Exploratory analysis of spatiotemporal patterns of cellular automata by clustering compressibility

Frank Emmert-Streib*

Computational Biology and Machine Learning, Center for Cancer Research and Cell Biology, School of Medicine, Dentistry and Biomedical Sciences, Queen's University Belfast, 97 Lisburn Road, Belfast BT9 7BL, United Kingdom

(Received 3 April 2009; published 8 February 2010)

In this paper we study the classification of spatiotemporal pattern of one-dimensional cellular automata (CA) whereas the classification comprises CA rules including their initial conditions. We propose an exploratory analysis method based on the normalized compression distance (NCD) of spatiotemporal patterns which is used as dissimilarity measure for a hierarchical clustering. Our approach is different with respect to the following points. First, the classification of spatiotemporal pattern is comparative because the NCD evaluates explicitly the difference of compressibility among two objects, e.g., strings corresponding to spatiotemporal patterns. This is in contrast to all other measures applied so far in a similar context because they are essentially univariate. Second, Kolmogorov complexity, which underlies the NCD, was used in the classification of CA with respect to their spatiotemporal pattern. Third, our method is semiautomatic allowing us to investigate hundreds or thousands of CA rules or initial conditions simultaneously to gain insights into their organizational structure. Our numerical results are not only plausible confirming previous classification attempts but also shed light on the intricate influence of random initial conditions on the classification results.

DOI: 10.1103/PhysRevE.81.026103 PACS number(s): 82.40.Ck, 89.75.Fb

I. INTRODUCTION

The analysis of complex systems generating spatiotemporal pattern has a long tradition in physical, chemical, and biological sciences because many natural phenomena exhibit an intriguing variety of such patterns. A very important mathematical model capable of generating discrete spatiotemporal patterns is cellular automata (CA). Since first introduced and studied by Von Neumann and Burks [1] CA have been investigated numerously, e.g., for analyzing fluid dynamics [2], pattern formation [3,4], or universality [5,6]. A meanwhile classic topic concerns the classification of CA with respect to either the behavior of their spatiotemporal pattern [7–10] or their transition (lookup) table [11–13]. Over the past decades several approaches have been suggested to classify cellular automata. Most prominently, Wolfram [8] proposed a phenomenological classification of CA based on their observed behavior and entropic measures.

The classification of CA is closely related to a more general topic, namely, the characterization of the complexity of an object itself. This problem has attracted much attention and many measures have been suggested [14–21] over the past almost three decades. An intrinsic problem with such a measure is that there are various ways to define or characterize complexity leading to complementing complexity measures. For example, Kolmogorov complexity [18,20,22] is based on algorithmic information theory considering objects as individual symbol strings whereas the measures, such as *effective measure complexity* [17], *excess entropy* [23], or *predictive information* [24], relate objects to random variables. Interestingly, despite considerable differences among all these complexity measures $\mathcal{M}$ they all have in common, that they assign a complexity value to each individual object x under consideration, $C_{\mathcal{M}}(x)$. It is of importance to note that there is a conceptually different measure recently introduced by Cilibrasi and Vitányi that evaluates the complexity *distance* among objects x and y instead of their absolute values. This measure is called the *normalized compression distance* (NCD) [25], NCD(x,y), and is based on Kolmogorov complexity [26]. So far the NCD has been applied to various different problems [27–30] but not to investigate spatiotemporal patterns. It is crucial to remark that the classification of CA is intimately related to measures of complexity because in order to be able to classify meaningfully the, e.g., spatiotemporal patterns of the CA one needs to clarify what are simple, complex, or random patterns?

The major purpose of this paper is to demonstrate the use of the normalized compression distance for the classification of the behavior of spatiotemporal patterns of complex systems. More precisely, we suggest an exploratory approach by clustering many different systems under investigation to unravel the relationship among the object's complexity. This comparative analysis will not allow us to assign absolute values of *complexity* to the individual objects but, instead, provides insights into their inter-relational structure. We argue that this is not only interesting for a practical application but also for theoretical investigations. Numerically, we exemplify our approach by application to one-dimensional cellular automata. As pointed out by Dhar *et al.* [31] it is crucial to include in the study of CA their initial conditions because the behavior of a rule is significantly affected by its initial state. For this reason we based our classification on the behavior (the spatiotemporal patterns) of the CA solely not taking the lookup table into account explicitly.

This paper is organized as follows. In Secs. II–V we present a detailed definition of the problem we are studying. Section VI provides our numerical results and this paper finishes in Sec. VII with a discussion and conclusions.

II. CELLULAR AUTOMATA

In this section we will provide the necessary definitions we need for our investigations. One-dimensional cellular au-

*v@bio-complexity.com

tomata consist of a lattice of N identical sites or cells, s_i, $i \in \mathcal{N}$, and $\mathcal{N}=\{0,\ldots,N-1\}$. Each cell s_i can assume values from an alphabet $\mathcal{S}$ consisting of $k=|\mathcal{S}|$ discrete states or symbols, e.g., $\mathcal{S}=\{0,1\}$. All cells are updated synchronously by a local update function

$$s_i^{t+1} = \phi(s_{i-r}^t, \ldots, s_{i+r}^t) \tag{1}$$

involving only the nearest $2r+1$ neighbors, including cell i itself. The local update function ϕ can be written as lookup table encoding the value of a cell at time step $t+1$. The evolution of the system $\mathbf{s}^t=(s_0^t,\ldots,s_{N-1}^t) \in \mathcal{S}^N$ is given by the global update function $\Phi:\mathcal{S}^N \to \mathcal{S}^N$ defined by

$$\Phi(\mathbf{s}^{t+1})_i = \phi(s_{i-r}^t, \ldots, s_{i+r}^t). \tag{2}$$

In the following we consider only cellular automata with $k=2$, $r=1$ and periodic boundary conditions for the finite lattice. This implies that we have a total of $256=2^{2^3}$ different update functions ϕ that we call in the following update rules. The update rules of the cellular automata are coded according to the scheme introduced by Wolfram [8] which can be explicitly written [32] as

$$R = \phi(0,0,0)2^0 + \phi(0,0,1)2^1 + \cdots + \phi(1,1,1)2^7, \tag{3}$$

with $R \in \mathcal{R}=\{0,\ldots,255\}$.

III. CLASSIFICATION OF CELLULAR AUTOMATA

The first attempt to classify one-dimensional cellular automata systematically was presented by Wolfram [7,8]. The basic idea behind Wolfram's classification is to group the rules of cellular automata according to their observed behavior into four disjoint classes. The four suggested classes are [8]

(1) evolution leads to a homogeneous state,

(2) evolution leads to a set of separated simple stable or periodic structures,

(3) evolution leads to a chaotic pattern, and

(4) evolution leads to complex localized structures, sometimes long lived.

