Complexity of classical dynamics of molecular systems. I. Methodology

Dmitry Nerukh, George Karvounis, and Robert C. Glen

View Online   Export Citation

ARTICLES YOU MAY BE INTERESTED IN

Complexity of classical dynamics of molecular systems. II. Finite statistical complexity of a water–Na^+ system
The Journal of Chemical Physics **117**, 9618 (2002); https://doi.org/10.1063/1.1518011

The Journal
of Chemical Physics 2018 EDITORS' CHOICE READ NOW!

Complexity of classical dynamics of molecular systems. I. Methodology

Dmitry Nerukh, George Karvounis, and Robert C. Glen
Unilever Centre for Molecular Informatics, Department of Chemistry, Cambridge University, Cambridge CB2 1EW, United Kingdom

Methods for the calculation of complexity have been investigated as a possible alternative for the analysis of the dynamics of molecular systems. "Computational mechanics" is the approach chosen to describe emergent behavior in molecular systems that evolve in time. A novel algorithm has been developed for symbolization of a continuous physical trajectory of a dynamic system. A method for calculating statistical complexity has been implemented and tested on representative systems. It is shown that the computational mechanics approach is suitable for analyzing the dynamic complexity of molecular systems and offers new insight into the process. © *2002 American Institute of Physics.* [DOI: 10.1063/1.1518010]

I. INTRODUCTION

The motions of atoms and molecules in the condensed phase is so complicated that it is often considered as random. However, when rigorously defined this motion is *chaotic*. That is, the underlying dynamics of seemingly very complicated motion can be completely deterministic and even very simple. There is direct evidence of the chaotic nature of molecular motions.[1,2] The first criterion of the chaotic character of a dynamic process is its Lyapunov exponents. It has been shown[1,2] that biomolecular systems have positive Lyapunov exponents, which is the condition for a system to be chaotic. Even a simple linear triatomic molecule possesses chaotic dynamics, which is rigorously proved in Ref. 3 by the existence of the transversal homoclinic points in a Poincaré map of the system. Together with these numerical simulations there is experimental evidence of chaos on the microscopic level[4] showing positive Lyapunov exponents of the motion of a Brownian particle immersed in liquid.

Another distinctive feature of molecular systems is their multiparticle nature. The dynamics of the system is made up of the motion of a large number of small, relatively simple interacting particles. Such systems, sometimes referred to as large systems, are capable of producing very complex dynamics. Their time evolution has a self-organizing character—a qualitatively new complex behavior emerges from simple laws of interactions between the constituent parts.[5] It is also now being recognized that this type of behavior is quite generic for the class of nonlinear chaotic systems.[6]

Molecular systems are Hamiltonian nonlinear dynamical systems made of a large number of simply interacting parts. This, therefore, makes them potentially capable of exhibiting complex, self-organizing, emergent behavior. The diversity of physical–chemical processes from phase transitions to protein folding is a consequence of this characteristic. It is thus of great interest to study the emergent nature of molecular systems in more detail.

The key point in understanding emergent behavior is to find a way of analyzing the complexity of the system. The complexity and, particularly, the dynamical complexity is ultimately related to self-organization and emergence. The term "complexity" is loosely defined. There are many approaches to calculating complexity that vary considerably in their definition and implementation. Nevertheless, they all aim to estimate the same characteristic of the system, namely, how sophisticated are the dynamical laws governing the time evolution of the system.

In order to make complexity analysis a practical approach, a method for calculating complexity must be chosen. It should be noted that complexity measures are mostly applied to abstract mathematical models exhibiting chaotic dynamics.[7] For our purposes, however, a practical method for estimating the complexity of a continuous physical trajectory is desired. If we devise a method for calculating complexity quantitatively we can apply it to specific molecular characteristics. This should then describe the key features of the system and be a suitable descriptor of the phenomena under study.

The mathematical abstraction used to analyze complexity is in describing the system in terms of "symbolic dynamics"; a signal is replaced by a sequence of symbols from an "alphabet" of finite size. In the simplest case the alphabet consists of zeros and ones and "symbolization" becomes a binary coding. Even though complexity has only recently become an active field of research it rests on such well-known approaches as Shannon entropy and Kolmogorov–Chaitin algorithmic complexity.

Shannon entropy is widely used for characterizing the informational content of a signal and in fact was originally introduced to describe the information capability of a communication channel.[8] Its close connection with thermodynamic entropy and the theory of information makes it popular in the analysis of many physical processes.[9,10] The "excess entropy" concept[11] is a step towards a quantitative measure of complexity. It gives a measure of the "memory" of a dynamical system thus giving an estimate of the sophistication of the dynamical law that defines the behavior of the system. It was first introduced by Crutchfield and Packard[11]

9611

and recognized as "an effective measure of complexity" by Grassberger.[12]

Kolmogorov–Chaitin (KC) complexity is the length of the minimal program for a Universal Turing Machine which when run reproduces the process at hand. This is a mathematically well-developed approach which is unfortunately not realizable in practise. Nevertheless, it provides the foundation for many variations and extensions. This measure of complexity is related to Shannon entropy because they both estimate the size of the informational description of the process.

