

How the Dimension of Space Affects the Products of Pre-Biotic Evolution: The Spatial Population Dynamics of Structural Complexity and The Emergence of Membranes

Steve T. Piantadosi^{1,2,*} and James P. Crutchfield^{2,3,†}

¹*Department of Brain and Cognitive Science,
Massachusetts Institute of Technology, 43 Vassar Street, Cambridge, MA 02139*

²*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501*

³*Complexity Sciences Center and Physics Department,
University of California at Davis, One Shields Avenue, Davis, CA 95616*

(Dated: November 9, 2010)

We show that autocatalytic networks of ϵ -machines and their population dynamics differ substantially between spatial (geographically distributed) and nonspatial (panmixia) populations. Generally, regions of spacetime-invariant autocatalytic networks—or *domains*—emerge in geographically distributed populations. These are separated by *functional membranes* of complementary ϵ -machines that actively translate between the domains and are responsible for their growth and stability. We analyze both spatial and nonspatial populations, determining the algebraic properties of the autocatalytic networks that allow for space to affect the dynamics and so generate autocatalytic domains and membranes. In addition, we analyze populations of intermediate spatial architecture, delineating the thresholds at which spatial memory (information storage) begins to determine the character of the emergent auto-catalytic organization.

Keywords: population dynamics, structural complexity, spatial dimension, autocatalytic network

PACS numbers: 87.18.-h 87.23.Cc 87.23.Kg 05.45.-a 89.75.Kd

I. INTRODUCTION

Almost certainly, the sophisticated mechanisms for self-replication found today in living cells were not present in the earliest replicators [1–3]. Instead, some hypothesized that the first replicators were autocatalytic networks of functional molecules that collectively were capable of self-reproduction [4, 5]. Numerous studies have been devoted to these and analogous networks with the hope of understanding pre-biotic evolution [6–8]. A particular class of such models represented network elements with mathematical constructs, including regular expressions, the λ -calculus [9], and, recently, ϵ -machines [10]. ϵ -Machines are especially useful since they support well defined and computable measures of structural complexity [11] and, equally important, these measures extend directly to networks of interacting ϵ -machines.

Early models often assumed a *Turing gas* in which every network element has an equal chance of interacting with every other, thereby ignoring a network’s spatial configuration. However, natural and engineered evolutionary systems are not architected this way since elements of physical systems always have some spatial relationship that determines which components interact. This observation led to studies of spatial pattern formation in evolutionary and autocatalytic systems [12–14].

Here, we consider networks of single-state

ϵ -machines—a simplifying initial focus that helps to highlight the role of population architecture. We show that the behavior of spatially distributed populations differs substantially from that of the nonspatial Turing gas. More to the point, we determine the algebraic properties of the network that lead to the emergence of distinctive organizations.

II. BACKGROUND

ϵ -Machines serve as models of stochastic finite and infinite computation [15]. An ϵ -machine consists of a set \mathcal{S} of *causal states* and a set of *transitions* between those states. In the particular variation used here (ϵ -transducers), each transition is labeled by both an input symbol $x \in \mathcal{A}$ and an output symbol $y \in \mathcal{A}$. An ϵ -machine in state $\sigma \in \mathcal{S}$ reads an input symbol x and chooses from all transitions from σ the one labeled x . The ϵ -machine follows the chosen transition to a next state $\sigma' \in \mathcal{S}$ while emitting the output symbol y corresponding to that transition. In this way, ϵ -machines can be viewed as mapping an input language to an output language, perhaps probabilistically.

Following Ref. [10], we focus on single-state ϵ -machines over the input-output alphabet $\mathcal{A} = \{0, 1\}$. (Results for multi-state ϵ -machines appear in a sequel. Our goal here is to highlight the effects of space, uncomplicated by the richness that comes with using multi-state ϵ -machines.) There are 16 such ϵ -machines and we denote the set of all them by \mathbb{T} . Each ϵ -machine can be represented as a 2×2 binary matrix M , where $M_{ij} = 1$ means that the ϵ -machine reads in symbol $i-1$ while emitting symbol $j-1$. We number the 16 ϵ -machines— $\mathbb{T} \equiv \{T_0, \dots, T_{15}\}$ —

*Electronic address: piantado@mit.edu

†Electronic address: chaos@ucdavis.edu

by finding the decimal equivalent of the binary number $M_{11}M_{12}M_{21}M_{22}$ for each ϵ -machine. Thus, for example, ϵ -machine T_{11} has the matrix representation:

$$M = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (1)$$

ϵ -Machine pairs interact in a population by composition, under which \mathbb{T} is closed and forms a monoid [16]. With the matrix representation, ϵ -machine composition— $T_b \circ T_a$ —is simply matrix multiplication where, after multiplying, any positive matrix element is set to 1. From this, it is straightforward to compute the *interaction matrix* \mathbb{M} , where $T_a \circ T_b = T_c$ if and only if $\mathbb{M}_{a+1,b+1} = c$, where $a, b, c \in \{0, \dots, 15\}$:

$$\mathbb{M} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 1 & 2 & 3 & 1 & 1 & 3 & 3 & 2 & 3 & 2 & 3 & 3 & 3 & 3 & 3 \\ 0 & 4 & 8 & 12 & 0 & 4 & 8 & 12 & 0 & 4 & 8 & 12 & 0 & 4 & 8 & 12 \\ 0 & 5 & 10 & 15 & 0 & 5 & 10 & 15 & 0 & 5 & 10 & 15 & 0 & 5 & 10 & 15 \\ 0 & 4 & 8 & 12 & 1 & 5 & 9 & 13 & 2 & 6 & 10 & 14 & 3 & 7 & 11 & 15 \\ 0 & 5 & 10 & 15 & 1 & 5 & 11 & 15 & 2 & 7 & 10 & 15 & 3 & 7 & 11 & 15 \\ 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 & 8 & 8 & 8 & 8 & 12 & 12 & 12 & 12 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 0 & 0 & 0 & 5 & 5 & 5 & 5 & 10 & 10 & 10 & 10 & 15 & 15 & 15 & 15 \\ 0 & 1 & 2 & 3 & 5 & 5 & 7 & 7 & 10 & 11 & 10 & 11 & 15 & 15 & 15 & 15 \\ 0 & 4 & 8 & 12 & 4 & 4 & 12 & 12 & 8 & 12 & 8 & 12 & 12 & 12 & 12 & 12 \\ 0 & 5 & 10 & 15 & 4 & 5 & 14 & 15 & 8 & 13 & 10 & 15 & 12 & 13 & 14 & 15 \\ 0 & 4 & 8 & 12 & 5 & 5 & 13 & 13 & 10 & 14 & 10 & 14 & 15 & 15 & 15 & 15 \\ 0 & 5 & 10 & 15 & 5 & 5 & 15 & 15 & 10 & 15 & 10 & 15 & 15 & 15 & 15 & 15 \end{bmatrix}. \quad (2)$$

