Complexity Measurement Based on Information Theory and Kolmogorov Complexity

Leong Ting Lui[**]
University of Nottingham

Germán Terrazas[†]
University of Nottingham

Hector Zenil[‡]
University of Sheffield

Cameron Alexander[§]
University of Nottingham

Natalio Krasnogor[*,¶]
Newcastle University

Abstract In the past decades many definitions of complexity have been proposed. Most of these definitions are based either on Shannon's information theory or on Kolmogorov complexity; these two are often compared, but very few studies integrate the two ideas. In this article we introduce a new measure of complexity that builds on both of these theories. As a demonstration of the concept, the technique is applied to elementary cellular automata and simulations of the self-organization of porphyrin molecules.

Keywords
Complex systems, measures of complexity, open-ended evolution, cellular automata, molecular computation

1 Introduction

The notion of *complexity* has been used in various disciplines. However, the very definition of complexity[1] is still without consensus. One of the reasons for this is that there are many intuitive interpretations to the concept of complexity that can be applied in different domains of science. For instance, the Lorentz attractor can be simply described by three interaction differential equations, but it is often seen as producing complex chaotic dynamics. The iconic fractal image generated by the Mandelbrot set can be seen as a simple mathematical algorithm having a complex phase diagram. Yet another characteristic of complex systems is that they consist of many interacting components such as genetic regulatory networks, electronic components, and ecological food chains. Complexity is also associated with processes that generate complex outcomes, evolution being the best example. These wildly different perceptions and contextual dependences make it difficult to come up with a unifying definition of complexity that captures all these different aspects.

[*] Contact author.

[**] School of Computer Science, University of Nottingham, Nottingham NG8 1BB, United Kingdom. E-mail: Leong.Lui@nottingham.ac.uk

[†] Institute for Advanced Manufacturing, Faculty of Engineering, University of Nottingham, Nottingham NG7 2RD, United Kingdom. E-mail: german.terrazas@nottingham.ac.uk

[‡] Behavioural and Evolutionary Theory Lab, Department of Computer Science, University of Sheffield, Sheffield S10 1TN, United Kingdom. E-mail: hzenilc@gmail.com

[§] School of Pharmacy, University of Nottingham, United Kingdom. E-mail: pazca1@exmail.nottingham.ac.uk

[¶] Interdisciplinary Computing and Complex BioSystems (ICOS) Research Group, School of Computing Science, Newcastle University, Newcastle NE1 7RU, United Kingdom. E-mail: Natalio.Krasnogor@Newcastle.ac.uk

[1] We do not refer here to *computational complexity*, which is a very well refined concept [17, 5].

Artificial Life 21: 1–20 (2015) doi:10.1162/ARTL_a_00157

Of the many definitions of complexity [1, 15] that have been proposed over the years, most are, to a large extent, derived from one or the other of two powerful mathematical theories—Shannon information theory [18] or Kolmogorov complexity [10, 2].

The complexity of a system is often expressed in terms of "the amount of information produced by the system." If we measure the outputs of a system and encode them into a series of strings, then the complexity of the system can be assessed by the information contained within any such string; Kolmogorov complexity measures the amount of information required to generate this string. Another way of evaluating the complexity of a system is by the diversity of its outputs, that is, the probability distributions of the different symbols in the string; this approach is related to Shannon's information theory. The two approaches are often compared [7]. It has been shown that for a recursive probability distribution, the average value of Kolmogorov complexity of the distribution is similar to the value of the corresponding Shannon entropy. However, Teixeira et al. [19] proved that the relationship does not hold for the two generalizations of Shannon entropy—Renyi entropy and Tsallis entropy—except for the special case of $\alpha = 1$.

Several studies combine both Kolmogorov complexity and Shannon information theory to derive a more general definition of complexity. Let us consider a system that can be in an ensemble E of states, with each state e having a corresponding probability P_e. Zurek [26, 27] proposed the concept of physical entropy. It has two components, (i) $K(d)$, the algorithmic randomness (Kolmogorov complexity) of the known data d about the system, (ii) the missing information about the system, measured by Shannon entropy $H(d) = -\sum_e P_{e|d} \log_2 P_{e|d}$, where $P_{e|d}$ is the conditional probability of state e given data d. The physical entropy S_d of the system is the sum of the two terms, $S_d = H(d) + K(d)$.

Gell-Mann and Lloyd [6] introduced effective complexity. Similarly to physical entropy, they define the total information Σ of the system to be the sum of two terms, $\Sigma = Y + I$. The effective complexity Y is the algorithmic information content (AIC, or Kolmogorov complexity) of the regularities among all the possible states of the system, $Y = K(E)$. The term I represents the random aspect of the states, $I \approx \sum_e P_e K(e|E)$, where $K(e|E)$ is the contingent AIC of each member e given E. Then Y can be the considered as the most concise theory that describes the system, whereas I captures the accuracy of the theory. For instance, a simple system can be described by a simple theory (which has a low value of K) with high certainty (low I), and therefore has a low overall total information; on the other hand, a complex system would require either a complex theory (high K), or one with a low predictability (high value of I), or both.

Different from, and complementary to, the above approaches, our definition of complexity is based on the relative difference between the Shannon information in the ensemble of possible input (initial) states of a system and the entropy of the ensemble of the resulting output states for such a system. A system is considered to be less complex if it preserves lower information while mapping input ensembles to the output ensembles. A more complex system would generate more information in the resultant output ensembles. The output states are categorized by their estimated Kolmogorov complexity (the method is described in the following sections).

In this article we refine our recently proposed definition of complexity [16], which is based on a few intuitive but less studied concepts of complexity. The new definition also takes into account studies that used Kolmogorov complexity [25, 21, 12] to characterize and classify images, which essentially are 2D strings.

In the next section we give a brief overview of both theories. Our proposed complexity definition is introduced in Section 3. As a demonstration of its utility, we apply it to the analysis of elementary cellular automata (Section 4) and to the simulation of porphyrin self-assembly (Section 5).

2 Background Theory

2.1 Kolmogorov Complexity

Kolmogorov complexity, or algorithmic complexity, has been widely used to analyze the complexity of strings. A string is considered to be simple if it consists of a lot of repetitive patterns or has simple structures, in which case it can rewritten as a much shorter sequence of symbols that represent those patterns. Therefore the Kolmogorov complexity measures the regularity in a string. In a formal definition, the complexity of a string s is given by the shortest program p that when input into a universal Turing machine U will produce the original string s:

$$K_U(s) = \min\{|p|, U(p) = s\}.$$

In this context, a string is considered to be *simple* if a short program is sufficient to regenerate the string; on the other hand, a string is deemed *complex* if a long program is required instead.

One should note that Kolmogorov complexity is not computable—there is no existing algorithm that can calculate the shortest description for any given string. Lossless compression algorithms can be used instead to calculate an upper bound of the Kolmogorov complexity. In this study we take such an approximation as the estimate of K [13]. Another concern regarding Kolmogorov complexity is that it assigns long programs to both random strings and genuinely complex ones.

2.2 Shannon Information Theory

Whereas Kolmogorov complexity was used to measure the information contained in a *single* message, Shannon proposed in his classic article a mathematical framework that measures the information contained in a *distribution*. Consider the following scenario: A message M is sent from source S to receiver R. The message is chosen from a fixed set of messages, known to both source and receiver, with given probability distribution $p(m)$. Then, Shannon's information theory measures the *gained information* once a message is received as

$$I(m) = p(m)\log\frac{1}{p(m)}$$

with respect to the probability distribution $p(m)$.

