Effective normalization of complexity measurements for epoch length and sampling frequency

P. E. Rapp*
Department of Pharmacology and Physiology, Medical College of Pennsylvania Hahnemann University, Philadelphia, Pennsylvania 19129 and The Arthur P. Noyes Clinical Research Foundation, Norristown, Pennsylvania

C. J. Cellucci
Department of Physics, Ursinus College, Collegeville, Pennsylvania 19426 and The Arthur P. Noyes Clinical Research Foundation, Norristown, Pennsylvania

K. E. Korslund
Department of Psychiatry and Behavioral Sciences, University of Washington, Seattle, Washington

T. A. A. Watanabe
Department of Pharmacology and Physiology, Medical College of Pennsylvania Hahnemann University, Philadelphia, Pennsylvania 19129 and The Arthur P. Noyes Clinical Research Foundation, Norristown, Pennsylvania

M. A. Jiménez-Montaño
Department of Physics and Mathematics, Universidad de las Américas, Puebla, Mexico
(Received 22 November 2000; revised manuscript received 21 March 2001; published 14 June 2001)

The algorithmic complexity of a symbol sequence is sensitive to the length of the message. Additionally, in those cases where the sequence is constructed by the symbolic reduction of an experimentally observed wave form, the corresponding value of algorithmic complexity is also sensitive to the sampling frequency. In this contribution, we present definitions of algorithmic redundancy that are sequence-sensitive generalizations of Shannon's original definition of information redundancy. In contrast with algorithmic complexity, we demonstrate that algorithmic redundancy is not sensitive to message length or to observation scale (sampling frequency) when stationary systems are examined.

DOI: 10.1103/PhysRevE.64.016209 PACS number(s): 05.45.−a

I. INTRODUCTION

While there are many different kinds of complexity measures [1–4], they all share the common property of providing a quantitative assessment of the structure of a symbol sequence. Complexity measures complement an examination of the symbol distribution by identifying sequence-sensitive patterns that would be destroyed by a random shuffle. In a previous publication [5], we have constructed an eight celled taxonomic classification of complexity measures based on a sequence of three dichotomous classifiers: (i) randomness finding vs rule finding, (ii) probabilistic vs non-probabilistic, and (iii) formula based vs model based. Randomness finding measures of complexity give the highest value of complexity to random sequences. The original definitions of symbolic complexity introduced by Kolmogorov [6] and Chaitin [7] were of this type. An alternative assessment of complexity is one that attempts to establish a quantitative characterization of the rules used to generate a symbol sequence. Examples of rule finding measures of complexity include forbidden word complexity [8–10], effective measure complexity [3], and ε-machines [11].

Probabilistic measures of complexity, for example, metric entropy [12] and effective measure complexity [3], are sensitive to both the number of distinct subgroups (words) in the message and the frequency of their appearance. In our nomenclature we describe a complexity measure as being non-probabilistic if it depends only on the number of distinct words and is insensitive to the frequency of their appearance. Examples include topological entropy [13] and the previously cited forbidden word complexities. Formula-based measures of complexity are those like topological entropy, metric entropy, and effective measure complexity that can be expressed in an equation. An alternative is one that constructs a model of the symbol sequence, for example, a series of instructions that permit the reconstruction of the original symbol sequence, and assigns a value of complexity based on the size of the model. Examples of model-based complexity include Kolmogorov-Chaitin complexity [6,7] and the Lempel-Ziv complexity [14].

This contribution focuses on a specific class of complexity measure, the algorithmic complexity that in this classification system would be an example of a non-probabilistic, model based, randomness finding measure of complexity. Two representative examples are considered. The first is the Lempel-Ziv measure of complexity [14]. The second, the context free grammar complexity, has been described elsewhere [15,16]. In both cases an upper bound of the complexity of a symbol sequence is found by first constructing an instruction sequence that can reproduce the message. The complexity is based on a measure of the length of that in-

*Corresponding author. Clinical Research Center, Building 52, Norristown State Hospital, 1001 Sterigere St., Norristown, PA 19401-5397. Email address: Paul.E.Rapp@Drexel.edu

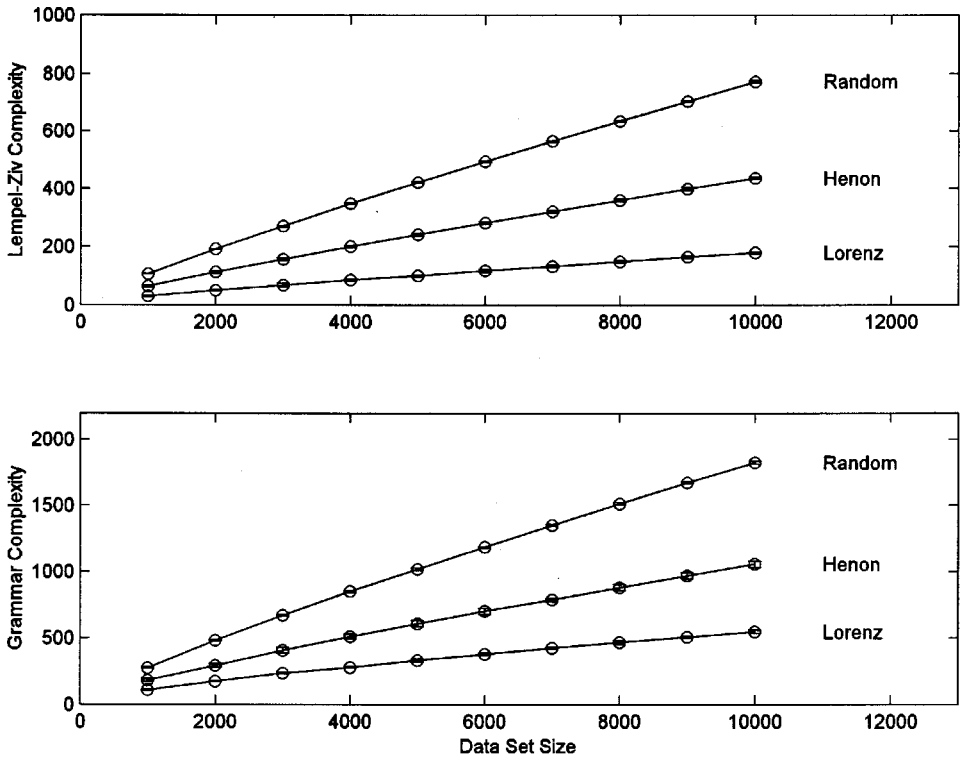

FIG. 1. Algorithmic complexity as a function of data set size for three model systems. Average values were calculated from five independent determinations obtained with different initial conditions. The corresponding symbol sequences were binary partitions about the median. System specifications are given in the text. The upper box shows the Lempel-Ziv complexity. The lower box displays the context free grammar complexity.

struction set. A brief didactic presentation of each definition is given in the appendixes.

