IOP Publishing Physica Scripta

Phys. Scr. 94 (2019) 045103 (17pp) https://doi.org/10.1088/1402-4896/aaf36a

Experimentally probing the algorithmic randomness and incomputability of quantum randomness

Alastair A Abbott[1], **Cristian S Calude**[2], **Michael J Dinneen**[2] and **Nan Huang**[2]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, Institut Néel, F-38000, Grenoble, France
[2] Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand

E-mail: alastair.abbott@neel.cnrs.fr

Received 6 July 2018, revised 29 October 2018
Accepted for publication 23 November 2018
Published 1 February 2019

CrossMark

Abstract

The advantages of quantum random number generators (QRNGs) over pseudo-random number generators (PRNGs) are normally attributed to the nature of quantum measurements. This is often seen as implying the superiority of the sequences of bits themselves generated by QRNGs, despite the absence of empirical tests supporting this. Nonetheless, one may expect sequences of bits generated by QRNGs to have properties that pseudo-random sequences do not; indeed, pseudo-random sequences are necessarily computable, a highly nontypical property of sequences. In this paper, we discuss the differences between QRNGs and PRNGs and the challenges involved in certifying the quality of QRNGs theoretically and testing their output experimentally. While QRNGs are often tested with standard suites of statistical tests, such tests are designed for PRNGs and only verify statistical properties of a QRNG, but are insensitive to many supposed advantages of QRNGs. We discuss the ability to test the incomputability and algorithmic complexity of QRNGs. While such properties cannot be directly verified with certainty, we show how one can construct indirect tests that may provide evidence for the incomputability of QRNGs. We use these tests to compare various PRNGs to a QRNG, based on superconducting transmon qutrits and certified by the Kochen–Specker theorem, to see whether such evidence can be found in practice. While our tests fail to observe a strong advantage of the quantum random sequences due to algorithmic properties, the results are nonetheless informative: some of the test results are ambiguous and require further study, while others highlight difficulties that can guide the development of future tests of algorithmic randomness and incomputability.

Keywords: quantum randomness, algorithmic randomness, incomputability, quantum random number generators

(Some figures may appear in colour only in the online journal)

1. Introduction

Randomness is an important resource in a diverse range of domains: it has uses in science, statistics, cryptography, gambling, and even in art and politics. In many of these domains, it is crucial that the randomness be of high quality. This is most directly the case in cryptography, where good randomness is vital to the security of data and communication, but is equally, albeit more subtly, true in other areas such as politics, where decisions of consequence may be made based on scientific and statistical studies relying crucially on randomness.

For a long time, people have predominantly relied on pseudo-random number generators (PRNGs)—that is, computer algorithms designed to simulate randomness—to serve such needs. Problems with various PRNGs, often only uncovered when it is already too late, are all too common and can have serious consequences[3]. This has driven a recent surge of interest in RNGs exploiting physical phenomena, and more particularly in quantum RNGs (QRNGs) that utilise the inherent randomness in quantum mechanics [2–5]. QRNGs are generally considered to be, by their very nature, better than classical RNGs (such as PRNGs), but how (or can) one test this in practice?

In fact, as we will see, there are two distinct aspects of randomness we may consider when asking this question. Firstly, one may look at the randomness of the process upon which a QRNG operates. Testing this aspect of randomness purely from the output of a device (rather than *a priori* considerations of its workings) is extremely difficult; classically it is not even possible and it is remarkable that, if one has access to an initial source of randomness, it is possible for certain QRNGs [4, 6]. Nonetheless, such tests are extremely demanding, often require additional assumptions and are applicable only to QRNGs designed for such tests.

In practice, instead, one is restricted to studying the randomness of individual strings produced by QRNGs by, e.g. conducting batteries of tests on (finite) sequences they have produced [7, 8], and it is this problem that we concern ourselves with here. Although this is a different aspect of randomness than that of the process itself, QRNGs should also provide an advantage here: PRNGs are guaranteed to produce computable sequences in stark contrast to the incomputability of QRNGs [9–12]. Standard tests, however, have focused on intuitive aspects of randomness, such as the frequencies of certain (strings of) bits, but human intuition about randomness is notoriously poor [13, 14] and many other symptoms of randomness remain untested. Indeed, the randomness of strings and sequences is an incomputable property and thus cannot be verified completely; moreover, it is characterised by an infinity of properties [15]. One may wonder, however, whether there are tests more appropriate for analysing QRNGs and perceiving the advantage they can provide.