The classification is qualitative, but motivated by the dynamical behavior dynamical systems can exhibit in general. We want to emphasize that so far no final agreement regarding the classification of all 256 rules has been reached. This is not only due to the missing mathematical rigor in the definition of the four classes but also due to the difficulty to demonstrate that a rule shows all required properties to belong to a certain class. In addition, this is complicated by the fact that initial conditions can change the spatiotemporal pattern of the CA rule profoundly. For example, it is known that rule 90 produces a behavior that is not distinguishable from random if the initial condition was a random sequence of 0s and 1s [33]. However, starting from an unbalanced ratio between black (1) and white (0) cells favoring, e.g., the presence of black cells significantly results in a nonrandom behavior. The reason for this is that rule 90 preserves the randomness present in the initial conditions [33]. This is just one example to show that the initial conditions can have a significant influence on the behavior of the cellular automata. For this reason, we are not aiming to classify rules of CA themselves, like Wolfram, but their spatiotemporal behavior. This implies that in addition to the rule of the CA we also need to specify its initial state because only the combination of both defines the behavior of the CA. From this explanations it becomes clear that we are not trying to reproduce the classification of Wolfram [8] (or any other classification schema suggested so far, e.g., [10,12,13]) not only because we are taking explicitly initial condition into account for this classification but also because we are using a different measure to identify the different classes as described in Sec. IV.

IV. KOLMOGOROV COMPLEXITY AND DERIVED MEASURES

The method we use to cluster CA rules according to their spatiotemporal behavior is the normalized compression distance [25,34] which is based on Kolmogorov complexity [18]. In general, the Kolmogorov complexity $K(x)$ of a string x is the length $l(p)$ of the shortest binary program p^*, with $p^*=\arg\min_{p:\mathcal{U}(p)=x} l(p)$, which computes x on a universal computer (Turing machine) $\mathcal{U}$ without input

$$K(x) = \min_{p:\mathcal{U}(p)=x} l(p). \tag{4}$$

A practical problem is that $K(x)$ is not computable but only upper semicomputable [34]. Li *et al.* introduced in [34] a normalized and universal metric

$$d(x,y) = \frac{\max\{K(x|y),K(y|x)\}}{\max\{K(x),K(y)\}}, \tag{5}$$

called normalized information distance (NID), whereas the conditional Kolmogorov complexity is given by [44]

$$K(x|y) = \min_{p:\mathcal{U}(p,y)=x} l(p). \tag{6}$$

It has been argued [34] that NID can be approximated by

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}}, \tag{7}$$

called the NCD. Here, $C(x)$ denotes the compression size of string x and $C(xy)$ denotes the compression size of the concatenated stings x and y [34]. The quantities $C(\,)$ are obtained by compressors such as gzip or bzip2 (see Sec. V). Recently, several approaches have been suggested to utilize concepts based on Kolmogorov complexity in general [35,36] or the NCD measure [25,29] to problems from data analysis. For example, the NCD has been used to study texts, music files (corresponding to time series) (see [30] and references therein), DNA and protein sequences [37,38], protein structures [29], and metabolic networks [28].

Clustering the behavior of CA

In this paper we use the NCD [as defined in Eq. (7)] as dissimilarity measure to evaluate the distance between spatiotemporal patterns generated by different CA. More precisely, we generate for all $|\mathcal{R}|=256$ one-dimensional cellular automata with N cells, $r=1$ and $k=2$ spatiotemporal patterns

$Y^f=(S_{ti})$, $t\in\{1,\ldots,T_f\}$, and $i\in\mathcal{N}$ of length T_f. From these patterns we use only the last $T=T_f-t_{trans}$ time steps because the first t_{trans} time steps represent a transient that the cellular automata need to reach their final behavior. In general, the influence of this transient should only have marginal influence on the final results for $T_f\to\infty$; however, for practical reason it is important to ensure that t_{trans} is sufficiently long because T is finite. This aspect will be discussed in more detail in Sec. V. The resulting spatiotemporal pattern Y has N space and T time dimensions. For $\{Y_i\}$ we calculate now the dissimilarity matrix NCD according to Eq. (7)—details regarding this procedure are presented in Sec. V. This dissimilarity matrix NCD is then used to apply a hierarchical clustering method resulting in a dendrogram that represents a grouping of CA rules. A proper classification could be obtained from this dendrogram by cutting this tree structure at a certain level. Because the determination of such a level is difficult to derive theoretically we investigate the structure of the dendrogram exploratory.

V. PRACTICAL NOTES

In order to use the NCD one needs to estimate the compression size $C(x)$ of a string x. Practically, this is done by a compression algorithm such as gzip or bzip2, which we used primarily for our study. The principle working mechanism of both compression algorithms is as follows. Gzip uses the deflate algorithm, which is a variant of LZ77 [39], for preprocessing the data and a statistical compressor (usually Huffman [40]) for postprocessing. Bzip2 uses a Burrows-Wheeler transformation [41] and a Huffman coding. In the preprocessing step duplicated strings in the input data are replaced by a pointer to the previous string providing information about the distance and length of the string. In the postprocessing step frequently used symbols are replaced with shorter representations and less frequently used symbols with longer representations by application of Huffman coding. Regarding the used compressor there is one important characteristic of the compression algorithms that needs to be considered. Because NCD is a metric when the compressor is *normal* [25,34] one needs to make sure that the normality property is fulfilled. This implies that the following condition,

$$C(xx)=C(x), \tag{8}$$

holds. The compressors gzip and bzip2 fulfill this condition to a certain degree when the size of string x is below a compressor specific threshold called the block size. For gzip this is about 64 kbytes and for bzip2 it is about 900 kbytes. In [42] this behavior has been studied numerically. If x is larger than this block size the compressor packs x in smaller pieces which may lead to strong violations of the normality condition. In order to avoid this violation one needs to make sure that the investigated strings are in size smaller than block size/2 because otherwise the assumptions used for the proof that NCD is a metric do no longer hold. For our problem this is easy to ensure because we can chose an appropriate length T of the time series for a given size N of the CA. For our simulations we used gzip as well as bzip2 and found that both compression algorithms give similar results. We emphasize that the purpose of this paper is not to estimate Kolmogorov complexity as accurate as possible but to apply the NCD to cluster spatiotemporal patterns. The difference is that a proper clustering does not depend solely on the accuracy of the absolute values of Kolmogorov complexity but on the conservation of *relative* distances among these objects.