Viewed in isolation, a single measure of algorithmic complexity provides limited information. A normalization that could facilitate comparisons between different measurements would be helpful. The process of constructing an effective normalization of complexity turns on two problems. First, as will be shown presently, the value of algorithmic complexity increases with the length of the message. An improved measure should be invariant with respect to message length provided that the underlying dynamical process was stationary throughout the observation period. Second, the complexity of symbol sequences constructed from observations of continuous dynamical systems is sensitive to sampling frequency. This problem should also be addressed by a successful renormalization.

We are, therefore, seeking a measure that would retain the essential features of algorithmic complexity in providing a sequence-sensitive measure of randomness, but would not depend on how long or how often observations are made. Finally, a means of assessing the uncertainty of the measurement would be of particular value in the analysis of experimental data.

II. SENSITIVITY TO EPOCH LENGTH

Figure 1 shows the values of complexity obtained from three model systems. In the first example, random numbers uniformly distributed on (0,1) were calculated with a Park-Miller random number generator that incorporated a Bays-Durham shuffle [17–19]. The resulting sequence of real numbers is reduced to a binary symbol sequence by partitioning the random numbers about the median. A symbol ''0'' is assigned if the number is less than or equal to the median value, and symbol ''1'' is assigned if the number is greater than the median. In this example, the median and the mean are essentially identical. However, it should be noted that in those instances where they are significantly different, a partition about the median should be used rather than a partition about the mean [20].

The Hénon data sets used to generate the results in Fig. 1 were generated from the Hénon system using different initial conditions and the parameters $a=1.4$ and $b=0.3$,

$$x_{t+1}=1-ax_t^2+y_t,$$

$$y_{t+1}=bx_t.$$

As in the previous case, a binary symbol sequence was generated about the median.

The Lorenz data sets were generated from the ordinary differential equation

$$dx/dt=10(x-y),$$

$$dy/dt=x(28-z)-y,$$

$$dz/dt=xy-(8/3)z.$$

The system was integrated with a sixth order Runge-Kutta-Huta algorithm [21] with a step length of $\Delta t=0.01$.

The values of complexity displayed in Fig. 1 were obtained by averaging five determinations generated from different initial conditions. The mean values are displayed with the corresponding standard deviations, which were typically on the order of 2% of the value of complexity. For both the Lempel-Ziv complexity and the context free grammar com-

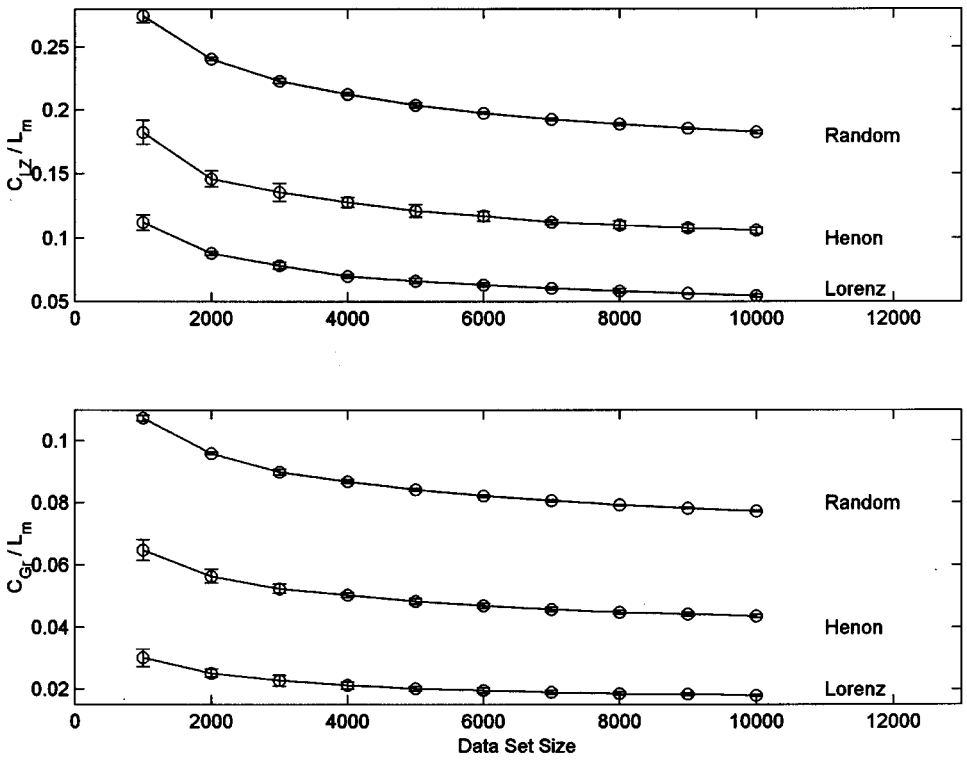

FIG. 2. Algorithmic complexity divided by L_m, the number of symbols in the message, for the model systems of Fig. 1. The upper box shows results obtained with the Lempel-Ziv definition of complexity. The lower box displays results obtained with the context free grammar complexity. Average values were calculated from five independent determinations obtained with different initial conditions.

plexity, algorithmic complexity is seen to increase monotonically with data set size. The two measures were found to be highly correlated. The Pearson linear correlation coefficient is $R=0.998$. The Spearman's rank correlation is $r_s=0.994$ and the Kendall nonparametric correlation coefficient is $\tau=0.963$.

The simplest possible normalization for data set size would be obtained by dividing the value of complexity by the length of the message L_m. The results in Fig. 2 show that this does not result in an L_m-independent measure.