III. ϵ -MACHINE SOUPS

A population, or simply a *soup*, Γ is a configuration of an $n \times n$ regular toroidal lattice. At each time $t = 0, 1, 2, \dots$, every lattice location (i, j) contains a single ϵ -machine, denoted $\Gamma_{i,j}^t \in \mathbb{T}$. The population size is $N = n^2$. Each location (i, j) is initialized to contain an ϵ -machine uniformly chosen at random from \mathbb{T} .

The population dynamics is specified by ϵ -machine composition, interaction, and update. We define a function $\theta : \mathbb{T} \times \mathbb{T} \times \mathbb{T} \rightarrow \mathbb{T}$ by:

$$\theta(T_a, T_b, T_c) = \begin{cases} T_a \circ T_c & \text{if } T_a \circ T_c \neq T_0 \\ T_b & \text{otherwise} \end{cases}. \quad (3)$$

We write $\theta(A, B, C)$ for sets $A, B, C \subseteq \mathbb{T}$ as shorthand for the set $\{\theta(a, b, c) : a \in A, b \in B, c \in C\}$. We must use θ to prevent the non- ϵ -machine T_0 from being produced since the fact that $T_0 \circ T_a = T_a \circ T_0 = T_0$ for all $T_a \in \mathbb{T}$ implies that if T_0 could be produced, it comes to dominate any population.

Each time step t we choose a location (i, j) at random

and set:

$$\Gamma_{i,j}^{t+1} = \begin{cases} \theta(\Gamma_{i-1,j}^t, \Gamma_{i,j}^t, \Gamma_{i+1,j}^t) & \text{with probability } \frac{1}{4} \\ \theta(\Gamma_{i+1,j}^t, \Gamma_{i,j}^t, \Gamma_{i-1,j}^t) & \text{with probability } \frac{1}{4} \\ \theta(\Gamma_{i,j-1}^t, \Gamma_{i,j}^t, \Gamma_{i,j+1}^t) & \text{with probability } \frac{1}{4} \\ \theta(\Gamma_{i,j+1}^t, \Gamma_{i,j}^t, \Gamma_{i,j-1}^t) & \text{with probability } \frac{1}{4} \end{cases} \quad (4)$$

and set $\Gamma_{k,l}^{t+1} = \Gamma_{k,l}^t$ for all $(k, l) \neq (i, j)$. Thus, two vertical or horizontal neighbors to $\Gamma_{i,j}$ are chosen, composed, and the ϵ -machine resulting from their composition is used to replace $\Gamma_{i,j}^t$, if it is not T_0 . This replacement scheme is meant to be locally analogous to the replacement scheme used in Ref. [10], which will facilitate direct comparisons in the following.

In addition, at each time step a certain amount of diffusion of ϵ -machines occurs due to spatial mixing. For this, let ζ_v be a Gaussian distribution with variance v and mean 0. At each time step, c ϵ -machines are chosen at random in Γ and each is swapped with a random ϵ -machine at a distance chosen from the distribution ζ_v . For more generality, we allow c to be any real number and swap c ϵ -machines per time step on average.

When v and c are large, there is considerable spatial mixing and one expects the dynamics to behave like a nonspatial population, where ϵ -machines have an equal chance of interacting with every other. However, when v and c are small, there is little spatial mixing and, as we shall see, the population dynamics change substantially.

To summarize, as a stochastic dynamical system the soup's state at time t is the population's *configuration* Γ^t . While Eq. (4) determines the local probabilistic update at each site, we use Θ to formally denote the global (probabilistic) update for the entire configuration over one time step:

$$\Gamma^{t+1} = \Theta \circ \Gamma^t. \quad (5)$$

And so, one goal is to understand the trajectories $\{\Gamma^0, \Gamma^1, \dots\}$. Another is to analyze the structure inside the Γ^t . For this, we use $\sigma_{\hat{n}}$ to denote a spatial shift of the soup configuration:

$$\Gamma' = \sigma_{\hat{n}} \circ \Gamma, \quad (6)$$

where $\hat{n} = (\Delta i, \Delta j)$ is the vector by which the configuration is shifted horizontally and vertically:

$$\begin{aligned} \Gamma'_{k,l} &= (\sigma_{\hat{n}} \circ \Gamma)_{k,l} \\ &= \Gamma_{(i+\Delta i) \bmod n, (j+\Delta j) \bmod n}. \end{aligned} \quad (7)$$

Finally, we view the population either as a configuration Γ of spatially positioned ϵ -machines or as a distribution \mathbf{f} of ϵ -machine types without regard to spatial location. The fractions $\mathbf{f} = (f^1, \dots, f^{15})$ of ϵ -machines of type T_a on Γ are $\mathbf{f}^a(\Gamma) = \Pr(T_a \in \Gamma)$.

IV. THE PANMIXIA SOUP

The least-spatial architecture of Γ , where $c \approx N$ and $v \approx n$, was previously studied in Ref. [10]. This is the case of a *panmixia* population, in which all ϵ -machines can interact with any other. When the size of Γ is large, $\mathbf{f}(t) \equiv \mathbf{f}(\Gamma^t)$ follows a simple population dynamics.

