Address reduction blindly identifies non-random data series

Thomas M. A. Fink,[1,2] Francis Brown,[3] Karen Willbrand,[4] and Sebastian E. Ahnert[1,2]

[1] *Theory of Condensed Matter, Cavendish Laboratory, Cambridge CB3 0HE, UK*
[2] *Institut Curie, CNRS UMR 144, 75005 Paris, France*
[3] *Departement de Mathematique, Ecole Normale Supérieure, 75005 Paris, France*
[4] *Laboratoire de Physique Statistique, Ecole Normale Supérieure, 75005 Paris, France*

We introduce a method of detecting data series (curves) which exhibit pattern without knowing what kind of pattern they contain. By partitioning the space of curves into neighbourhoods, we show that the curves with the shortest addresses are the most likely to result from simple underlying mechanisms. We show that address reduction is a bound on Kolmogorov complexity and is invariant over noise and one-to-one transformations. We use it to blindly identify gene expression profiles in yeast cell cycle and the segmentation clock, and to segregate human- and computer-generated random data.

The traditional basis of hypothesis formation in the physical and social sciences is the identification of trends or pattern in a series of data [1]. Typically, the pattern is incontrovertible and can be encapsulated by a concise mathematical relation between the data and the independent variable. Recently there has been surge of interest in collective systems [2–4], driven, for instance, by the proliferation of high throughput biology and the ability to retrieve market data over many different time scales. Many such systems — such as genetic networks [8, 15], physiological dynamics [4] and financial and social systems [3] — exhibit weak pattern, that is, the pattern does not look significantly different from a random curve. Moreover, because their dynamics are in general not understood beyond a statistical description, it is not clear what kind of pattern to look for.

Our goal in this Letter is twofold: to identify pattern without knowing what kind of pattern we are looking for; and to quantify it in such a way that it is independent of the systems studied and therefore comparable between them. Key to our approach is the assignment of more significance to data with a short description (or theory) than data with a long description. This is Occam's razor—choose the most concise theory that fits the data—in reverse—choose the data described by the most concise theory. It can be formalised using Kolmogorov complexity [6], also known as algorithmic information content [7]: The Kolmogorov complexity of a series of observations is the length in bits of the shortest possible algorithm, or description, of that data given a fixed (and universal) language. The shorter the description of a curve, the more pattern it contains; conversely, a curve whose shortest description is as long as the data itself is said to be random [3, 6, 7].

Our method enables us to approach a large number of data series without preconceptions of what sort of behaviour is significant. It can be used to study any series in which the pattern is faint or clouded by noise, even if the number of data points is small. Moreover it provides us with a universal currency by which we can compare the significance of curves of different lengths, from different experiments or exhibiting different forms of pattern.

The Kolmogorov complexity of a data series is, in general, fundamentally uncomputable [6], and at best we can bound it from above. One way of doing so, and one which we will later see satisfies several invariance properties, is as follows. We first convert a curve to a permutation by relabelling the data points with their ascending rank. We then segregate the permutations into clusters of different sizes with respect to some map: permutations mapped to the same number are assigned to the same cluster. From this we can write an alternative description of any curve, from which the original curve can be fully recovered. The length of this description is a bound on its Kolmogorov complexity. The difference between this bound and the length of the original curve is the number of bits k by which the curve can be reduced. We use the reduction k to order a collection of curves in decreasing order of significance. A curve with a high k is less likely to arise by chance and more likely to be the output of a simple underlying mechanism than curves with low k.

Address length — Without loss of generality, we take a curve f to be composed of N points, each taken from the interval $(0, 1]$ with resolution T, that is, there are T possibilities for each point. (We take T large enough such that no two points i and j are the same; otherwise, faint noise can be added to break any degeneracies.) For $N = 5$ and $T = 100$, f might be, for example, $0.77, 0.84, 0.51, 0.30, 0.26$.

Suppose we wish to store an arbitrary curve f on a computer. The size of the file in bits, or Shannon information [3], is

$$H(f) = -\sum_f T^{-N} \log_2 T^{-N} = \log_2 T^N. \qquad (1)$$

Instead of storing the curve directly, we could write down instructions for generating it, and store this instead. If the size of this file in bits is less than the Shannon infor-