With this goal, we formulate and study several tests of randomness based on algorithmic information theory. In particular, we consider tests based on Borel normality [16, 17] as well as novel tests based on the Solovay–Strassen probabilistic primality test [18, 19]—an algorithm which can be made deterministic when given access to algorithmic randomness [20]. These latter tests allow one to probe indirectly the algorithmic randomness—and consequently also the incomputability—of sequences produced by a RNG, and thus have the potential to identify differences between QRNGs and PRNGs that are not captured by more traditional statistical tests. We test several classical RNGs as well as a semiconductor-based QRNG [21] using these tests to assess their utility for real RNGs. While the first few tests we consider fail to find any significant difference between the quantum random sequences and those produced by the PRNGs, they bring to light certain issues useful for guiding future tests of algorithmic randomness and incomputability. Our final test finds some significant differences between the QRNG and PRNGs, but it is unclear whether these are really due to algorithmic properties of the strings; limitations of this test mean that further study is needed.

2. Randomness

In order to guide the development of tests for QRNGs, it is important to understand what randomness is and thus what one should test. Historically, the quest to develop a formal understanding of randomness focused on the problem of determining whether a given (finite) string or (infinite) sequence of bits is random. One of the first attempts to formalise such a notion of randomness is due to Borel, who defined the concept of *Borel normality* for infinite sequences [16]. Borel normality formalises the notion that bits should be evenly and equally distributed within a sequence. Although this captures one of the most intuitive features of randomness, it does not alone capture fully the desired concept. For example, the *Champernowne sequence* 01 00 01 10 11 000 001 011 100... [22] contains every string of length k with the same limiting frequency of 2^{-k}, and yet the sequence has a simple description: concatenate the binary representation of all the strings of length k in lexicographical order for $k = 1, 2, \ldots$. Given this description, it is clear that the Champernowne sequence is not random, but highly ordered.

The study of algorithmic information theory, developed in the 1960s by Solomonoff, Kolmogorov and Chaitin, provides more robust and acceptable definitions of a random sequence. In this framework, random strings and sequences are those that are incompressible [23]. The incompressibility of strings depends on the choice of universal Turing machine; this shortcoming disappears when the definition is extended to infinite sequences [15, 24]. Notions of randomness—both for finite strings and infinite sequences—defined in terms of incompressibility are generically called *algorithmic randomness*.

Let us briefly give some technical details useful later in the paper; we refer the reader to [15] for further details. Consider Turing machines operating on binary strings. A Turing machine U is universal if for every Turing machine M there exists a prefix p (depending only on U and M) such that $U(px) = M(x)$, for every program x. The *Kolmogorov* (or *algorithmic*) *complexity* of a Turing machine M is defined by $K_M(x) = \inf\{|s|: M(s) = x\}$, where by $|s|$ we denote the length of the string s. We can see that U is universal if and only if for every Turing machine M there exists a constant c such that $K_U(x) \leqslant K_M(x) + c$, for every string x. For this notion of complexity, the running time and the amount of storage required for computation are irrelevant. One can prove that for every M the maximum value of $K_M(x)$ over all strings x of a fixed length $|x| = n$ is $n + O(1)$. Furthermore, the overwhelming majority of strings x of length n have $K_M(x)$ very close to n. This means that almost all strings of length n are incompressible by M: more formally, very few such strings have $K_M(x) < n$ (i.e. are compressible). If U is a

[3] An example is the discovery in 2012 of a weakness in the encryption system used worldwide for online shopping, banking and email; the flaw was traced to the numbers a PRNG had produced [1]. As of 2018, Java still relies on a linear congruential generator, a low quality PRNG.

universal Turing machine, then the condition $K_M(x) < |x|$ means that $K_U(x) < |x| - c$, that is, x is *c-incompressible* (or *c-Kolmogorov random*). These incompressible strings are highly random, patternless and typical. It is easy to prove that less than 2^{n-c} strings of length n are not c-incompressible. An infinite sequence $\mathbf{x}$ is called *Martin-Löf random* if there exists a constant C such that infinitely many prefixes of $\mathbf{x}$ are C-Kolmogorov random. This definition is equivalent to the condition that $\mathbf{x}$ passes all Martin-Löf tests of randomness [25]; see section 7.2 for more details.