Currently a number of approaches are being developed. Among them is the "approximate entropy" approach,[13] a method exploring Fisher information[14] and even some "simple" algorithms[15] (which are however open to criticism[16,17]).

One of the most valuable insights offered by the application of complexity analysis is that it can reveal the mechanism of the appearance of qualitatively new, complex behavior from simple elementary events, in other words, an explanation of emergent behavior in physical systems. This set of problems also plays a significant role in evolution theory and is extensively developed by Kauffman.[5] One of the main postulates is that in this class of dynamical systems which show self-organization, the ability to process information is most effective when the system is "on the edge of chaos." In other words, if a system exhibits both types of dynamics: deterministic and chaotic depending on the parameters, the richest informational content of its behavior is found for the intermediate values of the parameters.

The same ideas are present in the more formal field of nonlinear chaotic dynamics. For example, emergence may be elucidated from an abstract map at the onset of chaos.[18] This is, however, only a hypothesis and there are doubts on how general this rule is.[19]

For our purposes we adopted the approach by Crutchfield *et al.* termed "computational mechanics." [20–23] This approach combines and implements the ideas from both Shannon entropy and KC algorithmic complexity theories. Here a symbolic sequence is used to reconstruct an algorithmic automaton that propagates the system from one state (the so-called "causal state") to the next one. "Computational" signifies that the complexity of the system (a "statistical complexity" in this case) is equal to the complexity of this automaton. Being well developed from the formal mathematical point of view this approach provides a practical algorithm for calculating the complexity of real systems. It has been applied to a number of systems, both mathematical and real physical models.[22,24–26]

If we are to calculate the statistical complexity of molecular systems we have to resolve a number of issues. Most importantly, an algorithm for converting a continuous trajectory of the systems to a symbolic sequence should be devised. A general solution to the generation of a symbolic representation (the concept of a generating partition) is, unfortunately so far applicable to only a very limited number of abstract mathematical maps.[7] The other difficulty in implementing the original computational mechanics methodology is that the reconstruction of the causal states automata is not
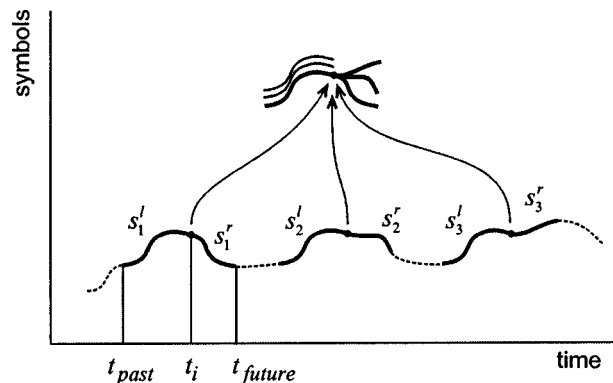

FIG. 1. A schematic representation of the equivalence relations. The left ("past") subsequences s_1^l, s_2^l, and s_3^l (all symbols on the $[t_{past}, t_i]$ interval) are the same. They lead to a distribution of right ("futures") subsequences s_1^r, s_2^r, and s_3^r ($[t_i, t_{future}]$).

always straightforward and may require nontrivial methods. That is why we use, at least at the current stage of this investigation, a variation of the original Crutchfield method suggested by Perry and Binder.[27]

This paper primarily deals with a methodology for the calculation of statistical complexity of a molecular system. The theory, computational details and testing of the algorithm on simple model signals is described. The accompanying paper,[28] the application of the method to Na^+–water classical MD simulation is discussed. In the forthcoming papers we will deal with more complex biochemical and other systems that exhibit self-organizing features.

II. THEORY

In the following symbolic dynamics is considered, i.e., the signal consists of discrete symbols assigned to discrete time steps. Let a set of symbols corresponding to each time step t_i form a sequence S. To calculate the statistical complexity S is decomposed into a set of left s_i^l (past) of length l and right s_i^r (future) of length r halves joined together at time points t_i. Consider a particular left subsequence s_1^l and all left subsequences equivalent to it: s_2^l and s_3^l. Collect a set of all right subsequences following this unique left subsequence (Fig. 1). Each right subsequence has its probability conditioned on the particular left one: $\Pr(s^r|s_i^l)$. The equivalence relation between any two left subsequences can now be defined. Two unique left subsequences s_i^l and s_j^l are equivalent if their right distributions are the same up to some tolerance value δ: $\Pr(s^r|s_i^l) = \Pr(s^r|s_j^l) + \delta$. A set of all equivalent left subsequences forms an "equivalence class." The equivalence classes have their own probabilities (A_i) equal to the sum of probabilities of the constituent left subsequences.