3 Controllability Complexity

3.1 Definition

Korb and Dorin [11] argued that when analyzing the emergence of complex events one should consider not just the complexity of the event itself, but also the complexity of the system that generated it. They suggested that a minimum-message-length theory (MML) [22] would provide the most suitable basis for investigating the emergence of complexity, especially in the biological context. MML divides messages into two components, of which one describes the hypothesis under consideration and the other describes the sample statistics available. Inspired by their insight, we try to relate the probabilities of the input states of the system (U) to these of the corresponding output states (E). We define an index of *emergent*, or excess, complexity (EC) of output states in relation to their corresponding input states, that is, the amount of additional complexity a system adds to the initial states measured. This index relates the probability of observing particular outputs, $p(E\,|\,U)$, with the probability of finding the system in some initial state, $p(U)$.

Our definition is based on the following six boundary conditions for what a useful complexity measure needs to have:

1. $p(E\,|\,U)$ and $p(U)$ are normalized and proper probabilities such that $0 \leq p(E\,|\,U), p(U) \leq 1$.
2. The parameter EC is an unbounded positive number.

3. $\lim_{p(E\,|\,U)\to 1,\quad p(U)\to 1} EC(p(E\,|\,U), p(U)) = 0$, that is, probable initial conditions that lead to probable events receive the lowest rank: no surprises can be expected from this universe under the given (highly probable) initial conditions. This is marked as scenario A in Figure 1.
4. $\lim_{p(E\,|\,U)\to 1,\quad p(U)\to 0} EC(p(E\,|\,U), p(U)) = K$ with $K > 0$, that is, improbable initial conditions that lead to probable events are ranked slightly higher than 0 (the previous case). Intuitively this is a “needle in a haystack” universe with highly predictable dynamics (scenario B).
5. $\lim_{p(E\,|\,U)\to 0,\quad p(U)\to 0} EC(p(E\,|\,U), p(U)) = L$ with $L > K$, that is, an improbable initial state that leads to improbable events ranks even higher than scenario B, as it clearly represents an unexpected observation emerging from an unexpected initial condition (“Garden of Eden,” scenario C).
6. $\lim_{p(E\,|\,U)\to 0,\quad p(U)\to 1} EC(p(E\,|\,U), p(U)) = M$ with $M > L$: a probable set of initial conditions throws out surprising outputs, a simplicity-begets-complexity situation, thus ranking at the top of the scale (“elegant Garden of Eden,” scenario D).

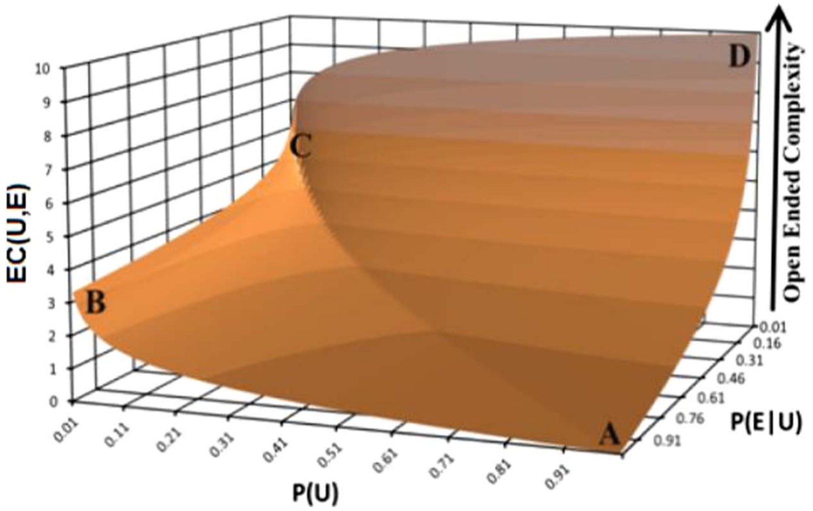
The above six intuitive constraints for our measure of complexity lead to the following equations:

$$EC(E, U) = -\frac{1}{2}\log_2(p(E|U)\cdot p(U)) + \log_2\frac{\max[p(E|U), p(U)]}{p(E|U)}. \tag{1}$$

The first term of Equation 1 captures the MML as suggested by Korb and Dorin, whereas the second term accounts for the added value in the complexity of events that the system’s dynamics has introduced. The overall emergent (or open-ended) complexity (OEC) of system is the calculated by the sum all possible combinations between input states and output states.

$$\text{OEC} = \sum_{\bar{U}}\left(\sum_{\bar{E}} EC(u, e)\right), \tag{2}$$

where $\bar{U}$ and $\bar{E}$ represent all possible inputs and outputs, respectively.



Q2 Figure 1. The idealized surface for the proposed emergent complexity measure. A, B, C, and D mark different scenarios, as explained in the text.

Figure 1 shows the four *EC* values of the four extreme scenarios. As one moves from A to B to C and finally D, *EC* increases.

Whereas probabilistic information is fundamentally related to Shannon's information theory, we are also interested in how much Kolmogorov complexity is contained in the outputs. We define the output states E not as the precise final states of the system, but as the amount of Kolmogorov complexity they contain. Therefore, $p(E|U)$ should be understood as the probability of a system generating outcomes with a specific value (or within a certain range) of Kolmogorov complexity, given the initial state U.

3.2 Method

The procedures for applying the measure are the following:

(i) The input states U for the complex system under study (cellular automata or porphyrin self-assembly) are defined as the different possible input parameters (initial conditions) of the system. This is straightforward for systems with discrete input parameters; for systems that take continuous parameters, the input parameter space is discretized and each resultant unit is considered as one input state U_i.

(ii) Once the input states are identified, their corresponding probabilities $p(U_i)$ can then be calculated.

(iii) Next we want to specify the output states E. For each input state U, multiple results are, in general, produced by a combination of system stochasticity and nonlinear behavior. As mentioned in the previous section, the Kolmogorov complexity of a string can be estimated by its compressed size. We follow the approach we presented in our previous works [25, 20]. The generated outputs of the system are converted into images. These images are then compressed into PNG format and further optimized using pngcrush, which has been shown to achieve good compression ratios [9]. The Kolmogorov approximations of the results are taken to be the final sizes of the images after the compressions. Thus, for each input state we have a collection of K-values from their corresponding outputs.

Q3 (iv) The range of measured K is divided into a series of discrete intervals, whose size is given by ΔK. The intervals are defined as the output states E in our complexity measure, which represent a group of results that have similar levels of Kolmogorov complexity. The size of the intervals indicates the sensitivity in detecting different outputs. As it is intuitive for us to consider a system that gives rich outputs as "complex," the calculated value of OEC is affected by the size of the intervals we can reliably measure.

(v) After we have defined E, we can then calculate $p(E|U)$ for all combinations of E and U, using the value of K for the images generated by each input state U.

(vi) Once we have calculated the $p(U)$ and $p(E|U)$, the complexity of the system (OEC) can be calculated using Equations 1 and 2.

4 Analysis of Cellular Automata

4.1 Experiment Setting and Procedures

Having discussed the concept and implementation method of our complexity measurement, we proceed to analyze the complexity of the different rule set of elementary cellular automata. Numerous studies have focused on the classification of cellular automata. One of the most notable investigations was reported by Wolfram in *A New Kind of Science* (*NKS*) [24]. He classified the rules of cellular automata into four classes, namely:

Class 1: The differences between the different initial states quickly diminish and result in homogeneous states.

Class 2: The output states evolve into periodic states.

Class 3: The outputs have random patterns.

Class 4: The outputs exhibit both ordered and random patterns.