First, for large N the discrete-time dynamic is well approximated by continuous time, since each discrete time updates only a single lattice location and this means that changes to Γ^t and so to $\mathbf{f}(t)$ are relatively small ($\propto N^{-1}$). Second, the probability of adding an ϵ -machine T_i is that of picking two ϵ -machines T_a and T_b , such that $T_a \circ T_b = T_i$, times the probability that the ϵ -machine replaced is not T_i . Third, the probability of removing an ϵ -machine T_i is that of picking two ϵ -machines, T_a and T_b , such that $T_a \circ T_b \neq T_i$ and also $T_a \circ T_b \neq T_0$, times the probability of picking a T_i to replace. Thus, the rate of change of $\mathbf{f}^i(t)$ is given by the differential equation:

$$\frac{d\mathbf{f}^i}{dt} = (1 - \mathbf{f}^i) \sum_{T_a \circ T_b = T_i} \mathbf{f}^a \mathbf{f}^b - \mathbf{f}^i \sum_{\substack{T_a \circ T_b \neq T_i \\ T_a \circ T_b \neq T_0}} \mathbf{f}^a \mathbf{f}^b, \quad (8)$$

where the sums run over all pairs satisfying their subscripted condition. That is, $\sum_{T_a \circ T_b = T_i}$ runs over all ordered pairs (T_a, T_b) such that $T_a \circ T_b = T_i$. Equation (8) also determines the steady-state probability distributions of ϵ -machine types, which are simply solutions of:

$$\frac{d\mathbf{f}}{dt} = 0. \quad (9)$$

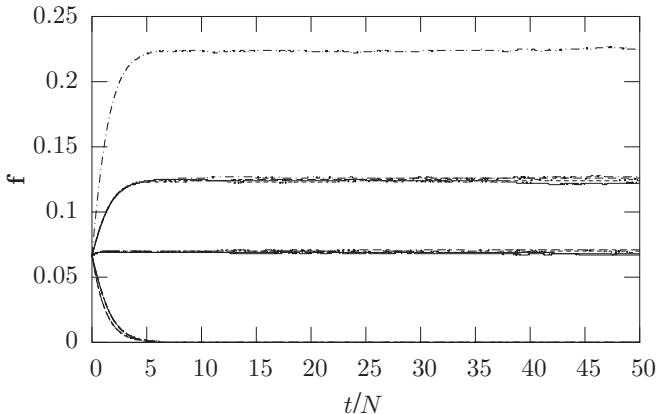


FIG. 1: Panmixia population evolution: ϵ -Machine-type distribution \mathbf{f} as a function of replication time (t/N), with $n = 10^3$, $N = 10^6$, $c = 3000$, and $v = 1000$.

Figure 1 shows the distribution $\mathbf{f}(t)$ over ϵ -machine types as a function of time (replication) for a simulation with $c = 3000$ and $v = 1000$. This is essentially the same population behavior reported in Ref. [10]. It was confirmed numerically that Eq. (8) closely predicts the results in Fig. 1. In Fig. 1, all 16 single-state ϵ -machines

ϵ -Machine Type	Behavior Class
T_{15}	Fast Growth
T_3, T_5, T_{10}, T_{12}	Medium Growth
T_1, T_2, T_4, T_8	No Growth
$T_6, T_7, T_9, T_{11}, T_{13}, T_{14}$	Fast Decay

TABLE I: ϵ -Machine-type behavior classes in the panmixia soup of Fig. 1.

are shown. They partition themselves into four classes such that those in the same class behave similarly; see Table I. Six ϵ -machines die away, while nine persist in a closed, self-maintaining, and dynamically stable *metamachine*, as shown in Ref. [10].

V. GENERAL REPLICATORS

While these ϵ -machine classes can be understood by examining the solutions of Eq. (8) as done in Ref. [10], the results can also be directly predicted by examining the algebraic structure of the set \mathbb{T} under the composition operator determined by \mathbb{M} . And this observation is critical to predicting the behavior of alternate population architectures. First, we address the panmixia population. The central idea is to find subsets of ϵ -machine types that map onto themselves under the population dynamics; in other words, to identify \mathbb{M} -invariant subsets of \mathbb{T} .

Definition 1. A set S of ϵ -machines is a general replicator (GR) if for all $a \in \mathbb{T}$ we have:

$$\theta(\mathbb{T}, a, S) \cup \theta(S, a, \mathbb{T}) \subseteq S \cup \{a\}$$

and

$$\theta(\mathbb{T}, a, S) \cup \theta(S, a, \mathbb{T}) \neq \{a\}.$$

This parallels the definition of an *ideal* in semigroup theory, except that we must be more careful (and less elegant) with the definition since θ is not a binary relation [16].

In \mathbb{T} there are many GRs, including \mathbb{T} itself. We intentionally used θ to exclude the non- ϵ -machine T_0 from being considered a GR. By design, T_0 cannot be produced. GRs are important because they are the simplest form of replicator. Suppose S is a GR. As evolution progresses, elements in the soup are replaced with some element of $\theta(\mathbb{T}, a, S)$ or $\theta(S, a, \mathbb{T})$. By the definition of a GR, though, $\theta(\mathbb{T}, a, S)$ and $\theta(S, a, \mathbb{T})$ contain elements of S and so generally we expect elements of S to produce more elements of S .

However, not all GRs are equal since some are subsets of others and therefore are produced more readily. There are, in fact, many GRs in \mathbb{T} , but there is one GR, call it Ω , that is *minimal* in that no subset of Ω is a GR, but every GR contains Ω . (Note that, generally, such a

set is not guaranteed to exist in a semigroup with a zero element [16].)

Proposition 1. $\Omega = \{T_1, T_2, T_3, T_4, T_5, T_8, T_{10}, T_{12}, T_{15}\}$ is the minimal general replicator in \mathbb{T} .

Proof. By direct verification of Def. 1 and observing that Ω , short any one of its ϵ -machines, is not a GR.

In the panmixia population, we expect that elements of Ω come to dominate the soup. And this is what is observed in the simulations (Fig. 1 and Table I). While Eq. (8) explains the specific values of \mathbf{f} , an understanding of \mathbb{M} 's structure leads to a direct explanation for why the set Fast Decay is removed from the soup: None of the ϵ -machines in Fast Decay are in Ω .

VI. THE SPATIAL SOUP

The population that we consider next is that of a spatially configured soup where $c = 0$ or $v = 0$. In these limits, no spatial mixing is added to the system and the model behaves essentially as an asynchronous, probabilistic cellular automaton. Specifically, each element is replaced by the composition of two of its neighbors if that composition does not result in T_0 . The pair of neighbors composed is chosen with uniform probability, according to Eq. (4).

