
Article

Not peer-reviewed version

Curious Coincidences and Kolmogorov Complexity

[Kamal Dingle](#) *

Posted Date: 11 July 2025

doi: [10.20944/preprints202507.0968.v1](https://doi.org/10.20944/preprints202507.0968.v1)

Keywords: Kolmogorov complexity; algorithmic probability; Solomonoff induction; coincidences; information theory; time series



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Curious Coincidences and Kolmogorov Complexity

Kamal Dingle

Centre for Applied Mathematics and Bioinformatics, Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, Kuwait; dingle.k@gust.edu.kw

Abstract

Coincidences are surprising concurrences of rare or unlikely events, with no apparent cause. Paradoxically, coincidences are reported quite often, which is explained by different factors such as psychological biases, hidden causes, but also mathematically. For example, it can be shown from combinatorics that some 'surprising' events like matching birthdays are expected under common circumstances, which illustrates how our intuition is not good at estimating these chances. In this chapter, I examine another mathematical factor that may explain some coincidences, namely Kolmogorov complexity and algorithmic probability. In brief, I show how the varying complexity of different numbers or patterns can make certain outcomes more likely than others, thereby reducing the entropy of the probability distributions, and hence increasing the chances of coincidences. I study the index of coincidence (probability of matches) for integers and patterns, convergent evolution in biology, and time series patterns. For the latter, I further study coincidences in anomalies, that is, two independent series simultaneously defying an observed trend. Additionally, I point to an intriguing connection to uncomputability, namely that in some cases it would not be possible to know how likely a given coincidence was, adding a peculiar twist to curious coincidences.

Keywords: Kolmogorov complexity; algorithmic probability; Solomonoff induction; coincidences; information theory; time series

1. Introduction

We often hear of strange coincidences or the occurrence of seemingly unlikely events, such as someone winning the lottery twice, or being struck by lightning multiple times, or finding that a new work colleague has the same last name. Such curious coincidences may be intriguing, possibly unsettling, or even profoundly meaningful [1,2]. In addition to academic interest, studying coincidences can also be important practically, because, for example, they may point to some deviation from statistical independence, are useful in cryptography [3], are relevant in the analysis of pseudo-random number generators [4], and may be misconstrued as evidence in law courts [5].

Naturally, we seek explanations for coincidences, and such explanations might come from multiple directions including hidden or undetected causes [2]. Another approach to explaining the high frequency of reported coincidences is from psychology, that is, the mind may search for causal structure and remember or highlight certain events while ignoring or forgetting others, or as a consequence of rational causal learning mechanisms [6]. Unfortunately, because most coincidences are noted only anecdotally or in personal communications, it is hard to make a thorough scientific study of the phenomenon. Seeking to address this lacuna, Spiegelhalter has created a catalogue ¹ of surprising events with contributions from the general public, but this catalogue is by no means exhaustive.

From a different perspective, the ubiquity of coincidences can be understood, at least partially, from mathematics [2]. For example Hand explains coincidences and other unlikely events in daily life via *The Improbability Principle* [7], that is, due to the large number of events occurring over many years,

¹ <http://understandinguncertainty.org/coincidences/>



and many possible opportunities for coincidences, even extremely ‘unlikely’ events are almost certain to occur, which is perhaps unintuitive. Fisher [8] argues similarly that “the *one chance in a million* will undoubtedly occur, with no less and no more than its appropriate frequency, however surprised we may be that it should occur to us.” In other words, if we perform the mathematical calculations and consider the associated combinatorics, and the very many potential situations that coincidences could occur, we would find that even without invoking psychological or other factors we should *expect* to see events like multi-lottery winners. Additionally, Ramsey theory [9] is a branch of mathematics that studies how some degree of order or pattern *must* appear in sufficiently large combinatorial structures. Such patterns can lead to coincidences, which might be deemed remarkable, but which arise from purely mathematical origins.

In this chapter, I pursue the direction of uncovering mathematical causes for coincidences, which may explain why they occur more often than we might have expected. In particular, I will invoke ideas from the theory of Kolmogorov complexity and the closely related notion of algorithmic probability [10]. While mathematical in nature, Kolmogorov complexity is not part of combinatorics, and hence distinct from these earlier mathematical approaches. Essentially, I will show that different numbers, patterns, and outcomes are expected to appear with different probabilities due to their intrinsic complexity, or information content. Further, that numbers, patterns, and outcomes with lower complexity values are *a priori* more likely to appear, while at the same time these low complexity outcomes are rare within the space of all possible outcomes. Together, these imply that coincidences are more likely. I explore this topic in the context of the *index of coincidence*, especially in connection to integers and input-output map patterns, as well as examples of convergent evolution from biology, pattern matches and simultaneous anomalies in time series, and connections to uncomputability.

2. Brief Review of the Mathematics of Coincidences

2.1. Index of Coincidence

Perhaps the simplest type of coincidence to analyse mathematically is the probability of two independent samples matching, known as the *index of coincidence* (IC) [3]. If $P(x)$ is the probability of outcome x , with $\sum_x P(x) = 1$ and $x \in \mathcal{X}$, where \mathcal{X} may be finite or infinite, then IC for the distribution P is

$$IC(P) = \sum_x P^2(x) \quad (1)$$

If we take for example $P_D(x)$, the probability that the outcome of a fair dice shows face x , with $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$, then on rolling a dice twice, the probability of two matching scores is

$$IC(P_D) = \sum_x \left(\frac{1}{6}\right)^2 = \frac{1}{6} \quad (2)$$

It is easy to see that for any discrete distribution P ,

$$IC(P) \geq \frac{1}{|\mathcal{X}|} \quad (3)$$

with equality if the probabilities are uniform. Further, using the Shannon entropy $H(P)$ in bits, given by

$$H(P) = - \sum_x P(x) \log_2 P(x) \quad (4)$$

IC can be lower-bounded [11] as follows

$$IC(P) \geq 2^{-H(P)} \quad (5)$$

Naturally, coincidences are more likely if there are non-uniform probabilities, and Eq. (5) confirms and quantifies this by showing that lower entropy distributions have higher IC values, and hence matches (coincidences) are more likely.

2.2. The Birthday Problem

A famous problem in probability theory, studied in the context of coincidences, is *The Birthday Problem*, which was introduced by Von Mises. This problem asks how many people are required in a room for there to be a 50% chance of at least one matching birthday date. This example is also famous because it highlights how poor our intuition is regarding estimating these probabilities. The solution to this problem is obtained by calculating the probability that there are *no* matches first, which is

$$\prod_{i=1}^{N-1} \left(1 - \frac{i}{365}\right) \quad (6)$$

for N people, and then finding the probability of at least one match by subtracting this value from 1. Counterintuitively, with only $N = 23$ people, there is already a roughly 50% chance of at least one pair sharing a common birthday. While this formula is accurate, it is somewhat unwieldy, so Diaconis and Mosteller [2] give a neat and simple formula for a $\sim 50\%$ chance of at least one match when N balls are thrown randomly into c categories

$$N = 1.2\sqrt{c} \quad (7)$$

In the original birthday problem, we have $c = 365$ days (categories), and so the formula gives $N = 1.2\sqrt{365} = 22.9 \approx 23$ people as before.

