Active Self-Assembly of Algorithmic Shapes and Patterns in Polylogarithmic Time*

Damien Woods[1,2] Ho-Lin Chen[1,2,7] Scott Goodfriend[3]
Nadine Dabby[4] Erik Winfree[1,4,5,6] Peng Yin[1,5,6,8]
[1]Computer Science, [2]Center for Mathematics of Information, [3]Chemical Engineering,
[4]Computation and Neural Systems, [5]Bioengineering, [6]Center for Biological Circuit Design,
California Institute of Technology, Pasadena, CA 91125, USA

Present addresses: [7]Department of Electrical Engineering, National Taiwan University, Taiwan
[8]Wyss Institute for Biologically Inspired Engineering, 3 Blackfan Circle, Boston, MA 02445, USA

woods@caltech.edu, holinc@gmail.com, goodfriend.scott@gmail.com,
ndabby@caltech.edu, winfree@caltech.edu, peng_yin@hms.harvard.edu

ABSTRACT

We describe a computational model for studying the complexity of self-assembled structures with active molecular components. Our model captures notions of growth and movement ubiquitous in biological systems. The model is inspired by biology's fantastic ability to assemble biomolecules that form systems with complicated structure and dynamics, from molecular motors that walk on rigid tracks and proteins that dynamically alter the structure of the cell during mitosis, to embryonic development where large scale complicated organisms efficiently grow from a single cell. Using this active self-assembly model, we show how to efficiently self-assemble shapes and patterns from simple monomers. For example we show how to grow a line of monomers in time and number of monomer states that is merely logarithmic in its length. Our main results show how to grow arbitrary connected two-dimensional geometric shapes and patterns in expected time polylogarithmic in the size of the shape plus roughly the time required to run a Turing machine deciding whether or not a given pixel is in the shape. We do this while keeping the number of monomer types logarithmic in shape size, plus monomers required by the Kolmogorov complexity of the shape or pattern. This work thus highlights the fundamental efficiency advantage of active self-assembly over passive self-assembly and motivates experimental effort to construct active molecular self-assembly systems.

Categories and Subject Descriptors

F.1.1 [**Computation by abstract devices**]: Models of Computation—Computability theory

Keywords

Self-assembly; model of computation; molecular programming; reconfigurable robotics

1 Introduction

One of the main inspirations for our model comes from biology. Embryonic development showcases the capability of molecules to compute efficiently. A human zygote cell contains within it a program that encodes the geometric structure of an organism with roughly 10^{14} cells, that have a wide variety of forms and roles, with each such cell containing up to tens of thousands of proteins and other molecules with their own intricate arrangement and functions. Early stages of embryonic development demonstrate exponential growth rates in the number of cells over time, showing remarkable time efficiency. Not only this, but the developmental path from a single cell to a functioning organism is an intricately choreographed, and incredibly robust, temporal and spatial manufacturing process that operates at the molecular scale. Development is possibly the most impressive example of the ubiquitous process of molecular self-assembly, where relatively simple components (such as nucleic and amino acids, lipids, carbohydrates) organize themselves into larger systems with extremely complicated structure and dynamics (cells, organs, humans).

Molecular programming, where nanoscale engineering is thought of as a programming task, provides our second motivation. The field has progressed to the stage where we can design and synthesize a range of programable self-assembling molecular systems, all with relatively simple laboratory techniques. For example, short DNA strands that form 'tiles' can be self-assembled into DNA tile crystals [6] that are algorithmically patterned into counters and Sierpinski triangles. This form of *passive self-assembly* is theoretically capable of growing arbitrarily complex algorithmically described shapes and patterns. DNA origami can be used to create uniquely addressable shapes and patterns upon which objects can be localized within six nanometer resolution [4]. These systems are static, in the sense that after formation their structure is essentially fixed. However, DNA nanotechnology has seen increased interest in the fabrication of *active* nanostructures that have the ability to dynamically change their structure. Examples include DNA-based walkers, reconfigurable DNA origami, and molecular motors that transition between a small number of discrete states. In these systems the interplay between structure and dynamics leads to behaviors and capabilities not seen in static structures.

Here we suggest a model to motivate engineering of molecular structures that have complicated active dynamics of the kind seen in living biomolecular systems. Our model combines features seen in passive DNA-based tile self-assembly, molecular motors and other active systems, molecular circuits that evolve according to well-mixed chemical kinetics, and even

*Supported by NSF grants CCF-1219274, CCF-1162589, and 0832824—the Molecular Programming Project, an NSF Graduate Fellowship, and The Caltech Center for Biological Circuit Design. Many thanks to Niles Pierce and Patrick Mullen for valuable discussions.

Copyright is held by the author/owner(s).
ITCS'13, January 9–12, 2013, Berkeley, California, USA.
ACM 978-1-4503-1859-4/13/01.