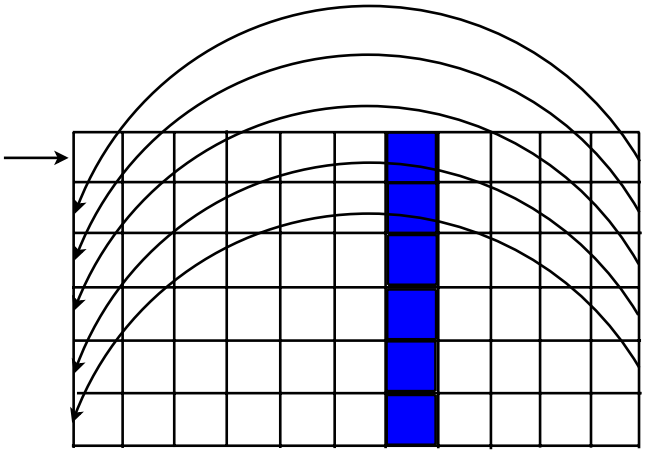

FIG. 1. (Color online) Mapping from a one-dimensional spatiotemporal pattern Y to a string x. Y represents a fixed point and x starts at the top left corner and is connected as the arrows indicate.

It is important to use an appropriate long burn-in time t_{trans} to avoid an evaluation of transients that are not characteristic for the long-time behavior of the CA. For example, when started from a random initial condition rule 97 takes in most cases 200 time steps to reach a periodic behavior but for certain random initial conditions twice as many time steps are necessary. For this reason we used for $N=50$ a burn-in phase of $t_{trans}=1000$ time steps.

In order to apply the NCD which utilizes a compression algorithm to evaluate $C(x)$ we need to transform the spatiotemporal pattern Y of the CA into a string x. The spatiotemporal pattern can be seen as matrix Y whereas a history of a cell is indexed by $t\in\{1,\ldots,T\}$ and $i\in\{1,\ldots,N\}$, $Y=(S_{ti})$. A natural mapping from S to x consists of a concatenation of, e.g., the rows of Y, $Y\to x=(S_1,S_2,\ldots,S_T)$. However, this simple transformation introduces a behavior not present in the spatiotemporal pattern that may lead to unwanted results. This problem is visualized in Fig. 1. For example, if Y represents a fixed point then the mapping defined above leads to a periodic pattern in the resulting string x, $Y\to x=(00000010000\ 00000010000\ 00000010000\ldots)$ [45]. To avoid this we map S to the difference of two consecutive time steps, $Y\to x=(S_1-S_2,S_2-S_3,\ldots,S_{T-1}-S_T)$. This mapping leads to an extension of the alphabet $\mathcal{S}$ of x consisting now of three symbols, $\mathcal{S}=(-1,0,1)$.

VI. RESULTS

A. Definition of initial conditions

We start our analysis by re-emphasizing that it is of importance to consider the initial condition as parameter that

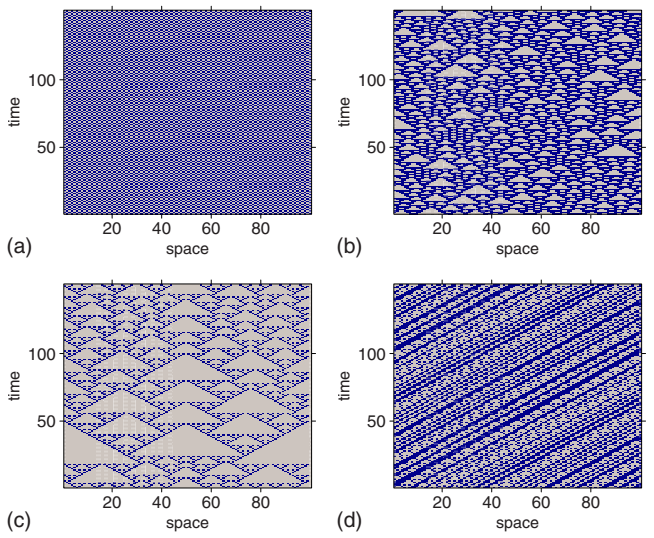

FIG. 2. (Color online) Spatiotemporal patterns of rule 122 (first row) and rule 210 (second row). First column: simple initial condition. Second column: random initial condition.

can strongly influence the behavior of the CA, i.e., its spatiotemporal pattern. In the following we define a simple initial condition as

$$s_i = \begin{cases} 1 & \text{for } i = \lfloor N/2 \rfloor \\ 0 & \text{else} \end{cases} \tag{9}$$

and a random initial condition as $s_i \sim \text{prob}(p, 1-p)$ for all $i \in \{1, \ldots, N\}$, with a probability of $p=0.5$ to observe a 1 or a 0 at each site s_i. We want to note that we intentionally restrict the simple initial condition to just one element whereas for the random initial conditions there exist 2^N different initial states.

To demonstrate the general dependence of the CA on the initial conditions we present in Fig. 2 two examples. The first row shows rule 122 and the second rule 210. For the patterns shown in the first column a simple initial condition was used, whereas in the second column we used a random initial condition as starting states. One can clearly see that it is possible that a periodic behavior for a simple initial condition can change to a complex pattern for a random initial condition (rule 122) but also the opposite behavior can occur (rule 210). In Sec. VI C we will continue this line of argument demonstrating that initial conditions are an intricate parameter that needs to be defined with care.

B. Simple initial condition

For the following simulations we generated spatiotemporal pattern for each CA rule for $N=50$ and $T=100$. As burn-in time we used $t_{trans}=1000$ time steps. Each of these spatiotemporal patterns was transformed as described in Sec. V and then the normalized compression distance was calculated according to Eq. (7). Then, NCD was used as a dissimilarity matrix for a hierarchical clustering for which the Ward method [43] has been used as distance measure. The resulting hierarchical clustering allows us to gain insights into the compressibility of the patterns because patterns of similar compressibility joining successively the same cluster.

The result of this clustering is shown in Fig. 3. The cluster on the left-hand side involves all 256 CA and consists of three major branches. The CA in these branches are labeled with "–," "––," or "+" and Table I provides the rule numbers for – and –– which are in the table termed 1 and 2. The right figure in Fig. 3 shows a magnification of the 76 CA rules labeled with a + in the left figure. In this figure there are two major branches but the CA rules are categorized in three different classes for reasons that will be discussed below. The category labels of these classes are "––," "–––," and "––––." Table II provides a summary for these CA rule numbers. Visual inspection of the CA rules categorized in classes 1 (–) and 2 (––) reveals that all CA rules (respectively their spatiotemporal pattern) in 1 are fixed points and in 2 are periodic. For the four sub-branches of the leftmost branch in the right of Fig. 3 we present four examples in Fig. 4. From these patterns the categorization becomes clear because only the top left pattern which was obtained from rule 30 shows random behavior whereas the other three exhibit complex behavior.