The importance of the notion of equivalence classes is that they represent the states of the system that define the dynamics at future moments—the "causal states." The time evolution of the system can be viewed as traversing from one causal state to the other with a probability defined by $\Pr(s^r|s_i^l)$. The set of the causal states together with the tran-

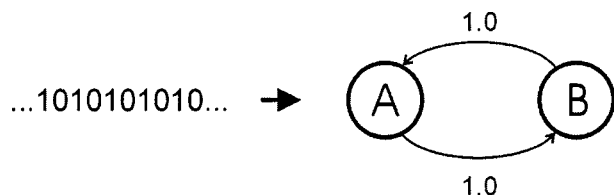FIG. 2. ϵ-machine reconstruction. A and B are two causal states of the system. Numbers on the arrows show the transition probability between the states.
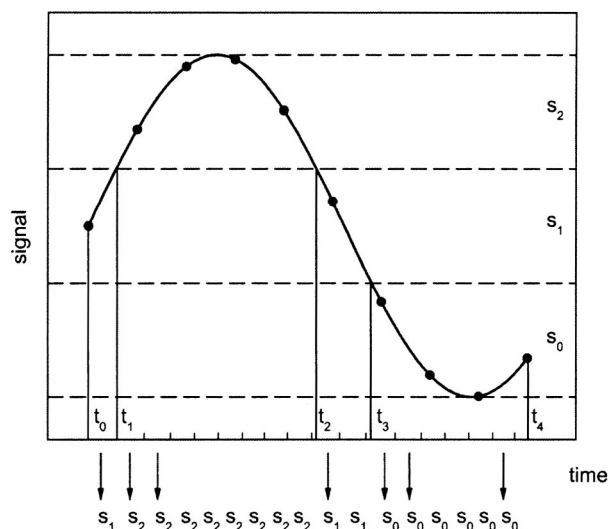

FIG. 3. Symbolization of a continuous signal. Only discrete data points (the dots) representing the continuous signal (the solid curve) are available.

sition probabilities constitute a so-called "ϵ-machine." ϵ-machines represent the minimal computation necessary to reproduce the dynamics of the system.

For example, a simple sequence consisting of alternating zeros and ones has an ϵ-machine shown in Fig. 2. Here state A is an equivalence class consisting of just one left subsequence ...01010. State B consists of a subsequence ...10101. The transition probabilities are equal to 1.0 because each state always leads to the other.

The statistical complexity is defined as the informational size of the ϵ-machine. The measure of this is the Shannon entropy of the causal states,

$$C \equiv -\sum_{A_i} \Pr(A_i)\log_2 \Pr(A_i), \qquad (1)$$

where A_i are causal states. In contrast to KC complexity this measure provides a zero complexity for *both* extremes—a constant signal and a purely random process. The maximum value of complexity lies somewhere in between these two limits.

This approach to calculating complexity gives an extensive opportunity for analyzing the intrinsic mechanism of the dynamics. In particular, the ϵ-machines can be classified by the algorithmic languages required to construct them. Their hierarchy represents various levels of complexity. It is suggested[22] that the transition from one level to the other upwards in the hierarchy represents the event of emergence. Thus, the analysis of the reconstructed ϵ-machine is a key point in studying dynamical complexity of the system.

In this work we use a simplified version of statistical complexity due to Perry and Binder[27] denoted "finite statistical complexity." It avoids the explicit reconstruction of the ϵ-machine but converges to statistical complexity. This, on the one hand, simplifies the algorithm and makes it more robust, however on the other it lacks the detailed explanatory power of the original approach.

The approach considers substrings of length l for the left subsequence and of length r for the right. It then estimates the occurrence frequencies of the left subsequences $P(x_l)$ and for each x_l the occurrence frequencies of the right subsequences $P(x_r|x_l)$. Then the equivalence classes $\{x_l\}_i$ are formed by comparing the distributions of the right strings. Finally the finite statistical complexity is calculated by the formula,

$$C = -\sum_i P(\{x_l\}_i)\log_2 P(\{x_l\}_i), \qquad (2)$$

where $P(\{x_l\}_i)$ is a probability of each equivalence class.

III. COMPUTING FINITE STATISTICAL COMPLEXITY

A. Signal symbolization

The crucial part in the implementation of the methodology is converting a continuous real signal into a sequence of symbols, "symbolization." There is a review on symbolization[29] reflecting the current state of affairs in this field. The rigorous approach to symbolization is to use a "generating partition."[7] There is a mathematical foundation for this type of encoding of the trajectory.[30] Unfortunately, there is no practical algorithm for constructing a generating partition for an arbitrary n-dimensional signal. The generating partitions are known for some, low dimensional systems, for example the Henon map.[31] Various criteria for partitioning are discussed in Ref. 32. For practical applications the partitioning is often chosen arbitrarily. This, however, may lead to erroneous conclusions about the dynamics of the system. Some of the problems arising when a misplaced binary partition is used are discussed in Ref. 33.