While algorithmic information theory provides a sound notion of randomness for strings and sequences, two important points must be mentioned. Firstly, it is not effectively decidable whether a string or sequence is random, so the notion does not provide a practical way to test the randomness of a finite or infinite sequence of bits. Secondly, it is possible to define ever stronger notions of randomness: from an algorithmic perspective, no notion of 'true', 'perfect' or 'absolute' randomness exists, only degrees of randomness [15, 26, 27]. This should temper any desire to verify the randomness of a RNG by tests on its output. Instead, we can only hope to compare the quality of strings produced.

As interest in *generating* random numbers soared, the concept of randomness received increased philosophical attention and it became clearer that the algorithmic notion of randomness fails to capture aspects of randomness important for RNGs [28]. Indeed, as von Neumann noted, 'there is no such thing as a random number—there are only methods to produce random numbers' [29]. The insight of von Neumann is not that the algorithmic notion of randomness is problematic—indeed, it is highly satisfactory as a notion of random *objects*—but that there is a dual concept of randomness, that of random *processes* [28, 30, 31]. Such a concept has historically received little attention, but the most convincing attempts to make it rigorous are perhaps those which define it as a form of maximal unpredictability: the outcome of such a process should be unpredictable for any physical observer [32, 33].

The randomness of a process is often quantified in terms of entropy, but it is important to note that, entropy being a function of the probability distribution associated to a process, such a quantification requires (i) knowing that the process is indeed unpredictable, and (ii) knowing the probability distribution modelling its behaviour. Although one can empirically estimate the distribution from the output of a RNG, the entropy calculated from such data can only be interpreted as a measure of randomness if (i) is satisfied, and this cannot be directly verified from the empirical data alone.

There are thus two legitimate notions of randomness to be reconciled: that of *process randomness* (which is applicable to RNGs—viewed as processes—themselves), and that of *product randomness* (which is applicable to the strings—i.e. objects—obtained from RNGs). The distinction between these notions is important for understanding tests of randomness.

3. Random number generators (RNGs)

An ideal random number generator is normally taken to be a random process producing the same probability distribution as the ideal (but unphysical) unbiased coin. It thus produces bits sequentially, thereby generating a sequence $\mathbf{x} = x_1x_2\ldots$ with each bit x_i being equiprobable, i.e. $p(x_i = 0) = p(x_i = 1) = 1/2$, and with successive bits produced independently. Hence, all strings x of length k have probability $p(x) = 2^{-k}$ and, in the infinite limit, one obtains the Lebesgue measure over all infinite sequences [15]. This type of ideal source has maximal entropy. It is important to recognise that this conception of an ideal RNG embodies the notion of random processes, not products, and concerns the distribution produced by said process and not its output.

If one tries to implement such a device in practice, two issues immediately become apparent.

Firstly, how is one to know that the process exploited is really random and actually produces the expected ideal distribution? This issue touches on the interpretation of probability [34] (although this is beyond the scope of the present article). For example, a physical process thought to be represented by the uniform distribution might only exhibit epistemic randomness, and a more precise, deterministic model of the process might be possible which reveals its non-randomness. The most direct way to avoid such possibilities is to harness an indeterministic process to ensure its unpredictability [33].

Secondly, how does one test or verify the randomness of a RNG given that one only has access to (finite) strings produced by it? Although the concepts of process and product randomness are indeed distinct, they are nonetheless related: long enough strings produced by an ideal RNG will, with high probability, be incompressible, while in the infinite limit the sequences produced will be Martin–Löf random (and thus also incomputable) with probability 1 *but not with certainty*: an ideal coin can in principle produce non-random or even computable sequences. However, as mentioned earlier, the randomness of sequences is already an incomputable property. Thus, one can do no better than verifying finitely many properties of randomness to gain confidence in a RNG.