An alternative can be constructed by generalizing the definition of redundancy introduced by Shannon [22]. Suppose a message of L_m symbols is constructed from an alphabet of N_a characters where p_i is the probability of the ith character. The entropy of the message, H_m, is defined as

$$H_m=-\sum_{i=1}^{N_a} p_i \log_2 p_i .$$

The maximum entropy is obtained when the sequence of symbols is equiprobable, that is $p_i=1/N_a$ for all i. $H_{\max}$, the maximum possible value of entropy, is given by $H_{\max}=\log_2 N_a$

Shannon defines informational redundancy R_S as

$$R_S=1-H_m/H_{\max}.$$

If $p_i=1/N_a$ for all i, then $H_m=H_{\max}$, and $R_S=0$; that is, when the sequence is equiprobable, the redundancy of the message is zero. The observation of each symbol in the message is informative. Alternatively, suppose $p_j=1$ for some j and that $p_i=0$ for all $i\neq j$. In this case, $H_m=0$; no information is obtained by observing the next symbol in the sequence since it is always the jth symbol, and $R_S=1$.

A generalization of the idea of redundancy to sequence-sensitive measures of complexity is implicit in the content of Kolmogorov's 1965 paper [6]. A sequence-sensitive redundancy R_K can be defined by:

$$R_K=1-C_m/L_m,$$

where C_m is the algorithmic complexity of the message. Chavoya-Aceves, Garcia de LeBarrera, and Jiménez-Montaño [23] defined a redundancy for complexity measurements as

$$R=1-C_m/C'_m,$$

where

$$C'_m=\min\{L_m,[\tfrac{1}{2}(L_m+1)+2N_a^2]\}.$$

However, when L_m is much greater than N_a, as is the case in the examples considered in this paper, $C'_m\approx L_m/2$, and the Chavoya-Aceves definition is essentially R_K. R_K as a function of data set size is shown in Fig. 3 for both Lempel-Ziv complexity and the context free grammar complexity. As expected given the results presented in Fig. 2, R_K changes with data set size. It is also important to note that R_K is not equal to zero for random numbers, which is contrary to the spirit of the definition of redundancy.

For the specific case of the Lempel-Ziv complexity measure, an additional possible normalization should be considered. Lempel and Ziv [14] show that their definition of complexity, denoted C_{LZ}, is bounded by N_1,

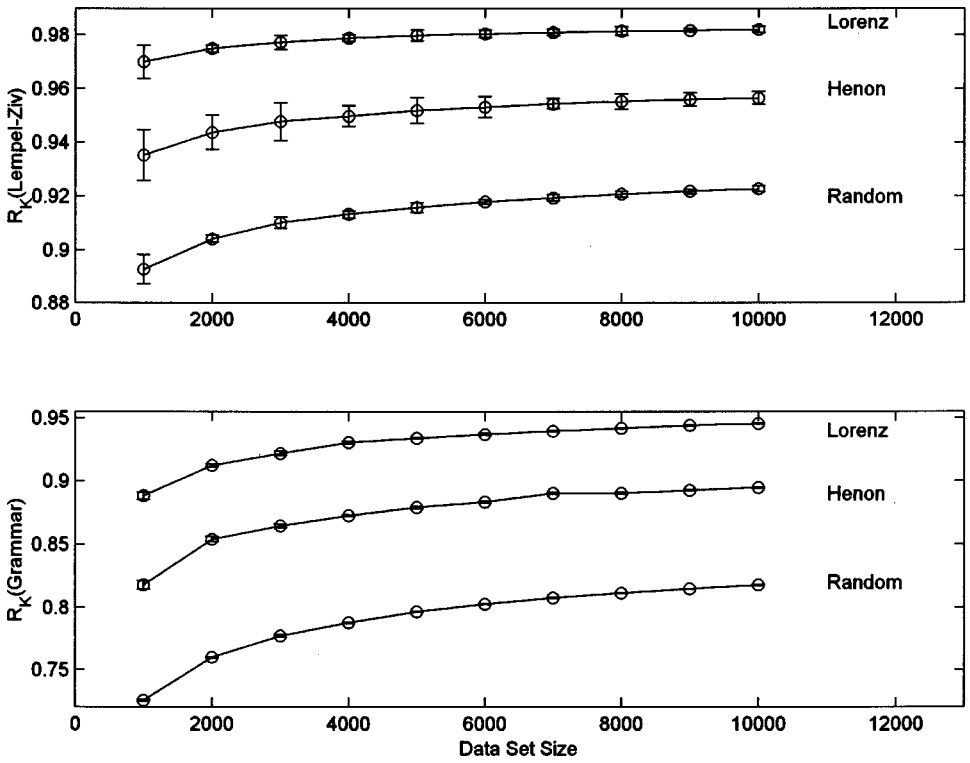

FIG. 3. R_K, redundancy constructed by normalizing against data set length, for the model systems of Fig. 1. Note that the redundancy obtained with random sequences is not zero. The upper box shows results obtained with the Lempel-Ziv definition of complexity. The lower box displays results obtained with the context free grammar complexity. Average values were calculated from five independent determinations obtained with different initial conditions.

$$C_{LZ} < N_1 = \frac{N_{\mathrm{Data}}}{(1-\varepsilon_N)\log_\alpha N_{\mathrm{Data}}},$$

where to clarify notation we set $\alpha = N_a$, the number of symbols in the alphabet.

$$\varepsilon_N = 2\{1+\log_\alpha \log_\alpha(\alpha N_{\mathrm{Data}})\}/\log_\alpha N_{\mathrm{Data}}$$

If N_{Data} is large, ε_N is small and a simplified normalization against $N_{\mathrm{Data}}/\log_\alpha N_{\mathrm{Data}}$ can be considered, as was done in Zhang *et al.* [24]. C_{LZ}/N_1 is shown in Fig. 4 and is seen to be sensitive to the value of N_{Data}.