Mackay [12] suggests that it is instructive also to estimate the expected number of matches using the number of pairs and the probability that any one pair matches, which is obtained from the following formula

$$\frac{N(N-1)}{2} \frac{1}{c} \quad (8)$$

The expression shows clearly why varying N matters so much, namely, because it is squared: If N^2 is large relative to c , then the expected value will be high, and many matches will occur, or we could say there is a high probability of being at least one. On the other hand, if N^2 is small relative to c , such that $N^2/c \approx 0$, then few coincidences (if any) are expected. With $c = 365$ and $N = 23$, the fraction is 0.7, which is close to 1.

There are many other types of birthday problems also, with variations on the original [2,5,13–16]. The birthday problem has also been solved for non-uniform probabilities [17], but the analytic form of the solution is rather involved. For such non-uniform probabilities, here I can propose roughly estimating the expected number of common birthday matches as

$$\frac{N(N-1)}{2} \frac{1}{2^{H(P)}} \quad (9)$$

This relation can be argued based on the fact that $2^{H(P)}$ is taken as the effective number of categories for a distribution with entropy H bits [11,18], and therefore $1/2^{H(P)}$ is a rough estimate for the probability that two samples match. Because lower entropy implies effectively fewer categories, then coincidences are more common, and the expected number of coincidences increases. So, both Eq. (5) and Eq. (9) imply that lower entropy distributions will yield more frequent coincidences.

3. Kolmogorov Complexity and Algorithmic Probability

Before exploring how complexity is related to coincidences, I will briefly recap some relevant theory.

3.1. Kolmogorov Complexity

Within theoretical computer science, *algorithmic information theory* [10,19–22] (AIT) connects computation, computability theory, and information theory. The central quantity of AIT is *Kolmogorov complexity*, $K(x)$, which measures the complexity of an individual object x as the amount of information required to describe or generate x . Intuitively, $K(x)$ is a measure of how many bits are required to store the compressed version of a data object. Objects containing simple or repeating patterns like 0101010101 will have low complexity, while random objects lacking patterns will have high complexity. $K(x)$ is more technically defined as the length of a shortest program which runs on an optimal prefix *universal Turing machine* (UTM) [23], generates x , and halts. (The use of prefix codes is a technical requirement, needed to satisfy Kraft's inequality.) Mathematically, we can write the Kolmogorov complexity $K_U(x)$ of a string x with respect to a universal Turing machine [23] (UTM) U as

$$K_U(x) = \min_p \{ |p| : U(p) = x \} \quad (10)$$

where p is a binary program for a prefix (optimal) UTM U , and $|p|$ indicates the length of the (halting) program p in bits. Due to the invariance theorem [10] for any two optimal UTMs U and V , $K_U(x) = K_V(x) + O(1)$ so that the complexity of x is independent of the machine, up to additive constants. Hence we conventionally drop the subscript U in $K_U(x)$, and speak of 'the' Kolmogorov complexity $K(x)$. This invariance property is a key aspect of Kolmogorov complexity, because it says the complexity of some output x is a universal quantity, agreed upon by different Turing machines, except up to an asymptotically irrelevant additive $O(1)$ constant.

Note that Kolmogorov complexity is somewhat related to, but also very distinct from, computational complexity. The latter concerns the computational resources (i.e., time and space) required by algorithms, and not the size of the program required to produce some object. The term *algorithmic complexity* is (confusingly) sometimes used to refer to Kolmogorov complexity, and sometimes for computational complexity.

The quantity $K(x)$ is uncomputable, meaning that in general it is not possible, even in principle, to calculate its value, by reduction to the Halting Problem [10]. However, it can often be successfully approximated for practical purposes [10,24,25], or bounded from above, using for example standard data compression algorithms like gzip (but see [26–28] for a fundamentally different approach). Moreover, an increasing number of studies show that AIT and Kolmogorov complexity can be successfully applied in the natural sciences, including thermodynamics [29–31], entropy estimation [32,33], biology [34–36], other natural sciences [37], as well as engineering and other areas [10,24,38].

Although $K(x)$ is uncomputable, many analytical results can be derived. For example, it is well known that almost all data objects (like integers or graphs) are incompressible [10], and so cannot be described except via programs like 'print x ' with x written out literally in full detail. Also, it is well known that for $K(x)$, a typical (random) integer x has [10] complexity

$$K(x) \leq \log_2(x) + \log_2 \log_2(x) + \log_2 \log_2 \log_2(x) + \dots + O(1) \quad (11)$$

with the repeated logarithm continuing until the last positive term. Moreover, almost all integers have complexity values very close to this upper bound. It follows that a lower-order reasonable approximation to estimate the complexity of a typical integer x is

$$K(x) \approx \log_2(x) + 2 \log_2 \log_2(x) \quad (12)$$

However, there are also some integers which are highly compressible such that

$$K(x) \ll \log_2(x) \quad (13)$$

Following ref. [39], I will call these highly compressible values ‘special’, or low complexity. One example of a special number is

$$2^{2^{2^{2^2}}} \approx 2 \times 10^{19728} \quad (14)$$

which has almost 20,000 digits if written out literally in full, and yet can be described very succinctly: In the Python programming language we could use the following to generate this value

```
print(2**2**2**2**2) \quad (15)
```

This program has very few characters, far less than 20,000, implying that the value is highly compressible. Another example of a special number is

$$(8!)! \approx 3 \times 10^{168186} \quad (16)$$

which has almost 170,000 digits, and yet can be succinctly described in far fewer characters. Finally

$$\text{“The largest prime number less than } 10^{1000} \text{”} \approx 10^{1000} \quad (17)$$

Note how in the last case this short description does not invoke basic mathematical operations like factorials and stacked exponentials, but nonetheless precisely describes some specific integer, which has around 1000 digits.

These example values are *very* large numbers, and yet have relatively very low Kolmogorov complexity due to their short descriptions. Hence, these numbers are ‘easy’ to make from an information complexity sense. For simplicity, I have mentioned two categories of numbers, namely compressible and incompressible, but in reality the complexity values can take on a whole range of values, with $0 \lesssim K(x) \lesssim \log_2(x) + O(\log_2 \log_2(x))$.

A core result in AIT, and data compression theory more generally, is that there can be only very few highly compressible objects [10]. This result can be proved very simply via counting arguments: There are 2^n bit strings of length n , but only $2^{n'}$ of length $n' < n$. If x is to be compressed from the literal size of n bits to a much shorter size of n' bits, then there must be an available string of length n' bits to use as a program. But we have just seen that there are only few of these, and so only few objects can be substantially compressed. In other words, there are few short programs, and exponentially many data objects, so only a tiny fraction can be assigned short programs. It follows that only a very small fraction of objects are special.

3.2. Algorithmic Probability

An important result in AIT is Levin’s *coding theorem* [40], establishing a fundamental connection between $K(x)$ and probability predictions. Mathematically, it states that

$$m(x) = 2^{-K(x)+O(1)} \quad (18)$$

where $m(x)$ is the probability that an output x is generated by a (prefix optimal) UTM denoted by U , which is fed with a random binary program. Probability estimates $m(x)$ based on the Kolmogorov complexity of output patterns are called *algorithmic probability*. The intuition here is that low complexity outputs require only short programs to appear in the random input string, and hence such outputs are (exponentially) more likely to be generated. By contrast, a highly complex output which can only be generated via some long program will only be generated very rarely, because it is unlikely that a random input binary string will chance upon the correct program.