Wolfram identified the four classes simply by the visual appearance of the images of the automata. A recent investigation reported Kolmogorov complexity to be an effective classifier of elementary cellular automata [20, 19]. Since our goal is to examine the viability of the proposed measurement, not to systematically compare all of the 256 possible rules, only 32 rules covering all four classes are investigated. As our complexity measure relies on accurate estimation of $p(E\,|\,U)$, this means we need to sample a significant proportion of all possible images. In order to keep the required computation manageable, we decided to use a relatively small system size, 18 cells in width, and simulated for 100 steps. Thus, the number of possible images is 2^{18}.

The procedure is the following:

(i) For each selected rule, 10% of the possible initial states are randomly chosen and simulated, which generates over 20,000 images. The images are then converted into PNG format and further optimized.

(ii) The input states U are denoted by the number of 1s in the initial condition; for example, for cellular automata of width 6, initial conditions 000001 and 100000 both belong to U_1, and 000011 and 110000 belong to U_2. For a system with fixed width m the probability of each U_n can be easily calculated using the following equation:

$$p(U_n) = \frac{m!}{n!(m-n)!} \times \frac{1}{2^m}.$$

(iii) The probability distributions of K for each of the input states U, which are given by $p(E\,|\,U)$, are built using the corresponding images. Figure 2 shows the probability distribution of $p(E)$; each bar in the histogram is considered to be an output state E.

(iv) Once we have $p(E\,|\,U)$ and $p(U)$, we can calculate the complexity for each of the rules, using Equations 1 and 2.

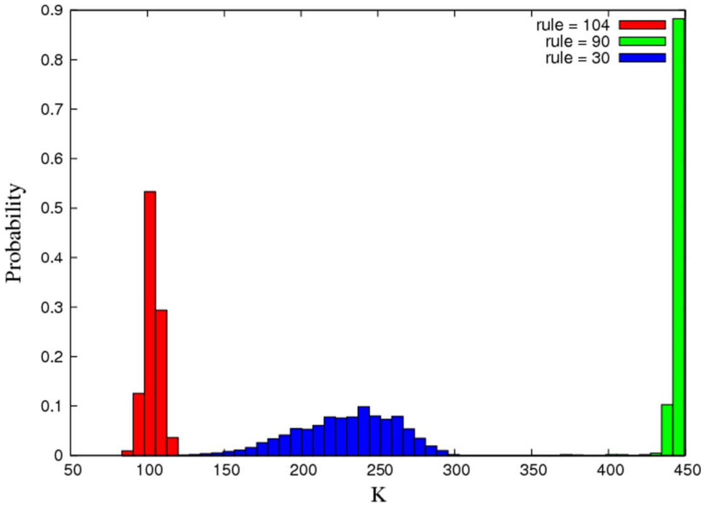


Figure 2. The distribution of K for cellular automata images generated using rules 30, 90, and 104.

Table 1. The examined rules ranked by according to their OEC values, from least complex (rank 1) to most complex (rank 32).

Rank	No. of rules	OEC	Rank	No. of rules	OEC
1	236	66.26	**17**	164	237.08
2	200	68.06	**18**	50	247.53
3	76	76.33	**19**	104	257.18
4	4	84.77	**20**	150	491.98
5	0	86.26	**21**	218	562.99
6	204	100.32	**22**	94	683.66
7	72	115.90	**23**	90	795.17
8	36	142.73	**24**	30	1735.59
9	128	150.76	**25**	54	1974.79
10	132	154.30	**26**	126	2453.35
11	32	167.01	**27**	18	2520.11
12	222	169.22	**28**	22	2691.10
13	232	170.41	**29**	110	2771.38
14	160	187.44	**30**	146	2818.84
15	108	188.46	**31**	122	2855.31
16	178	220.36	**32**	182	3063.92

4.2 Results

The 32 rules are ranked according to their calculated OEC values and are presented in Table 1. First of all we should note that, as stated above, the width of the intervals (ΔK) affects the values of the OEC; in order to show our classification to be consistent, we have calculated the OEC values using different sizes of ΔK. The results show that there is little effect on the comparison between the rules, that is, if, by using a particular width, rule A is considered to be more complex than rule B, then it is highly likely that we will see the same relationship when a different width is used. Thus, while the OEC absolute values of the rules are bin-width dependent, the relative values of complexity are unaffected.

We plot the value of the OEC against the corresponding rank among the examined rules (see Figure 3). From the figure we can roughly identify four regions according to the OEC values. However, it is not straightforward to justify the calculated values. We start with two of the simplest rules—rules 0 and 204. Rule 0 simply turns any state into 0; rule 204, on the other hand, maintains the previous row. Three example images produced using the two rules are shown in Figure 4. In the context of complexity theories, these images have very low algorithm complexity (can be described using short strings); therefore they occupy a very small region in the K space. This is indeed what we

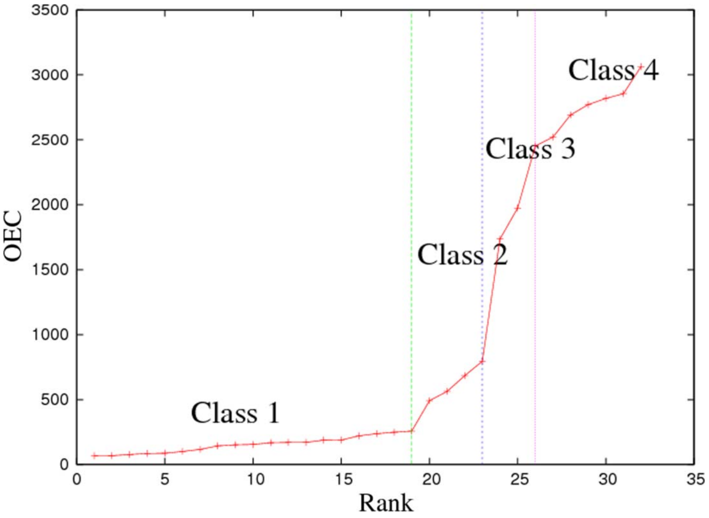


Figure 3. The value of the OEC at the corresponding rank among the 32 rules.

see from the probability distributions in Figure 5. As the images are concentrated in a small region in the *K* space, in comparison with other rules there are more pairs of (*U*, *E*) belonging to scenario A, which resulted in low OEC values. Similar *K* probability distributions are observed for the rules that have low OEC values.

Next we look at the higher end of the OEC spectrum. Here we classify rules with OEC higher than 2400 to be in class 4. Wolfram considers class 4 cellular automata to produce both random and structured patterns. Examples of the images from class 4 are displayed in Figure 6. Figure 7 shows the *K* probability distributions for rules 18, 110, and 146. All three of the distributions spread over a large range of *K*, from structured to random, which fits Wolfram's intuition. The differences between the rules that have high OEC values (class 4) and low ones (class 1) are apparent—the profiles of the simple rules occupy only a small region in the *K* space, whereas the ones with high

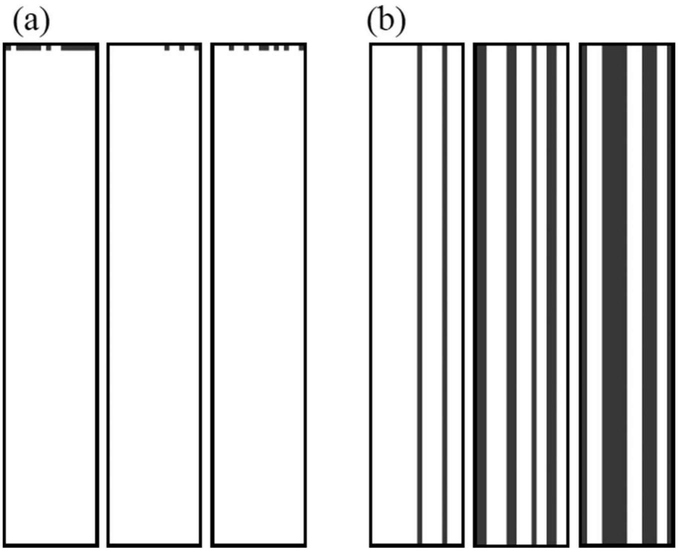


Figure 4. Three images generated using (a) rule 0, (b) rule 204.

