

UC Berkeley

UC Berkeley Previously Published Works

Title

A simple method for detecting chaos in nature

Permalink

<https://escholarship.org/uc/item/2703h4k4>

Journal

Communications Biology, 3(1)

ISSN

2399-3642

Authors

Toker, Daniel
Sommer, Friedrich T
D'Esposito, Mark

Publication Date

2020

DOI

10.1038/s42003-019-0715-9

Peer reviewed

ARTICLE

<https://doi.org/10.1038/s42003-019-0715-9>

OPEN

A simple method for detecting chaos in nature

Daniel Toker ^{1*}, Friedrich T. Sommer¹ & Mark D'Esposito¹

Chaos, or exponential sensitivity to small perturbations, appears everywhere in nature. Moreover, chaos is predicted to play diverse functional roles in living systems. A method for detecting chaos from empirical measurements should therefore be a key component of the biologist's toolkit. But, classic chaos-detection tools are highly sensitive to measurement noise and break down for common edge cases, making it difficult to detect chaos in domains, like biology, where measurements are noisy. However, newer tools promise to overcome these limitations. Here, we combine several such tools into an automated processing pipeline, and show that our pipeline can detect the presence (or absence) of chaos in noisy recordings, even for difficult edge cases. As a first-pass application of our pipeline, we show that heart rate variability is not chaotic as some have proposed, and instead reflects a stochastic process in both health and disease. Our tool is easy-to-use and freely available.

¹Helen Wills Neuroscience Institute, University of California, Berkeley, CA, USA. *email: danieltoker@berkeley.edu

A remarkable diversity of natural phenomena are thought to be chaotic. Formally, a system is chaotic if it is bounded (meaning that, like a planet circling a star, its dynamics stay inside an orbit rather than escaping off to infinity), and if it is deterministic (meaning that, with the exact same initial conditions, it will always evolve over time in the same way), and if tiny perturbations to the system get exponentially amplified (Glossary (Supplementary Information), Supplementary Figs. 1, 2). The meteorologist Edward Lorenz famously described this phenomenon as the butterfly effect: in a chaotic system, something as small as the flapping of a butterfly's wings can cause an effect as big as a tornado. This conceptually simple phenomenon—i.e., extreme sensitivity to small perturbations—is thought to appear everywhere in nature, from cosmic inflation¹, to the orbit of Hyperion², to the Belousov–Zhabotinskii chemical reaction³, to the electrodynamics of the stimulated squid giant axon⁴. These are only a few examples of the many places in nature where chaos has been found.

It is relatively simple to determine if a simulated system is chaotic: just run the simulation a few times, with very slightly different initial conditions, and see how quickly the simulations diverge (Supplementary Fig. 1). But, if all that is available are measurements of how a real, non-simulated system evolves over time—for e.g., how a neuron's membrane potential changes over time, or how the brightness of a star changes over time—how can it be determined if those observations come from a chaotic system? Or if they are just noise? Or if the system is in fact periodic (Glossary, Supplementary Figs. 1, 2), meaning that, like a clock, small perturbations do not appreciably influence its dynamics?

While a reliable method for detecting chaos using empirical recordings should be an essential part of any scientist's toolbox, such a tool might be especially helpful to biologists, as chaos is predicted to play an important functional role in a wide variety of biological processes (that said, we note that real biological systems cannot be purely chaotic in the strict mathematical sense, since they certainly contain some level of dynamic noise—see Glossary—but that researchers have long speculated that many biological processes are still predominantly deterministic, but chaotic⁵). For example, following early speculations about the presence of chaos in the electrodynamics of both cardiac⁶ and neural⁷ tissue, the science writer Robert Pool posited in 1989 that “chaos may provide a healthy flexibility for the heart, brain, and other parts of the body.”⁸ Though this point has been intensely debated since the 1980s^{5,9}, a range of more specific possible biological functions for chaos have since been proposed, including potentially maximizing the information processing capacity of both neural systems¹⁰ and gene regulatory networks¹¹, enabling multistable perception¹², allowing neural systems to flexibly transition between different activity patterns¹³, and boosting cellular survival rates through the promotion of heterogeneous gene expression¹⁴. And there is good reason to expect chaos to exist in biological systems, as a large range of simulations of biological processes¹⁵, and in particular of neural systems⁹, show clear evidence of chaos. Moreover, unambiguous evidence of biological chaos has been found in a very small number of real cases that were amenable to comparison to good theoretical models; these include periodically stimulated squid giant axons⁴ and cardiac cells¹⁶, as well as the discharges of the *Onchidium* pacemaker neuron¹⁷ and the *Nitella flexillis* internodal cell¹⁸. But, beyond simulations and these select empirical cases, most attempts to test the presence or predicted functions of chaos in biology have fallen short due to high levels of measurement noise (Glossary) in biological recordings. For this reason, it has long been recognized that biologists need a noise-robust tool for detecting the presence (or absence) of chaos in their noisy empirical data^{9,15}.

Researchers also need a tool that can detect varying degrees of chaos (Glossary) in noisy recordings. In strongly chaotic systems, initially similar system states diverge faster than they do in weakly chaotic systems. And such varying degrees of chaos are predicted to occur in biology, with functional consequences. For example, a model of white blood cell concentrations in chronic granulocytic leukemia can display varying levels of chaos, and knowing how chaotic those concentrations are in actual leukemia patients could have important implications for health outcomes¹⁹. As another example, models of the human cortex predict that macro-scale cortical electrodynamics should be weakly chaotic during waking states and should be strongly chaotic under propofol anesthesia²⁰; if this prediction is true, then detecting changing levels of chaos in large-scale brain activity could be useful for monitoring depth of anesthesia and for basic anesthesia research. Thus, it is imperative to develop tools that can not only determine that an experimental system is chaotic, but also tools to assess changing levels of chaos in a system.

Although classic tools for detecting the presence and degree of chaos in data are slow, require large amounts of data, are highly sensitive to measurement noise, and break down for common edge cases, more recent mathematical research has provided new, more robust tools for detecting chaos or a lack thereof in noisy time-series recordings. Here, for the first time (to our knowledge), we combine several key mathematical tools into a single, fully automated Matlab processing pipeline, which we call the Chaos Decision Tree Algorithm²¹ (Fig. 1). The Chaos Decision Tree Algorithm takes a single time-series of any type—be it recordings of neural spikes, time-varying transcription levels of a particular gene, fluctuating oil prices, or recordings of stellar flux—and classifies those recorded data as coming from a system that is predominantly (or “operationally”²²) stochastic, periodic, or chaotic. The algorithm requires no input from the user other than a time-series recording, though we have structured our code such that users can also select from among a number of alternative subroutines (see Methods section, Fig. 1).

In this paper, we show that the Chaos Decision Tree Algorithm performs with very high accuracy across a wide variety of both real and simulated systems, even in the presence of relatively high levels of measurement noise. Moreover, our pipeline can accurately track changing degrees of chaos (for e.g., transitions from weak to strong chaos). With an eye toward applications to biology, the simulated systems we tested included a high-dimensional mean-field model of cortical electrodynamics, a model of a spiking neuron, a model of white blood cell concentrations in chronic granulocytic leukemia, and a model of the transcription of the NF- κ B protein complex. We also tested the algorithm on a wide variety of non-biological simulations, including several difficult edge cases; these included strange non-chaotic systems, quasi-periodic systems, colored noise, and nonlinear stochastic systems (see Glossary for definitions of these terms), which are all classically difficult to distinguish from chaotic systems^{23–26}. We also tested the algorithm on a hyperchaotic system (Glossary), which can be difficult to distinguish from noise²⁵, as well as on several non-stationary processes (Glossary) in order to test the robustness of the algorithm against non-stationarity. Finally, we tested the Chaos Decision Tree Algorithm on several empirical (i.e. non-simulated) datasets for which the ground-truth presence or absence of chaos has been previously established by other studies. These included an electronic circuit in periodic, strange non-chaotic, and chaotic states²⁷, a chaotic laser²⁸, the stellar flux of a strange non-chaotic star²⁹, the linear/stochastic North Atlantic Oscillation index³⁰, and nonlinear/stochastic Parkinson's and essential tremors²⁶. Overall, our pipeline performed with near-perfect accuracy in classifying these data as stochastic, periodic, or chaotic, as well as