Because $K(x)$ is uncomputable, algorithmic probability $m(x)$ is also uncomputable. Like $K(x)$, $m(x)$ can often be approximated quite effectively [27,41]. Also, due to the invariance theorem, algorithmic probability is only weakly dependent on the specific Turing machine used to define it. This means that there is a kind of universal nature to the probability $m(x)$, among other “miraculous” properties [42].

4. Algorithmic Probability and Index of Coincidence

4.1. IC for Algorithmic Probability

In this chapter, we will look at two main ways in which algorithmic probability can influence, and indeed increase, the probability of coincidences. I will begin by examining IC for the algorithmic probability function $m(x)$ given in Eq. (18). If we take the distribution to be over all positive integers $x = 1, 2, 3, \dots$ then IC is given by

$$IC(m) = \sum_{x=1,2,3,\dots} (2^{-K(x)+O(1)})^2 = \sum_{x=1,2,3,\dots} 2^{-2K(x)+O(1)} \quad (19)$$

This is not trivial to calculate, including because $K(x)$ is uncomputable, but also because the exact value depends on the Turing machine used. However, we can still attempt to make estimates in two ways: To begin, Eq. (5) gives a bound on IC using the entropy. Interestingly, the entropy of $m(x)$ is known to be infinite [10], which gives

$$IC(m) \geq 2^{-H(m)} = 2^{-\infty} = 0 \quad (20)$$

On the one hand, stating that IC should be non-negative appears trivial and uninformative. On the other hand, it hints that IC might be very low, due to the zero lower bound. Note that the support of $m(x)$ being countably infinite ($x = 1, 2, 3, \dots$) does not in any way imply that the entropy should be infinite, so the lower bound of 0 for IC is non-trivial and intriguing. For example, a Poisson distribution with mean parameter λ has non-zero probability for all $x = 0, 1, 2, 3, \dots$, and yet has entropy $\sim \log_2(\lambda^{\frac{1}{2}})$ for large λ , which is finite. Moving on from this discussion of infinite entropy, it is clear that $IC(m) > 0$ and not equal to zero because Eq. (19) contains the sum of positive terms.

A second approach to estimating IC is to use Eq. (12) for making a rough approximation, $m_{app}(x)$, to the functional form of algorithmic probability $m(x)$ as follows

$$m_{app}(x) = C 2^{-(\log_2(x) + 2 \log_2 \log_2(x+1))} = \frac{C}{x \log_2^2(x+1)} \quad (21)$$

with C a normalising constant, and where I have used $(x+1)$ in the logarithm to avoid $\log_2(1) = 0$. Note that this rough approximation treats all integers as maximally random or typical (which holds, by definition, for almost all integers). By computing the infinite sum $\sum m_{app}(x) = 1$, the normalising constant is found to be $C = 0.614$. It follows that the index of coincidence is

$$IC(m_{app}) = \sum_{x=1,2,3,\dots} m_{app}^2(x) = \sum_{x=1,2,3,\dots} \left(\frac{0.614}{x \log_2^2(x+1)} \right)^2 = \sum_{x=1,2,3,\dots} \frac{0.377}{x^2 \log_2^4(x+1)} \approx 0.40 \quad (22)$$

So the probability of a match between two random integers drawn according to $m_{app}(x)$ is about 40%, this is despite the fact that the range of possible values is countably infinite, and moreover that the entropy is infinite.

Because $m_{app}^2(1) = 0.38$, which is about 95% of 0.40, this means that most of the coincidences will come from $x \approx 1$, and much less for $x \gg 1$. Further, for $m_{app}(x)$ we can calculate that the probability of a coincidence for a value of $x \geq 5$ is $\approx 10^{-3}$, and for $x \geq 10$ it is $\approx 10^{-4}$, while for $x \geq 100$ it is $\approx 10^{-6}$, or one in a million. For comparison, the Poisson distribution with parameter $\lambda = 1$ also has a countably infinite support $x = 0, 1, 2, 3, \dots$, and also has a monotonically decaying probability distribution. For this distribution, IC is about 0.31, which is not very different from IC of $m_{app}(x)$ which is 0.40. Despite these similarities, for the Poisson distribution, the probability that two random samples match decays to zero very fast in comparison to $m_{app}(x)$, with 99% of matches expected for $x \leq 2$. The chances of a match for $x \geq 5$ is $\approx 10^{-5}$, and for $x \geq 10$ it is only $\approx 10^{-14}$, and finally for $x \geq 100$ it is only $\approx 10^{-317}$.

In summary, for this rough approximation to algorithmic probability, coincidences will occur with a high rate but mainly for small values of x , but at the same time coincidences for larger values are not extremely rare and are much more frequent than for a comparison distribution, that is, the

Poisson distribution with $\lambda = 1$. In other words, while IC is of similar magnitude for both, for $m_{app}(x)$ coincidences will occur for a much broader range of values than for the Poisson distribution. It is plausible that this broader range of values, which makes matches less predictable in terms of which values will coincide, may lead to coincidences being viewed as more surprising.

4.2. Curious Coincidences and Uncomputability

Notice that while $m_{app}(x)$ is monotonically decreasing, this is not true for $m(x)$ in Eq. (18). The reason $m(x)$ is not monotonically decreasing is that for special (compressible) values of x , we may have $m(x) > m(y)$ even though $x \gg y$. What this means is that while often the coincidences will be for $x \approx 1$, which have low complexity values, occasionally we can expect to have coincidences for $x \gg 1$, namely for special values of x which are large but highly compressible and hence highly probable.

One interesting implication is that if a system is governed by algorithmic probability, even only very roughly, it may be that coincidences for very large values would occur. We saw already for $m_{app}(x)$ that a broad range of value matches are not extremely unlikely, and $m(x)$ would yield even larger values. Now, while the average human would easily recognise values $x \approx 1$ as simple, and may not be too surprised to see such numbers appearing in coincidences, the average human would certainly not recognise such special numbers with e.g. 20,000 digits as being low complexity, thus making the coincidence appear more remarkable.

Further, because Kolmogorov complexity and hence $m(x)$ is uncomputable, this means that on observing a matching value x , in general it would not be possible to know the exact value of $m(x)$. This means that it would not in general be possible to know just how surprising the coincidence is. This is especially interesting in the case that the matching values are large. So for example on observing a coincidence of some large special number like $x = 10^{100}$, it can easily be recognised as special because it is just a 1 followed by many 0s. In this case, despite its large size, $m(x)$ would not be extremely small, and so this coincidence would not be especially remarkable, and hence perhaps less arresting. Suppose now that some other large number appeared in a coincidence, but this time not easily recognisable as being special, and we want to know whether this is truly a remarkable and unlikely coincidence or not. Should we be *very* surprised, or just mildly surprised? If $m(x)$ is very small, then the coincidence is really remarkable, but if $m(x)$ is not very small, then the coincidence is not very remarkable. Due to uncomputability, it could never be proved that the number is not special. If it is special, we may be able to show that; but if it is not, then we cannot prove this, even in principle. Hence in general it would not be possible to know if a truly amazing coincidence of large values did in fact occur, or if there is some hidden special property of the number, which caused $m(x)$ to be high, and therefore the event is less remarkable. This points to an intriguing connection between hidden causes and uncomputability, and adds a thought-provoking dimension to curious coincidences.