3.1. Pseudo RNGs (PRNGs)

The predominant approach to generating randomness is to use algorithms to produce 'pseudo-randomness', and such PRNGs are ubiquitous as a result of their practicality and speed. However, the very fact that such devices use computational methods to produce their outcomes distinguishes them from ideal RNGs. PRNGs typically use a short string from an external source—generally assumed to be random—as an initial 'seed' for an algorithm [35]. Thus, PRNGs can only produce computable sequences, whereas such sequences should be produced only with probability 0 by an ideal RNG. Instead, effort is made to make PRNGs difficult to distinguish from an ideal RNG given limited (typically polynomial time) computational resources [36], so that the PRNG appears to be a high entropy source. This provides a degree of security against cryptographic attacks, even if the resulting distribution (induced by the distribution over the initial seeds) is far from uniform in reality.

PRNGs generally produce sequences that satisfy many intuitive aspects of randomness—such as the equidistribution of the bits produced—and pass most standard statistical tests of randomness despite their computability. Nonetheless, deficiencies resulting from the non-randomness of PRNGs are regularly exploited (see, e.g. [37]) and much of the interest in quantum randomness has been driven by the potential to avoid the shortcomings of PRNGs.

4. Quantum randomness

For some time now, quantum mechanics has garnered interest as a potential source of randomness for RNGs. Such interest stems from the fact that certain quantum phenomena, such as the radioactive decay of an atom or the detection of a photon having passed through a beamsplitter, are generally taken to be 'intrinsically random' under the standard interpretation of quantum mechanics [38]. We will first discuss these claims in a little more detail—since it is important to base the randomness of QRNGs on more formal grounds rather than simply assuming such randomness—before discussing one approach to the generation of quantum randomness in more detail.

Claims about quantum randomness originate with the fact that, as a formal theory, quantum mechanics differs fundamentally from classical physics in that not all observable properties are simultaneously defined with arbitrary precision. Instead, quantum mechanics, via the Born rule, only specifies the probabilities with which individual measurement outcomes occur for the measurement of a physical quantity—i.e. a quantum *observable*. Formally, if a system is in a quantum state $|\psi\rangle$ and one measures an observable A with spectral decomposition $A = \sum_i a_i P_i$, where we adopt the notation $P_i = |i\rangle\langle i|$ for rank-1 projection observables, then one obtains outcome a_i with probability

$$P(a_i|\psi) = |\langle i|\psi\rangle|^2. \qquad (1)$$

Thus, whereas randomness in classical physics is due to incomplete knowledge of the precise initial conditions of a system (e.g. as in chaotic systems) [39], in quantum mechanics it is intrinsic to the standard interpretation of the formal theory.

Nonetheless, the Born rule is a purely formal statement, and interpreting the probability distribution specified by the Born rule remains the subject of ongoing debate. The orthodox interpretation, however, is that the distribution should be understood ontically as representing an indeterministic phenomenon [38]. Crucially, this interpretation is more than a mere assumption: several well-known no-go theorems rule out classical statistical interpretations of quantum randomness.

Bell's theorem [40] is the most well-known of these results, and shows that a classical, local hidden variable theory cannot reproduce the statistics of quantum correlations that are observed [41] between entangled particles. The Kochen–Specker theorem [42], although perhaps lesser known, pinpoints this breakdown indeterminism in a more precise way: it shows that, for any quantum system with more than two dimensions, it is logically impossible to predetermine all measurement outcomes prior to measurement in a noncontextual fashion (i.e. in a way which is independent of other compatible—and thus non-disturbing—measurements one can perform).

More recently, this theorem has been refined to show that the only observables that can be predetermined in a noncontextual way are those for which the Born rule assigns the probability 1 to a particular outcome [43, 44]. More precisely, we say that an observable A is *value definite* for a system prepared in a state $|\psi\rangle$ if it has a predetermined measurement outcome $v_\psi(A)$. The stronger result shows that for systems of more than two dimensions, if we assume that any such value definite observables should be noncontextual, then A is value definite if and only if $|\psi\rangle$ is an eigenstate of A; all other observables must be *value indefinite*.

This result makes the extent of quantum value indefiniteness—and thus indeterminism—clear and pinpoints which measurements are protected by such formal results. This not only allows some QRNGs to be based more rigorously on physical principles but also to clarify the link between quantum randomness and indeterminism. Crucially, this result also allows one to show that the measurement of such value indefinite observables satisfies a strong form of unpredictability [12], proving that one really cannot provide better predictions than the Born rule specifies, and thus giving a stronger theoretical grounding to claims about the form of quantum randomness proposed for QRNGs.