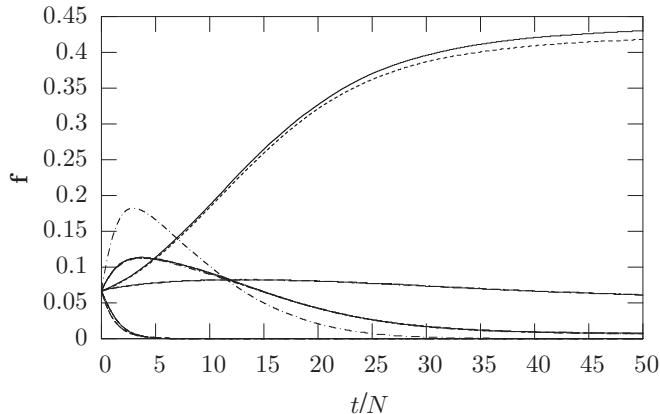


FIG. 2: Spatial population dynamics: ϵ -Machine-type distribution $\mathbf{f}(t)$ as a function of replication time with $n = 10^3$, $N = 10^6$, and $c = 0$.

Figure 2 shows the ϵ -machine-type distribution $\mathbf{f}(t)$ for $c = 0$ and $N = 10^6$. Note that initially the population behaves quite similarly to the panmixia case. This is to be expected since the simulations begin with identical initial configurations so at $t = 0$ the range of possible interactions is effectively the same for both architectures. As in the panmixia population, elements not in Ω are quickly removed from the soup.

By $t/N \approx 3.4$, however, the populations start to behave differently. For example, the ϵ -machine T_{15} that is most readily made in the panmixia case is also readily made in the spatial case, until $t/N = 3.4$, when it begins to be removed from the population. Again, the

ϵ -Machine Type	Early	Late
T_{15}	Fast Growth	Decay
T_3, T_5, T_{10}, T_{12}	Medium Growth	Decay
T_2, T_4	Logistic Growth	Saturation
T_1, T_8	No Growth	No Growth
$T_6, T_7, T_9, T_{11}, T_{13}, T_{14}$	Fast Decay	Removed

TABLE II: ϵ -Machine-type behavior classes in the two-dimensional spatial population for early and late times for Fig. 2. Early times: $t/N < 3.4$; late times: $t/N > 3.4$.

ϵ -machines partition themselves into subsets whose elements behave similarly. At early times before $t/N = 3.4$, these are identical to those seen in the panmixia population except that T_2 and T_4 undergo logistic growth, while T_1 and T_8 do not. Table II summarizes the overall behavior, for both early and late times.

More striking than the evolution of the ϵ -machine-type distribution $\mathbf{f}(t)$, however, are the spatial patterns that emerge in Γ^t . Figure 3 shows a 70×70 region of Γ^t at increasing times and one 500×500 region at a late time. Regions of T_2 and T_4 form stable domains that grow to dominate the soup. Within a region of T_2 or T_4 there exist ϵ -machines that cannot be replaced, typically due to an “elastic” collision—an interaction producing the non- ϵ -machine T_0 that, by θ , simply leaves the ϵ -machine at the lattice location alone.

When a region of T_2 meets a region of T_4 , ϵ -machines T_1 and T_8 form on the boundary, since $T_2 \circ T_4 = T_1$ and $T_4 \circ T_2 = T_8$. Moreover, $T_1 \circ T_1 = T_1$ and $T_8 \circ T_8 = T_8$ and so the boundaries are self-sustaining along the interface.

The spatial soup exhibits many of the nontrivial spontaneous patternings common to reaction-diffusion systems that exhibit Turing instability [17–20]. Finding a set of reaction-diffusion partial differential equations equivalent to this model and the spatial analog of Eq. 8, however, remains an open problem.

Over a large number of time steps, the spatial population looks essentially like that at $t/N = 20.0$ in Fig. 3. If enough time passes, though, the soup eventually divides itself into two connected regions of T_2 and of T_4 , or one will take over completely. This requires an extremely large number of replications. Which ϵ -machine region dominates in the long run appears to be randomly determined. The overall process is highly reminiscent of spinodal decomposition in which a mixed solution separates into stable component phases [21]. A more direct connection to the predictions of that theory awaits further effort.

VII. SPATIAL-REPLICATOR DOMAINS AND THEIR MEMBRANES

After time $t/N \approx 3.4$, the characters of the spatial and panmixia populations begin to diverge substantially, with patterns emerging in the spatial Γ . Those patterns consist of *domains* of nearly homogeneous ϵ -machines of type T_2 or of type T_4 . They are separated by domain