4.3. Real-World Data Examples

I have argued that compressible, special numbers, may appear with atypically high probability, and therefore lead to coincidences. But is there any evidence of numbers with special properties being overrepresented in the real-world? There are some examples, as I survey now.

Cilibrasi and Vitanyi [43] studied the probability of integers from 1 to 120 appearing in Google searches, and found that “even multiples of ten and five stand out” by having much higher frequencies of occurrence compared to other integers of similar sizes. Their study was explicitly in the context of Kolmogorov complexity, but was not connected to coincidences. Their study gives evidence for special numbers appearing with high probability in natural world data. It follows that coincidences in these numbers would be more likely than without such a bias for special numbers.

More recently, Fink [44] made an investigation into integers which have many factors, and points out that some of these values are more likely to appear in technology applications. These numbers with more factors are special and rare in the space of possible integers, and hence will have lower complexity. Fink found that in design and technology, these integers appear in the screen pixel resolutions of watches, phones, cameras and computers, as well website and typesetting parameters. Hence this is a

real-world case where integers with special properties appear more frequently. Clearly these highly divisible numbers are intentionally chosen for design purposes, and not randomly appearing due to simplicity, but at the same time this does give another real-world example of natural data showing a preference for special integers.

Benford's Law [45] states that in many real-world data sets, the leading digit is often small, e.g., ≈ 1 in favour of ≈ 9 . This law also speaks of a preference for smaller, and in a sense simpler numbers. However, while related to our arguments here, this phenomenon is more limited in scope, and not necessarily related to algorithmically simple or special integers.

Speculating, the ubiquity of numbers like e and π in many mathematical formulae may also be related to their low complexity. Despite their infinite decimal expansions, both have simple short programs, or ways to calculate them, showing that they are highly compressible and hence 'easy' to make from an information perspective.

5. Simplicity Bias

5.1. Upper Bound

Algorithmic probability is beautiful and powerful, but as discussed above it is difficult to apply directly to real-world systems, at least due to the uncomputability of Kolmogorov complexity, but also due to the fact that it is defined in terms of programs running on Turing machines. This latter restriction concerns the fact that algorithmic probability assumes universal computational power, equivalent to a universal Turing machine. While Wolfram [46] has argued that in nature (e.g., the weather) non-trivial systems should be able to perform arbitrary computations (known as the Principle of Computational Equivalence), this fascinating claim is far from being proven true. Therefore, as far as we know the capacity for universal computation is rare or non-existent in nature. In ref. [47] the authors discuss further issues and difficulties related to the application of algorithmic probability to natural systems.

Due to these difficulties, approximations to algorithmic probability in input-output maps have been developed, leading to the observation of a phenomenon called *simplicity bias* [48]. Simplicity bias is captured mathematically as

$$P_{SB}(x) \leq 2^{-a\tilde{K}(x)-b} \quad (23)$$

where $P_{SB}(x)$ is the probability of observing output x on a random choice of inputs, and this probability is computable (i.e., it can be calculated). The function $\tilde{K}(x)$ is the approximate Kolmogorov complexity of output x which in general is some compression-based complexity measure (such as Lempel-Ziv complexity [49]), but the bound was also introduced and tested on other more general complexity metrics. In words, Eq. (23) says that complex outputs from input-output maps have lower probabilities, and high probability outputs are simpler. The constants $a > 0$ and b can be fit with little sampling and often even predicted without recourse to sampling [48].

Under some quite generic and not too onerous conditions [48], including that the outputs are patterns which are 'computed' in some way from the inputs, we expect the simplicity bias bound, given in Eq. (23), to apply. For example, we can say (as an approximation) that biomolecular shapes are 'computed' from input gene sequences and the laws of biochemistry and biophysics act as the map/function (while being aware that the mapping in reality does not operate in a vacuum, but instead under micro-environmental factors, such as chaperones or cofactors). Similarly, we can say that the solution of an ordinary differential equation system is 'computed' from the system parameters. Examples of systems exhibiting simplicity bias are diverse and wide ranging, and include:

- (1) Molecular shapes such as natural protein quaternary structures [35], and RNA secondary structures [35,48,50]. (2) Polyomino self-assembled 2D tile shapes [35,51].
- (3) Tooth shapes from intricate and realistic computational models of dental morphology [51].
- (4) Plant shapes in computational simulations of plant growth [48].
- (5) Output binary strings from finite state transistors [52].

- (6) Solution curves of differential equation systems [35,48], and a model of financial market time series [48].
- (7) Output patterns in deep neural networks from machine learning [53–55].
- (8) Natural time series data [56], in particular digitised up-down series patterns.
- (9) Digitised time series output patterns from dynamical systems [57,58].

In all these cases, the high probability output patterns are simple, while complex outputs have low probability. The ways in which simplicity bias differs from $m(x)$ include that Eq. (23) does not assume the presence of Turing machines, uses approximations of Kolmogorov complexity, and for many outputs, their respective probabilities fall far below the upper bound, i.e.,

$$P_{SB}(x) \ll 2^{-a\bar{K}(x)-b} \quad (24)$$

Hence the abundance of low complexity, low probability outputs [47,52] is a signature of simplicity bias. These low complexity, low probability outputs are said to be patterns which are not intrinsically complex, but yet are ‘hard’ to make for the specific system [47,52].

Perhaps counter-intuitively, while many output patterns fall below the upper bound, almost all the probability mass is associated to outputs which are close to the upper bound [52]. This means that a randomly chosen input is very likely to map to an output which is close to the bound.

5.2. IC and Simplicity Bias

So far we have studied the index of coincidence (IC) for integers, but now we turn to patterns, and in particular digitised patterns like binary strings, shapes, or natural structures. Distributions which follow the simplicity bias Eq. (23) for a range of complexities will show non-uniform probabilities, and hence have lower entropy distributions. Therefore, we can expect low entropy and high IC scores for outcomes for which the categories are ‘computed’ patterns (under certain conditions [48]). As we saw above, there are already very many systems which have been shown to exhibit simplicity bias, and so we can expect it to be widely applicable.

It is not possible to estimate the entropy or IC directly from the form of $P_{SB}(x)$, because a whole range of different distributions can display simplicity bias while not violating the upper bound. Nonetheless, this is not problematic because I am arguing merely that simplicity bias is a source of bias (low entropy distributions). Lower entropy distributions imply higher IC values, and therefore more chance of coincidences. So, in all the examples mentioned, from plant shapes to dynamical systems outputs, we can expect higher IC scores, and hence higher chances of coincidences in these shapes and outputs.

5.3. Convergent Evolution

In biology, different species can share traits or characteristics, such as lions and tigers which both have four limbs, sharp claws and teeth, hair, etc. These similarities, known in evolution theory as *homology*, are due to having evolved from common ancestors ‘recently’ in natural history. These similarities are in some sense trivial, have a clear cause, and would not be described as coincidental.

On the other hand, many similarities in biology are not ascribed to common ancestry, but to *convergent evolution* [59,60], where the same trait appears independently in different species lineages. Such convergences can be viewed as coincidences of biological traits, and some cases are slightly mysterious [61,62]. A famous example of convergence is the camera-type eye which appears in organisms as diverse as an octopus and a cat, even though the common ancestor of cats and octopuses lived many millions of years ago, and did not itself have a camera eye. Therefore the same trait has appeared independently more than once in life history. While natural selection operating in similar environments can explain some of these cases of convergence, a full explanation for convergent evolution is lacking [60], which motivates looking for other causes.