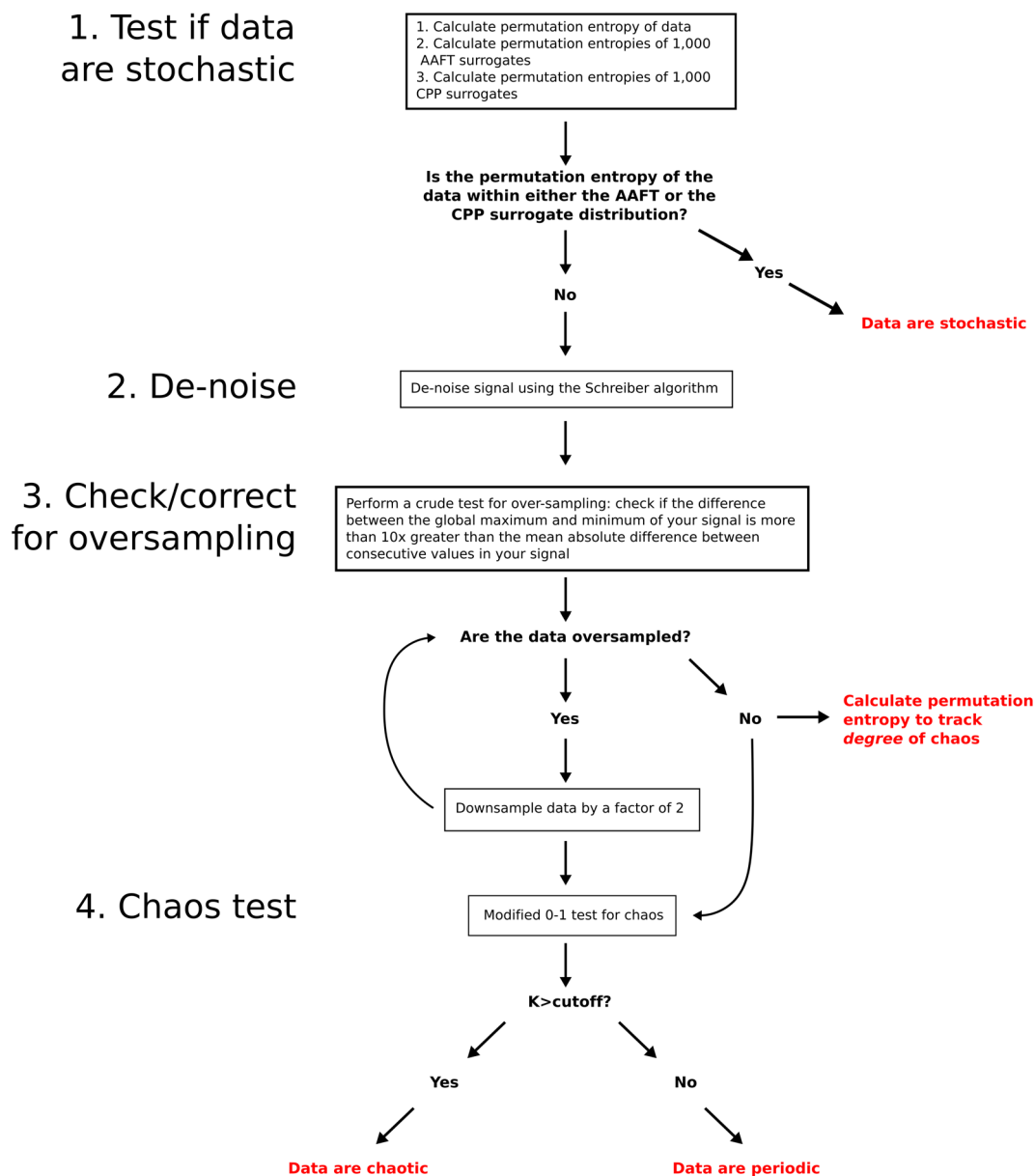


Fig. 1 The Chaos Decision Tree Algorithm²¹. The first step of the algorithm is to test if data are stochastic. The Chaos Decision Tree Algorithm uses a surrogate-based approach to test for stochasticity, by comparing the permutation entropy of the original time-series to the permutation entropies of random surrogates of that time series. If the user does not specify which surrogate algorithms to use, the Chaos Decision Tree Algorithm automatically picks a combination of Amplitude Adjusted Fourier Transform³³ surrogates and Cyclic Phase Permutation³⁵ surrogates—see Supplementary Tables 2, 3. If the permutation entropy of the original time-series falls within either surrogate distribution, the time-series is classified as stochastic; if the permutation entropy falls outside the surrogate distributions, then the algorithm proceeds to denoise the inputted signal. Several de-noising subroutines are available, but if the user does not specify a subroutine, the pipeline will use Schreiber’s noise-reduction algorithm³⁶ (Supplementary Table 5). The pipeline then checks for signal oversampling; if data are oversampled, the pipeline iteratively downsamples the data until they are no longer oversampled (note that an alternative downsampling method proposed by Eyébé Fouda and colleagues⁶⁸ may be selected instead—see Supplementary Table 6). Finally, the Chaos Decision Tree Algorithm performs the 0-1 chaos test on the input data, which has been modified from the original 0-1 test to be less sensitive to noise, suppress correlations arising from quasi-periodicity, and normalize the standard deviation of the test signal (see Methods section). The value for the parameter that suppresses signal correlations can be specified by the user, but is otherwise automatically chosen based on ROC analyses performed here (Supplementary Fig. 4). The modified 0-1 test provides a single statistic, K , which approaches 1 for chaotic systems and approaches 0 for periodic systems. Any cutoff for K may be inputted to the pipeline, and if no cutoff is provided, the pipeline will use a cutoff based on the length of the time-series (Supplementary Fig. 6). If K is greater than the cutoff, the data are classified as chaotic, and if they are less than or equal to the cutoff, they are classified as periodic.

in tracking changing degrees of chaos in both real and simulated systems. Finally, we applied our algorithm to electrocardiogram recordings from healthy subjects, patients with congestive heart failure, and patients with atrial fibrillation³¹, and provide

evidence that heart rate variability reflects a predominantly stochastic, rather than chaotic process.

We have made our Matlab code freely and publicly available at <https://figshare.com/s/80891dfb34c6ee9c8b34>.

Results

The Chaos Decision Tree Algorithm²¹ is depicted graphically in Fig. 1. The pipeline consists of four steps: (1) Determine if the data are stochastic using permutation entropy³² and a combination of Amplitude Adjusted Fourier transform surrogates^{33,34} and Cyclic Phase Permutation surrogates^{34,35} (Glossary), (2) De-noise the data using the Schreiber de-noising algorithm³⁶ (Glossary), (3) Correct for possible signal oversampling, and (4) Test for chaos using a modified 0–1 test for chaos^{23,37–40} (Glossary). For each step of the processing pipeline, we compared the performance of different available tools (i.e., different surrogate-based tests for stochasticity, different de-noising methods, different downsampling methods, and different chaos-detection methods), and chose the tools with the highest classification performance (Supplementary Tables 1–14). Note that with user input, the Chaos Decision Tree Algorithm can use any of the alternative tools tested here, and that with no user input other than a time-series recording, the algorithm will automatically use the tools we found maximized its performance. All results reported in the main body of this paper are for this automated set of high-performing tools. See Supplementary Fig. 3 for example time-traces illustrating each step of the algorithm.

We tested the performance of the (automated) Chaos Decision Tree Algorithm in detecting the presence and degree of chaos in a wide range of simulated and empirical systems for which the ground-truth presence of chaos, periodicity, or stochasticity has already been established. Details about each dataset and how the ground-truth presence or absence of chaos in those systems was previously determined are included in the Methods. Note that some systems are labeled “SNA,” which is an abbreviation for “strange non-chaotic attractor” (Glossary). These are systems whose attractors in phase space (Glossary) are fractal (like chaotic systems), but which are periodic (i.e., non-chaotic). Among these, we included the only known non-artificial strange non-chaotic system, the stellar flux of the so-called golden star KIC 5520878, as recorded by the Kepler space telescope²⁹. All simulated datasets consisted of 10,000 time-points, and all initial conditions were randomized. For systems with more than one variable, we here report results for linear combinations of those variables (see Methods section), under the assumption that in most real-life cases, empirical recordings will contain features of multiple components of the system of study; that said, we also confirmed that the Chaos Decision Tree Algorithm has very high performance for individual system variables (Supplementary Table 15).

Results for simulations of biological systems are reported in Table 1, and results for non-biological simulations are reported in Table 2. Note that no measurement noise was added to the colored noise signals in Table 2, as doing so would flatten their power spectra. Because the datasets in Tables 1 and 2 were used to choose between alternative methods for detecting stochasticity (Supplementary Tables 1–4), de-noising (Supplementary Table 5), downsampling (Supplementary Table 6), and alternative

tests of chaos (Supplementary Tables 8–13), as well as to optimize the 0–1 test for chaos (Supplementary Figs. 4–6), we further tested the Chaos Decision Tree Algorithm on held out datasets, which were not used to adjudicate between alternative tools. These held out datasets included both simulated systems (Table 3) and recordings from real (non-simulated) systems (Table 4). Several of these held out datasets were of direct biological relevance: the periodically stimulated Poincaré oscillator in Table 3 is thought to be a good model of cardiac cell electrodynamics⁴¹, which, like the Poincaré oscillator, are chaotic when periodically stimulated with certain delays between stimulation pulses¹⁶; the integrated circuit in Table 4 is a physical implementation of equations that are based on the Hodgkin-Huxley neuron model⁴²; and the tremor signals in Table 4 are direct recordings from patients. The Chaos Decision Tree Algorithm classified the systems in Tables 1–4 as stochastic, periodic, or chaotic with near-perfect accuracy even at high levels of measurement noise, with the exception of the noise-driven sine map (Table 2)—see Discussion section. Finally, we tested the performance of the Chaos Decision Tree Algorithm on sub-samples of all systems in Tables 1–3, and confirmed that it is still highly accurate for data with just 1000 time-points (Supplementary Table 16) or 5000 time-points (Supplementary Table 17), though we note that performance for some systems did go down with less data, which is to be expected⁴⁰.

Table 5 reports the accuracy of the Chaos Decision Tree Algorithm in detecting degree of chaos. Formally, a system’s degree of chaos is quantified by the magnitude of its largest Lyapunov exponent (Glossary). Unfortunately, largest Lyapunov exponents are very difficult to estimate from finite, noisy time-series recordings. But, directly estimating largest Lyapunov exponents may not be necessary for tracking changing degrees of chaos in real systems: following prior observations of a strong correlation between a quick-to-compute and noise-robust measure called permutation entropy (Glossary) and the largest Lyapunov exponents of several systems^{32,43}, the Chaos Decision Tree Algorithm approximates degree of chaos by calculating the permutation entropy of the inputted signal, after it has been de-noised and corrected for possible over-sampling. In agreement with prior findings, we found that permutation entropy tracked degree of chaos in the logistic map, the Hénon map, the Lorenz system, a high-dimensional mean-field model of the cortex, and an electronic circuit. See Methods for details on the parameters that were used to generate dynamics with different degrees of chaos in these systems, and for details on how ground-truth largest Lyapunov exponents were calculated. Note that without downsampling, the correlation between the largest Lyapunov exponents and permutation entropy breaks down in continuous systems (Supplementary Table 14), which is to be expected, as permutation entropy has only been analytically proven to track the degree of chaos in discrete-time systems^{32,44} (see Glossary).

Table 1 The Chaos Decision Tree Algorithm classified biological simulations as periodic or chaotic with near-perfect accuracy.

System	Measurement noise level (% of std. dev.)				
	0%	10%	20%	30%	40%
Cortical model ²⁰ (chaotic)	100/100	100/100	100/100	97/100	83/100
Cortical model ²⁰ (periodic)	100/100	100/100	100/100	100/100	100/100
Spiking neuron ⁴⁹ (chaotic)	100/100	100/100	100/100	100/100	98/100
Granulocyte levels ¹⁹ (chaotic)	100/100	100/100	100/100	100/100	100/100
Granulocyte levels ¹⁹ (periodic)	100/100	100/100	100/100	100/100	83/100
NF-κB transcription ¹⁴ (chaotic)	99/100	99/100	100/100	100/100	100/100
NF-κB transcription ¹⁴ (periodic)	100/100	100/100	97/100	100/100	100/100

Table 2 The Chaos Decision Tree Algorithm classified non-biological simulations as stochastic, periodic, or chaotic with high accuracy. These simulated systems include strange non-chaotic attractors (SNAs), linear stochastic processes, and nonlinear stochastic processes, all of which are classically difficult to distinguish from chaos.

System	Measurement noise level (% of std. dev.)				
	0%	10%	20%	30%	40%
Cubic map ⁵⁰ (chaotic)	100/100	100/100	100/100	100/100	100/100
Cubic map ⁵⁰ (periodic)	100/100	100/100	100/100	100/100	100/100
Cubic map ⁵⁰ (SNA HH)	100/100	100/100	100/100	100/100	100/100
Cubic map ⁵⁰ (SNA S3)	100/100	100/100	100/100	100/100	0/100
GOPY map ⁵¹ (SNA)	100/100	100/100	100/100	99/100	14/100
Logistic map ⁵² (chaotic)	100/100	100/100	100/100	100/100	100/100
Logistic map ⁵² (periodic)	100/100	100/100	100/100	100/100	100/100
Lorenz system ⁵³ (chaotic)	100/100	100/100	97/100	82/100	36/100
Generalized Hénon map ⁵⁴ (hyperchaotic)	100/100	100/100	100/100	100/100	93/100
Freitas map ⁵⁵ (nonlinear stochastic)	78/100	83/100	98/100	98/100	74/100
Noise-driven sine map ⁵⁵ (nonlinear stochastic)	55/100	3/100	22/100	5/100	78/100
Bounded random walk ⁵⁶ (nonlinear stochastic)	100/100	97/100	59/100	95/100	100/100
Cyclostationary process ⁵⁷ (linear stochastic)	99/100	100/100	99/100	100/100	100/100
ARMA process (linear stochastic)	85/100	98/100	99/100	99/100	100/100
Trended random walk (linear stochastic)	100/100	89/100	90/100	98/100	100/100
Random walk (linear stochastic)	100/100	98/100	100/100	100/100	100/100
Violet noise ⁵⁸ (linear stochastic)	99/100				
Blue noise ⁵⁸ (linear stochastic)	100/100				
White noise ⁵⁸ (linear stochastic)	100/100				
Pink noise ⁵⁸ (linear stochastic)	100/100				
Red noise ⁵⁸ (linear stochastic)	100/100				

Table 3 Classification accuracy in held out simulated systems. While the datasets in Tables 1, 2 were used to optimize the Chaos Decision Tree Algorithm, these datasets were not. Performance was near-perfect.