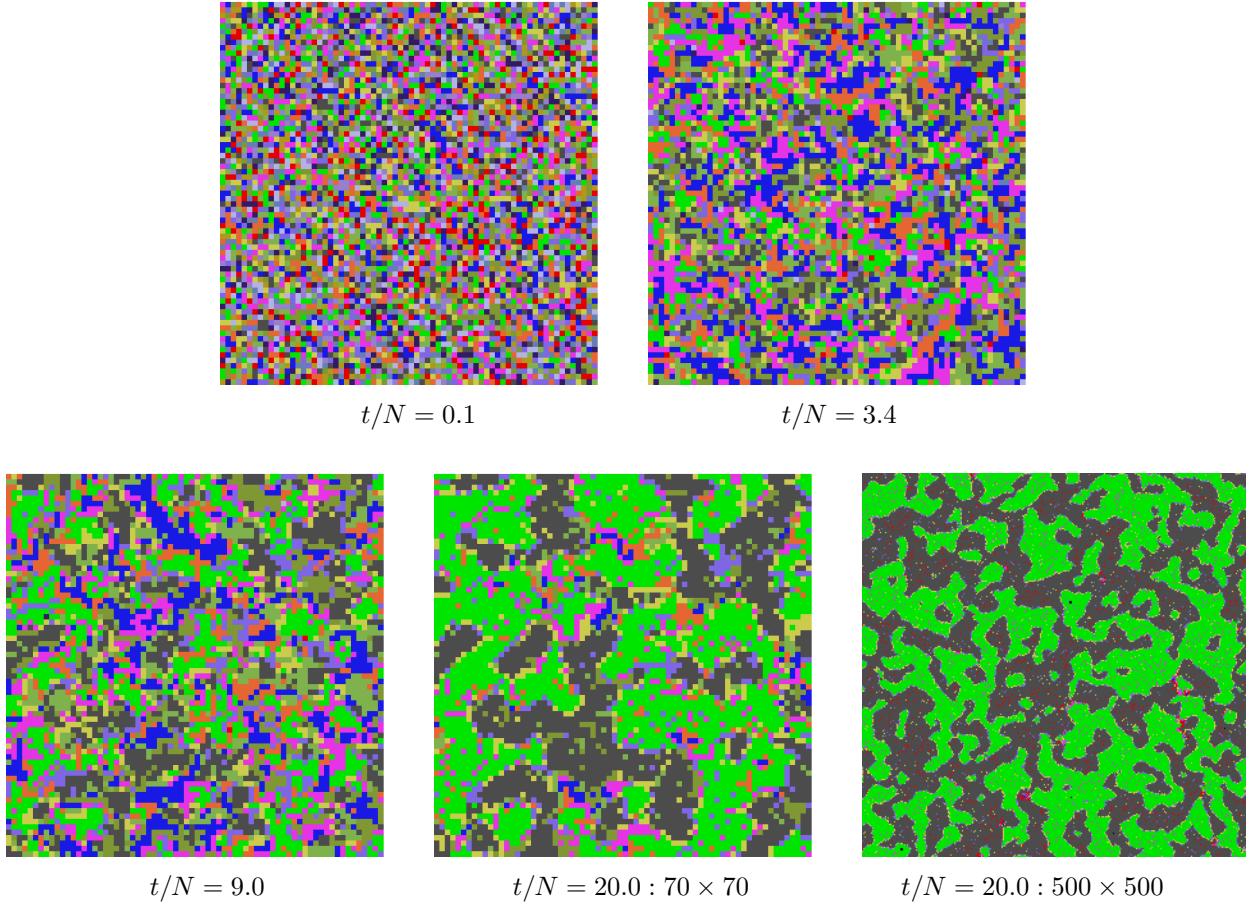


FIG. 3: Emergent spatial replicators and their membranes: A 70×70 region of Γ with $n = 10^3$, $N = 10^6$, and $c = v = 0$. The last image, however, is of a 500×500 region. Each color corresponds to one of the 16 ϵ -machine types: T_2 is green, T_4 is dark gray, T_1 is yellow, T_8 is purple, and T_{15} is blue.

walls that consist of ϵ -machines T_1 and T_8 . As we shall see, these walls play the role of functional membranes that actively translate between the domain ϵ -machines.

To summarize, then, the first observation about the spatial population is that it is completely differently organized. The persistent set of ϵ -machines differs markedly from those found with panmixia, as do their roles and interactions. Why did they emerge when the population is embedded in space? We can explain the emergence of these structures by defining a mechanism in the interaction network \mathbb{M} that affects the population dynamics when there is sufficient “spatial memory”. This mechanism is called a *spatial replicator* (SR).

Definition 2. *A set S of ϵ -machines is a spatial replicator (SR) if*

$$\begin{aligned} \theta(S, S, \theta(S, S, \mathbb{T})) &\subseteq S, \\ \theta(S, S, \theta(\mathbb{T}, S, S)) &\subseteq S, \\ \theta(\theta(S, S, \mathbb{T}), S, S) &\subseteq S, \text{ and} \\ \theta(\theta(\mathbb{T}, S, S), S, S) &\subseteq S. \end{aligned}$$

Notice that this is similar to the definition of a GR, except that SRs require two applications of θ to produce an element of S . Due to this, SRs only replicate when there is spatial information storage: If S is a SR, an element of S must compose with an ϵ -machine in \mathbb{T} to produce an ϵ -machine T_y . T_y must then compose with an element of S again to produce a new element of S .

In the panmixia population there is no notion of adjacency. Therefore, T_y does not have an increased chance of interacting with an element of S again. However, space allows a way for SRs to replicate: If by chance a sufficiently large domain of Γ consists only of elements of some SR S , ϵ -machines on the domain border will be replaced by elements of $\theta(S, S, \mathbb{T})$ or $\theta(\mathbb{T}, S, S)$. Later, these boundary ϵ -machines will be replaced with elements of

$$\begin{aligned} \theta(S, S, \theta(S, S, \mathbb{T})) \cup \theta(S, S, \theta(\mathbb{T}, S, S)) \\ \cup \theta(\theta(S, S, \mathbb{T}), S, S) \cup \theta(\theta(\mathbb{T}, S, S), S, S) , \end{aligned} \quad (10)$$

which are all elements of S . Thus, a domain of S will grow. This argument lays out how the algebraic structure that space adds to \mathbb{M} leads to domain growth. In