C. Random initial condition

For the following simulations from random initial conditions we generated spatiotemporal pattern for each CA rule for $N=50$ and $T=100$. As burn-in time we used again $t_{trans}=1000$ time steps. Each of these spatiotemporal patterns was transformed as described in Sec. V and then the normalized compression distance was calculated according to Eq. (7). Because different random initial conditions may result in a different clustering we averaged over E different initial conditions,

$$\overline{\text{NCD}} = \frac{1}{E}\sum_{j}^{E} \text{NCD}(j), \tag{10}$$

to obtain a mean normalized compression distance that is characteristic for the ensemble of all initial conditions. Then, $\overline{\text{NCD}}$ was used as a dissimilarity matrix for a hierarchical clustering for which the Ward method [43] has been used as distance measure.

The result of this hierarchical clustering is shown in Fig. 5. The left figure shows again the clustering of all 256 CA and the right figure provides a magnification of the CA rules labeled with a +. In Tables III and IV a summary of this clustering can be found providing the categorization of the CA rules in the four classes (1–4). Figure 6 visualizes some spatiotemporal pattern for some rules in the right of Fig. 5 that justify the categorization. Periodic patterns (class 2) are again clearly separated from classes 3 and 4 and also the latter classes can be distinguished.

Heterogeneous initial conditions

Finally, we demonstrate that the use of random initial conditions, as defined in Sec. VI C, is problematic with respect to a classification of spatiotemporal patterns because it can result in behavior that apparently belongs to different catego-

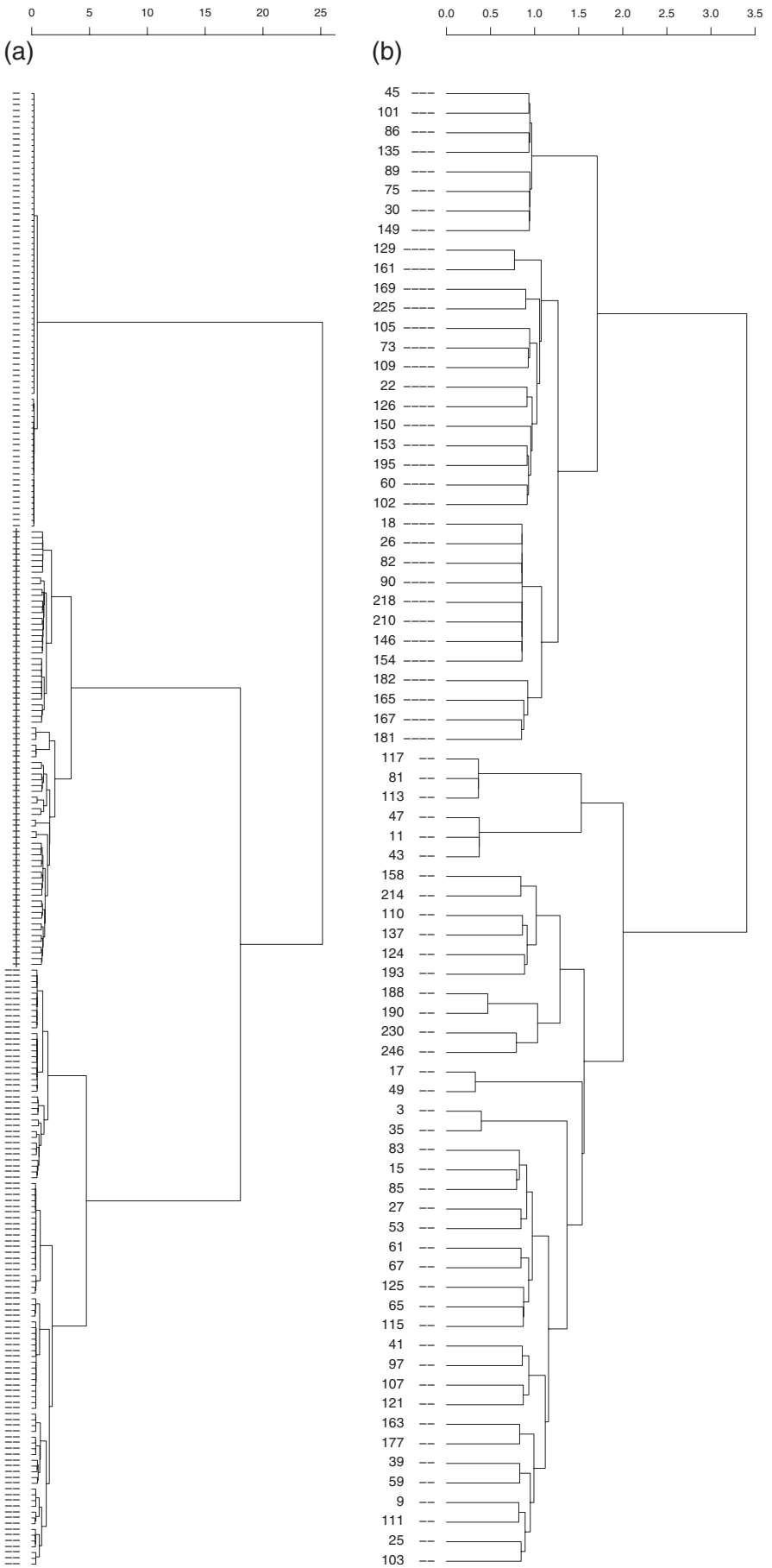

FIG. 3. Left: hierarchical clustering of all 256 CA started from a simple initial condition. Right: hierarchical clustering of 76 CA labeled with a + in the left figure. A summary of the categorization is provided by Tables I and II.

TABLE I. Classification of the behavior of CA started from a simple initial condition as shown in the left figure in Fig. 3.

Class	CA rule
1	0, 4, 8, 12, 13, 28, 32, 36, 40, 44, 64, 68, 69, 70, 72, 76, 77, 78, 79, 92, 93, 94, 96, 100, 104, 108, 128, 132, 133, 136, 140, 141, 151, 156, 157, 159, 160, 164, 168, 172, 183, 191, 192, 196, 197, 198, 199, 200, 203, 204, 205, 206, 207, 215, 217, 219, 220, 221, 222, 223, 224, 228, 232, 233, 235, 236, 237, 238, 239, 247, 249, 251, 252, 253, 254, 255
2	1, 2, 5, 6, 7, 10, 14, 16, 19, 20, 21, 23, 24, 29, 31, 33, 34, 37, 38, 42, 46, 48, 50, 51, 52, 54, 55, 56, 57, 58, 62, 63, 66, 71, 74, 80, 84, 87, 88, 91, 95, 98, 99, 106, 112, 114, 116, 118, 119, 120, 122, 123, 127, 130, 131, 134, 138, 139, 142, 143, 144, 145, 147, 148, 152, 155, 162, 166, 170, 171, 173, 174, 175, 176, 178, 179, 180, 184, 185, 186, 187, 189, 194, 201, 202, 208, 209, 211, 212, 213, 216, 226, 227, 229, 231, 234, 240, 241, 242, 243, 244, 245, 248, 250