System	Measurement noise level (% of std. dev.)				
	0%	10%	20%	30%	40%
Rössler system ⁵⁹ (chaotic)	70/100	90/100	96/100	100/100	100/100
Ikeda map ⁶⁰ (chaotic)	100/100	100/100	100/100	100/100	14/100
Hénon map ⁶¹ (periodic)	100/100	100/100	100/100	100/100	100/100
Cubic map ⁶¹ (period-doubled)	100/100	100/100	100/100	100/100	100/100
Poincaré oscillator ⁴¹ (periodic)	100/100	100/100	100/100	100/100	57/100
Poincaré oscillator ⁴¹ (quasi-periodic)	100/100	100/100	100/100	100/100	100/100
Poincaré oscillator ⁴¹ (chaotic)	100/100	100/100	100/100	100/100	100/100
Multivariate AR model (linear stochastic)	100/100	100/100	100/100	100/100	100/100

Finally, as a first-pass implementation of our method, we applied the Chaos Decision Tree Algorithm to recordings of human heart rate variability, made available by Physionet³¹. There has been considerable debate over whether or not the irregularities of heart rate signals (in either health or disease) reflect a predominantly chaotic process. While many classic chaos-detection methods have identified heart rate variability as chaotic (see Glass⁵ for a review), other studies have argued that this is an erroneous classification, suggesting that heart rate variability is, in fact, a nonlinear stochastic process^{45,46}, and that prior classifications of heart rate signals as chaotic simply reflect the shortcomings of classic chaos-detection methods. In agreement with this view, we here show that the Chaos Decision Tree Algorithm classified heart rate signals from healthy subjects, congestive heart failure patients, and atrial fibrillation patients as stochastic, rather than chaotic, with the exception of two congestive heart failure patients (Table 6).

Discussion

In this paper, we have introduced a processing pipeline, called the Chaos Decision Tree Algorithm²¹, that can accurately detect

whether a time-series signal is generated by a predominantly stochastic, periodic, or chaotic system, and can also accurately track changing levels of chaos within a system using permutation entropy. The pipeline makes no assumptions about the input data. The Chaos Decision Tree Algorithm consists of four broad steps: (1) testing for stochasticity using surrogate data methods, (2) de-noising, (3) downsampling if data are oversampled, and (3) testing for chaos using the modified 0–1 test. We tested the performance of several different surrogate data generation algorithms, de-noising algorithms, downsampling algorithms, and parameters for the modified 0–1 test. Each alternative algorithm and parameter choice has its relative strengths and weaknesses, and we have structured our code such that a user can specify which algorithms and parameters to use for each step of the pipeline. If a user only inputs a time-series recording without specifying any sub-algorithms or parameters, then our pipeline will automatically use the methods and parameters we found yielded the most accurate results across a large and diverse set of data. All analyses reported in the main body of this paper are for this automated set of subroutines.

We tested the (automated) Chaos Decision Tree Algorithm on a diverse range of simulations of biological systems, non-

biological simulations, and empirical (non-simulated) data recordings. Empirical data were recorded from an integrated circuit in a periodic, strange non-chaotic, and chaotic state, a chaotic laser, the stellar flux of a strange non-chaotic star, the North Atlantic Oscillation index, a Parkinson's tremor, an essential tremor, and heart rate variability from congestive heart failure patients, atrial fibrillation patients, and healthy controls. In the cases for which the ground-truth was known (i.e., all datasets other than heart rate variability), the Chaos Decision Tree Algorithm performed at very high accuracy even at relatively high levels of measurement noise. For heart rate variability, our results support the hypothesis that cardiac rhythm variability is stochastic^{45,46}. Overall, these findings make us confident that the Chaos Decision Tree Algorithm can be fruitfully applied to biological and non-biological signals contaminated by measurement noise.

We note a few limitations/shortcomings of our algorithm. First, the 0–1 test used in our pipeline might classify some very weakly chaotic systems (i.e., systems whose largest Lyapunov exponent is positive but very near zero) as periodic if the length of the time-series provided is short; but, with longer time-series, the test is guaranteed to provide accurate results⁴⁰. We also note that the algorithm performed poorly for the noise-driven sine map, which was consistently mis-classified as chaotic (Table 2). It is possible that this system was not classified as stochastic because its level of intrinsic noise was very low; in support of this, we found that the Chaos Decision Tree Algorithm classified nonlinear dynamical systems with very low levels of intrinsic noise as deterministic, and that classifications of stochasticity became more frequent as

the level of intrinsic noise was increased (Supplementary Table 18). It is also possible that this system is, in fact, an example of noise-induced chaos⁴⁷. Finally, although the choice of system observables did not appreciably affect the performance of our method (Supplementary Table 15), we agree with Letellier and colleagues⁴⁸ that some system observables are better representations of a system's dynamics than others, and that this can have important consequences for the accuracy of nonlinear time-series analysis methods such as this one. In light of these potential limitations, it bears re-emphasizing that the absence, presence, and degree of chaos can only be determined with absolute certainty in a computer model that is free of measurement noise, by running multiple simulations and seeing how quickly initially similar states diverge. Thus, although the Chaos Decision Tree Algorithm pipeline performs at very high accuracy, it should, when possible, be used in conjunction with analyses of a computer model of the system at hand.

We hope that the Chaos Decision Tree Algorithm will help advance the decades-old effort to bring the insights of chaos theory to biology. While a diverse range of biological simulations and a small number of real biological cases have been shown to be chaotic, detecting the presence and degree of chaos in biological recordings has been difficult. The Chaos Decision Tree Algorithm overcomes the difficulties of prior tests, by being fast, highly robust to measurement noise, and, unless the user specifies specific alternative subroutines, fully automated. We welcome any efforts to identify edge cases for which our pipeline systematically breaks down; given that our pipeline is a modular decision tree, new subroutines can be added to accommodate such edge cases. We hope that our pipeline (and perhaps future iterations of it) will be useful to any of the domains of science—and in particular of biology—in which chaos has been invoked, but not tested.

Table 4 Classification accuracy in empirical (non-simulated) data from stochastic, periodic, strange non-chaotic (SNA), and chaotic systems. As was the case for the data in Table 3, these datasets were not used to optimize algorithm performance, and so are also held out datasets. Algorithm performance was perfect.

System	
Neuron integrated circuit ²⁷ (chaotic)	10/10
Neuron integrated circuit ²⁷ (SNA)	10/10
Neuron integrated circuit ²⁷ (periodic)	10/10
Laser ²⁸ (chaotic)	1/1
Stellar flux ²⁹ (SNA)	1/1
North Atlantic Oscillation index ³⁰ (linear stochastic)	1/1
Essential tremor ²⁶ (nonlinear stochastic)	1/1
Parkinson's tremor ²⁶ (nonlinear stochastic)	1/1

Methods

The Chaos Decision Tree Algorithm. To understand the logic of the Chaos Decision Tree Algorithm²¹, we begin with the final test in the decision tree. The crux of the Chaos Decision Tree Algorithm is the 0–1 test for chaos. The 0–1 test for chaos was originally developed by Gottwald and Melbourne²⁷, who later offered a slightly modified version of the test, which can cope with moderate amounts of measurement noise³⁸. Several years later, Dawes and Freeland further modified the test, such that it could suppress correlations induced by quasi-periodic dynamics, and thus more effectively distinguish between chaotic and strange non-chaotic dynamics, which are difficult to distinguish given only a time-series recording²³. The modified 0–1 test involves taking a one-dimensional time-series of interest ϕ , and using it to drive the following two-dimensional system:

$$\begin{aligned} p(n+1) &= p(n) + \phi(n) \cos cn \\ q(n+1) &= q(n) + \phi(n) \sin cn \end{aligned} \quad (1)$$

where c is a random value bounded between 0 and 2π . For a given c , the solution to

Table 5 The Chaos Decision Tree Algorithm uses permutation entropy, calculated from data that have been de-noised and downsampled (if oversampled), to track the degree of chaos in a system, which might change as the state of the system changes.

System	Measurement noise level (% of std. dev.)				
	0%	10%	20%	30%	40%
Logistic map ⁵²	0.93***	0.80***	0.93***	0.93***	0.91***
Hénon map ⁶¹	0.92***	0.93***	0.94***	0.92***	0.88***
Lorenz system ⁵³	0.81***	0.71***	0.73***	0.69***	0.62***
Cortical model ²⁰	0.69***	0.55***	0.39***	0.33***	0.24***
Neuron integrated circuit ²⁷	0.94***				

We here show Spearman correlations between permutation entropy and largest Lyapunov exponents, which measure degree of chaos but which are difficult to estimate from empirical data. Data include four simulated systems and recordings from an integrated circuit in different states. See Methods for how ground-truth largest Lyapunov exponents were calculated for these systems
*** $p < 0.001$ (two-tailed) after Bonferroni-correcting for multiple comparisons to the same set of ground-truth largest Lyapunov exponents

Table 6 The Chaos Decision Tree Algorithm consistently classifies heart rate recordings, across conditions, as stochastic.

Condition	Classification		
	Stochastic	Periodic	Chaotic
Healthy controls ³¹	5/5	0/5	0/5
Congestive heart failure ³¹	3/5	2/5	0/5
Atrial fibrillation ³¹	5/5	0/5	0/5

The only exceptions were the heart rate signals recorded from two patients with congestive heart failure, which were classified as periodic

Eq. (1) yields:

$$p_c(n) = \sum_{j=1}^n \phi(j) \cos jc \quad (2)$$

$$q_c(n) = \sum_{j=1}^n \phi(j) \sin jc$$

Gottwald and Melbourne show that if the inputted time-series ϕ is regular, the motion of \mathbf{p} and \mathbf{q} is bounded, while \mathbf{p} and \mathbf{q} display asymptotic Brownian motion if ϕ is chaotic. The time-averaged mean square displacement of \mathbf{p} and \mathbf{q} , plus the noise term proposed by Dawes and Freeland²³, is

$$M_c(n) = \frac{1}{N} \sum_{j=1}^N [(p_c(j+n) - p_c(j))^2 + (q_c(j+n) - q_c(j))^2] + \sigma \eta_n \quad (3)$$

where η_n is a uniformly distributed random variable between $[-\frac{1}{2}, \frac{1}{2}]$ and σ is the noise level. Finally, the outputted K -statistic of the 0–1 test uses a correlation coefficient to measure the growth rate of the mean squared displacement of the two-dimensional system in Eq. (1):

$$K_c = \text{corr}(n, M_c(n)) \quad (4)$$

K is computed for 100 different values of c , randomly sampled between 0 and 2π , and the final output of the test is the median K across different values of c . For chaotic systems, this median K value will approach 1, and for periodic systems, K will approach 0^{23,37–40}.

There are two parameters in this modified 0–1 test: the parameter σ that controls the level of added noise in Eq. (3), and the cutoff for what K -statistic values are classified as indicating chaos or periodicity in a finite time-series. We performed ROC-curve analyses for different values of σ and found that $\sigma = 0.5$ maximized classification performance across systems and noise levels (Supplementary Fig. 4), and so our pipeline automatically sets σ to 0.5 if σ is not specified by the user. Note that for non-zero values of σ , K approaches zero as the standard deviation of a test signal approaches zero (Supplementary Fig. 5), and so the Chaos Decision Tree Algorithm multiplies a test signal by a constant to fix its standard deviation at 0.5 before applying the 0–1 test. A cutoff for K can also be inputted to our Matlab script, such that data that yield a K value greater than that cutoff are classified as chaotic and data that yield a K value less than or equal to that cutoff are classified as periodic. If no cutoff is provided, a cutoff is chosen based on an analysis of optimal cutoffs as a function of time-series length (Supplementary Fig. 6). If the automatically selected cutoff is greater than 0.99, the cutoff is set to $K = 0.99$, as K is upper-bounded by 1. We have confirmed that this automated cutoff selection yields highly accurate results for sub-samples of both test and held-out datasets (Supplementary Tables 16, 17).