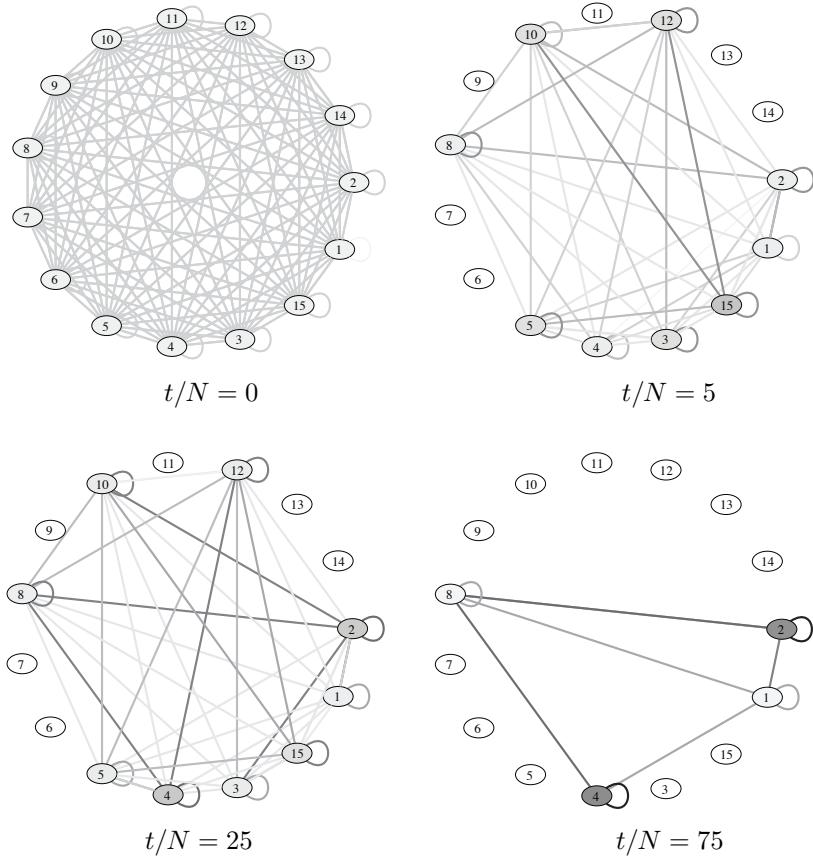


FIG. 4: The effective ϵ -machine interactions at various times. The nodes are the 15 possible ϵ -machines. The darkness of their interiors indicates their relative fractions \mathbf{f}^i in the populations. The edges connecting them indicate the frequency of their interactions; darker is more frequent.

addition, since $\theta(S, S, S) = S$, domains of S are self-maintaining.

As with a GR, one expects that if S' is a SR and is also a subset of some other SR S , then S' eventually will come to dominate the soup. For the spatial population of single-state ϵ -machines, it is easy to check that $\{T_1, T_2, T_4, T_8\}$ is a SR. Dynamically, the membranes of T_1 and T_8 grow their respective domains of T_4 and T_2 . Unsurprisingly, then, T_2 and T_4 come to dominate Γ . In contrast, T_{15} —an ϵ -machine made so readily by the panmixia population—begins to dominate the simulation until “spatial memory” begins to act. Beyond that point T_{15} is replaced by SRs.

Figure 4 shows how these interactions develop over time by giving the effective ϵ -machine interactions on an undirected graph, where the darkness of the ϵ -machine nodes indicates their relative frequencies \mathbf{f}^i and the darkness of the connecting edges indicates the relative frequency of interactions. As a complement to this, Fig. 5 gives the ϵ -machine interaction network at long times. This should be compared to the nine- ϵ -machine metamachine in the panmixia soup: see Fig. 3 of Ref. [10].

The explanation for why a spatial population organizes

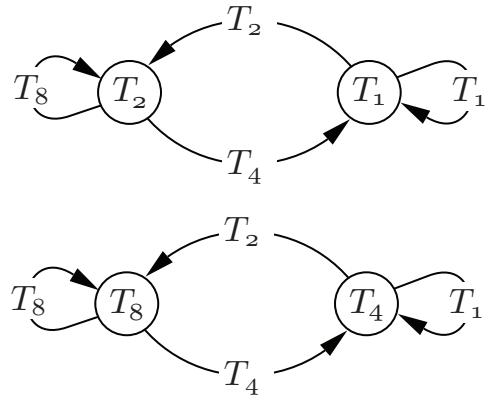


FIG. 5: The ϵ -machine interaction network at long times in the spatial soup. An arrow labeled T_b going from node T_a to node T_c denotes $T_c = T_b \circ T_a$.

differently and, in effect, selects different individuals for different roles relies on SRs, a local mechanism. The domains and walls, however, are nonlocal structures. Given a local dynamic that produces them, how do we describe their global structure?

To do this we adapt the computational mechanics analysis of emergent domains and particles in deterministic, synchronous cellular automata (CAs) [22] to the stochastic, asynchronous spatial population dynamics here. Speaking informally, a domain is a patch of spacetime that has the same “texture” when shifted in time, space, or both. How much one must shift each domain so that its texture maps onto itself is one crude measure of the structural complexity of the domain.

Definition 3. A domain is a set of configurations $\Gamma^S = \{\Gamma_{i,j} \in S\}$, where S is an SR, that satisfies:

1. *Temporal-shift invariance*: $\Gamma^S = \Phi\Gamma^S$, and
2. *Spatial-shift invariance*: $\Gamma^S = \sigma^{\widehat{n}}\Gamma^S$ for some spatial offset \widehat{n} .

It is straightforward to see that, in the spatial population, homogeneous regions of SRs— $\{T_2\}$ and $\{T_4\}$ —are domains. Determining which additional ϵ -machines—the nonreplaceable ones mentioned above—can be embedded in these domains, such that the regions are still domains, is a more difficult calculation that we will not attempt here.

The membranes separating the domains are complementary ϵ -machine types that actively function to translate between domains on either side, being composable with the ϵ -machine types in those domains. And, over time, sets of membrane ϵ -machines map back into themselves. In the spatial population, as noted above (Fig. 5), $T_1 = T_2 \circ T_4$ and $T_8 = T_4 \circ T_2$. The definition of a membrane replicator and a membrane express these formally.

Definition 4. A set M of ϵ -machines is a membrane replicator (MR) if there are two domain SRs, S and S' , such that

$$\begin{aligned} \theta(S, M, \theta(S, M, S')) &\subseteq M, \\ \theta(S, M, \theta(S', M, S)) &\subseteq M, \\ \theta(\theta(S, M, S'), M, S') &\subseteq M, \text{ and} \\ \theta(\theta(S', M, S), M, S') &\subseteq M. \end{aligned}$$

Notice that this is similar to the definition of an SR, except that MRs require the bounding domain SRs to produce ϵ -machines in the MR.

Definition 5. A membrane between two domain SRs S and S' is a set of configurations:

$$\Gamma^{SS'} = \Gamma^S M \Gamma^{S'}, \quad (11)$$

where M is a membrane replicator, that is temporal-shift invariant:

$$\Gamma^{SS'} = \Phi\Gamma^{SS'}. \quad (12)$$

Observe that membranes, like domains, are temporally invariant, but, unlike domains, they are *not* spatially shift invariant. This is what it means, in fact, for a structure to have a location in space.