ries. In order to make this point more clear we show in Fig. 7 an example. In Fig. 7 spatiotemporal patterns of rule 54 are shown. Each row corresponds to one random initial condition and each column shows a different time window. The second column shows the complete pattern of length $T=2650$ whereas the first column shows only the last 150 time steps thereof. One can clearly see that the first random initial condition (first row) leads after about 600 time steps to a periodic behavior of the CA whereas the second random initial condition (second row) does not. To make sure that the reason therefore is not just a too short burn-in time we continued the simulations for 100 000 time steps (from the same initial condition used in the second row) and found that the behavior is still not periodic. This reveals that there are two different regimes of *random initial conditions* that can lead to a completely different behavior of the spatiotemporal behavior of rule 54. From this one can conclude that it should be possible to categorize rule 54 in two different categories depending on the random initial condition. We demonstrate this by generating 200 spatiotemporal patterns from rule 54 all belonging to different random initial conditions and calculated for these patterns the normalized compression distance. The result of the hierarchical clustering is shown in the left of Fig. 8. One can see that there are two clearly separated branches in the tree. Inspection of the patterns from these branches reveals that the right branch corresponds to periodic behavior whereas the patterns in the left branches are nonperiodic.

TABLE II. Classification of the behavior of CA started from a simple initial condition as shown in the right of Fig. 3.

Class	CA rule
2	3, 9, 11, 15, 17, 25, 27, 35, 39, 41, 43, 47, 49, 53, 59, 61, 65, 67, 81, 83, 85, 97, 103, 107, 110, 111, 113, 115, 117, 121, 124, 125, 137, 158, 163, 177, 188, 190, 193, 214, 230, 246
3	30, 45, 75, 86, 89, 101, 135, 149
4	18, 22, 26, 60, 73, 82, 90, 102, 105, 109, 126, 129, 146, 150, 153, 154, 161, 165, 167, 169, 181, 182, 195, 210, 218, 225

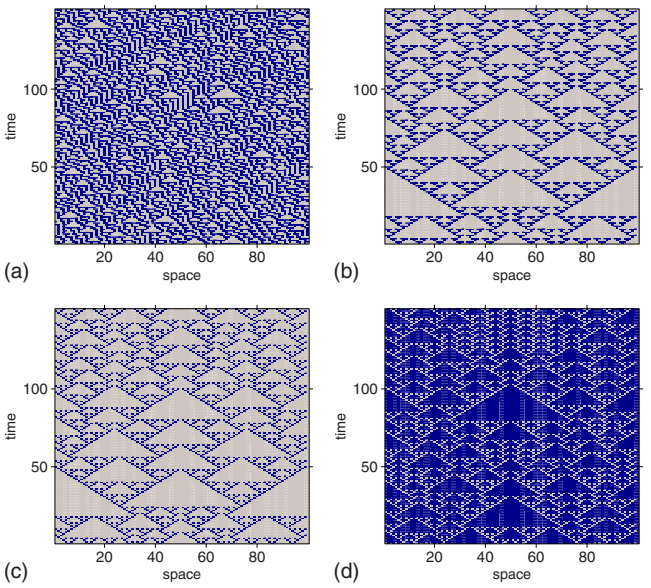

FIG. 4. (Color online) Spatiotemporal patterns of different rules started from simple initial conditions. Top left: rule 30. Top right: rule 22. Bottom left: rule 18. Bottom right: rule 182.

These results suggest to subdivide the random initial conditions by refining its definition to obtain a unique classification of rule 54. However, the purpose of this paper is not to provide such a specified definition but to point out that this is one source of complexity that makes the classification of CA rules so challenging.

In order to show that there are also rules that behave uniformly for all random initial conditions we present in the right of Fig. 8 an example showing the results for rule 30. There one can see that the distances between the branches are much smaller (see scale on the abscissa), indicating that the results for all initial conditions are very homogeneous suggesting not to cut the tree at any level which would lead to subcategories.

We want to point out that this observation corresponds to the behavior of rules 122 and 210, as shown in Fig. 2, when using simple or random initial conditions. However, the crucial difference is that it is less intuitive that the random initial conditions themselves, as defined in Sec. VI A, do not lead to a constant class behavior for the spatiotemporal patterns of CA.

We finish Sec. VI by a remark regarding the identify rule (204) and the shift rule (170). The identity rule (204) gives a fixed point and a shift rule (e.g., 170) gives a periodic pattern. According to Wolfram these two rules would be classified in class 1 (rule 204) and class 2 (rule 170). Hence, they would be considered different to each other but closer to each other than rules that show chaotic or complex behavior. Application of the Kolmogorov complexity measure leads to very similar results. This can be seen not only from Table I