The 0–1 test described above only yields accurate results for data that are deterministic^{24,40,62,63}. A system is considered deterministic if, given the exact same initial conditions, it always evolves over time the same way, whereas a system is considered stochastic if there is appreciable randomness built in to its evolution over time (Glossary, Supplementary Fig. 1, 2). Not only are all chaotic systems (predominantly) deterministic—and thus the possibility of chaos can be automatically rejected if a system is found to be stochastic (though we note that a mathematically rigorous definition of chaos has recently been extended to the domain of stochastic systems, under the framework of the Supersymmetric Theory of Stochastics⁴⁷)—but the 0–1 test is also known to incorrectly classify stochastic dynamics as chaotic^{24,62,63}. Thus, the Chaos Decision Tree Algorithm first rules out the possibility that data are predominantly stochastic before applying the modified 0–1 test. To do so, it uses a noise-robust method recently developed by Zunino and Kulp⁶⁴, which tests for determinism using surrogate statistics³³, with permutation entropy³² as the test statistic. The calculation of permutation entropy relies on two parameters: permutation order and time-lag. We follow the recommendation from Bandt and Pompe³² and set the time-lag to 1, and found that a permutation order of 8 maximized stochasticity detection accuracy (Supplementary Tables 2, 3). Moreover, we use a combination of amplitude adjusted Fourier transform

surrogates³³ and Cyclic Phase Permutation surrogates³⁵, unlike Zunino and Kulp, who used iterative amplitude adjusted Fourier transform³³ surrogates, because we found that this combination led to far higher classification accuracy (Supplementary Tables 2, 3). The Chaos Decision Tree Algorithm classifies data as stochastic (and thus does not proceed to subsequent steps) if the permutation entropy of the original data falls within either surrogate distribution. The algorithm uses the Toolboxes for Complex Systems implementation of the permutation entropy algorithm, written by Andreas Müller⁶⁵. Surrogates are generated using the Matlab toolbox recently released by Lancaster and colleagues³⁴. Note that because Fourier-based surrogates are strictly stationary, surrogate-based tests that use only Fourier-based algorithms are only valid if the test time-series is also stationary^{34,57}; that said, we found that non-stationarity did not affect the accuracy of a stochasticity test that uses a combination of amplitude adjusted Fourier transform and Cyclic Phase Permutation surrogates (Supplementary Tables 1–4). We also did not find that a normality transformation of the data improved the performance of our surrogate-based stochasticity test (Supplementary Table 2), counter to what has been suggested elsewhere²².

If data “pass” the stochasticity test described above and are deemed operationally deterministic, then the Chaos Decision Tree Algorithm automatically denoises the inputted signal. We compared three de-noising algorithms: a moving average filter (using Matlab’s `smooth.m` function), the Matlab Chaotic Systems Toolbox’s⁶⁶ implementation of Schreiber’s noise-reduction algorithm³⁶ (Glossary), and wavelet de-noising using an empirical Bayesian method with a Cauchy prior (using Matlab’s `wdenoise.m` function). Although it is considerably slower to run, Schreiber denoising markedly outperforms the other two approaches in recovering the deterministic component of signals contaminated by measurement noise (Supplementary Table 5), and markedly improves the performance of the modified 0–1 test (Supplementary Table 6, Supplementary Fig. 4). Thus, the Chaos Decision Tree Algorithm automatically uses Schreiber de-noising before testing for chaos, unless the user specifies one of the other two de-noising algorithms tested here to be used instead.

The final step of the Chaos Decision Tree Algorithm before applying the 0–1 test is to check if data are oversampled and to downsample them if they are. Gottwald and Melbourne have shown³⁹ that the 0–1 test can give inaccurate results for continuous (i.e., non-discrete-time) systems sampled at a very high frequency, but that it can accurately differentiate between periodic dynamics and chaotic dynamics in continuous deterministic systems when data are properly downsampled. In light of this, the Chaos Decision Tree Algorithm utilizes the (crude) test for oversampling used by Matthews⁶⁷, by calculating a measure η , which is the difference between the global maximum and global minimum of the data divided by the mean absolute difference between consecutive time-points in the data. If $\eta > 10$, then the data are deemed to be oversampled, and the Chaos Decision Tree Algorithm iteratively downsamples the data by a factor of 2 until $\eta \leq 10$ or until there are fewer than 100 time-points left in the signal. We compared this approach both to no downsampling and to an alternative method, suggested by Eyébé Fouda and colleagues⁶⁸ to improve 0–1 test performance, which downsamples by taking just the local minima and maxima of oversampled signals. We found that downsampling after de-noising yields more accurate results than either alternative approach when oversampled signals are contaminated by measurement noise (Supplementary Table 6). We also note that recorded experimental data may be unlikely to be oversampled (Supplementary Table 7), and that this problem may be more likely to arise in simulated continuous systems. If the data are not oversampled, or if they have been downsampled, the Chaos Decision Tree Algorithm then applies the modified 0–1 test to the data, as described above.

Finally, the algorithm uses the permutation entropy of the inputted signal as a proxy for the degree of chaos in the system. Though the algorithm uses permutation entropy to establish whether or not a signal is predominantly deterministic (see above), permutation entropy has also been shown to tightly track the largest Lyapunov exponent (and therefore the degree of chaos) of the logistic map³², the tent map⁶⁹, and the Duffing oscillator⁴³. We should in general expect a close correspondence between permutation entropy and Lyapunov exponents, in light of the equivalence in discrete-time systems between permutation entropy and Kolmogorov-Sinai entropy^{44,70–72}, which is upper-bounded by the sum of a system’s positive Lyapunov exponents—a relationship known as the “Pesin identity”⁷³. When calculating permutation entropy to track degree of chaos (rather than for determinism testing as above), we follow Bandt and Pompe’s³² recommendation and simply set the time-lag to 1 and the permutation order to 5, which we showed tracks degree of chaos in all systems tested (Table 5). Because this equivalence is only known to hold for discrete-time systems⁴⁴, permutation entropy is only calculated after the inputted signal has been de-noised and, if oversampled, downsampled; this considerably improves its ability to track degree of chaos in continuous systems (Table 5, Supplementary Table 14).

The full decision tree of our algorithm is depicted graphically in Fig. 1.

Data

Biological simulations. The following describes the simulations of biological systems analyzed in this paper. We only picked biological simulations for which the presence or absence of chaos has been established in prior work. Initial conditions were randomized in all simulations. We also tested the effect of measurement noise

on the accuracy of the Chaos Decision Tree Algorithm in classifying systems, by adding white noise to our simulated data, the amplitude of which was up to 40% the standard deviation of the original data. For each simulated system and level of measurement noise, we created 100 datasets with 10,000 time points.

Chaotic mean-field cortical model. Steyn-Ross, Steyn-Ross, and Sleight²⁰ describe a mean-field model of the cortex based on the equations first introduced by Liley and colleagues^{74,75}, which includes electrical gap-junction synapses in addition to the standard chemical synapses used in the earlier models. The model contains both inhibitory and excitatory neural populations communicating locally through gap junctions and chemical synapses and communicating over long ranges via myelinated axons. The dynamics of each neural population in the model are determined by two first-order and six second-order partial differential equations, which is equivalent to 14 first-order differential equations. The primary output of the model is the mean excitatory firing rates of 120 neural populations, which approximates the large-scale cortical signals that might be measured through electrocortigraphy, magnetoencephalography, or electroencephalography. Steyn-Ross, Steyn-Ross, and Sleight²⁰ show that just by varying the inhibitory gap-junction diffusive-coupling strength parameter in their model, they can produce dynamics ranging from periodicity to strong chaos. In their simulation of “waking” cortical dynamics, Turing (spatial) and Hopf (temporal) instabilities interact to produce chaotic, low-frequency spatiotemporal oscillations. For chaotic dynamics, we simulated 2,000,000 time-points of their “wake” simulation, with the inhibitory gap-junction diffusive-coupling strength parameter set to 0.4, and then downsampled the data to 10,000 time-points. We only applied our algorithm to the mean excitatory firing rate of one neural population, i.e. to just one out of 14 variables describing the dynamics of just one out of 120 interacting such 14-dimensional systems (though the variable is biologically well-defined). The Matlab code for the simulations is available in the Supplementary Material of Steyn-Ross, Steyn-Ross, and Sleight²⁰.

Periodic mean-field cortical model. Steyn-Ross, Steyn-Ross, and Sleight show that their cortical mean-field model enters a periodic, seizure-like state dominated by a Hopf instability when the inhibitory gap-junction diffusive-coupling strength parameter is set to 0.1. Just as in the chaotic case, we simulated 2,000,000 time-points and then downsampled to 10,000 time-points. Note that Steyn-Ross, Steyn-Ross, and Sleight estimate the largest Lyapunov exponent of their model to be around zero when the inhibitory gap-junction diffusive-coupling strength parameter is 0.1, whereas our own estimate (using an automated version of their same method—see below) placed the largest Lyapunov exponent more clearly in the periodic regime, at -2.1 .

Chaotic spiking neuron. Izhikevich^{49,76} describes a simple neuron model that can display both spiking and bursting behavior. The model consists of a neuron’s membrane potential v , a membrane recovery variable u , an input current I , and parameters a, b, c , and d :

$$\begin{aligned} \frac{dv}{dt} &= 0.004v^2 + 5v + 140 - u + I \\ \frac{du}{dt} &= a(bv - u) \end{aligned} \tag{5}$$

with the auxiliary after-spike resetting:

$$\text{if } v \geq +30 \text{ mV, then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d. \end{cases} \tag{6}$$

When $a = 0.2$, $b = 2$, $c = -56$, $d = -16$, and $I = -99$, the neuron’s membrane potential v (which is the variable we analyze) displays chaotic spikes^{49,76}. We simulated the Izhikevich neuron using a first-order Euler method, with an integration step of 0.25 ms. We generated 50,000 time points, and downsampled by a factor of 5 (to avoid over-sampling).

Periodic white blood cell concentration. Inspired by the finding that chronic granulocytic leukemia involves apparently aperiodic oscillations in the concentration of circulating white blood cells⁷⁷, Mackey and Glass¹⁹ study mathematical models of oscillating physiological control systems. They describe a simple mathematical model of the concentration of circulating white blood cells:

$$\frac{dx}{dt} = a \frac{x_t}{1 + x_t^\tau} - bx \tag{7}$$

where $a = 0.2$, $b = 0.1$, and $c = 10$. The parameter τ represents the delay between white blood cell production in bone marrow and the release of those cells into the blood stream. Since this cellular generation delay time is increased in some patients with chronic granulocytic leukemia, Mackey and Glass study the behavior of this system as a function of the delay time τ . They find that as τ increased, the resulting oscillations produced by this equation became aperiodic. Through formal analysis of Lyapunov exponents of this system, Farmer⁷⁸ later confirmed that for $\tau = 10$, the oscillations of this system are periodic. We simulated 100,000 time-points of the periodic Mackey-Glass system using a first-order Euler method with an integration step of 1, and then downsampled by a factor of 10 (to avoid over-sampling).

Chaotic white blood cell concentration. For $\tau = 30$, Farmer confirmed⁷⁸ that the Mackey-Glass equation (Eq. (7)) for the concentration of circulating white blood cells yields a chaotic oscillation. We simulated 100,000 time-points of the chaotic

Mackey-Glass system using a first-order Euler method with an integration step of 1, and then downsampled by a factor of 10 (to avoid over-sampling).

Periodic NF- κ B transcription. Heltberg and colleagues¹⁴ recently described a five-dimensional mathematical model of oscillating concentrations of the transcription factor NF- κ B, which regulates several genes involved in immune responses and is widely studied in immunity and cancer research. They show that the dynamics of NF- κ B concentration are coupled to varying levels of a cytokin-like tumor necrosis factor (TNF). They show that when TNF oscillations have a low amplitude, NF- κ B oscillations are periodic. We simulated periodic NF- κ B oscillations using Heltberg and colleagues’ Matlab code, available at <https://github.com/MathiasHeltberg/ChaoticDynamicsInTranscriptionFactors>.

Chaotic NF- κ B transcription. Heltberg and colleagues¹⁴ show that by increasing the amplitude of the TNF signal, the oscillating number of NF- κ B molecules in their model becomes chaotic. We simulated chaotic NF- κ B oscillations using Heltberg and colleagues’ Matlab code, available at <https://github.com/MathiasHeltberg/ChaoticDynamicsInTranscriptionFactors>.