A more successful definition of redundancy defined by analogy with Shannon's R_S is

$$R_0 = 1 - C_m/\langle C_0 \rangle.$$

As before, C_m is the complexity of the original message. C_0 is the complexity of a randomly shuffled equiprobable symbol sequence of the same length where $p_i = 1/N_a$. The subscript ''0'' is used to indicate that this is the complexity of a random shuffle surrogate data set. As defined here, surrogates are not constructed by simply shuffling the original symbol sequence. The distribution of the original sequence is not necessarily $p_i = 1/N_a$. $\langle C_0 \rangle$ denotes the mean value of complexity found by averaging values from several independently constructed surrogates. The number of surrogates that should be used to estimate R_0 depends on the signal-to-noise ratio of the original data. This question can be addressed empirically by increasing the number of surrogates until a stable value of R_0 is obtained. Test calculations indicated that for the data sets examined in this paper, the values of

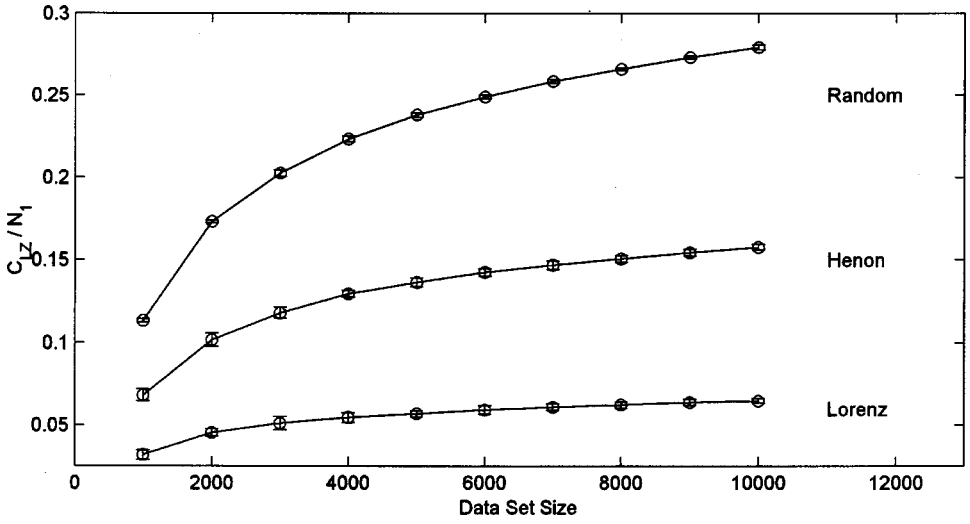

FIG. 4. Lempel-Ziv complexity normalized against the upper bound N_1 presented in the text as a function of data set size for the model systems of Fig. 1. Average values were calculated from five independent determinations obtained with different initial conditions.

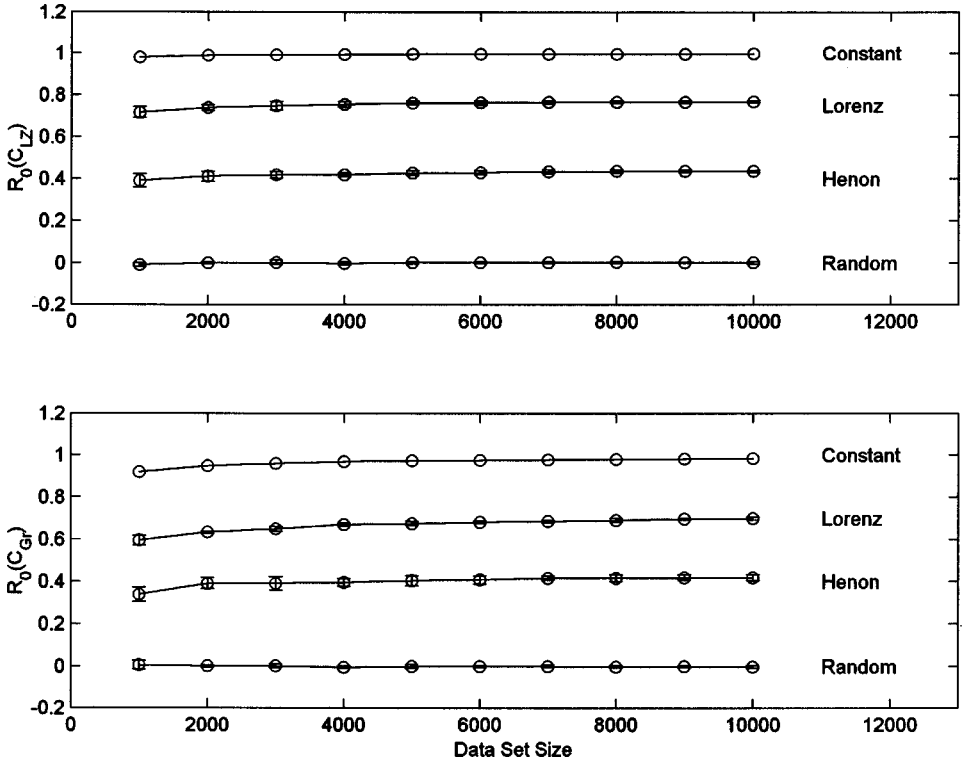

FIG. 5. R_0, the redundancy obtained by normalizing against random equiprobable binary symbol sequences. R_0 is shown for the three model systems of the preceding diagrams, and for a constant sequence constructed by repeating a single symbol. The upper box shows results obtained with the Lempel-Ziv definition of complexity. The lower box displays results obtained with the context free grammar complexity.

redundancy obtained with ten surrogates were within 1% of the values obtained with 50 surrogates. The values presented here were obtained with ten surrogates. Measures of complexity such as algorithmic complexity give the highest values to random sequences. $\langle C_0 \rangle$ is, therefore, an empirical estimate of $C_{\max}$ and R_0 is a sequence-sensitive generalization of R_S.

It is possible to estimate the uncertainty of R_0 as below:

$$\Delta R_0^2 = \left(\frac{\partial R_0}{\partial C_m} \right)^2 \Delta C_m^2 + \left(\frac{\partial R_0}{\partial \langle C_0 \rangle} \right)^2 \Delta C_0^2 = \frac{\Delta C_m^2}{\langle C_0 \rangle^2} + \frac{C_m^2}{\langle C_0 \rangle^4} \Delta C_0^2,$$

where ΔC_0 is the standard deviation of averaged surrogate complexities found when calculating $\langle C_0 \rangle$. In Fig. 5, ΔC_m can be estimated empirically by multiple determinations of C_m from different initial conditions. In some instances, for example, the examination of experimental data, multiple determinations may not be available. In those cases, ΔC_m can be approximated by determining C_A, the complexity of the first half of the message, and C_B, the complexity of the second half of the message. An approximation of ΔC_m is given by

$$\Delta C_m = \frac{|C_A - C_B|}{(|C_A| + |C_B|)/2} C_m,$$

where, as before, C_m is the complexity of the original message.

