of DNA Sierpinski triangles. *PLoS Biol.* **2**, e424. (doi:10.1371/journal.pbio.0020424)
67. Demaine ED, Demaine ML, Fekete SP, Ishaque M, Rafalin E, Schweller RT, Souvaine DL. 2008 Staged self-assembly: nanomanufacture of arbitrary shapes with $O(1)$ glues. *Nat. Comput.* **7**, 347–370. (doi:10.1007/s11047-008-9073-0)
68. Aggarwal G, Cheng Q, Goldwasser MH, Kao MY, De Espanes PM, Schweller RT. 2005 Complexities for generalized models of self-assembly. *SIAM J. Comput.* **34**, 1493–1515. (doi:10.1137/S0097539704445202)
69. Chandran H, Gopalkrishnan N, Reif J. 2013 Tile complexity of approximate squares. *Algorithmica* **66**, 1–17. (doi:10.1007/s00453-012-9620-z)
70. Winfree E. 1998 *Simulations of computing by self-assembly*. Technical report, CaltechCSTR: 1998.22. Pasadena, CA: Caltech.
71. Padilla JE, Patitz MJ, Pena R, Schweller RT, Seeman NC, Sheline R, Summers SM, Zhong X. 2013 Asynchronous signal passing for tile self-assembly: fuel efficient computation and efficient assembly of shapes. In *Unconventional computation and natural computation* (eds G Mauri, A Dennunzio, L Manzoni, AE Porreca), pp. 174–185. Berlin, Germany: Springer. (doi:10.1007/978-3-642-39074-6_17)
72. Padilla JE, Liu W, Seeman NC. 2012 Hierarchical self assembly of patterns from the Robinson tilings: DNA tile design in an enhanced tile assembly model. *Nat. Comput.* **11**, 323–338. (doi:10.1007/s11047-011-9268-7)
73. Hendricks J, Padilla JE, Patitz MJ, Rogers TA. 2013 Signal transmission across tile assemblies: 3D static tiles simulate active self-assembly by 2D signal-passing tiles. In *19th Int. Conf. on DNA Computing and Molecular Programming, Tempe, AZ, USA, 22–27 September 2013*, pp. 90–104. Berlin, Germany: Springer.
74. Mohammed AM, Šulc P, Zenk J, Schulman R. 2017 Self-assembling DNA nanotubes to connect molecular landmarks. *Nat. Nanotechnol.* **12**, 312–316. (doi:10.1038/nnano.2016.277)
75. Padilla JE, Sha R, Kristiansen M, Chen J, Jonoska N, Seeman NC. 2015 A Signal-passing DNA-strand-exchange mechanism for active self-assembly of DNA nanostructures. *Angew. Chem.* **127**, 6037–6040. (doi:10.1002/ange.201500252)
76. Xu H, Sun M, Zhao X. 2017 Turing mechanism underlying a branching model for lung morphogenesis. *PLoS ONE* **12**, e0174946. (doi:10.1371/journal.pone.0174946)
77. Patitz MJ. 2014 An introduction to tile-based self-assembly and a survey of recent results. *Nat. Comput.* **13**, 195–224. (doi:10.1007/s11047-013-9379-4)
78. Doty D, Lutz JH, Patitz MJ, Schweller RT, Summers SM, Woods D. 2012 The tile assembly model is intrinsically universal. In *2012 IEEE 53rd Annual Symp. on Foundations of Computer Science, New Brunswick, NJ, USA, 20–23 October 2012*, pp. 302–310. (doi:10.1109/FOCS.2012.76)
79. Bolos VJ, Benitez R, Eleta-Lopez A, Toca-Herrera JL. 2019 A probabilistic model for crystal growth applied to protein deposition at the microscale. *Materials* **12**, 479. (doi:10.3390/ma12030479)
80. Teran AV, Bill A, Bergmann RB. 2010 Time-evolution of grain size distributions in random nucleation and growth crystallization processes. *Phys. Rev. B* **81**, 075319. (doi:10.1103/PhysRevB.81.075319)
81. Yang F, Yuan K, Stack AG, Starchenko V. 2022 Numerical study of mineral nucleation and growth on a substrate. *ACS Earth Space Chem.* **6**, 1655–1665. (doi:10.1021/acsearthspacechem.1c00376)
82. Sui X, Cheng Y, Zhou N, Tang B, Zhou L. 2018 Molecular dynamics simulation of the solidification process of multicrystalline silicon from homogeneous nucleation to grain coarsening. *CrystEngComm* **20**, 3569–3580. (doi:10.1039/C8CE00767E)
83. Heuer-Jungemann A, Liedl T. 2019 From DNA tiles to functional DNA materials. *Trends Chem.* **1**, 799–814. (doi:10.1016/j.trechm.2019.07.006)
84. Ke Y. 2014 Designer three-dimensional DNA architectures. *Curr. Opin. Struct. Biol.* **27**, 122–128. (doi:10.1016/j.sbi.2014.07.010)
85. Durand-Lose J, Hendricks J, Patitz MJ, Perkins I, Sharp M. 2020 Self-assembly of 3-D structures using 2-D folding tiles. *Natural Comput.* **19**, 337–355. (doi:10.1007/s11047-019-09751-9)
86. Zhang DY, Hariadi RF, Choi HM, Winfree E. 2013 Integrating DNA strand-displacement circuitry with DNA tile self-assembly. *Nat. Commun.* **4**, 1965. (doi:10.1038/ncomms2965)
87. Morya V, Walia S, Mandal BB, Ghoroi C, Bhatia D. 2020 Functional DNA based hydrogels: development, properties and biological applications. *ACS Biomater. Sci. Eng.* **6**, 6021–6035. (doi:10.1021/acsbomaterials.0c01125)
88. Um SH, Lee JB, Park N, Kwon SY, Umbach CC, Luo D. 2006 Enzyme-catalysed assembly of DNA hydrogel. *Nat. Mater.* **5**, 797–801. (doi:10.1038/nmat1741)
89. West GB, Brown JH, Enquist BJ. 1997 A general model for the origin of allometric scaling laws in biology. *Science* **276**, 122–126. (doi:10.1126/science.276.5309.122)
90. Aguilar-Trigueros CA, Rillig MC, Crowther TW. 2017 Applying allometric theory to fungi. *ISME J.* **11**, 2175–2180. (doi:10.1038/ismej.2017.86)
91. Del Grosso E, Franco E, Prins LJ, Ricci F. 2022 Dissipative DNA nanotechnology. *Nat. Chem.* **14**, 600–613. (doi:10.1038/s41557-022-00957-6)
92. Fu D, Reif J. 2024 Data from: A biomimetic branching signal-passing tile assembly model with dynamic growth and disassembly. Github. (<https://github.com/dfu99/Y-STAM>)
93. Fu D, Reif J. 2024 A biomimetic branching signal-passing tile assembly model with dynamic growth and disassembly. *Zenodo*. (doi:10.5281/zenodo.10398468)
94. Fu D, Reif J. 2024 Supplementary material: A biomimetic branching signal-passing tile assembly model with dynamic growth and disassembly. Figshare. (doi:10.6084/m9.figshare.c.7232726)