Non-biological simulations. Because there are only a limited number of biological simulations for which the presence of chaos has already been established, we also applied the Chaos Decision Tree Algorithm to a wide range of mathematical systems previously studied in the chaos theory and nonlinear time-series analysis literatures:

Chaotic cubic map. Venkatesan and Lakshmanan⁵⁰ describe a quasiperiodically forced cubic map, which can exhibit a large diversity of periodic, chaotic, and strange non-chaotic dynamics. In particular, the map exhibits many different routes to chaos. Their system is described by the following:

$$\begin{aligned} x_{i+1} &= Q + f \cos(2\pi\theta_i) - Ax_i + x_i^3 \\ \theta_{i+1} &= \theta_i + \omega \pmod{1}, \end{aligned} \tag{8}$$

where $\omega = \frac{\sqrt{5}-1}{2}$ (the golden ratio). We set $f = -0.8$, $Q = 0$, and $A = 1.5$, which Venkatesan and Lakshmanan have shown lead to chaotic dynamics⁵⁰. For the results reported in Table 2, we followed Dawes and Freeland²³ in taking a linear combination of \mathbf{x} and θ : $\phi_i = x_i/6 + \theta_i/10$. Results for \mathbf{x} individually are reported in Supplementary Table 15 (results for θ on its own are not informative, as θ is an independent, quasi-periodic process).

Periodic cubic map. Venkatesan and Lakshmanan⁵⁰ show that the system in Eq. (8) exhibits periodic (one-frequency torus) dynamics when $f = 0$, $Q = 0$, and $A = 1$. We picked these parameters for periodic dynamics. To get a time-series ϕ from the cubic map, we again took a linear combination of \mathbf{x} and θ : $\phi_i = x_i/6 + \theta_i/10$. Results for \mathbf{x} and θ individually are reported in Supplementary Table 15.

Strange non-chaotic cubic map (HH). We set $f = 0.7$, $Q = 0$, and $A = 1.88697$ for one type of strange non-chaotic dynamics, which Venkatesan and Lakshmanan⁵⁰ have shown bring the forced cubic map into a strange non-chaotic regime via the Heagy-Hammill route (i.e., collision of a period-doubled quasi-periodic torus with its unstable parent). Results for \mathbf{x} individually are reported in Supplementary Table 15.

Strange non-chaotic cubic map (S3). We set $f = 0.35$, $Q = 0$, and $A = 0.35$ for a second type of strange non-chaotic dynamics, which Venkatesan and Lakshmanan⁵⁰ have shown push the forced cubic map into a strange non-chaotic regime via Type-3 Intermittency (i.e., inverse period-doubling bifurcation). Results for \mathbf{x} individually are reported in Supplementary Table 15.

Period-doubled cubic map. Venkatesan and Lakshmanan⁵⁰ show that the system in Eq. (8) exhibits period-doubled dynamics when $f = -0.18$, $Q = 0$, and $A = 1.1$. We picked these parameters for period-doubled dynamics. To get a time-series ϕ from the cubic map, we again took a linear combination of \mathbf{x} and θ : $\phi_i = x_i/6 + \theta_i/10$. Results for \mathbf{x} individually are reported in Supplementary Table 15.

Strange non-chaotic GOPY model. The first known strange non-chaotic system was described by Grebogi, Ott, Pelikan, and Yorke, commonly referred to as the GOPY model⁵¹. The GOPY model is described by the following:

$$\begin{aligned} x_{i+1} &= 2\sigma(\tanh x_i) \cos(2\pi\theta_i) \\ \theta_{i+1} &= \theta_i + \omega \pmod{1}, \end{aligned} \tag{9}$$

where $\sigma = 1.5$, $\theta = 0.5$, and $\omega = \frac{\sqrt{5}-1}{2}$ (the golden ratio). To get a time-series ϕ from the GOPY model, we followed Dawes and Freeland²³ in taking a linear combination of \mathbf{x} and θ : $\phi_i = x_i/6 + \theta_i/10$. Results for \mathbf{x} individually are reported in Supplementary Table 15 (as is the case for the cubic map, results for θ on its own are not informative, as θ is an independent, quasi-periodic process).

Chaotic logistic map. The logistic map is one of the simplest known systems that can exhibit both periodic and chaotic behavior. It was originally introduced by biologist Robert May⁵² as a discrete-time model of population growth. It is described by the following equation:

$$x_{i+1} = rx_i(1 - x_i) \tag{10}$$

where x_i represents the ratio of the population size at time i to the maximum possible population size. For chaotic dynamics⁵², we set $r = 4$.

Periodic logistic map. For periodic dynamics⁵² in the logistic map, we set $r = 3.5$ in Eq. (10).

Chaotic Lorenz system. Perhaps the most famous of all chaotic systems, the Lorenz model of atmospheric convection is described by the following system of equations⁵³:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \tag{11}$$

where \mathbf{x} is the rate of convection, \mathbf{y} is the horizontal temperature variation, and \mathbf{z} is the vertical temperature variation. Though the equations were initially meant to model atmospheric convection, identical equations have been found in models of a wide variety of physical systems, including lasers⁷⁹ and chemical reactions⁸⁰. We set $\sigma = 10$, $\rho = 30$, and $\beta = \frac{8}{3}$, for which the Lorenz system exhibits chaos (determined by calculating the largest Lyapunov exponent of the system with these parameters, using Ramasubramanian’s algorithm⁸¹). We integrated the equations for the Lorenz system using the Fourth Order Runge-Kutta method with an integration step of 0.01. To get a single time-series ϕ from the Lorenz model, we took a linear combination of \mathbf{x} and \mathbf{y} : $\phi = \mathbf{x} + \mathbf{y}$. Results for \mathbf{x} , \mathbf{y} , and \mathbf{z} individually are reported in Supplementary Table 15.

Hyperchaotic generalized Henon map. Data from hyperchaotic systems, which contain more than one positive Lyapunov exponent, can be difficult to distinguish from noise²⁵. As such, hyperchaotic systems present a challenge to tests of determinism from time-series data, which might mistake hyperchaos for stochasticity. To demonstrate the robustness of the Chaos Decision Tree Algorithm’s stochasticity test, we analyzed the Generalized Henon Map, which is described by the following equation:

$$x_{i+1} = a - x_{i-1}^2 - bx_{i-2} \tag{12}$$

We set $a = 1.76$ and $b = 0.1$, for which the Generalized Henon Map produces hyperchaos⁵⁴.

Noise-driven sine map. Freitas and colleagues⁵⁵ describe a non-chaotic, randomly driven system:

$$x_{i+1} = \mu \sin(x_i) + Y_i \eta_i \tag{13}$$

where $\mu = 2.4$, Y_i is a random Bernoulli process that equals 1 with probability 0.01 and 0 with probability 0.99, and η_i is a random variable uniformly distributed between -2 and 2 . Freitas and colleagues show that a chaos-detection technique called “noise titration”⁸² incorrectly classifies this system as chaotic.

Freitas map. Freitas and colleagues⁵⁵ also describe a nonlinear correlated noise process, which we here call the “Freitas map.” The Freitas map contains dynamic noise added to a nonlinear moving average filter:

$$v_{i+1} = 3v_i + 4v_{i-1}(1 - v_i) \tag{14}$$

where v_n is a uniform random variable distributed between 0 and 1. Freitas and colleagues show that the noise titration technique also incorrectly classifies this system as chaotic.

Bounded random walk. Nicolau⁵⁶ describes a bounded random walk (BRW), which is a globally stationary process with local unit roots (i.e. local non-stationarities):

$$X_t = X_{t-1} + e^k(e^{-\alpha_1(X_{t-1}-\tau)} - e^{\alpha_2(X_{t-1}-\tau)}) + \sigma_t \epsilon_t \tag{15}$$

where τ , k , α_1 , α_2 , and σ are parameters, and ϵ_t is a white noise error term. Note that the bounded random walk can be decomposed into a random walk, $X_t = X_{t-1} + \sigma_t \epsilon_t$, plus an adjustment function $e^k(e^{-\alpha_1(X_{t-1}-\tau)} - e^{\alpha_2(X_{t-1}-\tau)})$. The adjustment function serves to pull the random walk toward τ if the process deviates too far from τ . Though it is a stationary process (albeit with local non-stationarities), the bounded random walk is often mis-classified as non-stationary by stationarity tests⁸³. Following Nicolau⁵⁶ and Patterson⁸³, we set $\tau = 100$, $k = -15$, $\alpha_1 = 3$, $\alpha_2 = 3$, and $\sigma = 0.4$, which generates a random walk that remains roughly within the interval of 100 ± 5 .

Cyclostationary process. A cyclostationary autoregressive process is essentially a combination of a noise-driven linear damped oscillator and linear relaxators. Cyclostationary systems are non-stationary because their probability distributions vary cyclically with time. Following Timmer⁵⁷, we simulate a cyclostationary process described by the following:

$$X_t = a_1 X_{t-1} + a_2 X_{t-2} + \epsilon_t \tag{16}$$

where ϵ_t is a white noise error term and

$$\begin{aligned} a_1 &= 2 \cos(2\pi/T) e^{-1/\tau} \\ a_2 &= -e^{-2/\tau} \end{aligned} \tag{17}$$

τ is the relaxation time and T is the oscillation period. We set $\tau = 50$ and $T = 10$, which Timmer has shown leads to incorrect classification of this system as nonlinear or deterministic by surrogate tests that only use Fourier-based surrogates, which are strictly stationary.

ARMA process. A general autoregressive moving-average (ARMA) process is described by the following:

$$X_t = c + \epsilon_t + \sum_{i=1}^p \phi_i X_{t-i} + \sum_{i=1}^q \theta_i \epsilon_{t-i} \tag{18}$$

where c is a constant, ϕ_1, \dots, ϕ_p and $\theta_1, \dots, \theta_p$ are parameters, and $\epsilon_t, \epsilon_{t-1}, \dots$ are white noise error terms. An ARMA process with a lag of 1, or an ARMA(1) process, is:

$$X_t = c + \epsilon_t + \phi X_{t-1} + \theta \epsilon_{t-1} \tag{19}$$

When $\phi < 1$, the ARMA process is (weakly) stationary. When ϕ is close to but less than 1 and $\theta \neq 0$, ARMA processes, though stationary, are often mis-classified as non-stationary by stationarity tests⁸⁴. All ARMA processes simulated in this paper were lag 1, and we set $c = 0$ and $\phi = 0.99$. For the analyses in Supplementary Tables 1–4 and in Table 2, we drew θ from a random, normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$ for each simulation. We also tested ARMA(1) processes for θ values fixed at $-0.5, 0, 0.5$, and 0.9 (Supplementary Table 19).

Random walk. A random walk is modeled by an autoregressive process with a unit root:

$$X_t = X_{t-1} + \epsilon_t \tag{20}$$

where ϵ is a white noise error term with mean $\mu = 0$ and standard deviation $\sigma = 1$. Random walks are non-stationary.

Trended random walk. A trended random walk introduces a secondary non-stationarity, namely, a linear trend, to the random walk:

$$X_t = X_{t-1} + b + \epsilon_t \tag{21}$$

where ϵ is a white noise error term with mean $\mu = 0$ and standard deviation $\sigma = 1$ and b is the slope of the linear trend. For all trended random walks simulated in this paper, b was randomly drawn from a Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = 0.01$.

Colored noise. Colored noise refers to a stationary, stochastic process with a non-uniform power spectrum. White noise has a uniform power spectrum (meaning equal power at all frequencies); pink noise has a power spectral density proportional to $\frac{1}{f}$, where f is frequency; red noise has a power spectral density proportional to $\frac{1}{f^2}$; blue noise has a power spectral density proportional to f ; and violet noise has a power spectral density proportional to f^2 . All colored noise signals were simulated using Zhivomirov’s algorithm⁵⁸, available at <https://www.mathworks.com/matlabcentral/fileexchange/42919-pink-red-blue-and-violet-noise-generation-with-matlab>.