Here I propose that some cases of convergent evolution are due to high IC scores, deriving from simplicity bias. Indeed, Dingle et al. [63] and Louis [64] already suggest that some convergent

evolution (e.g., of naturally occurring RNA biomolecules) can be explained at least in part by low entropy distributions over the possible structures/shapes. That is, even though there are millions of different possible molecule shapes, due to a very biased (low entropy) distribution over these shapes, only a small fraction of these are likely to appear when sampling random gene sequences. Even though evolution is not thought to operate by simple random sampling of the gene sequence space, but closer to a (biased) random walk in gene space, these two processes in fact share many statistical properties [35]. Hence, if uniform random sampling of the gene space leads to a low entropy distribution over output structures, then so too would an evolutionary random walk through the space. As above, low entropy distributions imply higher IC values, which means that independently evolving populations have a relatively high chance of coincidences in shape patterns, what would be called convergent evolution in biology. Later, it was shown that many low entropy distributions over biological shapes are explained in large part by simplicity bias [35,48]. I therefore propose that simplicity bias may explain the convergent evolution of some biomolecules. Given that many biological traits are likely to be subject to simplicity bias [35,48,51], this suggests that other biological traits might also converge coincidentally for the same reason.

This argument resonates with what Wright [65] recently mentions in relation to simplicity bias and the convergent evolution of protein structures. He notes that convergence is more likely “for small, low complexity, or repetitive proteins that readily evolve across different genomes”, due to their simplicity, and therefore being inherently more likely to appear.

6. Coincidences in Time Series

Analysing and forecasting time series is a staple of statistics, machine learning, and dynamical systems research, with applications from finance to climate science, and beyond [66]. In this section, we study potential causes for coincidences in time series by invoking results from Kolmogorov complexity theory.

6.1. Time Series Patterns

Natural time series have been shown to display some form of simplicity bias [56,67], and additionally some prototypical maps from dynamical systems research have also shown simplicity bias [57,58]. In particular, this was observed for digitised series patterns, when continuous curves were converted into binary patterns, either as up-down patterns, or high-low patterns. It follows that the distribution over time series patterns in such settings will be low entropy, and hence we might expect to have coincidences in patterns. For example, the time series pattern “monotonic increasing” is highly likely to appear due to its simplicity, and hence is more likely to cause coincidences in time series patterns. Therefore, we can expect some coincidences in natural or dynamical systems time series patterns.

6.2. What's Next?

In the last part of this chapter, I will study coincidences in time series anomalies. The following analysis is somewhat speculative, and to my knowledge there is currently no recorded evidence of this type of coincidence occurring in real-world data, but that is not to say that it could not be found. I will first recap the relevant forecasting theory.

Consider observing the outcome of a binary series $x_1x_2x_3\dots x_n$, with $x_i \in \{0, 1\}$, and finding that the first five bits of the series are

$$x_1x_2x_3x_4x_5 = 11111 \quad (25)$$

What comes next? Most people would predict that the next symbol is $x_6 = 1$, but how confident should we be? Clearly we are more than 50% certain, but less than 100%, which still leaves a wide range of possible values. How can we proceed with this problem, and what is the basis of our inference?

Around 200 years ago, Pierre-Simon Laplace examined the question of induction, and proposed the Rule of Succession [68]. It states that the probability that the next bit is 1, after observing k 1s out of a total of n observations, is

$$\frac{k+1}{n+2} \quad (26)$$

assuming we have no other information about the induction problem, and we are extrapolating only from the observed data. Applying Laplace's Rule to the specific problem introduced above gives the probability of the next bit being 1 as

$$P_L(1|11111) = \frac{k+1}{n+2} = \frac{6}{7} \quad (27)$$

where $k = 5$ is the number of 1s, and $n = 5$ is the total number of digits in the series. So Laplace would be $6/7 \approx 86\%$ sure that the next bit is a 1. It follows that his method gives the probability that the next bit is 1 after seeing $k = n$ ones is

$$P_L(1|1^n) = \frac{n+1}{n+2} \approx 1 - \frac{1}{n} \quad (28)$$

where the notation 1^n means n 1s in a row. This problem could also be addressed using Bayes theorem [68], and indeed other methods [69,70], but now I will describe a very general method that will help us study anomalies.

6.3. Solomonoff Induction

If we know the mechanism or process that generates the series, then using that mechanism as the method of inference will yield the most accurate predictions. Suppose by contrast, that we do not know the mechanism and we wish to forecast solely based the available observed series data. It can be argued that it is best not to assume one particular mechanism, but following the Greek philosopher Epicurus, we should try to take into account all possible hypotheses which could have generated the time series. Additionally, following Occam we should prefer simpler hypotheses [71]. To take a concrete example, the series 2, 4, 6, 8 could be explained by the following hypotheses (among others)

$$h_1 : 2n \quad (29)$$

$$h_2 : -n^4 + 10n^3 - 35n^2 + 52n - 24 \quad (30)$$

$$h_3 : 2, 4, 6, 8, -9, 732 \quad (31)$$

Each of these hypotheses or explanations would give the starting values of 2, 4, 6, and 8 for the first four terms. While Occam's razor would suggest h_1 as the best candidate due to its simplicity, forecasting *only* according h_1 might be overly hasty, and preferably we should also incorporate the forecasts from the other hypotheses. How can this be done? The solution proposed in the 1960s by Ray Solomonoff [19,72] is called *Solomonoff induction* [10,39,71,73] and is based on algorithmic probability. More technically, Solomonoff induction uses the function $M(x)$ [71]

$$M(x) = \sum_{p:U(p)=x*} 2^{-|p|} \quad (32)$$

which gives the probability that the output of a (monotone) universal Turing machine has initial segment x , when run on a uniformly random input tape. Here U is some given fixed universal Turing machine, $|p|$ is the length of program p in bits, and $x*$ denotes an infinite series that begins with x . This quantity $M(x)$ can be viewed as a weighted sum of all possible explanations or models of the data, where simpler hypotheses are given more weight. As for predicting the next digit in a series, Solomonoff induction employs [71]

$$M(x_{n+1}|x_1x_2\dots x_n) = \frac{M(x_1x_2\dots x_nx_{n+1})}{M(x_1x_2\dots x_n)} \quad (33)$$

Essentially, this assigns the predicted probabilities based on how well the following bit (0 or 1) fits the preceding sequence, in terms of how well the sequence can be compressed (see also [74]).

Let us now return to the question of predicting the next bit in a series. Using $M(x)$, Hutter [39] has shown that the probability that the next bit is 1 can be estimated via Solomonoff induction as

$$P_S(1|1^n) = 1 - 2^{-K(n)+O(1)} \quad (34)$$

For typical random values of n , this relation can be simplified using Eq. (12) to

$$P_S(1|1^n) = 1 - 2^{-K(n)+O(1)} \approx 1 - \frac{1}{n \log_2^2(n)} \quad (35)$$

which is very similar to Laplace's estimate above in Eq. (28), the only difference being the log term. The log term implies that for typical random integers, Solomonoff is only slightly more confident in predicting the next bit as compared to Laplace, because

$$P_S(1|1^n) \approx 1 - \frac{1}{n \log_2^2(n)} > 1 - \frac{1}{n} \approx P_L(1|1^n) \quad (36)$$

More interestingly, for special numbers Solomonoff induction does not agree with Laplace, or to be more precise, the probabilities $P_S(1|1^n)$ and $P_L(1|1^n)$ are quite different: Observe that for highly compressible n

$$K(n) \ll \log_2(n) \Rightarrow P_S(1|1^n) = 1 - 2^{-K(n)+O(1)} < 1 - \frac{1}{n} \approx P_L(1|1^n) \quad (37)$$

So that at special values of n , Solomonoff is much less certain that the next bit will be 1, as compared to Laplace.