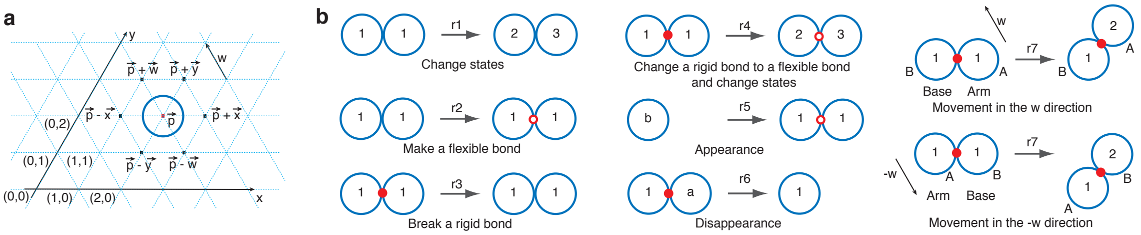

Figure 1: (a) A monomer in the triangular grid coordinate system. (b) Example monomer interaction rules. The movement rule ($r7$) nondeterministically applies one of two possible symmetric movements. When a movement rule is applied, any monomers attached to the arm are pushed along with it.

reaction-diffusion systems. The model is designed to capture the interplay between molecular structure and dynamics. In our model, simple molecular components form assemblies that can grow and shrink. Individual components can undergo state changes and move relative to each other.

The model consists of a two-dimensional grid of monomers, as shown in Figure 1. A specified set of rules, or a program, directs adjacent monomers to interact in a pairwise fashion. Monomers have internal states, and a pair of adjacent monomers can change their state with a single rule application. Monomers can appear and disappear from the grid. So far, the model can be thought of as a cellular automaton of a certain kind (rules are applied asynchronously and can be nondeterministic, and there is a notion of a growth front). An additional rule type allows monomers to move relative to each other. The movement rule is locally applied but propagates movement throughout the system in a very non-local fashion. This geometric and mechanical feature distinguishes our model, the *nubot* model, from previous molecular models and cellular automata, and crucially underlies its construction efficiency. The system evolves as a continuous time Markov process, with rules being applied to the grid asynchronously and in parallel analogous to standard chemical kinetics, modeling the distributed nature of molecular reactions.

The model can carry out local state changes on a grid, so it can easily simulate Turing machines, walkers, and cellular automata-like systems. We show examples of other simple programs such as robotic molecular arms that can move distance n in expected time $O(\log n)$, something that can not be done by cellular automata. By using a combination of monomer movement, appearance, and state change we show how to build a line of monomers in time that is merely logarithmic in the length of the line, something that is impossible in the (passive) abstract tile assembly model [1]. We go on to efficiently build a counter, that counts from 0 to n, within time and number of monomer states that are both logarithmic in n. We build on these results to show that the model is capable of building wide classes of shapes exponentially faster than passive self-assembly. We show how to build a computable shape of size $n \times n$ in time polylogarithmic in n, plus roughly the time needed to simulate a Turing machine that computes whether or not a given pixel is in the final shape. Our constructions are not only time efficient, but efficient in terms of their program-size: requiring at most polylogarithmic monomer types in terms of shape size, plus that required by the Kolmogorov complexity of the shape.

For shapes that have short algorithmic descriptions, but require a lot of time and space to compute their pixels, the previous computable shape construction necessarily requires (temporary) growth beyond the shape boundary. One can ask if there are interesting structures that we can build efficiently, but yet keep all growth confined within the boundary. It turns out that colored patterns, where the color of each pixel in the pattern is computable by a polynomial time Turing machine, can be computed extremely efficiently in this way. More precisely, we show that $n \times n$ colored patterns are computable in expected time $O(\log^{\ell+1} n)$ and using $O(s + \log n)$ states, where each pixel's color is computable by a program-size s Turing machine that runs in polynomial time in the length of the binary description of pixel indices (specifically, in time $O(\log^{\ell} n)$ where ℓ is $O(1)$). This entire construction is initiated by a single monomer and is carried out using only local information and in an entirely distributed and asynchronous fashion.

Our results are related to theoretical work in passive self-assembly [1, 6], chemical reaction networks [5], cellular automata, Lindenmayer systems [3] and reconfigurable robotics [2]—each such model sharing a subset of our active self-assembly model's features. In particular, reconfigurable modular robots with a similar long-range motion primitive to ours can achieve arbitrary reconfiguration in time logarithmic in shape size [2], so it will be interesting to compare such models to ours. Our constructions serve to show that exponentially large complicated shapes and patterns can be fabricated in polynomial time and using a linear number of states, besides the states that are required by the Kolmogorov complexity of the shape or pattern. Thus the model illustrates the power of active local growth along with non-local movement. We hope that the model can serve as the basis of further investigation into the computational properties of active self-assembly.

References

1. Leonard M. Adleman, Qi Cheng, Ashish Goel, and Ming-Deh Huang. Running time and program size for self-assembled squares. In *STOC 2001, pp 740–748.*
2. Greg Aloupis, Sébastien Collette, Erik D. Demaine, Stefan Langerman, Vera Sacristán, and Stefanie Wuhrer. Reconfiguration of cube-style modular robots using $O(\log n)$ parallel moves. In *Algorithms and Computation*, pages 342–353. Springer, 2008.
3. Aristid Lindenmayer. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *J. Theor. Biol.*, 18(3):280–299, 1968.
4. Paul WK Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.
5. David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with finite stochastic chemical reaction networks. *Natural Computing*, 7(4):615–633, 2008.
6. Erik Winfree, Furong Liu, Lisa Wenzler, and Nadrian C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394:539–544, 1998.