We have paid special attention to the symbolization of molecular trajectories. Because of the lack of rigorous definition of a partition for a general case, we simply divide the whole interval covered by the signal into k equal parts (Fig. 3) and investigate the behavior of the algorithm at different values of k. We also devised a special approach for generating the symbols based on the given partition.

The one-dimensional case is shown in Fig. 3. The signal is considered as a continuous function represented by the dots at the discrete experimental data points. The alphabet is constructed by the partitioning and for $k=3$, consists of three symbols: $\{s_0,s_1,s_2\}$ (Fig. 3). The resulting symbolic sequence is shown at the bottom row in Fig. 3. The algorithm for the symbolization is as follows:

(1) Find the intersection points of a signal with the partition lines (t_1, t_2, t_3) [take the first and the last points of the signal as well (t_0, t_4)].
(2) Find the smallest interval in time Δt (which is equal to the length of $[t_0,t_1]$ for this example).
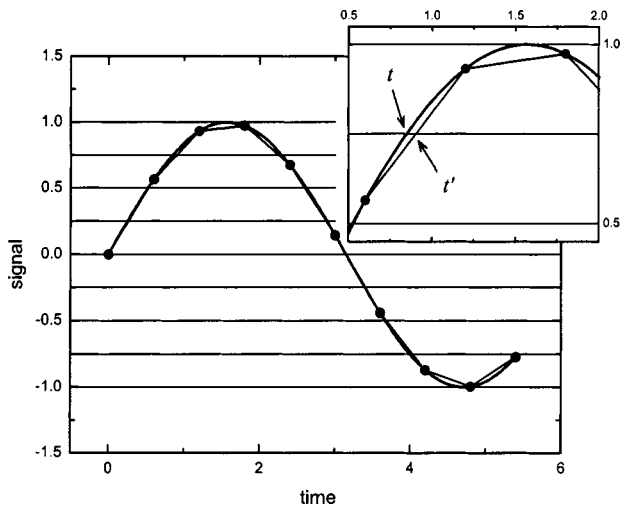
FIG. 4. Linear interpolation for finding the intersection points of the signal and the partition lines. t is a true intersection point, t' is the one used in the calculation.
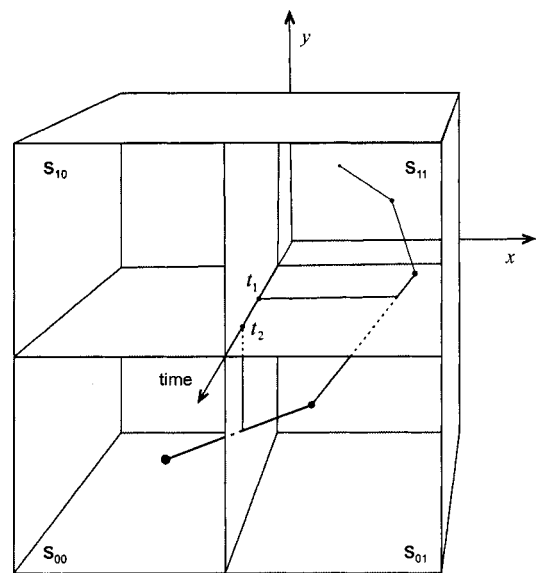

FIG. 5. Two-dimensional signal symbolization. The dots and the straight lines joining them represent the continuous signal. The time interval boundaries (t_1, t_2) are found by projecting the points of intersection of the signal with the partitioning planes. The symbolic alphabet consists of four symbols: $s_{00}, s_{01}, s_{10}, s_{11}$.

(3) For each interval find the number of symbols produced by this interval by dividing its length by Δt.

(4) Form the final sequence by choosing the symbol from the partition space where the signal falls between the intersection points (Fig. 3).

It is important to consider the intersection points t_i as time interval boundaries and not the data points themselves. Otherwise, if the data points do not fall in the points of natural periodicity an artificial randomness is introduced into the final symbolic sequence. Also, generating a sequence of repeating symbols like the ones on the $[t_1, t_2]$ interval preserves more information from the original signal.

In choosing Δt some tolerance was used and Δt was not allowed to be less than this tolerance (typically 10% of the time step). Also, when comparing different signals the same value of Δt was used for all signals.

A continuous trajectory is normally not available since we have only a discrete set of experimental points therefore an interpolation must be used. We used a simple linear interpolation as shown in Fig. 4. A higher order scheme may be introduced if necessary. However, if the points are dense enough the linear approximation does not introduce a significant error into the final result.

The algorithm is straightforwardly generalized to the n-dimensional case. The partition lines are now hyperplanes and the signal is a curve in $n+1$ dimensional space (Fig. 5). Also, there are k^n number of symbols in the alphabet, where k is the number of partitions. The intersection points t_i are the time coordinates of the intersections of the hyperplanes with the signal curve.