R_0 as a function of data set size is shown for the three model systems in Fig. 5. The results obtained with symbol sequences consisting of a single repeated symbol are also shown and labeled as ''Constant.'' It is seen that R_0 is largely unchanged as the size of the data set is increased. Thus, the first of our objectives has been met. The definition of R_0 also produces results consistent with our intuitive sense of the word redundancy. Random sequences have zero redundancy. Every observation of a random sequence provides information. In contrast, the redundancy of a constant sequence is 1. A constant symbol sequence is completely redundant. Lorenz and Hénon show intermediate values with the Hénon system having a lower redundancy; that is, it appears more random. This is also consistent with our expectations since the Hénon system is a map that rapidly distributes itself over its attractor. Each point is typically well displaced from its predecessor. In contrast, the Lorenz system is a flow; displacement on the attractor with each successive observation is smaller.

III. SENSITIVITY TO SAMPLING INTERVAL

The calculations presented in the previous section indicate that R_0 is essentially constant for stationary systems as the number of data points (the observed epoch) is increased. However, during the examination of continuous wave forms, for example, computed solutions of the Lorenz equations or experimental measurements of fluid flow, there is another issue. The investigator faces an operational question: how often should I sample the signal? While R_0 provides an effective normalization for epoch length (given assumptions of stationarity), it is sensitive to the sampling frequency. This is shown in Figs. 6 and 7. The Lorenz system was integrated at different sampling intervals ($\Delta t = 0.01$, 0.02, 0.04, and 0.08). The corresponding number of data points in each data set is 8192, 4096, 2048, and 1024. Thus the epoch length is the same in all four cases. When compared across constant epochs, complexity is seen to decrease with Δt (Figs. 6 and 7,

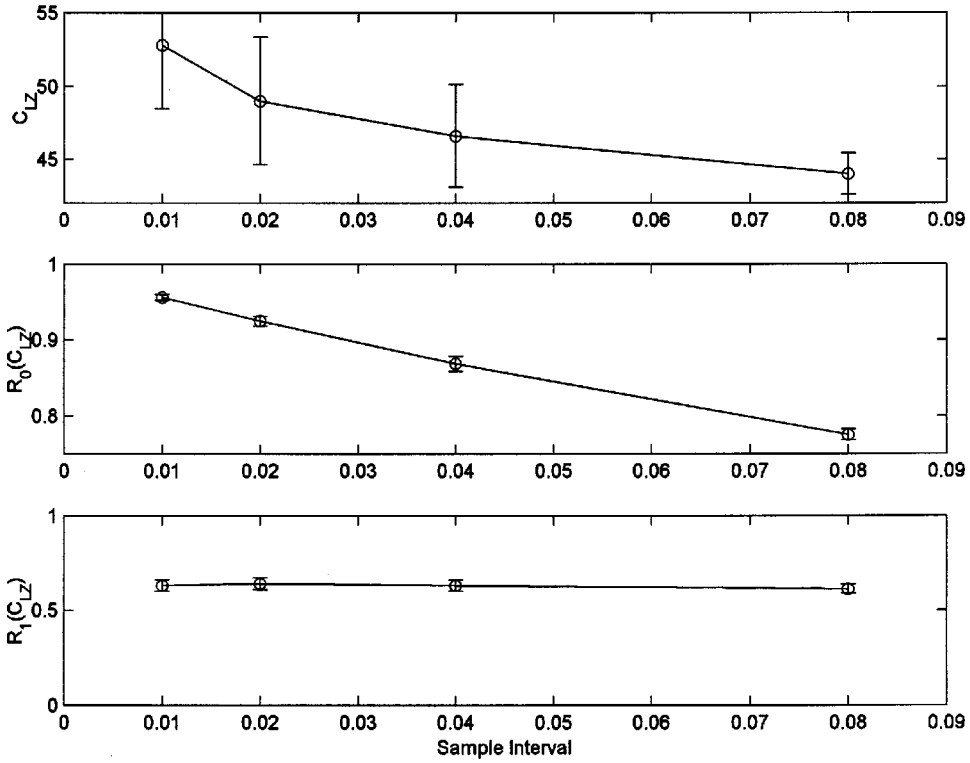

FIG. 6. An examination of the Lorenz system for constant-epoch time series (100 time units). The sampling interval, and hence the size of the data sets, changes with each case. In the top panel the Lempel-Ziv complexity is displayed. The middle panel shows R_0, the redundancy calculated with equiprobable randomly shuffled surrogates using the Lempel-Ziv definition of complexity. The third panel displays R_1, the redundancy calculated using random phase surrogates and Lempel-Ziv complexity. The averages of five determinations obtained with different initial conditions are shown.

top panels). This might suggest that the dynamical behavior of the observed data is less disordered for larger value of Δt. But in fact, decreasing complexity is a result of decreasing number of data points. The second panel of Figs. 6 and 7 shows R_0 as a function of Δt. Redundancy decreases as Δt increases; that is, the observed symbol sequence becomes more random as Δt increases. Upon reflection, this is as it should be. As Δt increases, the chaotic Lorenz system decorrelates and the relationship between successive observations becomes more disordered.