There are a number of curious implications of the influence of special numbers on predictions [39,68], among them being that having *more* data, can yield *less* certainty in the ability to forecast the next bit [10,58]. Under normal circumstances, more data which all conform to a trend should give more confidence in the ability to forecast the next bit (as Laplace would agree), however for a special value n^* and some larger typical random value n , we may have $n^* \gg n$, yet $P_S(1|1^{n^*}) < P_S(1|1^n)$. Hamzi and Dingle [58] give an example of how this counterintuitive result applies in the logistic map from dynamical systems.

6.4. Coincidence in Time Series Anomalies

Studying and detecting spikes, anomalies, and change points in time series is an active and important area of research [75–77]. I now examine the probability of coincidences regarding a break or anomaly in the pattern of a time series, using the preceding theory of Solomonoff induction.

Suppose Alice and Bob observe binary series s_A and s_B , respectively, from time $t = 1$ up to time $t = T - 1$. So far, both series behave according to some simple pattern. Alice's series is

$$s_A = \underbrace{000 \dots 000}_{T-1} \quad (38)$$

and Bob's is

$$s_B = \underbrace{111 \dots 111}_{T-1} \quad (39)$$

They now ask: What is the probability that at time $t = T$, they both simultaneously observe an anomaly, i.e., a break in the pattern? For Alice, this anomaly would mean observing a 1 instead of the expected 0; and for Bob a 0 in place of the expected 1. If Alice and Bob knew the mechanisms or processes that generated their respective series, then they could use that information (or other side information) to make estimates about this probability question. Here, by contrast, I will assume that

they have no information about the mechanisms at all, and only have the data observations to work with.

In this general setting, where no information is assumed, the probability can be estimated using Eq. (34), which tells that Solomonoff gives the probability of Alice's anomaly as $P_S(1|0^{T-1}) = 2^{-K(T)+O(1)}$. Similarly, for Bob the probability is also $2^{-K(T)+O(1)}$. Because these two series are independent, the probability that they both have an anomaly, occurring coincidentally, is

$$(2^{-K(T)+O(1)})^2 \quad (40)$$

This value depends on the complexity of integer T . If T is a typical (incompressible) integer, then the probability is very low, as quantified by Eq. (35). On the other hand, if T is a special number, then this probability may not be extremely small, and may be much higher than $1/T^2$, that Laplace would predict. Hence for special and highly compressible time points, coincidental anomalies in the next bit may be more likely than we would have thought without invoking Kolmogorov complexity theory. This is not to say that at special time points anomalies should actually be expected to occur, but that at these time points we should be more prepared for something unusual [39].

Interestingly, due to the universality of Kolmogorov complexity, this coincidence probability does not depend very sensitively on the specific pattern or sequence in the time series. For illustration purposes, the series s_A and s_B were chosen to be extremely simple, but the arguments apply to any simple patterns for the first $T - 1$ values. So for example, the same probability of coincidence would be valid for a series which appears to match the function $\sin(t)$, or $5 \cos(t^2 + 1)$ or some other kind of simple pattern.

Indeed, the theory even applies to pseudo-random deterministic series. If Alice had observed what appears to be the first $T - 1$ digits of $\pi = 3.141592\dots$, and Bob observed what appears to be the first $T - 1$ digits of $e = 2.718281\dots$ and they wanted to predict the probability that the next digit, in position T of their respective series, did not fit the pattern of the preceding digits, then the same estimate $(2^{-K(T)+O(1)})^2$ would apply.

A known weakness of Solomonoff induction for predicting the next bit is that if we have not observed the series from the start, or not counted, then we would not know the values of T [68]. Hence, for example, Alice would not know when to be prepared for an anomaly. Additionally, even if T is known, this does not imply knowing $K(T)$, which is the relevant quantity. Although this is a problem for prediction, it is not a problem for my argument here which relates to coincidences: what Solomonoff induction tells us is that at certain special time points — whether we know or not — anomalies are a priori more likely. Hence, coincidences are a priori more likely, even if we do not know the values of T .

Even though Alice and Bob's series are independent, if enough unexpected coincidents occurred, it may lead Alice and Bob to infer that their series are in reality correlated or linked in some way. The preceding theoretical framework hints that deterministic series may share some statistical properties, even when they are unrelated. These properties arise from the complexity of the time points, rather than any direct connection of the series. There is plenty of future work opportunities for exploring the consequences of these potential curious correlations.

6.5. Near Matches

Finally, I look at near matches in which Alice sees an anomaly at T , while Bob sees an anomaly at a near time point T' . Observe that

$$|T' - T| = k \Rightarrow |K(T') - K(T)| \leq \log_2(k) + O(\log_2 \log_2 k) \quad (41)$$

for integer k . This means that if k is small, in the sense of

$$\frac{\log_2(k)}{K(T)} \approx 0 \approx \frac{\log_2(k)}{K(T')} \quad (42)$$

then the values $K(T)$ and $K(T')$ will be close. Hence, if T is special and k is small, then $T' \approx T$ is also special, and so the probability of anomalies according to Solomonoff is still roughly $(2^{-K(T)+O(1)})^2$. This means that for special T , close matching anomalies are also a priori not extremely unlikely.

7. Discussion

I have studied the problem of explaining the frequent reports of coincidences across various aspects of life. Several possible lines of explanation have been suggested earlier by others, such as psychological factors, hidden or unnoticed causes, and mathematical reasons, namely the large number of opportunities available for coincidences to occur. Here I proposed that the varying Kolmogorov complexity values of different numbers, patterns, and outcomes makes the a priori probability of the different outcomes biased towards simpler, or special outcomes. This bias yields lower entropy distributions, which in turn make coincidences more likely. The many cited examples of such biases, including in technical design specifications and evolutionary biology, argue for the potential widespread applicability of the arguments. Finally, I suggested here that the uncomputability of Kolmogorov complexity means that on observing a coincidence of numbers or patterns, it might never be known for certain whether it was truly a remarkable and extremely unlikely event, or if instead a hidden pattern had a causal role in its occurrence such that it was not so very unlikely.

Another perspective on the main thrust of this work is to highlight that in patterns resulting from purely random processes, such as coin flips, regularities and repeated motifs seldom occur, which means that coincidences are relatively unlikely. By contrast, for patterns that result from some kind of computation, regularities and repeated motifs occur relatively often, which means that coincidences are more likely.