Rössler system. The Rössler system is described by the following system of differential equations⁵⁹:

$$\begin{aligned} \frac{dx}{dt} &= -wy - z \\ \frac{dy}{dt} &= wx + ay \\ \frac{dz}{dt} &= b + z(x - c) \end{aligned} \tag{22}$$

We set $a = 0.2$, $b = 0.2$, and $c = 5.7$, for which the Rössler system exhibits chaos⁵⁹. w controls the frequency of the system’s oscillations, and was set to 1. We integrated the equations for the Rössler system using the Fourth Order Runge-Kutta method with an integration step of 0.01. We generated 5,000,000 time-points, and then downsampled to 10,000 datapoints. To get a single time-series ϕ from the Rössler model, we took a linear combination of \mathbf{x} and \mathbf{y} : $\phi = \mathbf{x} + \mathbf{y}$. Results for \mathbf{x} , \mathbf{y} , and \mathbf{z} individually are reported in Supplementary Table 15.

Ikeda map. Ikeda and colleagues described a chaotic model of light passing through a nonlinear optical resonator⁸⁵. The model can be simplified into a two-dimensional map⁸⁶:

$$\begin{aligned} x_{i+1} &= 1 + u(x_i \cos t_i - y_i \sin t_i) \\ y_{i+1} &= u(x_i \sin t_i - y_i \cos t_i) \end{aligned} \tag{23}$$

where u is a parameter and

$$t_i = 0.4 - \frac{6}{1 + x_i^2 + y_i^2} \tag{24}$$

We set $u = 0.9$, for which the Ikeda map exhibits chaos⁸⁶. Table 3 reports results for a linear combination of the two variables, $\phi = \mathbf{x} + \mathbf{y}$. Results for \mathbf{x} and \mathbf{y} individually are reported in Supplementary Table 15.

Hénon map. The Hénon map⁶¹ is a two-dimensional system of equations:

$$\begin{aligned} x_{i+1} &= 1 - ax_i^2 + y_i \\ y_{i+1} &= bx_i \end{aligned} \tag{25}$$

We set $a = 1.25$ and $b = 0.3$, for which the Hénon map is periodic⁸⁷. Table 3 reports results for a linear combination of the two variables, $\phi = \mathbf{x} + \mathbf{y}$. Results for \mathbf{x} and \mathbf{y} individually are reported in Supplementary Table 15.

Periodic Poincaré oscillator. The Poincaré oscillator has been widely studied as a model of biological oscillations, particularly as a model of the effect of periodic stimulation on the dynamics of biological oscillators⁸⁸. The oscillator is described by the following equations:

$$\begin{aligned}\frac{dr}{dt} &= kr(1-r) \\ \frac{d\phi}{dt} &= 2\Phi\end{aligned}\quad (26)$$

where k is a positive value that controls the oscillator's relaxation rate. The phase of this system is described by its angular coordinate ϕ in a unit cycle. Periodic stimulation of the system is modeled as a perturbation of magnitude b away from the unit cycle, which leads to an instantaneous resetting of the phase of the oscillator, as determined by the following phase resetting curve:

$$g(\phi) = \frac{1}{2\pi} \arccos \frac{\cos 2\pi\phi + b}{\sqrt{1 + b^2 + 2b \cos 2\pi\phi}} \pmod{1} \quad (27)$$

The period of the stimulation is determined by a parameter τ . For periodic dynamics, we analyze the time-varying phase of the Poincaré oscillator with $b = 1.13$ and $\tau = 0.69$, which Guevara and Glass show leads to phase locking between the oscillator and the periodic perturbations⁴¹.

Quasi-periodic Poincaré oscillator. For quasi-periodic dynamics⁴¹ in the Poincaré oscillator, we set $b = 0.95$ in Eq. (20), with an inter-stimulus delay $\tau = 0.75$.

Chaotic Poincaré oscillator. For chaotic dynamics⁴¹ in the Poincaré oscillator, we set $b = 1.13$ and $\tau = 0.65$.

Stochastic Lorenz system. To study the effect of dynamic noise on our algorithm's classification of stochastic chaotic systems, we took the Lorenz system described in Eq. (11), and added intrinsic/dynamic Gaussian noise to the x component of the system (we found that the system was far less sensitive to noise being injected into the y variable):

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x) + A\eta \\ \frac{dy}{dt} &= x(\rho-z) - y \\ \frac{dz}{dt} &= xy - \beta z\end{aligned}\quad (28)$$

where η_i is a normally distributed random variable with mean 0 and standard deviation 1, and A is a parameter that controls the amplitude of the dynamic/intrinsic noise. As for the deterministic case, we set $\sigma = 10$, $\rho = 30$, and $\beta = \frac{8}{3}$. The stochastic Lorenz system was simulated using the Fourth Order Runge-Kutta method with an integration step of 0.01. Supplementary Table 18 reports results for different values of A , both for all system variables individually and for the linear combination $\mathbf{x} + \mathbf{y}$.

Stochastic Rössler system. We also took the Rössler system described in Eq. (22), and added dynamical Gaussian noise to the x component of the system:

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x) + A\eta \\ \frac{dy}{dt} &= x(\rho-z) - y \\ \frac{dz}{dt} &= xy - \beta z\end{aligned}\quad (29)$$

where η_i is a normally distributed random variable with mean 0 and standard deviation 1, and A is a parameter that controls the amplitude of the dynamic/intrinsic noise. As for the deterministic case, we set $a = 0.2$, $b = 0.2$, and $c = 5.7$, and simulated the model using the Fourth Order Runge-Kutta method with an integration step of 0.01. We generated 5,000,000 time-points, and then downsampled to 10,000 datapoints. Supplementary Table 18 reports results for different values of A , both for all system variables individually and for the linear combination of $\mathbf{x} + \mathbf{y}$.

Multivariate AR model. We generated random multivariate autoregressive (AR) models using the Multivariate Granger Causality (MVG) toolbox⁸⁹. To create random regression matrices, we created random 5-node dense positive definite matrices using Matlab's `sprandsym.m` function, with a graph density of 1. To ensure stationary dynamics, we used the MVG toolbox's `var_specrad.m` function to decay the coefficients of the random dense positive definite matrices so that their spectral radii were 0.8. To ensure uncorrelated noise in the resulting AR model, we created error matrices with diagonal elements set to 1 and off-diagonal elements set to 0. We then inputted these regression and error matrices into the MVG toolbox's `var_to_tsdata.m` function to create multivariate time-series with 5 nodes and 10,000 time-points. We then applied the Chaos Decision Tree Algorithm to just the univariate activity of the first node of the resulting multivariate signal.

Empirical data. We here describe the real-world data analyzed in this paper, and how these data were previously classified as stochastic, periodic, or chaotic:

A chaotic neuron integrated circuit. Data recorded from an integrated circuit were kindly sent to us by Seiji Uenohara and colleagues. The circuit is a physical implementation of a chaotic neuron model that is based on the Hodgkin-Huxley equations⁹⁰. The equations governing the circuit's behavior can be reduced to the following two-dimensional map:

$$\begin{aligned}\zeta(t+1) &= k_r \zeta(t) + af(\zeta(t) + b \cos(2\pi\theta(t))) + a \\ \theta(t+1) &= \theta(t) + \omega \pmod{1}\end{aligned}\quad (30)$$

where $f(\cdot)$ is a monotonically decreasing nonlinear output function, b controls the amplitude of the quasi-periodic forcing, and $\omega = \frac{\sqrt{5}-1}{2}$ (the golden ratio). The quasi-periodic external forcing was inputted to the circuit using an analog board PXI-6289, which was also used to record the circuit's output. Varying the parameter b can bring the circuit into periodic, strange non-chaotic, and chaotic states, which Uenohara and colleagues were able to classify by analyzing the consistency of the circuit's response to an external input²⁷. There are 10 datasets recorded from the circuit's chaotic state.

A strange non-chaotic neuron integrated circuit. There are 10 datasets recorded from the strange non-chaotic state of Uenohara and colleagues' circuit.

A periodic neuron integrated circuit. There are 10 datasets recorded from the periodic state of Uenohara and colleagues' circuit.

Chaotic laser. Hübner and colleagues²⁸ used phase portrait, correlation integral, and autocorrelation function analyses to detect chaos in the intensity pulsing of an unidirectional far-infrared NH3 ring laser. Laser data were downloaded from <https://www.pdx.edu/biomedical-signal-processing-lab/chaotic-time-series>.

Stellar flux of a strange non-chaotic star. We analyzed stellar flux from KIC 5520878, the only known non-artificial strange non-chaotic system²⁹. Data were sent to us by John F. Lindner, who, together with colleagues, determined the status of KIC 5520878 as a strange non-chaotic system using a series of spectral scaling analyses²⁹. Data were originally obtained from the Mikulski Archive for Space Telescopes. Because there are large shifts in the data due to the stellar flux being recorded in different pixels of the Kepler Space Telescope, we visually inspected the data to find a relatively stable period (i.e. a period in between large shifts) and then detrended the data. We thus exclusively analyzed time points 11,620 to 14,003 from the dataset analyzed in Lindner and colleagues' paper²⁹.

North Atlantic Oscillation Index. We analyzed the monthly mean North Atlantic Oscillation (NAO) Index from January 1950 to December 2018. The NAO Index is the difference in atmospheric pressure at sea level between the Azores high and the Icelandic low, and has been shown by several groups of researchers, employing a range of techniques, to be stochastic^{25,91-96}. Data were downloaded from the Climate Prediction Center website (<http://www.cpc.ncep.noaa.gov/>).

Parkinson's tremor. We analyzed recordings of a Parkinson's patient's hand acceleration, measured for 30 s at a sampling rate of 1000 Hz by piezoresistive accelerometers. Through analyses of correlation integrals, Poincaré and return maps, Lyapunov exponents, and the δ - ϵ method, Timmer and colleagues⁵⁷ showed that this Parkinson's tremor was a nonlinear stochastic oscillator. Data were downloaded from http://jeti.uni-freiburg.de/path_tremor/readme.

Essential tremor. We analyzed recordings of hand acceleration from a patient with an essential tremor, also measured for 30 seconds at a sampling rate of 1000 Hz by piezoresistive accelerometers. As they did with the Parkinson's tremor, Timmer and colleagues used correlation integrals, Poincaré and return maps, Lyapunov exponents, and the δ - ϵ method to show that this essential tremor was a nonlinear stochastic oscillator. Data were downloaded from http://jeti.uni-freiburg.de/path_tremor/readme.

Heart rate (healthy subjects). Five heart beat (RR-interval) time-series recordings from healthy subjects were downloaded from Physionet³¹: <https://www.physionet.org/challenge/chaos/>. The signals were recorded using continuous ambulatory (Holter) electrocardiograms, and are in sinus rhythm. Outliers were filtered out of the data using Physionet's WFDB software package. Though a full 24 h of data were available for each subject, we only took the first 2.78 h of data, corresponding to 10,000 time-points. This was both to save on computation time and to be consistent with the length of other time-series analyzed in this paper.

Heart rate (congestive heart failure patients). Five heart beat (RR-interval) time-series recordings from patients with congestive heart failure were downloaded from Physionet³¹. Like the healthy rate signals, these data were recorded using continuous ambulatory (Holter) electrocardiograms, and are in sinus rhythm, and were filtered for outliers. Though a full 24 h of data were available for each subject, we only took the first 2.78 h of data.