Normalization for both sampling interval and data set size can be constructed by using surrogates of length L_m (the length of the original message) that reflect the time scale of the observational process. The easiest way to do this is with

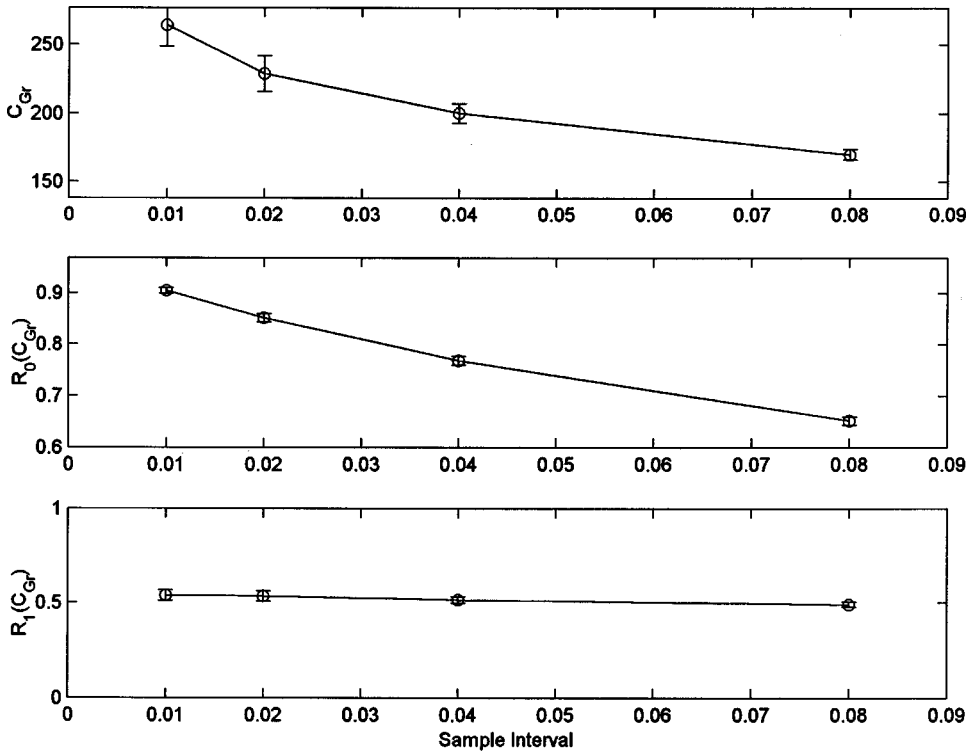

FIG. 7. An examination of the Lorenz system for constant-epoch time series (100 time units). The sampling interval, and hence the size of the data sets, changes with each case. In the top panel the context free grammar complexity is displayed. The middle panel shows R_0, the redundancy calculated with equiprobable randomly shuffled surrogates using grammar complexity. The third panel displays R_1, the redundancy calculated using random phase surrogates and grammar complexity. The averages of five determinations obtained with different initial conditions are shown.

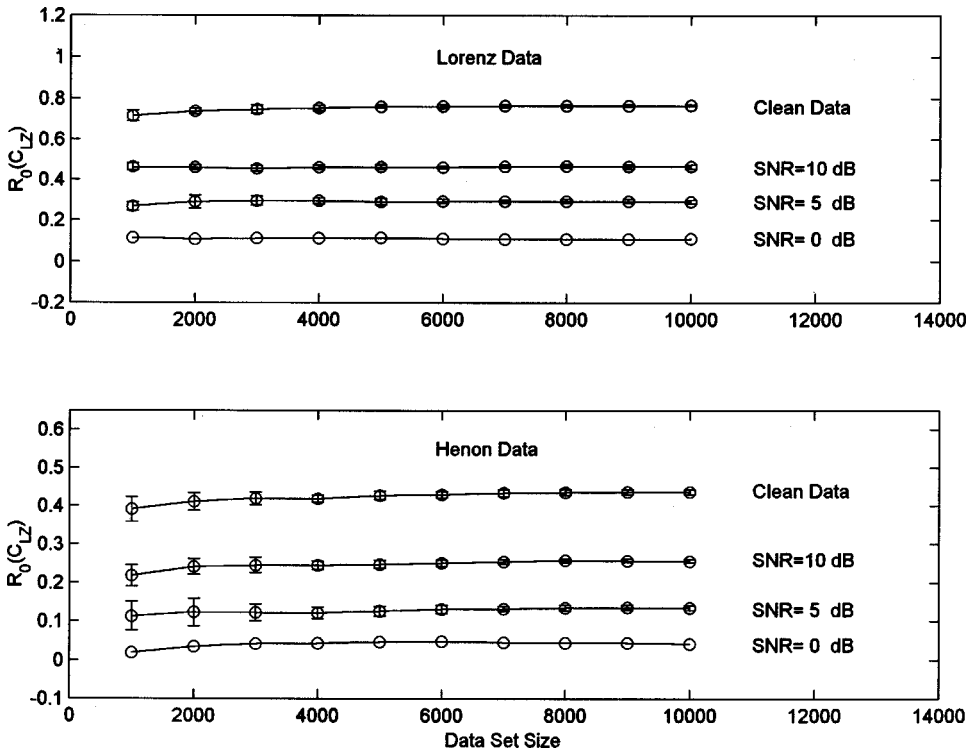

FIG. 8. R_0, the redundancy obtained by normalizing against random equiprobable binary symbol sequences and the Lempel-Ziv complexity measure. R_0 is shown for two of the model systems of the preceding diagrams. Normally distributed Gaussian noise of zero mean was added to reduce the signal-to-noise ratio.

random phase surrogates, which are also referred to as Algorithm 1 surrogates [25,26]. Random phase surrogates are constructed by calculating the Fourier transform of the original time series, randomizing the phases and calculating the inverse transform. The inverse transform is the surrogate. The surrogates and the original time series have the same power spectra, and thus by the Wiener-Khinchin theorem, they have the same autocorrelation function.

The complexity of random phase surrogates is calculated in the same manner as the complexity of the original time series. The surrogate is partitioned into a symbol sequence using an alphabet of N_a characters. The partition is about the median for the case $N_a=2$. An equiprobable generalized median partition ($p_i=1/N_a$, for all i) is used if $N_a>2$. Let $\langle C_1\rangle$ denote the mean value of complexity obtained from Algorithm 1 (random phase) surrogates. The random phase redundancy is defined as

$$R_1=1-C_m/\langle C_1\rangle.$$

The value of complexity obtained with a random phase surrogate is the maximum value of complexity compatible with the original signal's spectrum. The third panel of Figs. 6 and 7 shows R_1 as a function of sample interval. It is insensitive to changes in both data set size and sampling frequency.

Recall that algorithmic complexity is a measure that gives the greatest value of complexity to random sequences. Therefore, in general $\langle C_0\rangle$ is greater than $\langle C_1\rangle$, and it follows that $R_0>R_1$. This is confirmed for the specific example of the Lorenz system by comparing the second and third panels of Figs. 6 and 7. The introduction of R_1 indicates that there is no absolute definition of algorithmic redundancy. Redundancy can only be defined with respect to the underlying null hypothesis of the surrogates. In the case of R_0, the null hypothesis holds that the original signal is indistinguishable from noise. In the case of R_1, the null hypothesis holds that the original signal is indistinguishable from linearly filtered noise [25,26].