Naturally, there are several limitations to this type of work. To start, the information complexity arguments apply to outcomes for which a complexity value is meaningful, such as numbers, shapes, and digital sequences. On the other hand, outcomes like first names and colours do not have meaningful complexity values, and hence are out of the scope of this work. Also, while birthdates are technically numbers, there does not appear to be strong bias in the distribution of birthdates [17], and indeed we would not expect such dates to be affected by simplicity bias or algorithmic probability because they are not patterns computed from inputs. Another limitation is that I have invoked Kolmogorov complexity, technically an uncomputable quantity, which limits the applicability of related results to some extent. How much of a limitation this is, is an interesting question. On the one hand, I showed how simplicity bias theory, a computable analogue of algorithmic probability, can be used to argue for coincidences (therefore bypassing uncomputability). On the other hand, to what extent Kolmogorov complexity can be approximated is debated, with some arguing that for practical purposes uncomputability is not a major limitation and the theory can be applied quite straightforwardly [24,25], while other studies show that there are no good computable approximations for Kolmogorov complexity in general [78,79], and thereby emphasising the limitations of applying the theory.

This chapter is intended as a brief and discursive investigation of some cases where curious coincidences might be explained via information complexity arguments. No doubt there are many further areas of potential investigation. Among several possible lines of future work, one suggestion is to focus on dynamical systems research and connections to forecasting, anomalies, and change points, as briefly begun here. In particular, it would be very interesting to see if there is any evidence in natural data of the proposed coincidences in time series anomalies. Another line would be to explore coincidences in optima [80], especially in relation to biology, where it has been argued [81] via Kolmogorov complexity theory that optimising for one property may perhaps coincidentally aid optimising an unrelated quantity, which is a fascinating hypothesis for biophysics. Finally, finding new ways to estimate complexity and uncover patterns (e.g., ref. [82]) would be useful for the application of Kolmogorov complexity to studying coincidences, and more widely in mathematics and natural sciences.

Acknowledgments: I thank A. Hadi Erturk and the anonymous reviewers for useful comments and feedback.

References

1. Jung, C.G.; Hull, R. *Synchronicity: An Acausal Connecting Principle*. (From Vol. 8. of the Collected Works of C. G. Jung), rev - revised ed.; Princeton University Press, 1960.
2. Diaconis, P.; Mosteller, F. Methods for studying coincidences. *Journal of the American Statistical Association* **1989**, *84*, 853–861.
3. Friedman, W.F. *The index of coincidence and its applications in cryptography*; Aegean Park Press, 1922.
4. Hofert, M. Random number generators produce collisions: Why, how many and more. *The American Statistician* **2021**, *75*, 394–402.
5. Pollanen, M. A Double Birthday Paradox in the Study of Coincidences. *Mathematics* **2024**, *12*, 3882.
6. Johansen, M.K.; Osman, M. Coincidences: A fundamental consequence of rational cognition. *New Ideas in Psychology* **2015**, *39*, 34–44.
7. Hand, D.J. *The Improbability Principle: Why Coincidences, Miracles, and Rare Events Happen Every Day*; Macmillan, 2014.
8. Fisher, R. *The design of experiments*; Springer, 1971.
9. Graham, R.L.; Rothschild, B.L.; Spencer, J.H. *Ramsey theory*; John Wiley & Sons, 1991.
10. Li, M.; Vitanyi, P. *An introduction to Kolmogorov complexity and its applications*; Springer-Verlag New York Inc, 2008.
11. Cover, T.; Thomas, J. *Elements of information theory*; John Wiley and Sons, 2006.
12. MacKay, D.J. *Information theory, inference and learning algorithms*; Cambridge university press, 2003.
13. Holst, L. The general birthday problem. *Random Structures & Algorithms* **1995**, *6*, 201–208.
14. Henze, N. A Poisson limit law for a generalized birthday problem. *Statistics & Probability Letters* **1998**, *39*, 333–336.
15. Camarri, M.; Pitman, J. Limit distributions and random trees derived from the birthday problem with unequal probabilities. *Electronic Journal of Probability [electronic only]* **2000**, *5*, Paper-No.
16. Zhou, Q. Birth, Death, Coincidences and Occupancies: Solutions and Applications of Generalized Birthday and Occupancy Problems. *Methodology and Computing in Applied Probability* **2023**, *25*, 53.
17. Mase, S. Approximations to the birthday problem with unequal occurrence probabilities and their application to the surname problem in Japan. *Annals of the Institute of Statistical Mathematics* **1992**, *44*, 479–499.
18. Mezard, M.; Montanari, A. *Information, physics, and computation*; Oxford University Press, USA, 2009.
19. Solomonoff, R.J. A Preliminary Report on a General Theory of Inductive Inference (Revision of Report V-131). *Contract AF* **1960**, *49*, 376.
20. Kolmogorov, A. Three approaches to the quantitative definition of information. *Problems of information transmission* **1965**, *1*, 1–7.
21. Chaitin, G.J. A theory of program size formally identical to information theory. *Journal of the ACM (JACM)* **1975**, *22*, 329–340.
22. Bédard, C.A. Lecture Notes on Algorithmic Information Theory. *arXiv preprint arXiv:2504.18568* **2025**.
23. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. *J. of Math* **1936**, *58*, 5.
24. Vitányi, P.M. Similarity and denoising. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2013**, *371*, 20120091.
25. Vitányi, P. How incomputable is Kolmogorov complexity? *Entropy* **2020**, *22*, 408.
26. Veness, J.; Ng, K.S.; Hutter, M.; Uther, W.; Silver, D. A monte-carlo aixi approximation. *Journal of Artificial Intelligence Research* **2011**, *40*, 95–142.
27. Delahaye, J.; Zenil, H. Numerical Evaluation of Algorithmic Complexity for Short Strings: A Glance into the Innermost Structure of Algorithmic Randomness. *Appl. Math. Comput.* **2012**, *219*, 63–77.
28. Soler-Toscano, F.; Zenil, H.; Delahaye, J.P.; Gauvrit, N. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PLoS one* **2014**, *9*, e96223.
29. Bennett, C. The thermodynamics of computation – a review. *International Journal of Theoretical Physics* **1982**, *21*, 905–940.
30. Kolchinsky, A.; Wolpert, D.H. Thermodynamic costs of Turing machines. *Physical Review Research* **2020**, *2*, 033312.
31. Zurek, W. Algorithmic randomness and physical entropy. *Physical Review A* **1989**, *40*, 4731.
32. Avinery, R.; Kornreich, M.; Beck, R. Universal and accessible entropy estimation using a compression algorithm. *Physical review letters* **2019**, *123*, 178102.