Membranes translate between the, possibly distinct, kinds of information stored in the bounding SR domains. This view parallels that of, for example, the particles which perform computations in evolved cellular automata [23]. Specifically, when the spatial population dynamic is deterministic, then SR domains and membranes are directly analogous to the domains and particles of cellular automata computational mechanics [22]. The particular class of population dynamical systems considered here differ, however, in important ways: They have an asynchronous, stochastic spatial update. A more detailed framing of these differences and the methods to analyze them will be reported elsewhere.

In summary, Defs. 3, 4, and 5 specify algebraic constraints that, in principle, can be solved to find SR domains and membranes. Even in the simpler case of the computational mechanics analysis of deterministic CAs, solving the analogous set of constraints, though well defined, is at present difficult. And so, we will leave detailing the algorithms for these calculations to a sequel.

VIII. INTERMEDIATE SPATIAL POPULATIONS

The ϵ -machine soups here were specifically designed to permit investigation of populations that are neither completely spatial nor completely nonspatial by varying the mixing parameters c and v . To avoid presenting extremely high-dimensional data, we pick a metric that distinguishes the spatial versus nonspatial population dynamics and observe how that metric varies with c and v . For the metric we use $\mathbf{f}^{15}(t)$, the percentage of ϵ -machines of type T_{15} in Γ^t . In the nonspatial simulation, $\mathbf{f}^{15} \approx 0.223$ after $t/N = 50$ steps, while $\mathbf{f}^{15} \approx 0.0$ after $t/N = 50$ in the spatial model. In this way, Fig. 6 shows the transition between spatial and nonspatial behavior for the model as a function of c and v .

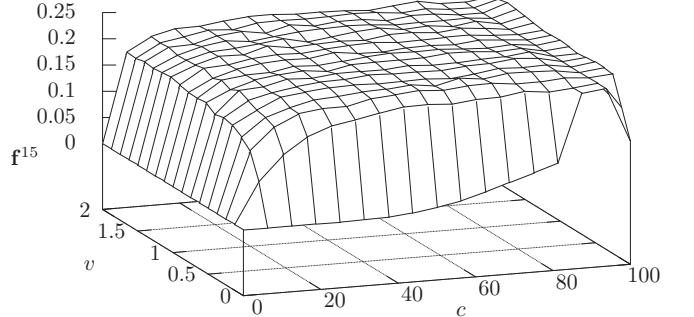


FIG. 6: $\mathbf{f}^{15}(t)$ after $t/N = 50$ time steps as a function of c and v with $N = 10^6$.

The two plots in Fig. 7 provide a clearer demarcation

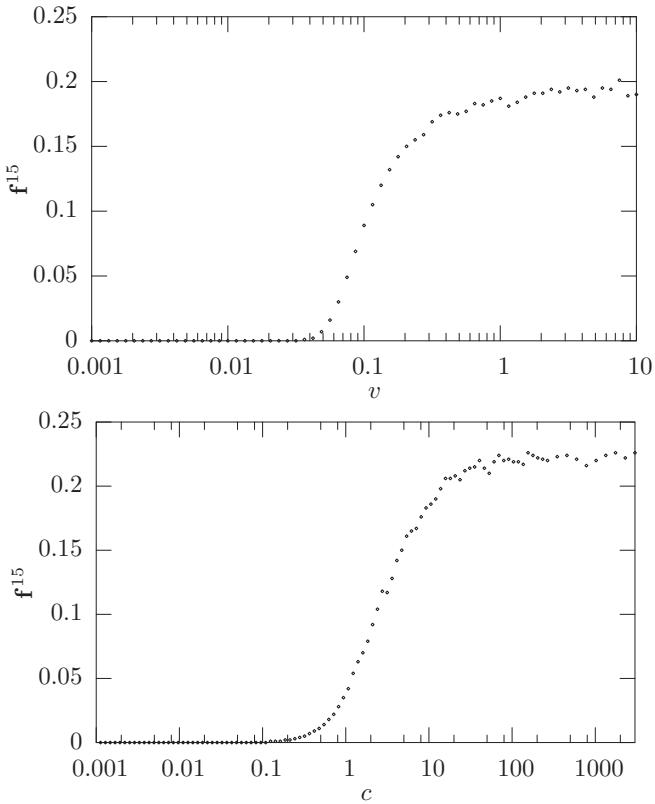


FIG. 7: Top panel: $f^{15}(t)$ after $t/N = 50$ time steps with $N = 10^6$ and $c = 10.0$. Bottom panel: f^{15} after $t/N = 50$ time steps with $N = 10^6$ and $v = 1.0$. Note that in both, the horizontal axis is logarithmic.

of the transition at which spatial memory is lost by showing slices along the v and c axes. Notably, the plots show that the “effect of space” is logistic in $\log c$ or $\log v$. This indicates that populations are sensitive to both parameters in similar ways, even though the physical realizations of varying c (number of ϵ -machines swapped) and v (spatial range of mixing) are quite different.

IX. CONCLUSION

The analysis demonstrated that the capacity for using spatial memory can be determined by examining the interaction network. Moreover, it also showed that the behavior of populations which are neither completely spatial nor completely nonspatial exhibit a systematic dependence on the degree of mixing. One can imagine several ways to generalize the definition of SRs—as the definition presented above is sufficient, but by no means necessary—for the spatial populations to behave differently from the nonspatial populations.

While the exact population dynamics of \mathbf{f} may be captured by an ODE such as Eq. (8), this coarse-graining is not useful to understanding the population dynamics’ key structural properties—the pattern formation highlighted

in the spatial case. Structural analysis of the interaction network, focusing specifically on its algebraic properties, was essential to understanding the mechanisms that drive the spontaneous emergence of organization. To this end, we drew a useful, if preliminary, connection to the computational mechanics analysis of domains and walls structures in spatially extended systems.

The membranes observed (and defined) here are not merely concentration gradients, as seen in familiar pattern formation systems. Rather, they are entities that, due to their interaction specificity, actively translate between replicators in neighboring domains. The resulting spatial organization suggests that one gets spontaneous compartmentalization without designing it in at the beginning. Recall that compartmentalization is often cited as one of the key early evolutionary steps on the road to increasing biological complexity [2]. It is not such a difficult step, after all. The ϵ -machine soup is a very simple system, built with a minimal set of physical, chemical, and biological assumptions.