B. Symbolic dynamics

The algorithm for computing the finite statistical complexity follows the method described in Ref. 27:

(1) Go through all left subsequences of length l (x^l) and the following right subsequences of length r (x^r). Calculate the occurrence frequencies for the left subsequences $P(x^l)$.

(2) Collect all unique left subsequences x_i^l. Form the sets of right subsequences $\{x^r\}_i$ for each left subsequence. For each set $\{x^r\}_i$ calculate the occurrence frequencies of each right subsequence within the set making them the distributions.

(3) Form the equivalence classes by comparing the right subsequence distributions. If they are equal, add the corresponding left subsequences to the same equivalence class.
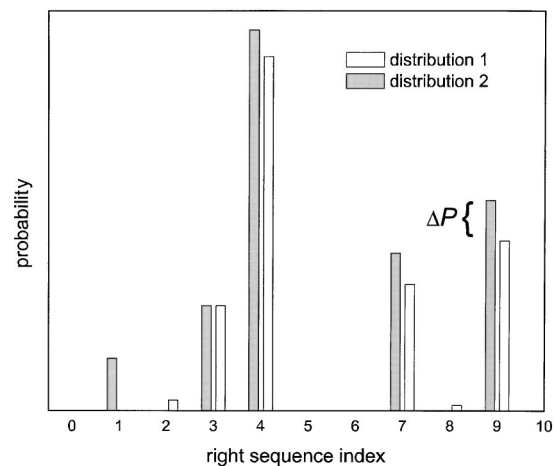

FIG. 6. The definition of the equivalence of two distributions of symbolic sequences. The probabilities number 2 and 8 from distribution 1 are smaller than the maximal difference between the equal sequences ΔP. Thus, these two distributions are equal.
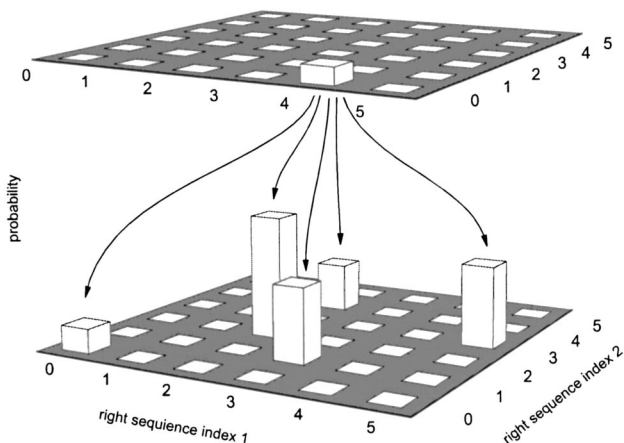
FIG. 7. An example of the left subsequence (upper panel) and following right subsequences (lower panel) probabilities. The subsequences length is equal to 2 and the alphabet consists of 6 symbols (see text).

FIG. 9. The dependence of the complexity on the number of partitions for various length of left and right subsequences for the f_{01} test signal.

(4) Calculate the equivalence class probabilities as a sum of the occurrence frequencies of the left subsequences $P(x^l)$ belonging to the class.

(5) The finite statistical complexity is calculated by the formula (2).

The important point is the criteria for comparing the distributions of right subsequences (step 3). According to the original approach by Crutchfield, they must be equal in the statistical sense up to a tolerance δ. In our algorithm we adopted rather loose criterion. First, the distributions should have common sequences. Second, the unequal ones should have a probability less then the biggest difference between the probabilities of the equal ones (Fig. 6).

Special attention must be paid to the cases of significantly random processes. Figure 7 shows a case of a 6-symbol alphabet and left and right subsequences of length
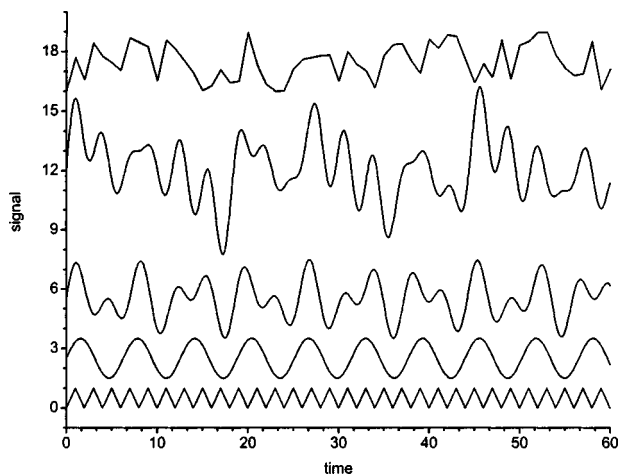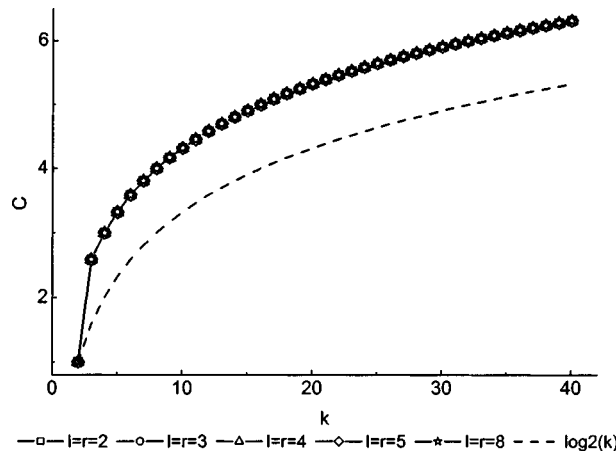