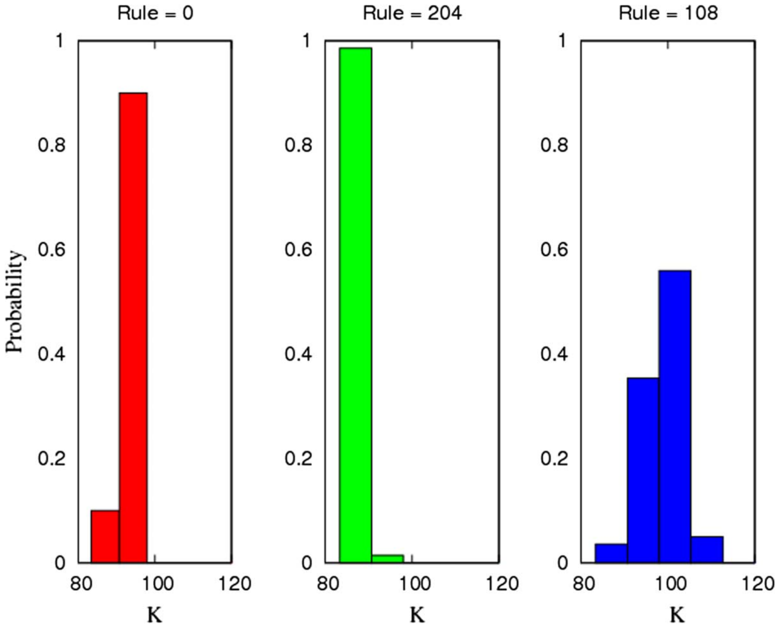


Figure 5. The probability distribution for *K* in the output images of rules 0, 204, and 108. Other rules with low OEC values have similar profiles. Each colored bar represent one output *E*.

values occupy a much larger region. This is a promising result, in that if one took an intuitive approach and evaluated the complexity of the rules using the average of the Kolmogorov complexity of the generated images, then one would not be able to correctly identify the class 4 rules as complex.

Now, let us consider rules that produce regular simple structures and were identified as class 2 by our measure. Among the 32 rules, rules 50 and 178 are examples of class 2: both of them have midrange OEC values. Three images for the two rules are presented in Figure 8. In addition to

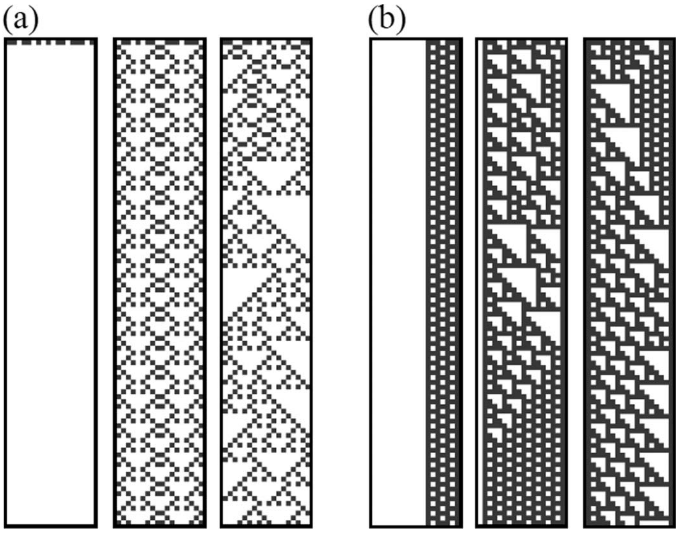
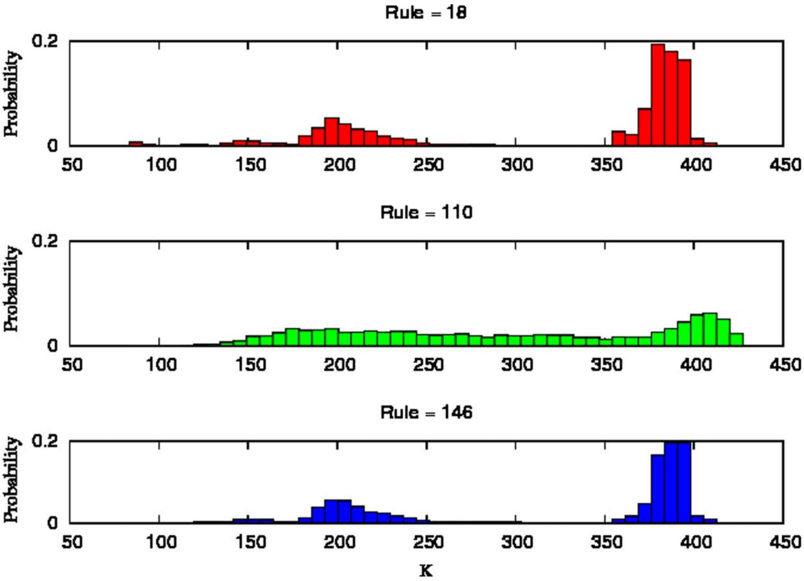


Figure 6. Three of the images generated using (a) rule 18, (b) rule 110.



Q4 Figure 7. The probability distribution of *K* for images generated using rules 18, 110, and 146. Similarly wide distributions are observed for other rules with high OEC values.

the two sets of images being very similar, it is not difficult to see that there are obvious simple patterns in them. Therefore, the images are considered to have low Kolmogorov complexity; in other words, they are simple structured objects. Their *K* distributions, as expected, are shifted to the lower end (Figure 9).

By closer inspection we can see differences between our OEC evaluation and Wolfram's classification. For instance, rule 150, which in general is considered to be a class 3 automaton, has a calculated OEC value lower than some of the class 2 rules such as rule 94. Figure 10a shows that the images generated in fact have complex patterns and are measured to have high Kolmogorov complexity. Again, we refer to the corresponding *K* probability profile (Figure 11) to see if

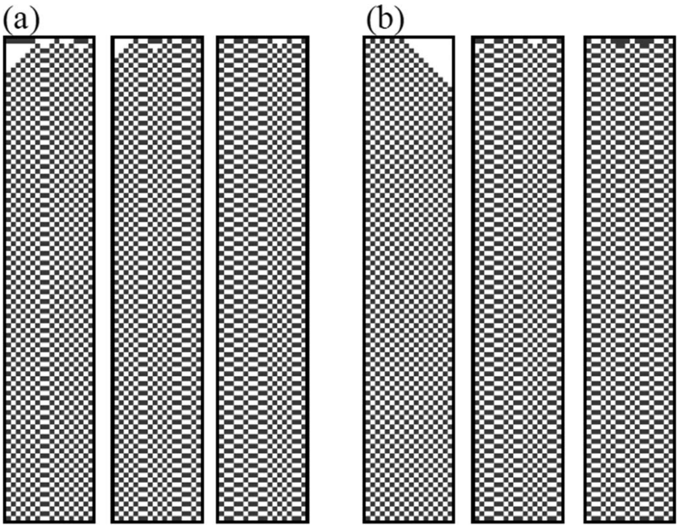


Figure 8. Three of the images generated using (a) rule 50, (b) rule 178.