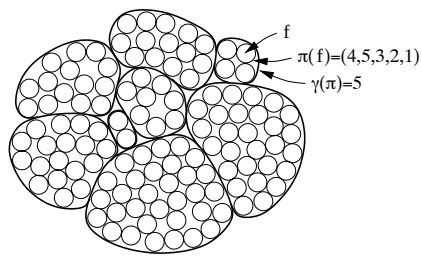


FIG. 1: The space of curves of N points can be equipartitioned into $N!$ equally sized balls (circles) by grouping together curves with the same permutation π. These balls are then segregated into clusters S (bold lines) by collecting circles with the same $\gamma(\pi)$. EXAMPLE: Let $N = 5$ and the curve $x = (0.77, 0.84, 0.51, 0.30, 0.26)$. Then $\pi(x) = (4, 5, 3, 2, 1)$, one of the 5! balls above. If we take the map γ to be the sum of the absolute values of the differences of consecutive points (the first difference operator, Δ_1), then $\gamma(4, 5, 3, 2, 1) = 1 + 2 + 1 + 1 = 5$. The probability that a random curve gives the same value is $P(5) = 4/120$, since 4 balls lie in the $\gamma(\pi) = 5$ cluster. The size of the image of γ, which is the total number of clusters, is $|\mathrm{Im}(\gamma)| = 8$. Then $k(f|\gamma) = \log_2 1/P(\gamma(\pi)) - \log_2 |\mathrm{Im}(\gamma)| = 1.92$ bits.

mation, then the curve is reducible by their difference.

Denote by $\pi(f)$ the permutation of the curve f; this is the permutation formed by replacing each point with its rank when ordered from lowest to highest. (The $N = 5$ example above has permutation $(4, 5, 3, 2, 1)$.) There are many curves with the same permutation π; in the limit of $T \gg N^2$ there are $T^N/N!$ of them. Let γ be a map from permutations to real numbers. We group together permutations π with the same number $\gamma(\pi)$ and call the resulting set of curves $S_{\gamma(\pi)}$. (In Figure 1, permutations are circles and groups of permutations are clusters.)

We encode the curve f in two parts: the coarse address of the cluster $S_{\gamma(\pi)}$ and the fine address of the curve f inside the cluster $S_{\gamma(\pi)}$. The number of bits necessary to store the address of $S_{\gamma(\pi)}$ is $\log_2$ of the total number of sets. The number of sets is simply the size of the image of the map γ, which we denote $|\mathrm{Im}(\gamma)|$. The number of bits necessary to store the address of f inside $S_{\gamma(\pi)}$ is $\log_2$ of the number of curves in $S_{\gamma(\pi)}$. Now the probability that a random curve f takes on the value $\gamma\big(\pi(f)\big)$ is the number of curves in $S_{\gamma(\pi(f))}$ divided by the total number of curves, that is, $P\big(\gamma(\pi(f))\big) = |S_{\gamma(\pi(f))}|/T^N$. Therefore to specify f within $S_{\gamma(\pi(f))}$ requires $\log_2 \big[T^N P\big(\gamma(\pi(f))\big)\big]$ bits. The bound on the Kolmogorov complexity of f is the sum of $\log_2$ of both addresses, that is,

$$I^{\mathrm{bnd}}(f|\gamma) = \log_2 \big[T^N P\big(\gamma(\pi(f)))\big)\big] + \log_2 |\mathrm{Im}(\gamma)|. \quad (2)$$

This means that from a string of length $I^{\mathrm{bnd}}(f|\gamma)$ bits, we can always reconstruct the original curve f.

The total reduction $k(f|\gamma) = H(f) - I^{\mathrm{bnd}}(f|\gamma)$, which by (1) and (2) is

$$k(f|\gamma) = \log_2(1/P\big(\gamma(\pi(f))\big) - \log_2 |\mathrm{Im}(\gamma)|; \quad (3)$$