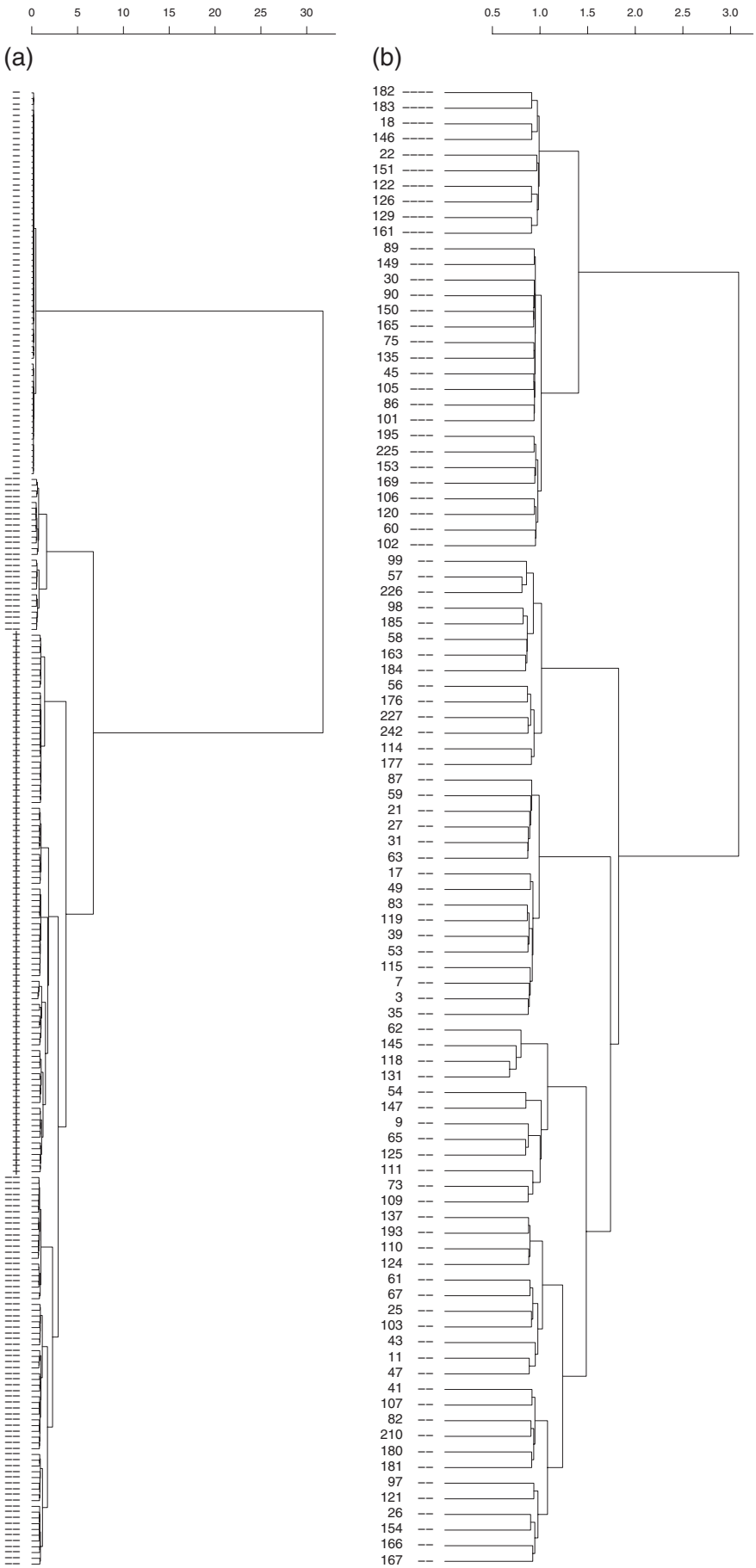

FIG. 5. Left: hierarchical clustering of all 256 CA started from random initial conditions. Right: hierarchical clustering of 95 CA labeled with a + in the left figure. A summary of the categorization can be found in Tables III and IV.

TABLE III. Classification of the behavior of CA started from random initial conditions averaged over $E=10$ runs as shown in the left of Fig. 5.

Class	CA rule
1	0, 4, 8, 12, 13, 32, 36, 40, 44, 64, 68, 69, 72, 76, 77, 78, 79, 92, 93, 96, 100, 104, 128, 132, 136, 140, 141, 160, 164, 168, 172, 192, 196, 197, 200, 202, 203, 204, 205, 206, 207, 216, 217, 218, 219, 220, 221, 222, 223, 224, 228, 232, 233, 234, 235, 236, 237, 238, 239, 248, 249, 250, 251, 252, 253, 254, 255
2	1, 2, 5, 6, 10, 14, 15, 16, 19, 20, 23, 24, 28, 29, 33, 34, 37, 38, 42, 43, 46, 48, 50, 51, 52, 55, 66, 70, 71, 74, 80, 81, 84, 85, 88, 91, 94, 95, 108, 112, 113, 116, 117, 123, 127, 130, 133, 134, 138, 139, 142, 143, 144, 148, 152, 155, 156, 157, 158, 159, 162, 170, 171, 173, 174, 175, 178, 179, 186, 187, 188, 189, 190, 191, 194, 198, 199, 201, 208, 209, 211, 212, 213, 214, 215, 229, 230, 231, 240, 241, 243, 244, 245, 246, 247

for simple initial conditions but also from Table III for random initial conditions (rule 204 is always in class 1 whereas rule 170 is always in class 2).

VII. DISCUSSION AND CONCLUSIONS

In this paper we introduced a semiautomatic method to classify spatiotemporal patterns of CA based on the normalized compression distance. The method is semiautomatic because it actually does not classify but cluster the rules of the CA leaving it to the investigator where to cut the tree obtained from the hierarchical clustering. This way the method allows an exploratory analysis studying many different conditions of CA and their resulting behavior efficiently. The spatiotemporal pattern of CA have been classified according to their compressibility. Kolmogorov complexity has been used previously [11], however, only for studying transition tables of CA but not with respect to their spatiotemporal pattern. Further, CA have been studied comparatively because the normalized compression distance NCD(x,y) compares always two strings x and y from, e.g., two different CA rules or initial conditions. Conventionally, CA have been studied one rule at a time with measures that are solely based on one rule or its spatiotemporal pattern [7,10].

TABLE IV. Classification of the behavior of CA started from random initial conditions averaged over $E=10$ runs as shown in the right of Fig. 5.

Class	CA rule
2	3, 7, 9, 11, 17, 21, 25, 26, 27, 31, 35, 39, 41, 43, 47, 49, 53, 54, 56, 57, 58, 59, 61, 62, 63, 65, 67, 73, 82, 83, 87, 97, 98, 99, 103, 107, 109, 110, 111, 114, 115, 118, 119, 121, 124, 125, 131, 137, 145, 147, 154, 163, 166, 167, 176, 177, 180, 181, 184, 185, 193, 210, 226, 227, 242
3	30, 45, 60, 75, 86, 89, 90, 101, 102, 105, 106, 120, 135, 149, 150, 153, 165, 169, 195, 225
4	18, 22, 122, 126, 129, 146, 151, 161, 182, 183

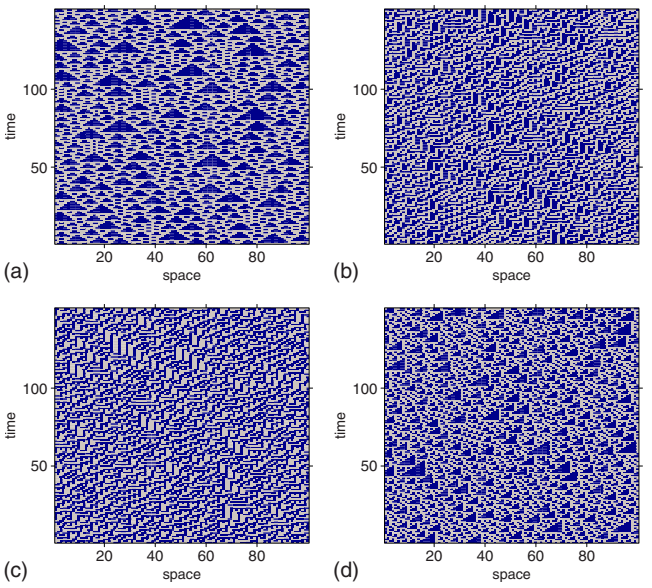

FIG. 6. (Color online) Spatiotemporal patterns of different rules started from random initial conditions. Top left: rule 161. Top right: rule 89. Bottom left: rule 101. Bottom right: rule 195.