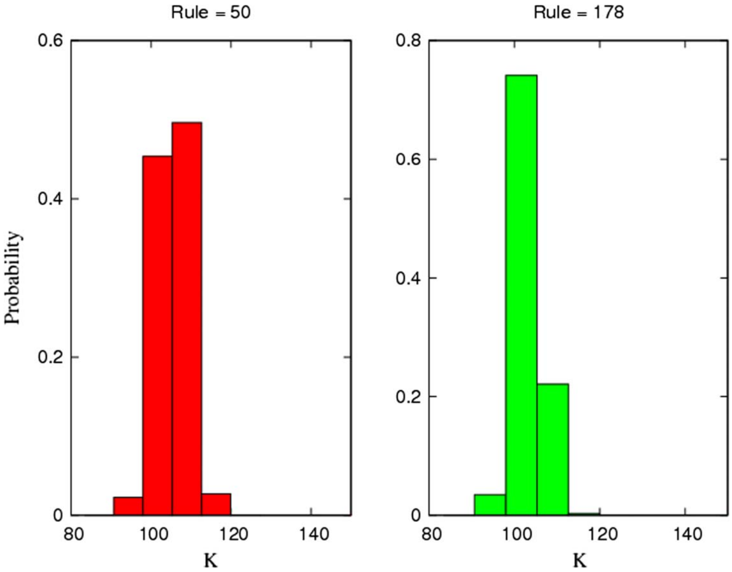


Figure 9. The probability distribution of *K* for images generated using rules 50 and 178.

the calculated OEC can be explained. The profiles agree with what we would have expected from the two rules. The probability profile of rule 150 shows that the majority of its images have high values of *K*. The reason that a low OEC value was assigned to this rule is that its distribution does not cover a wide range of *K*-values and hence the space of possible outputs is less surprising.

4.3 Modified OEC

A possible weakness of our measure is that the calculation does not properly weight the actual complexity of the output state E (given by the corresponding Kolmogorov complexity). In other

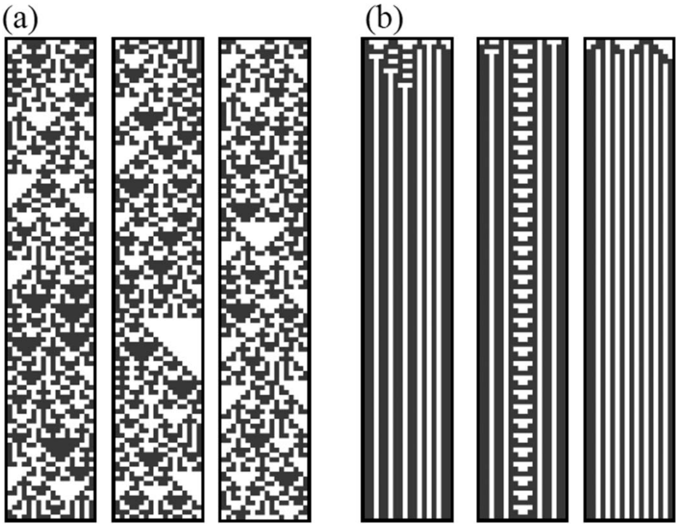


Figure 10. Three of the images generated using (a) rule 150, (b) rule 94.

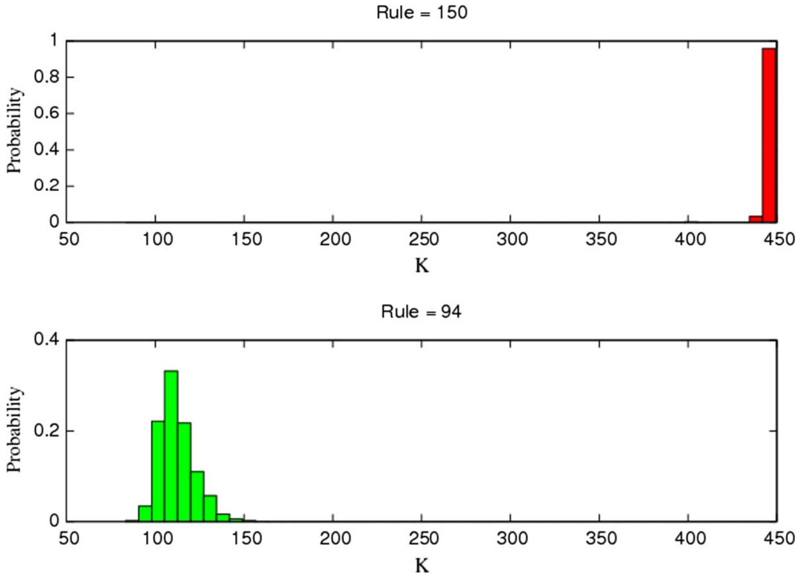


Figure 11. The probability distribution of *K* for images generated using rules 150 and 94.

words, two rules may end up having the same OEC as long as they have similar probability profiles $p(E|P)$, regardless of the information content in the images that they generate.

We can modify Equation 2 by introducing a weighting function that prioritizes the $EC(U,E)$ with higher complexity. The weighting function could take a number of forms, but for simplicity we consider a linear relation in *K*. The modified Equation 2 takes the following form:

$$W(K) = \frac{K - K_{\min}}{K_{\max} - K_{\min}}, \tag{3}$$

$$\mathrm{OEC} = \sum_{\bar{U}} \left(\sum_{\bar{E}} W(K) \cdot EC(u, e) \right). \tag{4}$$

The modification essentially gives extra weight to the rules that produce complex outputs.

The weighted OEC values for the rules are listed in Table 2.

The modified function addresses some of the differences between the two classifications. As complex images are weighted higher than simple ones, the calculated OEC values for class 2 and class 3 rules (such as rules 150, 30, and 90) are now in appropriate positions in the ranking, while the rankings for the rules that were identified as class 1 and class 4 are not affected. We would also like to point out that rule 110, the only rule proven to be Turing complete [3], was calculated to have one of the highest OEC values among the tested rules by our proposed complexity measure.

5 Porphyrin-Tile Kinetic Monte Carlo System

In this section we apply our complexity measure to a novel porphyrin-mediated molecular self-assembly simulation model. We first describe the components of a multi-agent system for the simulation of porphyrin molecule self-assembly. Porphyrins are planar molecules with fourfold symmetry and a chemical structure comprising four structural units that can be synthesized with substituent functional groups. Intermolecular binding, such as hydrogen bonding and van der Waals forces among such substituents, allows diverse self-assembly complexity together with a high degree of reversibility and highly dynamic pattern formation. We choose the Wang tile model [23] as an idealized model of porphyrins, since Wang tiles are square, with labeled edges, and undergo tile-to-tile interactions. Thus, there is not only a morphological correspondence to functionalized porphyrin molecules, but also a functional mapping to the intermolecular interactions between them. We refer to such models as *porphyrin-tile* models. A tile can be as classified *isofunctionalized* when its four sides are set with the same functional group, and as *heterofunctionalized* when its four sides are set with different functional groups. Figure 12 depicts the model correspondence and structural parts of a heterofunctionalized porphyrin molecule.