Heart rate (atrial fibrillation). Five heart beat (RR-interval) time-series recordings from patients with congestive heart failure were downloaded from Physionet³¹. Like the healthy rate signals, these data were recorded using continuous ambulatory (Holter) electrocardiograms and were filtered for outliers, but are not in sinus rhythm. We only took the first 2.78 h of data, corresponding to 10,000 time-points.

Parameters and largest Lyapunov exponents for data in Table 5. We here describe the methods used to generate data with different degrees of chaos for the analyses reported in Table 5, as well as the methods used to calculate largest Lyapunov exponents in these systems.

Logistic map. The logistic map has only a single parameter, r (see above). Following Bandt and Pompe³², we varied r between 3.5 and 4, in intervals of 0.001,

to generate 501 10,000 time-point signals with different levels of chaos. Ground-truth largest Lyapunov exponents were calculated using the derivative method, which does not involve generating time-series data.

Hénon map. To generate different degrees of chaos in the Hénon map, we varied its a parameter (see above) between 1 and 1.4, in intervals of 0.001, to generate 401 10,000 time-point signals with different degrees of chaos. Ground-truth largest Lyapunov exponents were calculated using code provided in *Dynamical Systems with Applications using Matlab*⁹⁷, available at <https://github.com/springer-math/Dynamical-Systems-with-Applications-using-MATLAB/>.

Lorenz system. For the Lorenz system, we varied its σ parameter between 5.75 and 15, in intervals of 0.05, and generated 10,000 time-points per simulation. Within this parameter range, the Lorenz system is chaotic, but displays varying degrees of chaos. To calculate largest Lyapunov exponents for each parameter, we used the algorithm provided by Ramasubramanian⁸¹, which, like the algorithms used for the logistic and Hénon maps, does not involve generating time-series data.

Mean-field cortical model. Following Steyn-Ross, Steyn-Ross, and Sleight²⁰, different levels of chaos in their mean-field cortical model were generated by varying two parameters: postsynaptic inhibitory response and inhibitory diffusion. The postsynaptic inhibitory response parameter (λ_i in their model) was varied between 0.98 and 1.018 in intervals of 0.001, and the inhibitory diffusion parameter (D_2 in their model) was varied between 0.1 and 0.8 in intervals of 0.05, producing a total of 585 parameter configurations. We simulated 25,000 time-points in the model, with no downsampling, so we could again test the effect of the Chaos Decision Tree Algorithm's automated downsampling on permutation entropy's ability to track level of chaos in continuous systems. Unfortunately, there are no tools for analytically estimating the largest Lyapunov exponent of the mean-field cortical model, and so largest Lyapunov exponents were approximated by running two noise-free simulations of the model for each parameter configuration, with very slightly different initial conditions, and fitting a line to the rate of divergence between the two simulations from the beginning of the simulations to the point when their divergence rate saturates and flattens out. The slope of the fitted line is taken as the estimate of the largest Lyapunov exponent¹⁰. While Steyn-Ross, Steyn-Ross, and Sleight's code for estimating largest Lyapunov exponents using this method requires subjective evaluation of where to fit the line (i.e. finding the non-saturated part of the divergence rate plot), we automated this process by fitting a line from the beginning of the divergence rate plot to the point where its mean abruptly changes, reflecting saturation; this point was determined using Matlab's `findchangepts.m` function. Simulations with approximated largest Lyapunov exponents less than -5 or greater than 5 were excluded from the analysis, as these are likely poor estimates; visual inspection confirmed that for these cases, there was often no clear point of saturation for the divergence rate between simulations, and so lines were often automatically fitted to particularly steep, short sub-segments of the plot. Visual inspection further confirmed that in most other cases, there was a clear, linear rate of divergence between simulations followed by saturation, and that the automatically fitted line was a good fit.

Neuron integrated circuit. The integrated circuit data analyzed in Table 5 are the same as those analyzed in Table 4. Because the circuit is a physical implementation of a known and simple two-dimensional system of equations, Uenuhara and colleagues used those equations to calculate the ground-truth largest Lyapunov exponents of the circuit in its three different states (periodic, strange non-chaotic, and chaotic), and report those largest Lyapunov exponents in their paper²⁷.

Statistics and reproducibility. In reporting the performance of the Chaos Decision Tree Algorithm, we only made recourse to statistical tests in Table 5 and Supplementary Table 14, where we report the (two-tailed) p -values of Spearman correlations between largest Lyapunov exponents and permutation entropies. Because these correlations were calculated against the same set of ground-truth largest Lyapunov exponents for each system, p -values were Bonferroni-corrected for multiple comparisons. Elsewhere, we only report the fraction out of all datasets that were correctly classified as stochastic, periodic, or chaotic (with exact sample sizes provided in all tables). Within the Chaos Decision Tree Algorithm itself, statistical tests appear in two locations. First, if the user chooses to test for stationarity, then the pipeline will only proceed to test for stochasticity if the test signal passes a (two-tailed) stationarity test with $\alpha = 0.05$. Moreover, the pipeline uses surrogate statistics to test for determinism: if the permutation entropy of the inputted signal lies outside the distribution of permutation entropies of 1000 surrogates (which is equivalent to a two-tailed statistical test with $\alpha = 9.99e-4$), then the data are classified as being generated by a predominantly deterministic system. No sample size calculation was performed before analyzing the data presented, as our results simply report classification accuracy for a large number of datasets (and no statistical analyses were performed beyond reporting classification accuracy). No data were excluded from the analysis. Data in Tables 1, 2 were used to optimize our algorithm, which was then re-tested on datasets in Tables 3, 4. Finally, note that no p -value is associated with the K -statistic outputted by the 0–1 test for chaos. To facilitate the reproducibility of our analyses, we have included code alongside our pipeline, as well as links to code in our Methods, for generating all simulated datasets tested in this paper. Links to most empirical datasets analyzed here have been provided in the Methods.

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Links to Matlab scripts for simulating the mean-field cortical model and NF- κ B transcription are provided in the Methods. Matlab scripts for simulating all other systems described in this paper are provided along with the code for the Chaos Decision Tree Algorithm at <https://figshare.com/s/80891dfb34c6ee9c8b34> (DOI: doi.org/10.6084/m9.figshare.7476362.v7). All empirical datasets except for the integrated circuit recordings are freely available online, and URLs to each data source are provided in the Methods.

Code availability

The code for the Chaos Decision Tree Algorithm is provided at <https://figshare.com/s/80891dfb34c6ee9c8b34> (DOI: doi.org/10.6084/m9.figshare.7476362.v7).

Received: 29 March 2019; Accepted: 26 November 2019;

Published online: 03 January 2020

References

- Linde, A. D. Eternally existing self-reproducing chaotic inflationary universe. *Phys. Lett. B* **175**, 395–400 (1986).
- Wisdom, J., Peale, S. J. & Mignard, F. The chaotic rotation of hyperion. *Icarus* **58**, 137–152 (1984).
- Hudson, J. & Mankin, J. Chaos in the belousov-zhabotinskii reaction. *J. Chem. Phys.* **74**, 6171–6177 (1981).
- Kaplan, D. T. et al. Subthreshold dynamics in periodically stimulated squid giant axons. *Phys. Rev. Lett.* **76**, 4074 (1996).
- Glass, L. Introduction to controversial topics in nonlinear science: Is the normal heart rate chaotic? *Chaos* **19**, 028501 (2009).
- Goldberger, A. L. & West, B. J. Applications of nonlinear dynamics to clinical cardiology. *Ann. N. Y. Acad. Sci.* **504**, 195–213 (1987).
- Skarda, C. A. & Freeman, W. J. How brains make chaos in order to make sense of the world. *Behav. Brain Sci.* **10**, 161–173 (1987).
- Pool, R. Is it healthy to be chaotic? *Science* **243**, 604–607 (1989).
- Korn, H. & Faure, P. Is there chaos in the brain? ii. experimental evidence and related models. *C. R. Biol.* **326**, 787–840 (2003).
- Destexhe, A. Oscillations, complex spatiotemporal behavior, and information transport in networks of excitatory and inhibitory neurons. *Phys. Rev. E* **50**, 1594 (1994).
- Lizier, J.T., Prokopenko, M. & Zomaya, A.Y. The information dynamics of phase transitions in random boolean networks. In *ALIFE*, 374–381 (2008).
- Nagao, N., Nishimura, H. & Matsui, N. A neural chaos model of multistable perception. *Neural Process. Lett.* **12**, 267–276 (2000).
- Rabinovich, M. & Abarbanel, H. The role of chaos in neural systems. *Neuroscience* **87**, 5–14 (1998).
- Heltberg, M. L., Krishna, S. & Jensen, M. H. On chaotic dynamics in transcription factors and the associated effects in differential gene regulation. *Nat. Commun.* **10**, 71 (2019).
- Glass, L. & Mackey, M.C. *From clocks to chaos: The rhythms of life* (Princeton University Press, 1988).
- Guevara, M. R., Glass, L. & Shrier, A. Phase locking, period-doubling bifurcations, and irregular dynamics in periodically stimulated cardiac cells. *Science* **214**, 1350–1353 (1981).
- Hayashi, H. & Ishizuka, S. Chaotic nature of bursting discharges in the onchidium pacemaker neuron. *J. Theor. Biol.* **156**, 269–291 (1992).
- Hayashi, H., Nakao, M. & Hirakawa, K. Chaos in the self-sustained oscillation of an excitable biological membrane under sinusoidal stimulation. *Phys. Lett. A* **88**, 265–266 (1982).
- Mackey, M. C. & Glass, L. Oscillation and chaos in physiological control systems. *Science* **197**, 287–289 (1977).
- Steyn-Ross, M. L., Steyn-Ross, D. A. & Sleight, J. W. Interacting Turing-Hopf instabilities drive symmetry-breaking transitions in a mean-field model of the cortex: a mechanism for the slow oscillation. *Phys. Rev. X* **3**, 021005 (2013).
- Toker, D. The chaos decision tree algorithm. <https://doi.org/10.6084/m9.figshare.7476362.v7> (2019).
- Chan, K.-S. & Tong, H. *Chaos: a statistical perspective* (Springer Science & Business Media, 2013).
- Dawes, J. H. P. & Freeland, M. C. The '0–1 test for chaos' and strange nonchaotic attractors. *preprint* <https://people.bath.ac.uk/jhp20/publications/sna.pdf> (2008).
- Hu, J., Tung, W.-w., Gao, J. & Cao, Y. Reliability of the 0-1 test for chaos. *Phys. Rev. E* **72**, 056207 (2005).