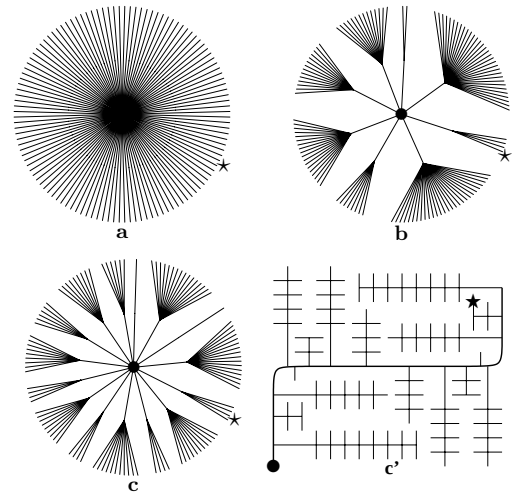


FIG. 2: Describing the rank order (permutation) of a curve is equivalent to navigating from the center of a maze • to some predetermined destination ⋆. In the mazes **a**–**c** the endpoints correspond to the 120 permutations for $N = 5$ points. How much information is needed to be certain of finding the destination? (**a**) In the case of the trivial partition map, it is necessary to choose between 120 paths, and $Q = \log_2 120 = 6.92$ bits; this is equivalent to the Shannon description. (**b**) For the γ_{Δ_1} partition map, one chooses between 8 paths, then 4 paths, requiring $3 + 2 = 5$ bits. (**c**) The map γ_{+-} requires $3.90 + 2 = 5.90$ bits; in (**c'**) the same maze is redrawn as a street map (*cf.* [5]). Thus, the permutation $4, 5, 3, 2, 1$ can be reduced by $0, 1.92$ and 1.02 bits using $\gamma_{\mathrm{trivial}}, \gamma_{\Delta_1}$ and γ_{+-}.

we say that the curve is reducible by at *at least* k bits. This can be expressed differently by noting that $\langle |S_{\gamma(\pi)}| \rangle = T^N/|\mathrm{Im}(\gamma)|$, where $\langle |S_{\gamma(\pi)}| \rangle$ is the average size of the set S. Substituting this into (3) yields

$$k(f|\gamma) = \log_2 \frac{\langle |S_{\gamma(\pi)}| \rangle}{|S_{\gamma(\pi(f))}|}. \quad (4)$$

Only when the size of S is less than its mean is k positive and the curve f reducible. Thus an effective map must partition the space of permutations in such a way that the clusters are of a wide variety of different sizes.

It now remains to choose the partition map γ from permutations to numbers. Different γ correspond to different mazes in Figure 2; *cf.* [5]. Their simplicity is deceptive; they can exhibit intricate mathematical behaviour [10]. Some of the simplest maps are: γ_{long}, the length of the longest increasing subsequence; γ_{opt}, the number of local optima [10]; γ_{+-}, the number of permutations with the same pattern of ups and downs [11]; γ_{Δ_n}, the sum of the absolute value of the nth difference operator, where for example $\gamma_{\Delta_1} = \sum_{i=1}^{N-1} |f_{i+1} - f_i|$ (Figure 1). Other maps can easily be imagined.

Applications — We applied our method of address reduction to three different experimental systems: yeast cell cycle [8], human- versus computer-generated random data [17, 18], and the segmentation clock in developmen-

tal segmentation [15, 16].

First we tested our method on the yeast cell cycle time series of Spellman *et al.* [8], comprising 6073 curves of 18 points sampled over 2 cell cycles. (We also studied experiments over 1 and 3 cycles, which gave qualitatively similar results.) By ordering the curves according to their address reduction, we would expect the top ranked genes to be those most correlated with time and thus (by virtue of the synchronised cell division in the experiment) cell cycle. We were able to verify this by comparing the intersection of our list with the validation set of 140 experimentally determined genes of Simon *et al.* [9]. This is shown in Fig. 3 (left) for five different maps γ; a perfect ordering would appear as a step function. Amongst the early part of our list, almost 40% of the genes were verified by Simon [9]. (Note that how many and which genes are associated with yeast cell cycle remains a partially open question; for example, the intersection of the set of Simon [9] with the independent set set collected by Spellman [8] is 35%.) For our comprehensive investigation of yeast cell cycle using address reduction, see [13].

Second, we attempted to de-mix forged (human-generated) from real (computer-generated) random data. The human data was collected by Towse and co-workers ([17, 18]; private communication). Subjects were asked to generate a random series of 100 numbers between 0 and 9, indicating digits at intervals of 1.5 seconds. We mixed 192 human series of length $N = 100$ with 6000 computer generated random series of the same length. Faint noise was added to both sets of curves so that no two numbers were the same. We ordered curves by their address reduction, with the view that deviations from randomness in the human-generated data series would emerge as increased reducibility. Unlike in the case of yeast, where two-cycle behaviour was anticipated, the sort of pattern implicit in human random number generation is poorly understood [17, 18]. Our result is an almost perfect de-mixing of the human and computer curves. One non-random aspect of the human generated data is an insufficient number of turning points. (But note that we have also identified curves with too many turning points: separate preparation of even and odd samples in yeast