We conducted our simulations for two different compression programs, gzip and bzip2, and found that the obtained results are very consistent indicating that the actual choice of the compressor to approximate $K(x)\sim C(x)$ has only a minor

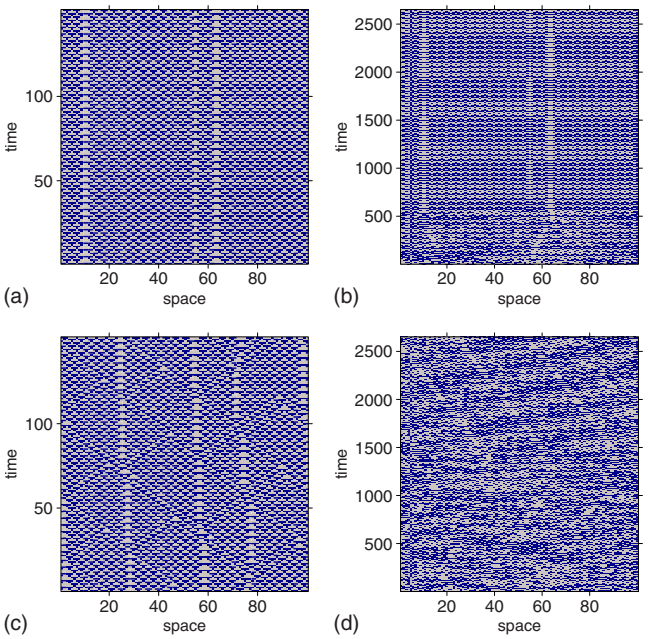

FIG. 7. (Color online) Spatiotemporal patterns of rule 54. Each row corresponds to one random initial condition and each column shows a different time window. The second column shows the complete pattern of length $T=2650$, whereas the first column shows only the last 150 time steps thereof.

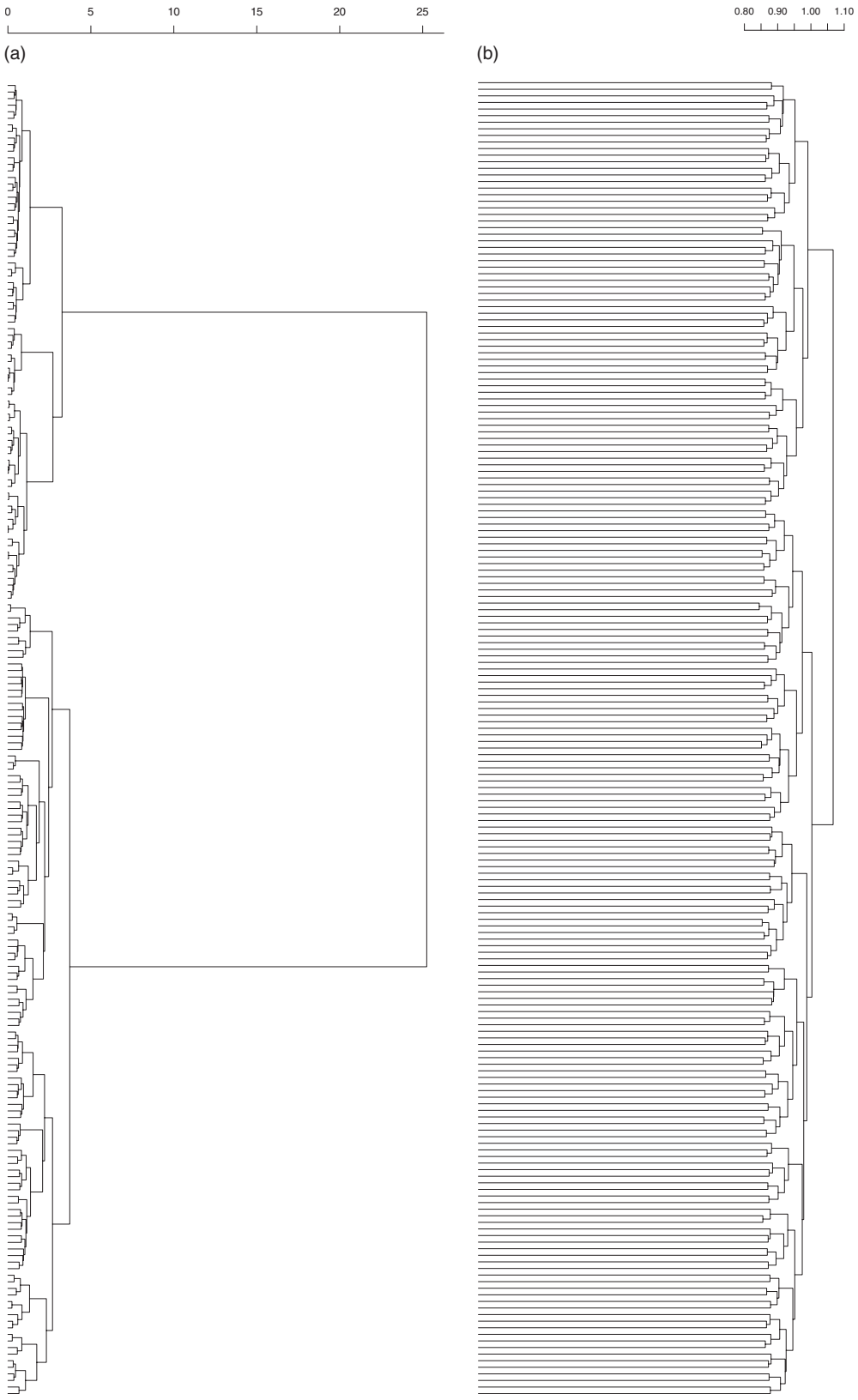

FIG. 8. Hierarchical clustering of rules 54 (left) and 30 (right) for 200 different random initial conditions.

influence on the results as long as the characteristic block size of the corresponding compressor is not exceeded. This is in contrast to the initial conditions which have a profound influence on the result of the clustering. For this reason we introduced an averaging over the NCD matrices obtained from different *random* initial conditions because we noticed from our simulations that some rules are sometimes labeled as class 2 and sometimes as class 3 or 4. Interestingly, as we

found out, averaging does not solve the problem fundamentally but provides just a majority decision with respect to the size of initial conditions that belong to a certain class (see, for example, rule 54 in Figs. 7 and 8). For this reason a statement such as "rule 54 is periodic" or "rule 54 is not periodic" is ill-posed for random initial conditions with respect to the compressibility measure we employed because it does not lead to a unique classification of a CA rule for all initial conditions. As remedy, a refinement of the definition for random initial conditions will lead to a partitioning of the space of random initial conditions in two (or more) disjoint regions $\mathcal{X}_i$ guaranteeing that each initial condition from $S_0 \sim \mathcal{X}_i$ leads to the same class label. The problem with such a partitioning is not only that it would be laborious to find the actual partitioning but also that two different rules may have two different partitions. But this would require a combinatorial treatment of the problem to identify initial conditions for which *all* CA rules show a constant class behavior. From this discussion it becomes clear that the *parameter* in form of initial conditions increases the complexity of the classification problem of CA considerably. For this reason the classification of one-dimensional CA with $k=2$ and $r=1$ depends actually on many more parameters and not just on k and r, hidden in the initial conditions, which are difficult to define precisely in advance without performing the simulations themselves. We want to point out that the problem of the dependence of the classification on the initial conditions has been already recognized and discussed in [31]. However, our study is different by providing additionally a tractable methodological approach leading to quantitative results.