The substrate where molecules are deposited and interact with one another, forming aggregates, is modeled as a two-dimensional square-site lattice. Such a lattice is subjected to periodic boundary conditions where each position is occupied by only one porphyrin tile at a time. The neighborhood of a porphyrin tile is of von Neumann type, and energy interactions among neighboring molecules are at the core of the system dynamics for capturing *deposition*, *motion*, and *rotation* of a molecule on the substrate. In particular, deposition models the arrival of a molecule at an empty position of the substrate, that is, the placement of a porphyrin tile in an unoccupied position (i, j) of the lattice. Motion models the translation of a molecule to one of its four neighboring empty positions

Table 2. The examined rules ranked according to their weighted OEC values.

Rank	No. of rules	Weighted OEC	Rank	No. of rules	Weighted OEC
1	200	0.49	**17**	178	12.42
2	236	0.50	**18**	104	13.16
3	76	0.53	**19**	50	13.92
4	4	0.57	**20**	94	85.06
5	0	0.68	**21**	218	179.54
6	204	1.14	**22**	150	435.18
7	72	1.75	**23**	30	546.27
8	36	1.95	**24**	90	684.58
9	132	3.66	**25**	54	726.44
10	128	4.16	**26**	18	1085.00
11	32	4.61	**27**	22	1197.63
12	108	4.64	**28**	146	1245.62
13	232	4.85	**29**	126	1248.47
14	222	5.46	**30**	110	1274.93
15	160	6.78	**31**	182	1428.42
16	164	11.21	**32**	122	1448.39

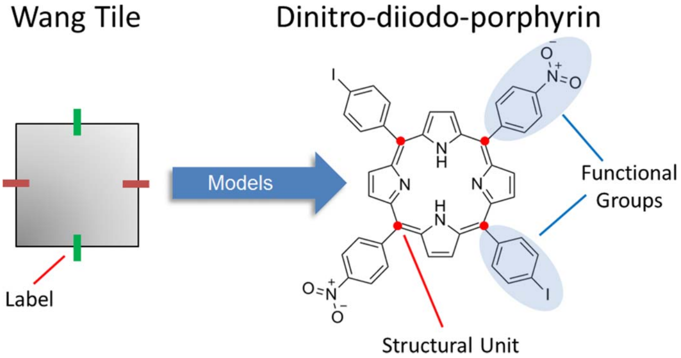


Figure 12. Structural units of a porphyrin molecule set with nitrogen and iodine give rise to a heterofunctionalized porphyrin molecule, which is represented by a Wang tile with different labels corresponding to different functional groups.

of the substrate, that is, the movement of a porphyrin tile located at position (*i,j*) to one of its four unoccupied nearest neighboring positions (*i*+*1*, *j*), (*i*, *j*+*1*), (*i*−*1*, *j*) or (*i*, *j*−*1*). For this, we consider three cases: the diffusion of a molecule across the lattice without interacting with neighboring molecules as shown in Figure 13b, diffusion along an aggregate as shown in Figure 13c,

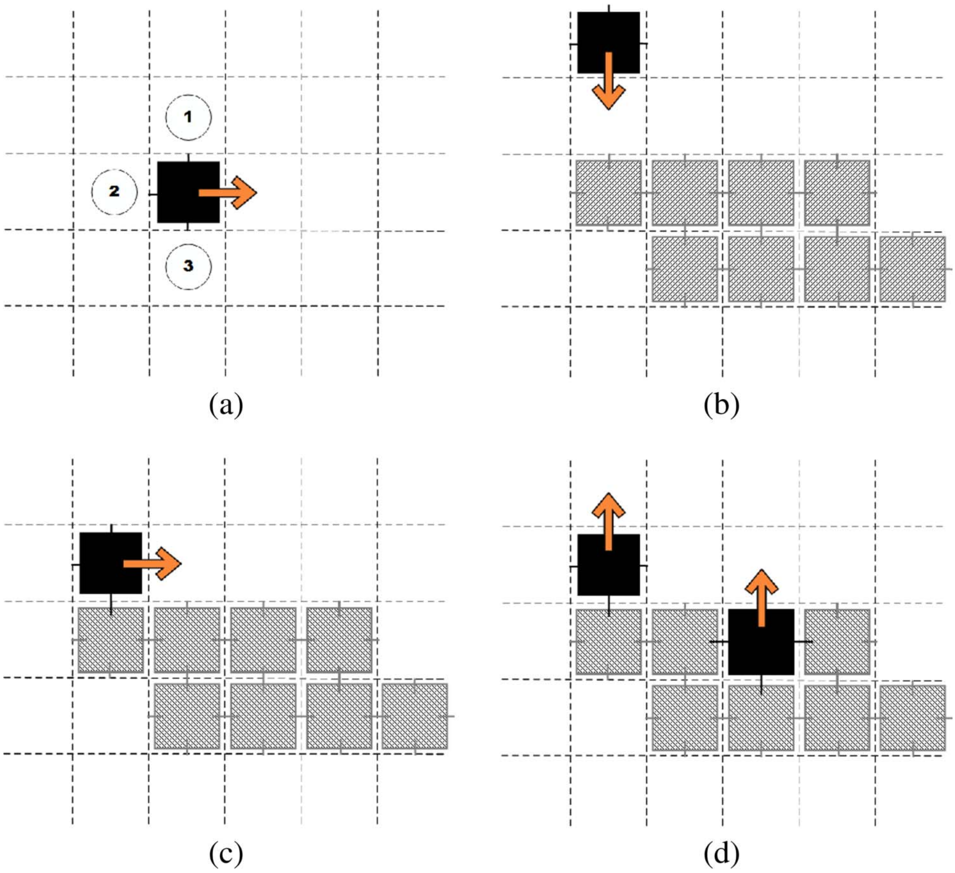


Figure 13. (a) Symbolic examples of a molecule hopping from position (*i,j*) to position (*i*, *j* + 1) together with its neighboring positions. (b) Diffusion of a molecule across the lattice without interacting with neighboring molecules. (c) Diffusion of a molecule along an aggregate. (d) Departure of a molecule from an aggregate.

and departure of a molecule from an aggregate as shown in Figure 13d. Rotation models spinning of a molecule about its center of mass, it consists of a 90-degree gyration of a porphyrin tile about its geometrical midpoint.

The Monte Carlo family of methods are generally good when coarse-grained modeling of the molecular entities and a fast approximation of the overall behavior of the system are needed [8, 14]. We implemented a *porphyrin-tile kinetic Monte Carlo* (kMC) system [21] in which a list of the possible transitions of the system (deposition, motion, and rotation of a porphyrin molecule) and their corresponding rates is compiled for each time step. Depositions take place at a constant rate (R_{Dep}), whereas diffusions and rotations are performed according to a diffusion rate (r_{ijkl}) calculated as

$$r_{ijkl} = \exp\left(\frac{-E_{ijkl}}{TT0}\right),$$

where E_{ijkl} is the activation energy a molecule needs to jump from position (i,j) to position (k,l) and $TT0$ is a fixed parameter capturing the temperature of the system and the Boltzmann constant. The activation energy is calculated in terms of the sum of intermolecular bindings and the binding with the substrate. Once the list of all possible transitions of the system and their rates is compiled, a Monte Carlo selection process follows, in which the transition with the best chance is chosen and performed. The chance of a transition is given according to the value of its associated rate, which is directly linked to the activation energy: the more neighboring molecules are present, the smaller the chance for a porphyrin tile to diffuse or rotate. After a transition is performed, the list is updated and the process is repeated for a fixed number of time steps.

We use this complex system model to run simulations that can be analyzed with our proposed complexity measure. Across this set of simulations, the porphyrin tile kMC system was configured with a lattice of 64 × 64 positions, $E_r = 1.3$ eV, $TT0 = 28 \times 10^{-3}$, $R_{\mathrm{Dep}} = 5 \times 10^{-5}$, a maximum coverage of 50%, a given binding energy between molecule and substrate (E_s), and six species comprising two isofunctionalized and four heterofunctionalized porphyrin tiles as shown in Figure 14.