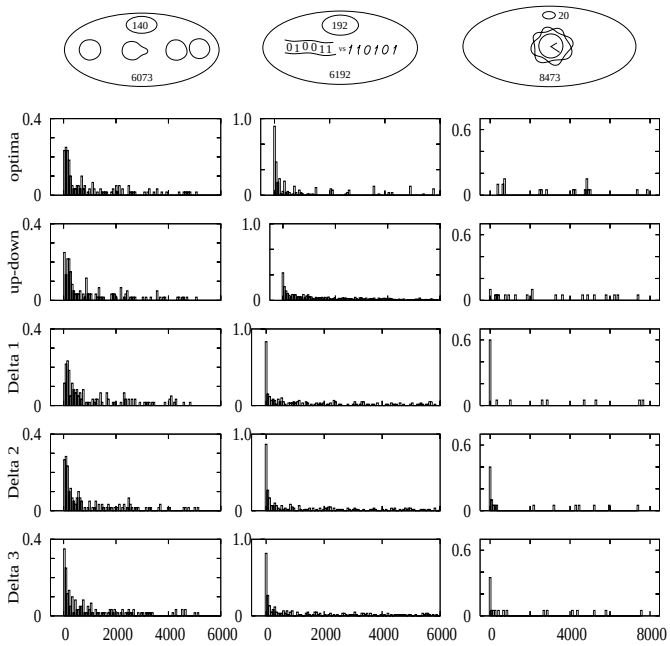
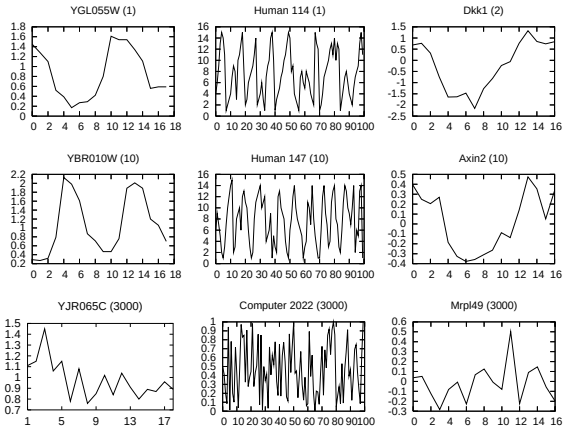


FIG. 3: We used address reduction to order curves from three experiments: (LEFT) yeast cell cycle; (CENTER) human- versus computer-generated random data; and (RIGHT) the developmental segmentation clock. We used five different maps γ: from top, the number of optima ($\gamma_{\rm opt}$); the pattern of ups and downs (γ_{+-}); the sum over the first difference operator (γ_{Δ_1}); the second difference operator (γ_{Δ_2}); and the third (γ_{Δ_3}). In each case, we compare our ordering of the curves with an experimentally derived validation subset: 140 cell cycle genes, out of 6073; 192 human-generated random series, out of 6192 human/computer random series; 20 genes prominent in the segmentation clock, out of 8473. The height of each histogram bar is the fraction of the histogram bin width that is included in the validation set. Note that the yeast and clock validation sets are themselves uncertain, whereas the human-generated data is known with complete confidence.



Yeast cell cycle			*Genetic clock*		
Rank	*Gene*	$k(f)$	*Rank*	*Gene*	$k(f)$
1	YGL055W	11.2	2	Dkk1	6.1
10	YBR010W	8.4	10	Axin2	4.0
3000	YJR065C	-0.8	3000	Mrpl49	-0.8

FIG. 4: TOP: The 1st, 10th and 3000th ranked curves for yeast cell cycle, human- versus computer-generated random data, and the segmentation clock (we show the 2nd ranked gene Dkk1 because of its biological importance). Address reduction was done using γ_{Δ_2}. BOTTOM: The reduction in bits for the yeast and segmentation clock genes. Being of similar length, we see that the yeast genes are more reducible, and therefore exhibit more pattern, than the segmentation clock genes. Note the variety of patterns identified. The known cell cycle gene YGL055W, which unexpectedly exhibits only one cycle, would be overlooked by less blind approaches.

led to systematic error in the form of hundreds of pure zig-zag curves, not shown.) Note that here, unlike in yeast above and the segmentation clock below, the confirmation set is exact.

Third, we studied gene expression curves associated with the segmentation clock, the landmark discovery of the molecular process responsible for converting embryonic time into spatial pattern [15, 16]. Genes associated with this oscillator drive the early segmentation of the embryo into somites, which later form the vertebrae, axial skeleton and skeletal muscles. Part of the difficulty in identifying the small number of associated cyclic genes is that the weakly signalled periodicity is easily confused with random fluctuations present in the thousands of other genes. Moreover, because experimentally it is only possible to measure a single period, traditional approaches like Fourier analysis are not applicable. In collaboration with Pourquie and co-workers, we ordered 8,473 mouse genes according to their address reduction. Our confirmation set was a list of 20 genes experimentally verified by Pourquie. As shown in Fig. 3, these gene are concentrated in the very top of our list. The map γ_{Δ_1} produced a 60% success rate, with $\gamma_{\Delta_{1,2}}$ not far behind; on the other hand, $\gamma_{\rm opt}$ and γ_{+-} performed no better than chance.