5. Quantum RNGs (QRNGs)

These properties of quantum measurements make them an ideal candidate for random number generation: if one measures an observable for which the Born rule predicts a uniform distribution, then the QRNG embodies a perfect coin. Moreover, the results discussed above show that—subject to very reasonable physical assumptions about how classical objects should behave—this distribution cannot be given an epistemic interpretation and the corresponding measurement outcomes are thus truly of indeterministic origin. The attractiveness of QRNGs is further enhanced by the possibility of obtaining high bitrates and the simplicity of their physical models. This is in contrast to RNGs based on classical physics, such as chaotic systems.

Early QRNGs relied on features such as radioactive decay [45], but simpler systems based, for example, on measuring the polarisation [3, 46, 47] or detection times [48] of photons, have become the norm due to the practical advantages they provide. Such approaches have led to the

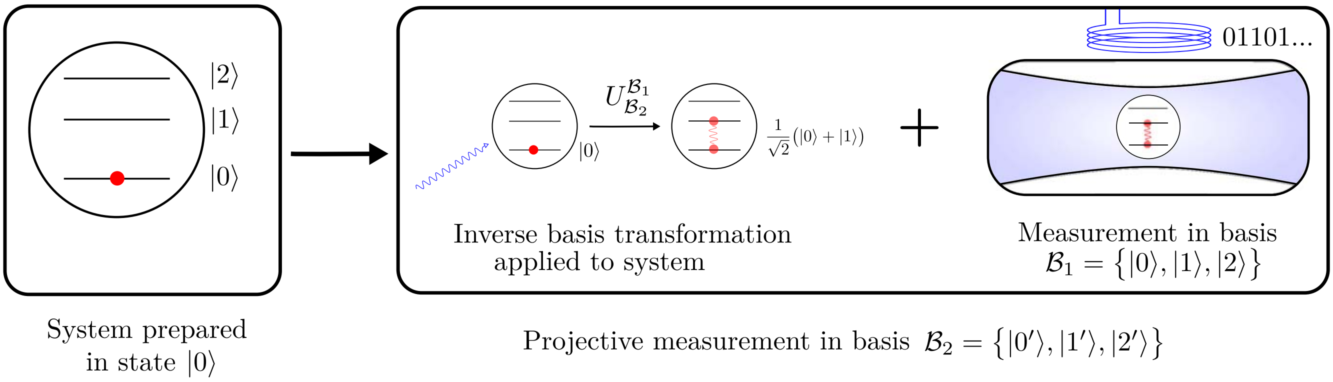


Figure 1. Schematic showing the QRNG based on the Kochen–Specker theorem as implemented in [21]. A transmon qutrit system is initially prepared in the state $|0\rangle$ (with respect to the computational basis $\mathcal{B}_1 = \{|0\rangle, |1\rangle, |2\rangle\}$ by thermal cooling. The system is then measured in the basis $\mathcal{B}_2 = \{|0'\rangle, |1'\rangle, |2'\rangle\}$ with $\langle 0|0'\rangle = 0$ and $\langle 0|1'\rangle = \langle 0|2'\rangle = \frac{1}{\sqrt{2}}$. In practice, this measurement is performed by first performing the inverse basis transformation on the system and measuring in the basis $\mathcal{B}_1$. Since $|\langle 0|0'\rangle|^2 = 0$, this outcome never occurs in an ideal implementation, so the outcomes a_1 and a_2 corresponding to $|1'\rangle\langle 1'|$ and $|2'\rangle\langle 2'|$ are mapped to a binary sequence.

development of commercial QRNGs, such as ID Quantique's Quantis [49].

Many successful QRNGs exploit two-dimensional systems to generate randomness (e.g. Quantis uses the polarisation of photons). This greatly simplifies the design and production of such devices but neither Bell's theorem (which requires entanglement) nor the Kochen–Specker theorem (which requires at least three-dimensional systems) are applicable, and these QRNGs thus lack the rigorous theoretical certification that quantum mechanics can provide, even if it may be reasonable to think that the measurements they exploit should still be indeterministic.