FIG. 8. Test signals. From bottom to top:

$$f_{01}=\begin{cases} t-2i, & i=\cdots-1,0,1,2\cdots \text{ for } t\in[2i,2i+1] \\ -t+2i, & i=\cdots-1,0,1,2\cdots \text{ for } t\in[2i-1,2i] \end{cases};$$

$f_s=\sin(t);f_{2s}=\sin(at)+\sin(bt); f_{5s}=\Sigma_{i=1}^{5}\sin(a_it); f_r$: straight lines joining the points $[i,r]$, where $i=\cdots-1,0,1,2...$ and r are random number in the interval $[0,3]$.

2. The upper panel contains the probabilities of the left subsequences and the bottom one those for the right. The probability of occurrence of a particular left subsequence (upper panel) is shown with the corresponding right subsequences (lower panel). For a purely random process each left subsequence leads to all possible right subsequences with the same probability, so that the lower panel in Fig. 7 is covered with bars of the same height. This means that the ϵ-machine consists of a single causal state which is visited an infinite number of times.

In reality, however, the finite number of subsequences leads to a situation in which the distribution on the lower panel is not uniform or some subsequences are even completely missing. This will make the algorithm produce artificial causal states and, consequently, increase the probability of those states. To avoid this, a sufficient set of subsequences must be accumulated, in other words, the original signal must be long enough to provide the correct approximation of the right subsequences distributions. For an alphabet of m symbols and right subsequences of length r it needs at least m^r symbols to cover all possible right subsequences. Keeping in mind the exponential dependence of the length of the alphabet on the dimensionality of the signal it is obvious that to get a correct zero value complexity for a purely random signal we need a very long data stream, especially for high-dimensional cases.

In the worst situation, when only one right sequence corresponds to each left sequence, we have m^r equivalence classes each with probability m^r. The formula for the complexity becomes

$$C=-\sum_{i=1}^{m^r}\frac{1}{m^r}\log_2\frac{1}{m^r}=r\log_2 m, \qquad (3)$$

and this indicates that for a large number of partitions the complexity is $\log_2 m$ dependent. It is worth stressing that this is only true for signals with a random component. For deterministic signals the logarithmic dependence may arise for another reason discussed in the next section.
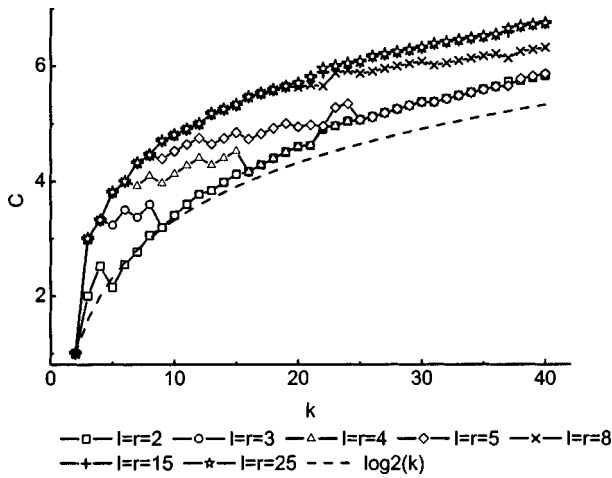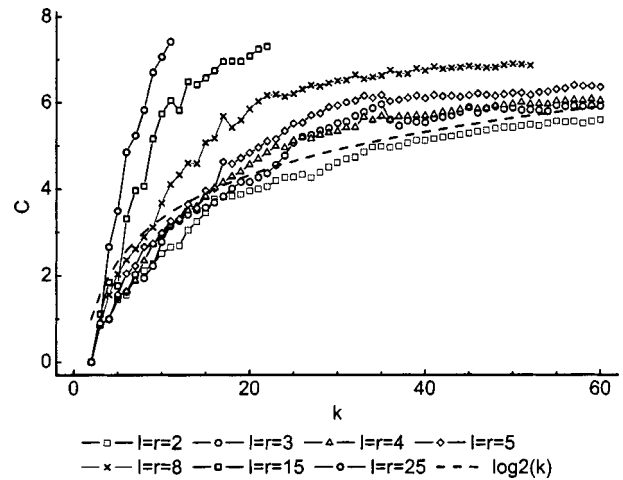
FIG. 10. Same as Fig. 9 but for f_s.