25. Kulp, C. & Zunino, L. Discriminating chaotic and stochastic dynamics through the permutation spectrum test. *Chaos: an Interdisciplinary. J. Nonlinear Sci.* **24**, 033116 (2014).
26. Timmer, J., Häußler, S., Lauk, M. & Lücking, C.-H. Pathological tremors: deterministic chaos or nonlinear stochastic oscillators? *Chaos: An Interdisciplinary. J. Nonlinear Sci.* **10**, 278–288 (2000).
27. Uenohara, S. et al. Experimental distinction between chaotic and strange nonchaotic attractors on the basis of consistency. *Chaos: an Interdisciplinary. J. Nonlinear Sci.* **23**, 023110 (2013).
28. Huebner, U., Abraham, N. & Weiss, C. Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH_3 laser. *Phys. Rev. A* **40**, 6354 (1989).
29. Lindner, J. F. et al. Strange nonchaotic stars. *Phys. Rev. Lett.* **114**, 054101 (2015).
30. Hurrell, J. W., Kushnir, Y., Ottersen, G. & Visbeck, M. An overview of the North Atlantic Oscillation. *North Atl. Oscillation: Climatic Significance Environ. Impact* **134**, 1–35 (2003).
31. Goldberger, A. L. et al. Physiobank, Physiokit, and Physionet: components of a new research resource for complex physiologic signals. *Circulation* **101**, e215–e220 (2000).
32. Bandt, C. & Pompe, B. Permutation entropy: a natural complexity measure for time series. *Phys. Rev. Lett.* **88**, 174102 (2002).
33. Theiler, J., Eubank, S., Longtin, A., Galdrikian, B. & Farmer, J. D. Testing for nonlinearity in time series: the method of surrogate data. *Phys. D: Nonlinear Phenom.* **58**, 77–94 (1992).
34. Lancaster, G., Iatsenko, D., Pidde, A., Ticcinelli, V. & Stefanovska, A. Surrogate data for hypothesis testing of physical systems. *Phys. Rep.* **748**, 1–60 (2018).
35. Jamšek, J., Paluš, M. & Stefanovska, A. Detecting couplings between interacting oscillators with time-varying basic frequencies: Instantaneous wavelet bispectrum and information theoretic approach. *Phys. Rev. E* **81**, 036207 (2010).
36. Schreiber, T. Extremely simple nonlinear noise-reduction method. *Phys. Rev. E* **47**, 2401 (1993).
37. Gottwald, G. A. & Melbourne, I. in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 460, 603–611 (The Royal Society, 2004).
38. Gottwald, G. A. & Melbourne, I. Testing for chaos in deterministic systems with noise. *Phys. D: Nonlinear Phenom.* **212**, 100–110 (2005).
39. Gottwald, G. A. & Melbourne, I. On the implementation of the 0–1 test for chaos. *SIAM J. Appl. Dynamical Syst.* **8**, 129–145 (2009).
40. Gottwald, G. A. & Melbourne, I. Comment on “reliability of the 0–1 test for chaos”. *Phys. Rev. E* **77**, 028201 (2008).
41. Guevara, M. R. & Glass, L. Phase locking, period doubling bifurcations and chaos in a mathematical model of a periodically driven oscillator: A theory for the entrainment of biological oscillators and the generation of cardiac dysrhythmias. *J. Math. Biol.* **14**, 1–23 (1982).
42. Horio, Y., Aihara, K. & Yamamoto, O. Neuron-synapse IC chip-set for large-scale chaotic neural networks. *IEEE Trans. Neural Netw.* **14**, 1393–1404 (2003).
43. Trostel, M. L., Misplon, M. Z., Aragonese, A. & Pattanayak, A. K. Characterizing complex dynamics in the classical and semi-classical duffing oscillator using ordinal patterns analysis. *Entropy* **20**, 40 (2018).
44. Amigó, J. M. The equality of Kolmogorov-Sinai entropy and metric permutation entropy generalized. *Phys. D: Nonlinear Phenom.* **241**, 789–793 (2012).
45. Baillie, R. T., Cecen, A. A. & Erkal, C. Normal heartbeat series are nonchaotic, nonlinear, and multifractal: New evidence from semiparametric and parametric tests. *Chaos: An Interdisciplinary. J. Nonlinear Sci.* **19**, 028503 (2009).
46. Zhang, J., Holden, A., Monfredi, O., Boyett, M. R. & Zhang, H. Stochastic vagal modulation of cardiac pacemaking may lead to erroneous identification of cardiac “chaos”. *Chaos* **19**, 028509 (2009).
47. Ovchinnikov, I. Introduction to supersymmetric theory of stochastics. *Entropy* **18**, 108 (2016).
48. Letellier, C. & Aguirre, L. A. Investigating nonlinear dynamics from time series: The influence of symmetries and the choice of observables. *Chaos* **12**, 549–558 (2002).
49. Izhikevich, E. M. Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **14**, 1569–1572 (2003).
50. Venkatesan, A. & Lakshmanan, M. Interruption of torus doubling bifurcation and genesis of strange nonchaotic attractors in a quasiperiodically forced map: Mechanisms and their characterizations. *Phys. Rev. E* **63**, 026219 (2001).
51. Grebogi, C., Ott, E., Pelikan, S. & Yorke, J. A. Strange attractors that are not chaotic. *Phys. D: Nonlinear Phenom.* **13**, 261–268 (1984).
52. May, R. M. Simple mathematical models with very complicated dynamics. *Nature* **261**, 459 (1976).
53. Lorenz, E. N. Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963).
54. Richter, H. The generalized henon maps: examples for higher-dimensional chaos. *Int. J. Bifurc. Chaos* **12**, 1371–1384 (2002).
55. Freitas, U. S., Letellier, C. & Aguirre, L. A. Failure in distinguishing colored noise from chaos using the “noise titration” technique. *Phys. Rev. E* **79**, 035201 (2009).
56. Nicolau, J. Stationary processes that look like random walks the bounded random walk process in discrete and continuous time. *Econom. Theory* **18**, 99–118 (2002).
57. Timmer, J. Power of surrogate data testing with respect to nonstationarity. *Phys. Rev. E* **58**, 5153 (1998).
58. Zhivomirov, H. A method for colored noise generation. *Rom. J. Acoust. Vib.* **15**, 14–19 (2002).
59. Rössler, O. E. An equation for continuous chaos. *Phys. Lett. A* **57**, 397–398 (1976).
60. Hammel, S., Jones, C. & Moloney, J. Global dynamical behavior of the optical field in a ring cavity. *JOSA B* **2**, 552–564 (1985).
61. Hénon, M. in *The Theory of Chaotic Attractors*, 94–102 (Springer, 1976).
62. Kulp, C. W. & Smith, S. Characterization of noisy symbolic time series. *Phys. Rev. E* **83**, 026201 (2011).
63. Bernardini, D. & Litak, G. An overview of 0–1 test for chaos. *J. Braz. Soc. Mech. Sci. Eng.* **38**, 1433–1450 (2016).
64. Zunino, L. & Kulp, C. W. Detecting nonlinearity in short and noisy time series using the permutation entropy. *Phys. Lett. A* **381**, 3627–3635 (2017).
65. Riedl, M., Müller, A. & Wessel, N. Practical considerations of permutation entropy. *Eur. Phys. J. Spec. Top.* **222**, 249–262 (2013).
66. Leontitsis, A. Chaotic systems toolbox. <https://www.mathworks.com/matlabcentral/fileexchange/1597-chaotic-systems-toolbox>. (2004).
67. Matthews, P. 0–1 test for chaos. <https://www.mathworks.com/matlabcentral/fileexchange/25050-0-1-test-for-chaos>. (2009).
68. EyébéFouda, J. S. A., Bodo, B., Sabat, S. L. & Effa, J. Y. A modified 0–1 test for chaos detection in oversampled time series observations. *Int. J. Bifurc. Chaos* **24**, 1450063 (2014).
69. Hawkins, R. Permutation entropies and chaos. <https://pdfs.semanticscholar.org/e5d9/65c915e62694a3a8a99863477911cc837992.pdf> (2015).
70. Politi, A. Quantifying the dynamical complexity of chaotic time series. *Phys. Rev. Lett.* **118**, 144101 (2017).
71. Bandt, C., Keller, G. & Pompe, B. Entropy of interval maps via permutations. *Nonlinearity* **15**, 1595 (2002).
72. Keller, K., Unakafov, A. M. & Unakafova, V. A. On the relation of KS entropy and permutation entropy. *Phys. D: Nonlinear Phenom.* **241**, 1477–1481 (2012).
73. Pesin, Y. B. Characteristic Lyapunov exponents and smooth ergodic theory. *Russian Math. Surv.* **32**, 55–114 (1977).
74. Liley, D. T., Cadusch, P. J. & Wright, J. J. A continuum theory of electrocortical activity. *Neurocomputing* **26**, 795–800 (1999).
75. Liley, D. T., Cadusch, P. J. & Dafilis, M. P. A spatially continuous mean field theory of electrocortical activity. *Netw.: Comput. Neural Syst.* **13**, 67–113 (2002).
76. Izhikevich, E. M. Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.* **15**, 1063–1070 (2004).
77. Gatti, A. et al. Cyclically changing phenomenon of cancer cells in an untreated hemophilia patient. *Blood* **41**, 771–773 (1973).
78. Farmer, J. D. Chaotic attractors of an infinite-dimensional dynamical system. *Phys. D: Nonlinear Phenom.* **4**, 366–393 (1982).
79. Haken, H. Analogy between higher instabilities in fluids and lasers. *Phys. Lett. A* **53**, 77–78 (1975).
80. Poland, D. Cooperative catalysis and chemical chaos: a chemical model for the Lorenz equations. *Phys. D: Nonlinear Phenom.* **65**, 86–99 (1993).
81. Ramasubramanian, K. & Sriram, M. A comparative study of computation of Lyapunov spectra with different algorithms. *Phys. D: Nonlinear Phenom.* **139**, 72–86 (2000).
82. Poon, C. S., & Barahona, M. Titration of chaos with added noise. *Proceedings of the national academy of sciences*, **98**, 7107–7112. (2001).
83. Patterson, K. *Unit Root Tests in Time Series Volume 2: Extensions and Developments*, vol. 2 (Palgrave Macmillan, 2012).
84. Patterson, K. *Unit Root Tests in Time Series Volume 1: Key Concepts and Problems* (Springer, 2011).
85. Ikeda, K., Daido, H. & Akimoto, O. Optical turbulence: chaotic behavior of transmitted light from a ring cavity. *Phys. Rev. Lett.* **45**, 709 (1980).
86. Galias, Z. Rigorous investigation of the Ikeda map by means of interval arithmetic. *Nonlinearity* **15**, 1759 (2002).
87. Ivancevic, V. G. & Ivancevic, T. T. *Computational mind: a complex dynamics perspective*, vol. 60 (Springer, 2007).
88. Glass, L. in *Nonlinear dynamics in physiology and medicine*, 123–148 (Springer, 2003).
89. Barnett, L. & Seth, A. K. The MVGC multivariate Granger causality toolbox: a new approach to Granger-causal inference. *J. Neurosci. methods* **223**, 50–68 (2014).

90. Aihara, K., Takabe, T. & Toyoda, M. Chaotic neural networks. *Phys. Lett. A* **144**, 333–340 (1990).
91. Fernández, I., Hernández, C. N. & Pacheco, J. M. Is the North Atlantic Oscillation just a pink noise? *Phys. A: Stat. Mech. its Appl.* **323**, 705–714 (2003).
92. Stephenson, D. B., Pavan, V. & Bojariu, R. Is the North Atlantic Oscillation a random walk? *Int. J. Climatol.* **20**, 1–18 (2000).
93. Collette, C. & Ausloos, M. Scaling analysis and evolution equation of the North Atlantic Oscillation index fluctuations. *Int. J. Mod. Phys. C* **15**, 1353–1366 (2004).
94. Lind, P. G., Mora, A., Haase, M. & Gallas, J. A. Minimizing stochasticity in the NAO index. *Int. J. Bifurc. Chaos* **17**, 3461–3466 (2007).
95. Martínez Santafé, M. D., Lana Pons, F. J., Burgueño, A. & Serra de Larrocha, C. Predictability of the monthly North Atlantic Oscillation index based on fractal analyses and dynamic system theory. *Nonlinear Proc. Geophys.* **17**, 93–101 (2010).
96. Fernández, I., Pacheco, J. M. & Quintana, M. P. Pinkness of the North Atlantic Oscillation signal revisited. *Phys. A: Stat. Mech. its Appl.* **389**, 5801–5807 (2010).
97. Lynch, S. *Dynamical systems with applications using Matlab®* (Springer, 2014).

Acknowledgements

We thank Seiji Uenohara and colleagues for kindly sending us data recorded from their chaotic neuron integrated circuit, and we thank John F. Lindner for sending us data on the stellar flux of KIC 5520878. We also thank Michael Jansson for his feedback on stationarity testing.

Author contributions

D.T. conceived of the project, wrote the code for the Chaos Decision Tree Algorithm, analyzed results, and wrote the manuscript. F.T.S. and M.D. contributed to discussing, editing, and revising the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at <https://doi.org/10.1038/s42003-019-0715-9>.

Correspondence and requests for materials should be addressed to D.T.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020