It is perhaps no surprise if we mention that analogous investigations of one-dimension ϵ -machine soups leads to similar results. This and the two-dimensional, intermediate dimensional, and effectively infinite dimensional (panmixia) soups lead one to wonder about the role of the three-dimensions in which we know that life arose. Is there something special?

Constructively, in concert with natural spontaneous pattern formation, evolution may very well commandeer structures of different spatial dimension as ways that insure various kinds of functionality or intrinsic computation. We appreciate that evolution is opportunistic and takes advantage of nature’s propensity to spontaneously organize. And so, is three-dimensional space more advantageous in, say, the range of structures that it supports, compared to one and two dimensions? Or does evolution take advantage of all possible dimensions? The empirical evidence speaks rather clearly, if not to the former question, then to the latter: genetic information is stored in one-dimensional structures, biological cells sport two-dimensional membranes, and organisms move and behave in three dimensions.

Acknowledgments

SP was supported via the Santa Fe Institute’s summer 2004 NSF Research Experience for Undergraduates Program. This work was partially supported by CSC’s Network Dynamics Program funded by Intel Corporation and by the Defense Advanced Research Projects Agency (DARPA) Physical Intelligence project. The views, opinions, and findings contained here are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the DARPA or the Department of Defense.

[1] E. Szathmáry and L. Demeter. Group selection of early replicators and the origin of life. *J. Theo. Bio.*, 128:463–486, 1987.

[2] J. Maynard Smith and E. Szathmary. *The Major Transitions in Evolution*. W. H. Freeman/Spektrum, Oxford, 1995.

[3] R. F. Gesteland, T. R. Cech, and J. F. Atkins, editors. *The RNA World*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, Massachusetts, 1999.

[4] M. Eigen. Self-organization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58:465–523, 1971.

[5] M. Eigen and P. Schuster. The hypercycle. A principle of natural self-organization. Part A: emergence of the hypercycle. *Naturwissenschaften*, 64:541–565, 1977.

[6] T. S. Ray. An approach to the synthesis of life. In C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume XI of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 371–408, Redwood City, California, 1991. Addison-Wesley.

[7] C. Adami and C. T. Brown. Evolutionary learning in the 2d artificial life system ‘Avida’. In *Artificial Life 4*, pages 377–381. MIT Press, 1994.

[8] S. Rasmussen, C. Knudsen, and R. Feldberg. Dynamics of programmable matter. In *Artificial Life II: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Santa Fe Institute Studies in the Sciences of Complexity, Vol. 10)*. Addison-Wesley, 1992.

[9] W. Fontana. Algorithmic chemistry. In C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, volume XI of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 159–209, Redwood City, California, 1991. Addison-Wesley.

[10] J. P. Crutchfield and O. Gornerup. Objects that make objects: The population dynamics of structural complexity. *J. Roy. Soc. Interface*, 3:345–349, 2006. O. Gornerup and J. P. Crutchfield. Hierarchical Self-Organization in the Finitary Process Soup. *Artificial Life Journal*, 14(3): 245–254, 2008. O. Gornerup and J. P. Crutchfield. Primordial Evolution in the Finitary Process Soup. *Physics of Emergence and Organization*. I. Licata and A. Sakaji, editors, World Scientific Publishers, New Jersey (2008), 297–311.

[11] C. J. Ellison, J. R. Mahoney, and J. P. Crutchfield. Prediction, retrodiction, and the amount of information stored in the present. *J. Stat. Phys.*, 136(6):1005–1034, 2009.

[12] T. Kirner, J. Ackermann, D. Steen, R. Ehricht, T. Ellinger, P. Foerster, and J. S. McCaskill. Complex patterns in a trans-cooperatively coupled DNA amplification system. *Chem. Eng. Sci.*, 55:245–256, 2000.

[13] T. Kirner, D. Steen, J. S. McCaskill, and J. Ackermann. Biochemical amplification waves in a one-dimensional microflow system. *J. Phys. Chem. B*, 106(17):4525–4532, 2002.

[14] J. Ackermann, T. Kirner, and S.H.L. Klapp. Complex pattern formation of simple biochemical amplification reactions in micro-structured flow reactors. *Zeitschrift fur Naturforschung A*, 61(1-2):60–68, 2006.

[15] J. P. Crutchfield and K. Young. Inferring statistical complexity. *Phys. Rev. Lett.*, 63:105–108, 1989. J. P. Crutchfield. The Calculi of Emergence: Computation, Dynamics, and Induction. *Physica D*, 75:11–54, 1994. J. P. Crutchfield and C. R. Shalizi. Thermodynamic Depth of Causal States: Objective Complexity via Minimal Representations. *Phys. Rev. E*, 59(1):275–283, 1999.

[16] W. M. L. Holcombe. *Algebraic Automata Theory*. Cambridge, UK, Cambridge University Press, 1982.

[17] A. M. Turing. The chemical basis of morphogenesis. *Trans. Roy. Soc., Series B*, 237:5, 1952.

[18] H. Meinhardt. *Models of Biological Pattern Formation*. Academic Press, London, 1982.

[19] P. Ball. *The Self-Made Tapestry: Pattern Formation in Nature*. Oxford University Press, New York, 1999.

[20] R. Hoyle. *Pattern Formation: An Introduction to Methods*. Cambridge University Press, New York, 2006.

[21] E. P. Favva and A. Ch. Mitropoulos. What is spinodal decomposition? *J. Eng. Sci. Tech. Rev.*, 1:25–27, 2008.

[22] J. P. Crutchfield and J. E. Hanson. Turbulent pattern bases for cellular automata. *Physica D*, 69:279–301, 1993. J. E. Hanson and J. P. Crutchfield. The Attractor-Basin Portrait of a Cellular Automaton. *J. Stat. Phys.*, 66: 1415–1462, 1992. J. E. Hanson and J. P. Crutchfield. Computational Mechanics of Cellular Automata: An Example. *Physica D*, 103: 169–189, 1997.

[23] J. P. Crutchfield and M. Mitchell. The evolution of emergent computation. *Proc. Natl. Acad. Sci.*, 92:10742–10746, 1995.