FIG. 12. Same as Fig. 9 but for f_r.

IV. RESULTS AND DISCUSSION

We tested the algorithm on the model signals shown in Fig. 8. Once the procedure for comparing the right sequence distributions is established the only parameters the algorithm depends on are the length of the subsequences l (we assumed the lengths of left and right subsequences are equal in all calculations) and the number of partitions k. We investigated the complexity of the test signals at various values of both these parameters.

The expected values of complexity are in this order: the lowest for the random signal, then f_{01}, f_s, f_{2s}, and the most complex f_{5s}.

Apparently, the more partitions that are used for symbolization, the more information is encoded in the symbolic sequence and the higher complexity we obtain. There is another reason for increasing complexity with k. It can be proven that Shannon entropy has $-\log_2 \Delta$ dependence in the $\Delta \to 0$ limit, where Δ is a discretisation interval.[34] As far as statistical complexity is a Shannon entropy measure, its limit should also have logarithmic dependence.

Dependence on the length l is important because, according to Ref. 27, finite statistical complexity converges to

statistical complexity with increasing l. Therefore, we have to make sure that our results converge with increasing l.

The k-dependence of the complexity of the test functions at various l are shown in Figs. 9–12. The logarithm of the number of partitions (k) is also included for comparison.

For nonrandom signals there is a clear convergence with increasing l. A very simple function like f_{01} does not exhibit any dependence on l which means that the correct dynamics is captured for the shortest possible left and right substrings, those of length 2.

The sine function displays an interesting feature: with increasing l the deviation from the converged value starts appearing at higher values of k. In other words it requires more information (the higher k the more information is transferred from the continuous signal to the symbolic sequence) to reach the true complexity as we increase the length of the time behavior that is analyzed.

The convergence for the sum of sine functions (not shown here f_{5s} graph is very similar to f_{2s}) is obvious even though the deviation from the logarithmic dependence become significant at high values of l. The nature of this is not clear, yet a possible reason is that a nonoptimal condition for comparison of right sequence distributions has been chosen. It is worth stressing that for nonrandom signals the logarithmic dependence on k comes from the natural limit of the Shannon entropy and not from the finiteness of the data stream [Eq. (3)].

The situation with the random signal is somewhat more complicated. The curves converge for low values of l. For lengths l higher then 5 the curves show high, divergent values of complexity that we attribute to the lack of statistics.
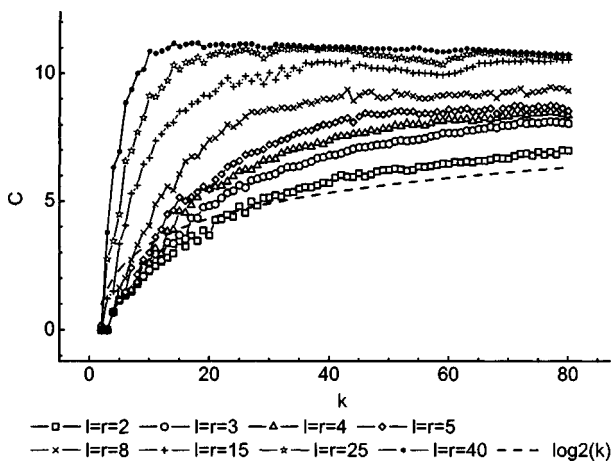

FIG. 11. Same as Fig. 9 but for f_{2s}.

TABLE I. Finite statistical complexities of the test functions.

Function	Finite statistical complexity
f_r	6.14
f_{01}	6.88
f_s	7.34
f_{2s}	10.88
f_{5s}	11.07

For small l, however, all the curves converge to approximately the same value which is significantly different from the other models.

Table I summarizes the data for the test function complexities at $k=60$. The converged value of l is taken for nonrandom functions and $l=5$ for the random one. The expected trend in complexity values of the signals with respect to each other is seen, i.e., the lowest is found for random values, then f_{01}, sine and sums of sines is observed.

The difference between f_{2s} and f_{5s} is not very big but it is clearly present. The small value of the difference is most probably caused by using criteria that are not sufficiently strict for the right side sequence distributions comparison.

V. CONCLUSIONS

Among the diversity of modern approaches for calculating dynamical complexity, computational mechanics by Crutchfield *et al.*[22] promises great opportunities in the investigation of emergent behavior in molecular systems. To apply them to real molecular trajectories, an algorithm for symbolization of a continuous trajectory has been developed. It is demonstrated that the algorithm reproduces the expected values of complexity for various test functions. The dependence on the number of partitioning intervals k of a real signal is investigated. It is shown that k-dependence has a logarithmic character as predicted by the theory. The approach allows us to apply it to a real molecular system. This is presented in the companion paper.[28] In future we plan to analyze in more detail the various algorithms for the comparison of right sequence distributions. This will most probably affect the resulting complexity, especially for systems with a high random component. It may also help to make the k-dependence of the complexity smoother.