IV. SENSITIVITY TO NOISE

As indicated in the Introduction, both of the complexity measures used in these computations are randomness-seeking measures. That is, they give the highest values of complexity to randomly constructed sequences. Therefore the complexity is expected to increase (the redundancy should decrease) as the stochastic component of a signal is increased.

This expectation is verified in the calculations presented in Fig. 8. Two of the three model systems considered in the previous sections, the Lorenz equations and the Hénon difference equations, were used. Normally distributed noise generated by the previously described Park-Miller random number generator of zero mean was added to the original signals. The variance of the added noise was adjusted to give signal-to-noise ratios of 10, 5, and 0 dB. Ten realizations using different seeds to the random number generator were produced for each case. The figure displays the average of these ten determinations. The redundancy decreases as the signal-to-noise ratio decreases. The invariance of redundancy with respect to data set length, however, is not affected by noise.

V. CONCLUSIONS

The results suggest that for stationary systems, appropriately defined versions of algorithmic redundancy are robust to changes in message length and sampling frequency. Re-

dundancy may therefore be a useful measure of the intrinsic structure of a symbol sequence that facilitates comparisons between cases. This is particularly important for longitudinal studies in which the object is to assess long term changes in dynamical systems.

ACKNOWLEDGMENTS

We would like to acknowledge support from the U.S. Department of Education Contract No. H235J000001 to the Krasnow Institute, from the Janssen Pharmaceutica Research Foundation, and from the Bristol-Myers Squibb Pharmaceutical Research Institute. We would also like to acknowledge the encouragement and leadership of R. C. Josiassen, Director of the Arthur P. Noyes Research Foundation at Norristown State Hospital. MAJ-M thanks CONACYT, Mexico (Project: 32201-E) for partial support.

APPENDIX A: THE LEMPEL-ZIV COMPLEXITY

According to Kolmogorov, the algorithmic complexity of a sequence of symbols is given by the number of bits of the shortest computer program that can generate that sequence [6]. However, a general algorithm that determines such a program cannot be given [27,7]. Instead, Lempel and Ziv developed a complexity measure that does not necessarily calculate the length n of the shortest program that generates a given symbol sequence, but rather a number $c(n)$, which is a useful upper bound of this length [14,28,29].

The Lempel-Ziv approach associates the complexity of a symbol sequence with the sequential appearance of new patterns within that sequence. They present an estimate of the complexity of a finite sequence from the point of view of a learning machine that, as it scans a n-digit sequence $S=s_1s_2\cdots s_n$ from left to right, adds a new word to its memory every time it encounters a substring not previously observed. The size of the compiled vocabulary, and the rate at which new words are encountered, serve as the basis for their complexity measure.

Let us introduce some definitions that are needed for this discussion: A denotes the alphabet set (i.e., the symbols that are used to compose the sequence); S denotes a finite length symbol sequence formed by A, whose complexity is to be measured; $S(i,j)$ indicates a substring of S that starts at position i and ends at position j, that is, when $i \leqslant j$, $S(i,j)=s_i,s_{i+1},\cdots s_j$ and when $i>j$, $S(i,j)=\{\}$, the null set; $V(S)$ the vocabulary of a sequence S. It is the set of all substrings, or *words*, $S(i,j)$ of S, (i.e., $S(i,j)$ for $i=1,2,\ldots n$; $j \geqslant i$). For example, Let $A=\{0,1\}$, and $S=001$, we then have

$$V(S)=\{0,1,00,01,001\}.$$

The Lempel-Ziv algorithm is essentially the parsing of the original sequence S into

$$H(S)=S(1,h_1)\oplus S(h_1+1,h_2)\oplus S(h_2+1,h_3)\cdots\oplus S(h_{m-1}+1,h_m),$$

which is called the production history of S. The symbol $\oplus$ denotes the concatenation operator. The m words $H_i(S)=S(h_{i-1}+1,h_i)$ $i=1,2,\ldots,m$, are called the components of S. The complexity $c(n)$ is the positive integer equal to the number of substrings, or components, required by this process. Understanding the procedure used by Lempel-Ziv to produce a unique production history constitutes an understanding of their algorithm.

As was stated earlier, this process follows a left-to-right scan of a sequence S. A substring $S(i,j)$ is compared to the vocabulary that is comprised of all substrings of S up to $j-1$, that is, $V(S(1,j-1))$. If the substring is present in $V(S(1,j-1))$, then $S(i,j)\rightarrow S(i,j+1)$, and $V(S(1,j-1))\rightarrow V(S(1,j))$, and the process repeats. If the substring is not present, then a dot is placed after $S(j)$ to indicate the end of a new component $S(i,j)\rightarrow S(j+1,j+1)$ that is the single symbol in the $j+1$ position, and $V(S(1,j-1))\rightarrow V(S(1,j))$, and the process continues. This parsing operation begins with $S(1,1)$ and continues until $j=n$, where n is the length of the symbol sequence.

Consider the following sequence of zeros and ones:

$$S=0100011010010011101100.$$

We begin with $S(1,1)=0$, and $V(S(1,0))=\{\}$. Since the substring $S(1,1)$ is not found in the vocabulary $V(S(1,0))$, we place a dot after the first element, $S(1,1)\rightarrow S(2,2)=1$, and $V(S(1,0))\rightarrow V(S(1,1))=\{0\}$. After this first step our sequence becomes

$$S=0\cdot 100011010010011101100.$$

Now $S(2,2)=1$, which is not found in the vocabulary $V(V(1,1))=\{0\}$, so we place a dot after the second element, $S(2,2)\rightarrow S(3,3)$, and $V(S(1,1))\rightarrow V(S(1,2))=\{0,1,01\}$. After this second step our sequence becomes

$$S=0\cdot 1\cdot 00011010011101100.$$

$S(3,3)=0$, is found in $V(S(1,2))$, therefore $S(3,3)\rightarrow S(3,4)$ and $V(S(1,2))\rightarrow V(S(1,3))=\{0,1,01,10,010\}$. $S(3,4)$ is not found in the vocabulary so we place a dot after the fourth element, $S(3,4)\rightarrow S(5,5)$ and $V(S(1,3))\rightarrow V(S(1,4))$. At the completion of our parsing process we get

$$S=0\cdot 1\cdot 00\cdot 011\cdot 01001\cdot 00111\cdot 01100\cdot.$$

(A dot is always placed after the last element in the symbol sequence.) The Lempel-Ziv complexity in this example would be $c(n)=7$.