The most direct approach to overcoming this shortcoming is to use higher dimensional systems for which the value indefiniteness of measurement outcomes is, via the Kochen–Specker theorem, provable [9, 10] to certify a QRNG. Such certification is necessarily 'device dependent'—that is, it relies on knowledge of the functioning of the QRNG—but nonetheless allows the randomness of the device to be more formally grounded. A simple example of a QRNG certified in this way was proposed in [9, 10] for spin-1 particles, but its principle is applicable to any three-dimensional system (i.e. an implementation of a qutrit). The approach proposed was to prepare a qutrit in the state $|0\rangle$ before measuring the observable $A = a_0|0'\rangle\langle 0'| + a_1|1'\rangle\langle 1'| + a_2|2'\rangle\langle 2'|$ for which the orthonormal basis $\{|0'\rangle, |1'\rangle, |2'\rangle\}$ is chosen such that $\langle 0|0'\rangle = 0$ and $\langle 0|1'\rangle = \langle 0|2'\rangle = \frac{1}{\sqrt{2}}$ (see figure 1). Since the state $|0\rangle$ is thus an eigenstate of the projection observable $P_{0'} = |0'\rangle\langle 0'|$, this observable is value definite with value $v(P_{0'}) = 0$—that is, the measurement outcome a_0 never occurs[4]. However, by the results of [10, 44], both $P_{1'} = |1'\rangle\langle 1'|$ and $P_{2'} = |2'\rangle\langle 2'|$ are value indefinite and, moreover, both outcomes a_1 and a_2 occur with probability $1/2$ according to the Born rule (1). Thus, the QRNG operates as an ideal coin certified by value indefiniteness.

A QRNG based on this proposal has recently been implemented experimentally [21], not with spin-1 particles but by exploiting a superconducting transmon coupled to a microwave cavity as a qutrit. Figure 1 shows a schematic of the QRNG proposed in [9, 10] based on the implementation used by Kulikov *et al* [21]. This implementation was used to generate a large number of bits, and in the subsequent sections we will analyse sample sequences produced by this QRNG implementation. In particular, we will look to detect differences between such sequences and pseudo-random sequences arising from algorithmic properties of the sequences.

This approach to certifying a QRNG via value indefiniteness implies some additional interesting algorithmic properties of the output sequences of the device if one is willing to accept slightly stronger physical assumptions (in particular, about whether being able to compute properties in advance implies well-defined physical properties). Specifically, it was shown in [10] that such a device, if used repeatedly *ad infinitum* to generate an infinite sequence $\mathbf{x}$ of bits, will produce a sequence that is strongly incomputable (technically, 'bi-immune' [24]) not just with probability 1, but *with certainty*. Although such a result will not alone lead to observable advantages for finite strings—recall that, from the Born rule, an ideal QRNG will produce an incomputable sequence with probability 1—this nonetheless highlights the differences between pseudo and quantum randomness in relation to computability.

More recently there has also been growing interest in a different type of QRNG which can provide a stronger form of certification but requires initial random seeds as input. (Such devices are thus technically randomness expansion devices, rather than RNGs.) Typically, such devices rely on violating a Bell inequality, which allows one to certify that the QRNG indeed uses a value indefinite system without assuming *a priori* anything about the workings of the device [4, 6, 50]. This type of certification is thus termed 'device independent', and allows one to place lower bounds on the entropy of the

[4] This is, of course, only true in the ideal case. In the non ideal scenario, any such outcomes can simply be discarded.

source; it is particularly important in cryptographic settings, where one perhaps does not wish to trust the workings of a given RNG. Such schemes are very costly, however: not only is an initial random seed required, but one also must separate the QRNG into two space-like separated (or at least isolated) components and the stringent requirements of loophole-free Bell tests reduce the obtainable bitrate by several orders of magnitude compared to 'standard' QRNGs [4].

Other related randomness expansion schemes have also been proposed which are less experimentally demanding but require additional physical assumptions [51]. In particular, we note that 'noncontextuality inequalities' [52, 53] obtainable from proofs of the Kochen–Specker theorem can be used to provide such a certification [54–56]. In doing so, rather than trusting outright that the QRNG uses a system in which the Kochen–Specker theorem applies, one actively verifies this under weaker physical assumptions about the workings of the device. Nonetheless, such schemes are still significantly more demanding than that described in figure 1. Here we thus focus on the similar device-dependent type scheme described in detail above. Indeed, our focus is on testing the algorithmic properties of individual strings produced by a QRNG; such tests are complementary to those aimed at certifying the indeterministic nature of the process itself, and the simplicity of this scheme, along with its high bitrate, facilities such an analysis.