Discussion — Address reduction allows us to quantify the presence of pattern, regardless of what kind of pattern it is. The reduction in bits k is a universal currency by which we can rank curves according to their significance, even if they are of different lengths (numbers of data points), or exhibit different kinds of pattern, or are the output of different experiments. We consider absolute reduction in bits, rather than relative reduction, because the presence of pattern is piecewise independent.

The reduction in bits k satisfies two invariance properties which help render our approach independent of details of the system. First, the distribution of $k(f|\gamma)$ is independent of the distribution P_f from which the f_i are drawn. Thus the behaviour of k for random curves does not depend on details of the experimental source, nor is it affected by the addition of i.i.d. noise. Second, let t be a continuous monotonic function, and $g = t(f)$ be the curve f transformed by t. Then $k(f|\gamma) = k(g|\gamma)$ for all t and maps γ. This is important because (often unknown) transformations are implicit in measuring and processing the data, such as normalisation or the logarithm.

Because we are not computing statistical averages over data points but rather an entire curve's exact reduction, we do not need many data points N to make definite conclusions. N depends only on the number of curves M under consideration; N should satisfy $N! \gg M$.

The role of the map γ can be considered in the following way. Transform the distribution P_f such that the f_i are uniformly distributed on the unit interval. Then an arbitrary curve f is a uniformly distributed point in the N dimensional hypercube. This hypercube can be cut into $N!$ polytopes of equal volume such that the curves in each polytope have the same permutation; thus the square divides into 2 triangles, the cube into 6 tetrahedra, and so on. The application of γ results in the merger of the $N!$ polytopes into fewer, larger polytopes of different sizes, in such a way that physically meaningful curves have polytope addresses short compared to $\log_2 N!$.

It should be noted that, for the reduction in bits k to be truly comparable across different maps γ, we should add to the address length the (minimal) amount of information necessary to describe γ itself. For all γ maps with constant length descriptions (*i.e.*, independent of N, as are those described above), the fractional lengths of γ are vanishingly small in the limit of large N.

Conclusion — We have presented here a general framework for detecting non-random data series in any system which exhibits random or near random fluctuations. Address reduction (i) is an unbiased, rigorous detector of pattern; (ii) provides a universal currency for comparing curves from different experiments; (iii) its implementation is independent of details of the experiment; and (iv) is applicable even when the number of data points is small. We believe our method will be of broad practical use in detecting pattern in genomic and proteomic expression profiles, financial and economic market indicators and medical diagnosis, complementing more traditional techniques like clustering and supervised learning.

[1] R. J. Solomonoff, *Inform. Control* **7**, 1–22, 224-254 (1964).
[2] Christoph Brandt and Bernd Pompe, *Phys. Rev. Lett.* **88**, 174102 (2002).
[3] Wojciech H. Zurek (editor), *Complexity, Entropy and the Physics of Information* (Addison-Wesley, CA, 1990).
[4] H. Kantz *et al.* (eds), *Nonlinear Analysis of Physiological Data* (Springer, Berlin, 1996).
[5] M. Rosvall, A. Trusina, P. Minnhagen and K. Sneppen, *Phys. Rev. Lett.* **94**, 028701 (2005).
[6] A. N. Kolmogorov, *Prob. Inform. Transmission* **1**, 4–7 (1965).
[7] G. J. Chaitin, *J. Assoc. Comput. Mach.* **13**, 547–569 (1966).
[8] Paul T. Spellman *et al.*, *Molecular Bio. of the Cell* **9**, 3273–3297 (1998).
[9] Itamar Simon *et al.*, *Cell* **106**, 697–708 (2001).
[10] D. I. Warren and E. Seneta, J. Appl. Prob. **33**, 101-114 (1996).
[11] Karen Willbrand *et al.*, Bioinformatics, **21**, 3859 (2005).
[12] Ziv Bar-Joseph *et al.*, Proc. Natl. Acad. Sci. USA **100**, 10146–10151 (2003).
[13] Sebastian Ahnert *et al.*, Bioinformatics, in press (2006).
[14] Sofia Wichert, Konstantinos Fokianos and Korbinian Strimmer, *Bioinformatics* **20**, 5–20 (2004).
[15] Pourquie, Olivier, *Science* **301**, 328–330 (2003).
[16] J.K. Dale *et al.*, *Nature* **421**, 275 (2003).
[17] J. N. Towse and J.D. Valentine, European J. of Cognitive Psychology **9**, 381-400 (1997).
[18] J. N. Towse, J. of Psychology, **89**, 77-101 (1998).