The simulations of this complex system involving these six porphyrin-tile species consider functional groups for which the intermolecular bindings are always positive. Therefore, all the possible combinations among E_{11}, E_{22}, E_{12}, and E_s were systematically given in turn, the first three taking values 0.1, 0.2, … , 0.5 eV, and the last taking values 0.5, 0.6, 0.7 eV.

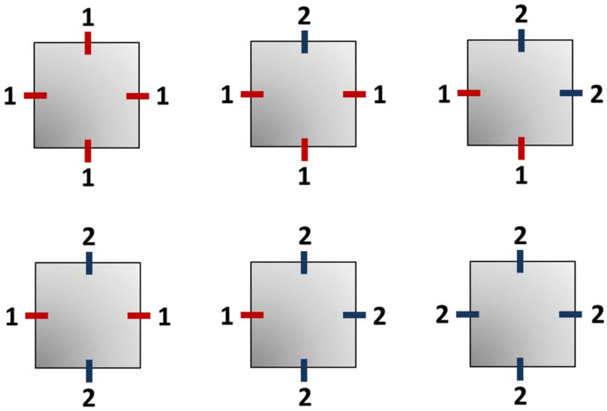


Figure 14. Six porphyrin-tile species comprising two isofunctionalized porphyrin tiles and four heterofunctionalized porphyrin tiles with functional groups labeled 1 and 2.

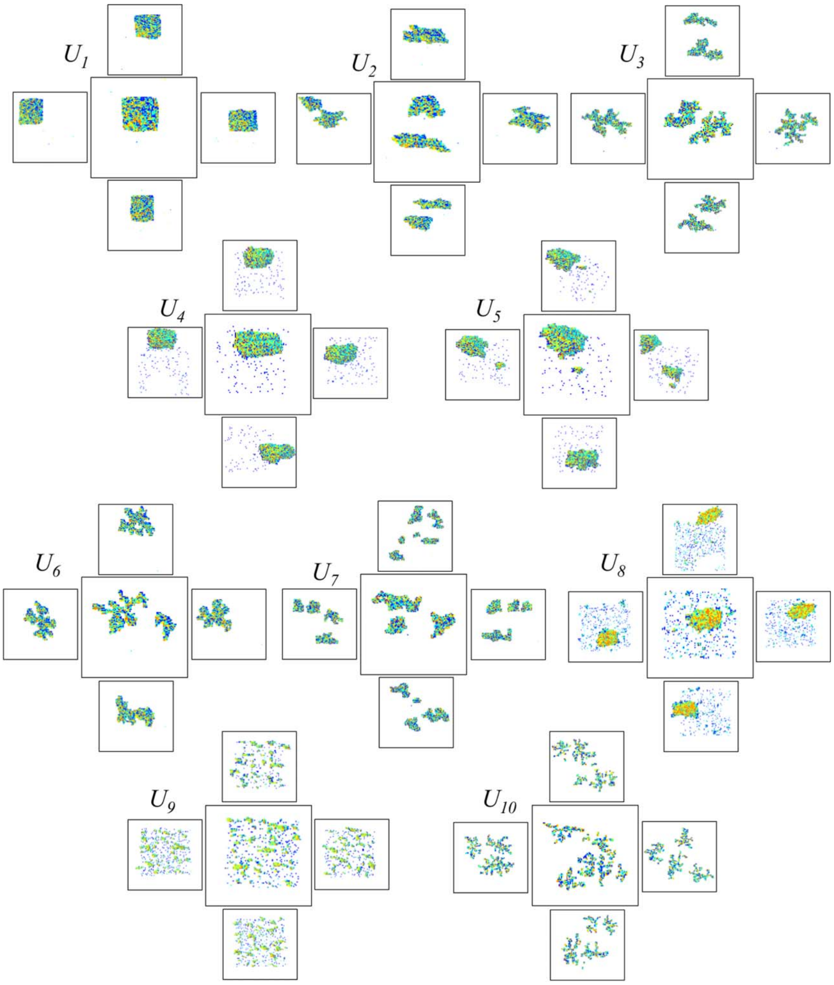


Figure 15. Sample images of the results of porphyrin self-assembly generated using different input parameters, and increasing average K-values from U_1 to U_{10}.

5.1 Complexity Analysis

Similarly to our analysis on cellular automata, the porphyrin simulation results are converted into PNG images and optimally compressed using pngcrush as before. The Kolmogorov complexities of the images are measured. Our initial set of simulations has explored a large area in the parameter space. Analyzing the images using the method described in the previous section gives us a rough estimation of the diversity of possible outputs of the system.

Let us assume the system has equal probability of taking each of 10 different sets of input parameters. For each of these input parameters, we run 100 independent runs and generate their associated images; sample images are shown in Figure 15. The values of these 10 sets of parameter are (ordered by increasing average K)

$U_1 = (0.5, 0.3, 0.3, 0.2)$,

$U_2 = (0.5, 0.3, 0.5, 0.2)$,

$U_3 = (0.5, 0.5, 0.5, 0.2)$,

$U_4 = (0.5, 0.1, 0.3, 0.2)$,

$U_5 = (0.5, 0.1, 0.4, 0.2)$,

$U_6 = (0.5, 0.1, 0.5, 0.5)$,

$U_7 = (0.7, 0.2, 0.3, 0.3)$,

$U_8 = (0.5, 0.1, 0.2, 0.1)$,

$U_9 = (0.5, 0.1, 0.5, 0.1)$,

$U_{10} = (0.7, 0.5, 0.5, 0.4)$.

Then, we measure and construct the corresponding K-distributions for each of the inputs. The distributions for the 10 inputs are shown in Figure 16. In this way we capture both the algorithmic complexity of the produced outputs and also the variety and range of K-values for these outputs.

Using the distributions, we can calculate the $p(E \mid U)$ for each pair of E and U. Since we know the probabilities of the input states, we can calculate how much each U contributes to the overall OEC value for the system (using Equation 4). This is shown in Table 3.

By carefully analyzing (i) the ranking obtained in Table 3, (ii) the probability distribution in Figure 16, and (iii) the actual numerical values used to calculate column B in Table 3, it is possible to observe that the OEC (Equation 4) is correctly distinguishing four cases, namely:

(a) Input-to-output mappings with relatively low complexity (K) and a low range of K-values (most morphologies are very similar to each other), sets U_1, U_2 and U_3.

(b) Input-to-output mappings with relatively low complexity but with varied (hence less predictable) morphologies, sets 7 and, notably, U_5. The latter has lower average K than U_7, but it has a wider distribution, that is, it produces a larger variety of outputs.

(c) Input-to-output mappings with relatively high algorithmic complexity but with a narrow variety of morphologies (and hence more predictable), sets 4 and, notably, 10. In the latter case, although U_{10} has the highest average algorithmic complexity, it produces the smallest range of possible values, thus making it highly predictable.

(d) Input-to-output mappings with relatively high algorithmic complexity and a wide range of morphologies (thus less predictable), sets U_8 and U_9.