We think that the classification of spatiotemporal patterns based on their compressibility, as measured by the normalized compression distance [25], is a very fruitful way to study complex systems in general and especially CA. For example, our exploratory analysis approach can be immediately used to study one-dimensional CA with $k>2$ and $r>1$ in an analog way. For the analysis of higher-dimensional CA one would need to extend our approach by studying how the d-dimensional spatiotemporal patterns are best mapped into a string representation needed for the NCD. As discussed in Sec. V, it could be involved to avoid the unintended introduction of symmetries that could lead the NCD astray because in higher dimensions accounting for such influences may become more challenging.

In summary, our results show that despite the complexity of single CA rules, which deserve to be studied individually, the analysis of a collective of CA is feasible with the help of our exploratory approach based on the normalized compression distance. When studied comparatively, the overall complexity may be tamable by unraveling latent dependencies governing the hierarchical organization of this multivariate problem.

ACKNOWLEDGMENTS

We would like to thank Mike Coleman, Matthias Dehmer, Earl F. Glynn, and Dan Thomasset for fruitful discussions and help on various aspects of this paper.

[1] J. Von Neumann and A. Burks, *Theory of Self-Reproducing Automata* (University of Illinois Press, Urbana, 1966).
[2] U. Frisch, B. Hasslacher, and Y. Pommeau, Phys. Rev. Lett. **56**, 1505 (1986).
[3] B. Madore and W. Freedman, Science **222**, 615 (1983).
[4] Y. Oono and M. Kohmoto, Phys. Rev. Lett. **55**, 2927 (1985).
[5] M. Cook, Adv. Complex Syst. **15**, 1 (2004).
[6] K. Lindgren and M. G. Nordahl, Complex Systems **4**, 299 (1990).
[7] S. Wolfram, Rev. Mod. Phys. **55**, 601 (1983).
[8] S. Wolfram, Physica D **10**, 1 (1984).
[9] P. Binder, J. Phys. A **24**, L31 (1991).
[10] A. Wuensche, Complexity **4**, 47 (1999).
[11] J.-C. Dubacq, B. Durand, and E. Formenti, Theor. Comput. Sci. **259**, 271 (2001).
[12] H. A. Gutowitz, Physica D **45**, 136 (1990).
[13] C. Langton, Physica D **42**, 12 (1990).
[14] R. Badii and A. Politi, *Complexity: Hierarchical Structures and Scaling in Physics* (Cambridge University Press, Cambridge, 1997).
[15] C. Bennett, in *The Universal Turing Machine—A Half-Century Survey*, edited by R. Herken (Oxford University Press, New York, 1988), pp. 227–257.
[16] M. Gell-Mann and S. Lloyd, Complexity **2**, 44 (1998).
[17] P. Grassberger, Int. J. Theor. Phys. **25**, 907 (1986).
[18] A. N. Kolmogorov, Probl. Inf. Transm. **1**, 1 (1965).
[19] S. Lloyd and H. Pagels, Ann. Phys. **188**, 186 (1988).
[20] R. Solomonoff, A Preliminary Report on a General Theory of Inductive Inference (Revision of Report No. V-131), Contract AF 49(639)-376, Report ZTB-138, Zator Co. (1960), http://www.world.std.com/~rjs.pubs.html
[21] H. Zenil, e-print arXiv:0910.4042, Complex Syst. (to be published).
[22] G. Chaitin, J. Assoc. Comput. Mach. **13**, 547 (1966).
[23] J. Crutchfield and N. Packard, Physica D **7**, 201 (1983).
[24] W. Bialek, I. Nemenman, and N. Tishby, Neural Comput. **13**, 2409 (2001).
[25] R. Cilibrasi and P. Vitányi, IEEE Trans. Inf. Theory **51**, 1523 (2005).
[26] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications* (Springer, New York, 1997).
[27] R. Cilibrasi and P. Vitányi, IEEE Trans. Knowl. Data Eng. **19**, 370 (2007).
[28] M. Nykter, N. D. Price, A. Larjo, T. Aho, S. A. Kauffman, O. Yli-Harja, and I. Shmulevich, Phys. Rev. Lett. **100**, 058702 (2008).
[29] N. Krasnogor and D. A. Pelta, Bioinformatics **20**, 1015 (2004).
[30] P. Vitányi, F. Balbach, R. Cilibrasi, and M. Li, in *Information Theory and Statistical Learning*, edited by F. Emmert-Streib and M. Dehmer (Springer, New York, 2008), pp. 45–82.
[31] A. Dhar, P. Lakdawala, G. Mandal, and S. Wadia, Phys. Rev. E **51**, 3032 (1995).
[32] N. Fatès, in Discrete Models for Complex Systems

(DMCS'03), edited by M. Morvan and É. Rémila, 2003, Vol. AB, pp. 155–166, URL: http://www.dmtcs.org/proceedings/html/dmAB0113.abs.html
[33] S. Wolfram, *A New Kind of Science* (Wolfram Media, Champaign, IL, 2002).
[34] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi, IEEE Trans. Inf. Theory **50**, 3250 (2004).
[35] S. Grumbach and F. Tahi, J. Info. Process. Manag. **30**, 875 (1994).
[36] J. S. Varre, J. P. Delahaye, and E. Rivals, German Conference on Bioinformatics, Koel, Germany, 1998 (unpublished).
[37] A. Kocsor, A. Kertesz-Farkas, L. Kajan, and S. Pongor, Bioinformatics **22**, 407 (2006).
[38] M. Li, J. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, Bioinformatics **17**, 149 (2001).
[39] J. Ziv and A. Lempel, IEEE Trans. Inf. Theory **23**, 337 (1977).
[40] D. Huffman, Proc. IRE **40**, 1098 (1952).
[41] M. Burrows and D. Wheeler, HP Report No. 124 (1994), http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-124.html
[42] M. Cebrian, M. Alfonseca, and A. Ortega, Commun. Inf. Syst. **5**, 367 (2005).
[43] S. Theodoridis and K. Koutroumbas, *Pattern Recognition* (Academic Press, New York, 2003).
[44] Here $\mathcal{U}(p,y)$ is a universal computer producing output x for input y and program p.
[45] Spaces are just included for a better perception of the pattern.