Finally, it will be very interesting to reconstruct the ϵ-machine implicitly and study a hierarchy of the algorithmic languages, getting a deeper understanding of the emergent and information processing nature of physical trajectories. This direction of research is currently in progress in our group.

ACKNOWLEDGMENT

[1] M. Braxenthaler, R. Unger, D. Auerbach, J.A. Given, and J. Moult, Proteins: Struct., Funct., Genet. **29**, 417 (1997).

[2] Huai-bei Zhou, J. Phys. Chem. **100**, 8101 (1996).

[3] T. Rage, A. Neumaier, and C. Schlier, Phys. Rev. E **50**, 2682 (1994).

[4] R. Gaspard, M.E. Briggs, M.K. Francis, J.V. Sengers, R.W. Gammon, J.R. Dorfman, and R.V. Calabrese, Nature (London) **394**, 865 (1998).

[5] S.A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution* (Oxford University Press, New York, 1993).

[6] J.H. Holland, *Emergence: From Chaos to Order* (Oxford University Press, New York, 1998).

[7] R. Badii and A. Politi, *Complexity: Hierarchical structures and Scaling in Physics* (Cambridge University Press, Cambridge, 1999).

[8] C.E. Shannon, Bell Syst. Tech. J. **27**, 379 (1948).

[9] R. Baierlein, *Atoms and Information Theory: An Introduction to Statistical Mechanics* (Freeman, New York, 1971).

[10] H.S. Robertson, *Statistical Thermophysics* (Prentice–Hall, Englewood Cliffs, 1993).

[11] J.P. Crutchfield and N.H. Packard, in *Evolution of Order and Chaos*, edited by H. Haken (Springer-Verlag, Berlin, 1982), pp. 215–227.

[12] P. Grassberger, Int. J. Theor. Phys. **25**, 907 (1986).

[13] S. Pincus, Chaos **5**, 110 (1995).

[14] P.-M. Binder, Phys. Rev. E **61**, R3303 (2000).

[15] J.S. Shiner, M. Davison, and P.T. Landsberg, Phys. Rev. E **59**, 1459 (1999).

[16] P.-M. Binder and N. Perry, Phys. Rev. E **62**, 2998 (2000).

[17] J.P. Crutchfield, D.P. Feldman, and C.R. Shalizi, Phys. Rev. E **62**, 2996 (2000).

[18] P. Lakdawala, Phys. Rev. E **53**, 4477 (1996).

[19] M. Mitchell, J.P. Crutchfield, and P.T. Hraber, in *Complexity: Metaphors, Models, and Reality, SFI Series in the Sciences of Complexity*, edited by G. Cowan, D. Pines, and D. Melzner (Addison–Wesley, Reading, 1994), Vol. XIX, pp. 497–513.

[20] J.P. Crutchfield and K. Young, Phys. Rev. Lett. **63**, 105 (1989).

[21] J.P. Crutchfield and K. Young, in *Entropy, Complexity, and Physics of Information, SFI Studies in the Sciences of Complexity*, edited by W. Zurek (Addison–Wesley, Reading, (1990)), Vol. VIII, pp. 223–269.

[22] J.P. Crutchfield, Physica D **75**, 11 (1994).

[23] David Feldman, A brief introduction to: Information theory, excess entropy and computational mechanics (http://hornacek.coa.edu/dave/Tutorial/index.html) (unpublished)

[24] J.E. Hanson and J.P. Crutchfield, J. Stat. Phys. **66**, 1415 (1992).

[25] J.E. Hanson and J.P. Crutchfield, Physica D **103**, 169 (1997).

[26] J.P. Crutchfield and D.P. Feldman, Phys. Rev. E **55**, R1239 (1997).

[27] N. Perry and P.-M. Binder, Phys. Rev. E **60**, 459 (1999).

[28] D. Nerukh, G. Karvounis, and R. C. Glen, J. Chem. Phys. **117**, 9618 (2002), following paper.

[29] C.S. Daw and C.E.A. Finney, Rev. Sci. Instrum. (submitted).

[30] V.M. Alekseev and M.V. Yakobson, Phys. Rep. **75**, 287 (1981).

[31] P. Grassberger and H. Kantz, Phys. Lett. **113A**, 235 (1985).

[32] P.E. Rapp, C.J. Cellucci, K.E. Korslund, T.A.A. Watanabe, and M.A. Jimenez-Montano, Phys. Rev. E **64**, 016209 (2001).

[33] E. Bollt, T. Stanford, Y.-C. Lai, and K. Zyczkowski, Physica D **154**, 259 (2001).

[34] T.M. Cover and J.A. Thomas, *Elements of Information Theory* (Wiley, New York, 1991).