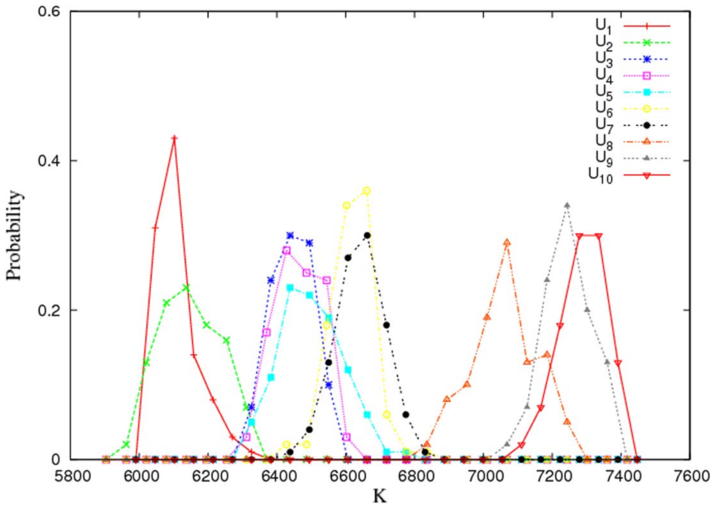


Figure 16. The probability distributions of the Kolmogorov complexities for images generated by the 10 sets of input parameters.

Table 3. The contribution of each input to the overall OEC for the system, ordered by magnitude of contribution.

A	B	C
Input state U_i	$\sum_{\bar{E}} W(K) \cdot EC(U, E)$	Average K
2	1.99	617720
1	2.97	613723
3	3.49	647249
4	5.44	649234
10	9.06	731923
6	10.94	664339
7	11.93	667083
5	12.67	652566
9	16.82	727454
8	16.88	708970
Total	92.21	668026

6 Conclusion and Discussion

One of the key concepts of this work is the merging of Shannon information and Kolmogorov complexity, into a new measure of emergent complexity, *EC* (Equations 1, 2, 3, and 4). We use the concept of Kolmogorov complexity to classify the outputs of a system according to their algorithmic complexity, then use MML-inspired equations to measure the amount of information contained in the *K* distribution of outputs. The proposed method has the advantage of taking into account both the information in the generated messages and the information given by the *diversity* of the messages. Apart from the information contained, our complexity measure also considers the correlation between input states and output states.

The method was applied to elementary cellular automata, where good agreement has been found between our formalized complexity classification and Wolfram's intuitive classification. We then introduced and demonstrated how the proposed measurement can be applied to a novel porphyrin self-assembly model, thus demonstrating its general utility in complex systems other than cellular automata.

The approximations we made for Equations 1–4 would benefit from better compression algorithms (more accurate estimates of Kolmogorov complexity—for example, approximations using the coding theorem method [4]). The derived equations (1–4) were simply designed to serve our basic intuitions of what a good measurement of complexity should do (i.e., conditions 1–6 in Section 3.1). Further investigations can aim to strike a balance between the different components of information content, that is, between the algorithmic information content in the objects produced and that in the distribution of those objects. The results we obtained with our new measure of overall complexity (Equation 4) are very intriguing indeed; notably, our measure ranks cellular automata rule 110, the only one known to be Turing complete, to be one of the most complex among all the tested rules.

Acknowledgments

We thank the UK EPSRC (grants EP/H010432/1, EP/G042462/1, and EP/J004111/1). We thank Prof. P. Moriarty and Prof. N. Champness for constructive discussions on molecular self-assembly.

References

1. Adami, C., & Cerf, N. J. (2000). Physical complexity of symbolic sequences. *Physica D*, *137*, 62–69.
2. Chaitin, G. J. (1969). On the length of programs for computing finite binary sequences: Statistical considerations. *Journal of the ACM*, *1*, 145–159.
3. Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, *15*(1), 1–40.
4. Delahaye, J. P., & Zenil, H. (2012). Numerical evaluation of the complexity of short strings: A glance into the innermost structure of algorithmic randomness. *Applied Mathematics and Computation*, *219*, 63–77.
5. Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman.
6. Gell-Mann, M., & Lloyd, S. (2004). Effective complexity. In M. Gell-Mann & C. Tsallis (Eds.), *Nonextensive entropy—interdisciplinary applications*. Oxford University Press.
7. Grünwald, P., & Vitányi, P. (2004). Shannon information and Kolmogorov complexity. arXiv:cs/0410002.
8. Hermann, B. A., Rohr, C., Balbás Gambra, M., Malecki, A., Malarek, M. S., Frey, E., & Franosch, T. (2010). Molecular self-organization: Predicting the pattern diversity and lowest energy state of competing ordering motifs. *Physical Review B*, *82*(16), 1–6.
9. http://pmt.sourceforge.net/pngcrush
10. Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems in Information Transmission*, *1*, 1–7.
11. Korb, K. B., & Dorin, A. (2011). Evolution unbound: Releasing the arrow of complexity. *Biology and Philosophy*, *26*(3), 317–338.
12. Krasnogor, N., & Pelta, D. A. (2004). Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, *20*(7), 1015–1021.
13. Li, M., & Vitanyi, P. (2008). *An introduction to Kolmogorov complexity and its applications* (3rd ed.). Springer.
14. Li, Y., & Nian, L. (2011). Combined scanning tunneling microscopy and kinetic Monte Carlo study on kinetics of Cu-coordinated pyridyl-porphyrin supramolecular self-assembly on a Au (111) surface. *Physical Review B*, *84*(12), 125418.
15. Lopez-Ruiz, R., Sañudo, J., Romera, E., & Calbet, X. (2011). Statistical complexity and Fisher-Shannon information: Applications. In K. D. Sen (Ed.), *Statistical complexity* (pp. 65–127).
16. Markovitch, O., Sorek, D., Lui, L. T., Lancet, D., & Krasnogor, N. (2012). Is there an optimal level of open-endedness in prebiotic evolution? *Origins of Life and Evolution of Biospheres*, *42*(5), 469–474.
17. Papadimitriou, C. H. (1994). *Computational complexity*. Addison-Wesley.
18. Shannon, C. E. (1948). The mathematical theory of communication. *Bell System Technical Journal*, *27*, 379–423, 623–656.
19. Teixeira, A., Matos, A., Souto, A., & Antunes, L. (2011). Entropy measures vs. Kolmogorov complexity. *Entropy*, *13*, 595–611.
20. Terrazas, G., Siepman, P., Kendal, G., & Krasnogor, N. (2007). An evolutionary methodology for the automated design of cellular automaton-based complex systems. *Journal of Cellular Automata*, *2*(1), 77–102.
21. Terrazas, G., Zenil, H., & Krasnogor, N. (2013). Exploring programmable self-assembly in non-DNA-based molecular computing. *Journal of Natural Computing*, *12*(4), 499–515.

22. Wallace, C. S. (2005). *Statistical and inductive inference by minimum message length*. Springer-Verlag.

23. Wang, H. (1961). Proving theorems by pattern recognition. *Bell Systems Technical Journal*, *40*, 1–42.

24. Wolfram, S. (2002). *A new kind of science* (p. 232). Wolfram Media.

25. Zenil, H. (2010). Compression-based investigation of the dynamical properties of cellular automata and other systems. *Complex Systems*, *19*(1), 1–28.

26. Zurek, W. (1989). Algorithmic randomness and the physical entropy. *Physical Review A*, *40*, 4731–4751.

27. Zurek, W. (1990). Algorithmic information content, Church-Turing thesis, physical entropy, and Maxwell's demon. In E. Jen (Ed.), *Lectures in Complex Systems, SFI Studies in the Sciences of Complexity*, Vol. II (pp. 49–65). Addison-Wesley.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

During the preparation of your manuscript, the questions listed below arose. Kindly supply the necessary information.

1. Please check if the proposed running title is okay.
2. Figure 1 contains poor quality of text. Please check.
3. Please check if (v) should be (iv).
4. Figure 7 contains pixelated text. Please check.

END OF ALL QUERIES