APPENDIX B: THE CONTEXT FREE GRAMMAR COMPLEXITY

This measure was constructed by Ebeling and Jiménez-Montaño [15]. The description given here follows the presentation in Rapp *et al.* [30] The definition is described most effectively by considering a specific application. In this example a binary symbol alphabet is used, but the definition is applicable to alphabets of arbitrary size. Consider symbol sequence M

$$M=0100011010010011101100.$$

The sequence is searched for repeated symbol pairs. The pair 0 1 is repeated six times. A new symbol $a=01$, is defined and substituted into message M resulting in its compression,

$$a=01,$$

$$M=a00a1a0a0a11a100.$$

M is again scanned for repeated pairs. The symbol pair a 0 is repeated three times and is replaced by symbol b,

$$a=01,$$

$$b=a0,$$

$$M=b0a1bba11a110.$$

The symbol pair $a1$ is repeated three times in the restatement of M. It is replaced by symbol c,

$$a=01,$$

$$b=a0,$$

$$c=a1,$$

$$M=b0cbbc1c10.$$

The search for repeated pairs has been exhausted. In the general case, the symbol sequence would be searched for repeated triples, which would be replaced by new symbols. Repeated four-letter elements would then be replaced, and so on. In the case of the present example, there are no higher order repeats. The compression has converged.

Using the definition of a, b, and c and the restatement of M, it is possible to reconstruct the original sequence exactly. In this example, the complexity of the original message is equal to the number of elements in its restatement. Symbols a, b, and c each consist of two symbols. Message M consists of ten symbols. Thus, using this definition of complexity,

$$(\text{Complexity of } M)=2+2+2+10=16.$$

Ebeling and Jiménez-Montaño [15] consider generalizations to include messages in which a single symbol is repeated three or more times in a row. When this occurs, the sequence of repeated symbols is replaced by its exponential representation. For example, *a a a a* becomes a^4. Under their definition, the exponent contributes to complexity logarithmically. The sequence a^4 would contribute one bit in recognition of symbol a and $\log_2 4$ in recognition of the exponent 4. Additional didactic examples are given in the earlier literature [30,31].

[1] M. Li, and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications* (Springer-Verlag, New York, 1993).

[2] *Kolmogorov Complexity and Computational Complexity*, edited by O. Watanabe (Springer-Verlag, Berlin, 1992).

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

[4] P. E. Rapp and T. I. Schmah, Mol. Psychiatry **1**, 408 (1996).

[5] P. E. Rapp and T. I. Schmah, in *Chaos in Brain?*, edited by K. Lenhertz, J. Arnhold, P. Grassberger, and C. E. Elger (World Scientific, Singapore, 2000).

[6] A. N. Kolmogorov, Probl. Inf. Transm. **1**, 1 (1965).

[7] G. J. Chaitin, J. Assoc. Comput. Mach. **13**, 547 (1966).

[8] J. Xu, Z.-R. Liu, and R. Liu, Chaos, Solitons Fractals **4**, 2111 (1994).

[9] G. D'Alessandro and A. Politi, Phys. Rev. Lett. **64**, 1609 (1990).

[10] S. Wolfram, Nature (London) **311**, 419 (1984).

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

[12] T. Schürmann and P. Grassberger, Chaos **6**, 414 (1996).

[13] J. P. Crutchfield and N. H. Packard, in *Evolution of Order and Chaos*, edited by H. Haken (Springer-Verlag, Berlin, 1983).

[14] A. Lempel and J. Ziv, IEEE Trans. Inf. Theory **IT22**, 75 (1976).

[15] W. Ebeling and M. A. Jiménez-Montaño, Math. Biosci. **52**, 53 (1980).

[16] P. E. Rapp, I. D. Zimmerman, E. P. Vining, N. Cohen, A. M. Albano, and M. A. Jiménez-Montaño, J. Neurosci. **14**, 4731 (1994).

[17] S. K. Park, and K. W. Miller, Commun. ACM **31**, 1192 (1988).

[18] D. E. Knuth, *The Art of Computer Programming* (Addison-Wesley, Reading, MA, 1981), Vol. 2, Secs. 3.2, 3.3.

[19] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes. The Art of Scientific Computing*. (Cambridge University Press, Cambridge, England, 1986).

[20] P. E. Rapp, A. M. Albano, I. D. Zimmerman, and M. A. Jiménez-Montaño, Phys. Lett. **192A**, 27 (1994).

[21] J. D. Lambert, *Computational Methods in Ordinary Differential Equations* (Wiley, New York, 1973).

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

[23] O. Chavoya-Aceves, F. Garcia de LaBarrera, and M. A. Jiménez-Montaño, *IX Reunion Nacional de Intelligencia Artificial* (SMIA Grupo Noriega Editores, Veracruz, 1992), pp. 243–254.

[24] X.-S. Zhang, Y.-S. Zhu, N. V. Thakor, and Z.-Z. Wang, IEEE Trans. Biomed. Eng. **46**, 548 (1999).

[25] J. Theiler, S. Eubanks, A. Longtin, B. Galdrikian, and J. D. Farmer, Physica D **58**, 77 (1992).

[26] P. E. Rapp, A. M. Albano, T. I. Schmah, and L. A. Farwell, Phys. Rev. E **47**, 2289 (1993).

[27] R. J. Solomonoff, Inform. Contr. **7**, 1 (1964).

[28] R. J. Solomonoff, Inform. Contr. **7**, 224 (1964).

[29] J. Ziv and A. Lempel, IEEE Trans. Inf. Theory **IT24**, 530 (1978).

[30] P. E. Rapp, M. A. Jiménez-Montaño, R. J. Langs, L. Thomson, and A. I. Mees, Math. Biosci. **105**, 207 (1991).

[31] M. A. Jiménez-Montaño, Bull. Math. Biol. **46**, 641 (1984).