33. Martiniani, S.; Chaikin, P.M.; Levine, D. Quantifying hidden order out of equilibrium. *Physical Review X* **2019**, *9*, 011031.
34. Ferragina, P.; Giancarlo, R.; Greco, V.; Manzini, G.; Valiente, G. Compression-based classification of biological sequences and structures via the Universal Similarity Metric: experimental assessment. *BMC bioinformatics* **2007**, *8*, 252.
35. Johnston, I.G.; Dingle, K.; Greenbury, S.F.; Camargo, C.Q.; Doye, J.P.; Ahnert, S.E.; Louis, A.A. Symmetry and simplicity spontaneously emerge from the algorithmic nature of evolution. *Proceedings of the National Academy of Sciences* **2022**, *119*, e2113883119.
36. Adams, A.; Zenil, H.; Davies, P.C.; Walker, S.I. Formal definitions of unbounded evolution and innovation reveal universal mechanisms for open-ended evolution in dynamical systems. *Scientific reports* **2017**, *7*, 1–15.
37. Devine, S.D. *Algorithmic Information Theory for Physicists and Natural Scientists*; IOP Publishing, 2020.
38. Cilibrasi, R.; Vitányi, P. Clustering by compression. *Information Theory, IEEE Transactions on* **2005**, *51*, 1523–1545.
39. Hutter, M. On universal prediction and Bayesian confirmation. *Theoretical Computer Science* **2007**, *384*, 33–48.
40. Levin, L. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii* **1974**, *10*, 30–35.
41. Zenil, H.; Badillo, L.; Hernández-Orozco, S.; Hernández-Quiroz, F. Coding-theorem like behaviour and emergence of the universal distribution from resource-bounded algorithmic probability. *International Journal of Parallel, Emergent and Distributed Systems* **2019**, *34*, 161–180.
42. Kirchherr, W.; Li, M.; Vitányi, P. The miraculous universal distribution. *The Mathematical Intelligencer* **1997**, *19*, 7–15.
43. Cilibrasi, R.; Vitanyi, P. Automatic meaning discovery using Google. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2006.
44. Fink, T.M. Recursively divisible numbers. *Journal of Number Theory* **2024**, *256*, 37–54.
45. Berger, A.; Hill, T.P. *An introduction to Benford's law*; Princeton University Press, 2015.
46. Wolfram, S. *A New Kind of Science*; Wolfram Media, 2002.
47. Alskandarani, M.; Dingle, K. Low complexity, low probability patterns and consequences for algorithmic probability applications. *Complexity* **2023**, *2023*, 9696075.
48. Dingle, K.; Camargo, C.Q.; Louis, A.A. Input–output maps are strongly biased towards simple outputs. *Nature communications* **2018**, *9*, 761.
49. Lempel, A.; Ziv, J. On the complexity of finite sequences. *Information Theory, IEEE Transactions on* **1976**, *22*, 75–81.
50. Dingle, K.; Batlle, P.; Owhadi, H. Multiclass classification utilising an estimated algorithmic probability prior. *Physica D: Nonlinear Phenomena* **2023**, *448*, 133713.
51. Dingle, K.; Hagolani, P.; Zimm, R.; Umar, M.; O'Sullivan, S.; Louis, A.A. Bounding phenotype transition probabilities via conditional complexity. *bioRxiv* **2024**, pp. 2024–12.
52. Dingle, K.; Pérez, G.V.; Louis, A.A. Generic predictions of output probability based on complexities of inputs and outputs. *Scientific reports* **2020**, *10*, 1–9.
53. Valle-Pérez, G.; Camargo, C.Q.; Louis, A.A. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522* **2018**.
54. Mingard, C.; Skalse, J.; Valle-Pérez, G.; Martínez-Rubio, D.; Mikulik, V.; Louis, A.A. Neural networks are a priori biased towards Boolean functions with low entropy. *arXiv preprint arXiv:1909.11522* **2019**.
55. Mingard, C.; Rees, H.; Valle-Pérez, G.; Louis, A.A. Deep neural networks have an inbuilt Occam's razor. *Nature Communications* **2025**, *16*, 220.
56. Dingle, K.; Kamal, R.; Hamzi, B. A note on a priori forecasting and simplicity bias in time series. *Physica A: Statistical Mechanics and its Applications* **2023**, *609*, 128339.
57. Dingle, K.; Alskandarani, M.; Hamzi, B.; Louis, A.A. Exploring simplicity bias in 1D dynamical systems. *Entropy* **2024**, *26*, 426.
58. Hamzi, B.; Dingle, K. Simplicity bias, algorithmic probability, and the random logistic map. *Physica D: Nonlinear Phenomena* **2024**, *463*, 134160.
59. Conway-Morris, S. Evolution: like any other science it is predictable. *Philosophical Transactions of the Royal Society B: Biological Sciences* **2010**, *365*, 133.
60. McGhee, G.R. *Convergent evolution: limited forms most beautiful*; MIT press, 2011.
61. Morris, S. *Life's solution: inevitable humans in a lonely universe*; Cambridge University Press, 2003.

62. Morris, S.C. *The deep structure of biology: is convergence sufficiently ubiquitous to give a directional signal?* Number 45, Templeton Foundation Press, 2008.
63. Dingle, K.; Schaper, S.; Louis, A.A. The structure of the genotype–phenotype map strongly constrains the evolution of non-coding RNA. *Interface focus* **2015**, *5*, 20150053.
64. Louis, A.A. Contingency, convergence and hyper-astronomical numbers in biological evolution. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences* **2016**, *58*, 107–116.
65. Wright, E.S. Tandem repeats provide evidence for convergent evolution to similar protein structures. *Genome Biology and Evolution* **2025**, *17*, evaf013.
66. Chatfield, C. *The analysis of time series: theory and practice*; Springer, 2013.
67. Zenil, H.; Delahaye, J. An algorithmic information theoretic approach to the behaviour of financial markets. *Journal of Economic Surveys* **2011**, *25*, 431–463.
68. Rathmanner, S.; Hutter, M. A philosophical treatise of universal induction. *Entropy* **2011**, *13*, 1076–1136.
69. Merhav, N.; Feder, M. Universal prediction. *IEEE Transactions on Information Theory* **1998**, *44*, 2124–2147.
70. Willems, F.M.; Shtarkov, Y.M.; Tjalkens, T.J. The context-tree weighting method: Basic properties. *IEEE transactions on information theory* **1995**, *41*, 653–664.
71. Hutter, M.; Legg, S.; Vitanyi, P.M. Algorithmic probability. *Scholarpedia* **2007**, *2*, 2572.
72. Solomonoff, R.J. A formal theory of inductive inference. Part I. *Information and control* **1964**, *7*, 1–22.
73. Hutter, M. *Universal artificial intelligence: Sequential decisions based on algorithmic probability*; Springer Science & Business Media, 2004.
74. Ryabko, B.; Astola, J.; Malyutov, M. *Compression-based methods of statistical analysis and prediction of time series*; Springer, 2016.
75. Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J.A. A review on outlier/anomaly detection in time series data. *ACM computing surveys (CSUR)* **2021**, *54*, 1–33.
76. Dasgupta, A.; Li, B. Detection and analysis of spikes in a random sequence. *Methodology and Computing in Applied Probability* **2018**, *20*, 1429–1451.
77. Horváth, L.; Rice, G. *Change Point Analysis for Time Series*; Springer, 2024.
78. Ishkuvatov, R.; Musatov, D. On approximate uncomputability of the Kolmogorov complexity function. In Proceedings of the Computing with Foresight and Industry: 15th Conference on Computability in Europe, CiE 2019, Durham, UK, July 15–19, 2019, Proceedings 15. Springer, 2019, pp. 230–239.
79. Ishkuvatov, R.; Musatov, D.; Shen, A. Approximating Kolmogorov complexity. *Computability* **2023**, *12*, 283–297.
80. Dingle, K. Optima and simplicity in nature. *arXiv preprint arXiv:2210.02564* **2022**.
81. Dingle, K. Fitness, optima, and simplicity. *Preprints* **2022**, p. 2022080402.
82. Wyeth, C.; Bu, D.; Yu, Q.; Gao, W.; Liu, X.; Li, M. Lossless data compression by large models. *Nature Machine Intelligence* **2025**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.