6. Testing RNGs

While it is crucial to have a good theoretical understanding of any RNG, there are several reasons why testing experimentally their outputs is nonetheless crucial. Firstly, one can never be sure that the implementation of a RNG matches its theoretical description, a fact that is equally as true for hardware RNGs as for software RNGs. Indeed, in the extreme limit, one might not wish to trust any theoretical claims about a given RNG, and thus confidence in the RNG can only be gained from performing carefully selected tests. Secondly, thorough testing gives one the opportunity to detect any issues with assumptions made in the theoretical analysis of a device or in its practical deployment (e.g. if the distribution of seeds does not match that assumed theoretically the performance of a RNG might be compromised).

It is nonetheless important to recognise that experimental testing can never allow one to perfectly characterise a device. Instead, with access to only finite strings produced by it and the ability to perform a finite number of tests, one can only ever gain increasing confidence in the operation of the device. One can never be certain, for instance, that the output obtained was not a simply atypical behaviour obtained purely by chance. This is doubly true since, as we discussed earlier, randomness is characterised by an infinity of properties, so one must carefully choose the tests one performs.

The issues arising when testing the outputs of RNGs can be illustrated pointedly with an example. Imagine a device which deterministically outputs the digits of the binary expansion of $\pi = \pi_1\pi_2\pi_3...$ starting from the 10^{10}th bit. If we are unaware of the behaviour of this device and believe it to be a RNG, its output will appear extremely random to us; indeed, π passes all standard statistical tests of randomness [57] despite the fact that it is not even known to be Borel normal [58, 59]. Nevertheless, the sequence produced by this box would be computable and thus not random at all. Similarly, any attempt to estimate the entropy of the source from the empirically observed distribution would lead one to believe the source to be a highly random process, despite the fact it is completely deterministic.

Standard statistical tests of randomness focus on properties of the distribution of bits or bit strings within sequences, properties more closely related to Borel normality than algorithmic complexity. Many such tests were developed with the aim of testing PRNGs, where reproducing such statistical predictions is a primary issue, particularly since failing to do so may leak information about the seed and thus break the security of the PRNG [1]. QRNGs[5] have generally been tested against similar tests, such as the NIST [8] and DIEHARD [7] batteries, and generally perform well. For example, Quantis is officially certified as passing these tests on 1000 samples of one million bits [49]. Such tests, however, far from confirm the randomness of the device; indeed, analysis of longer sequences (of 2^{32} bits) revealed (albeit it very small) bias and correlation amongst the output bits [60].

Such statistical non-uniformity is, however, to be expected in RNGs exploiting physical phenomena due to experimental imperfections and instability [9]. Inasmuch as this form of non-uniformity is small enough for the required application, this is not necessarily problematic as long as a QRNG remains certified by value indefiniteness: unlike for PRNGs, where non-equidistribution is often a symptom of deeper issues, the unpredictability of QRNGs is a result of the indeterministic nature of the device, and is thus assured even if the resulting distribution is biased [12]. Moreover, bias can be reduced by careful post-processing [29, 61, 62], allowing quantum indeterminism to still be exploited sufficiently. Although testing such properties is crucial in order to ensure any bias remains tolerably low, such tests do not directly probe crucial advantages of quantum randomness, such as the degree of algorithmic randomness or incomputability of their output.

Some authors have also looked at the compressibility of quantum random sequences using standard compression algorithms [63] as a proxy for direct tests of Kolmogorov complexity. In practice, however, just like the aforementioned tests, this approach fundamentally relies on statistical properties of the sequence and suffers from similar problems as the above tests (such as being fooled by computable sequences). Indeed, it is not possible to directly compute the Kolmogorov complexity since it is an incomputable quantity. Nevertheless, one may still ask whether there are useful tests that indirectly probe this to try and differentiate

[5] As we discussed in the previous section, we restrict our discussion henceforth to standard QRNGs, rather than device-independent randomness expansion schemes. These devices remain the standard approach to QRNGs in practice, and developing tests for their output remains a crucial problem despite the increased interest in device-independent QRNGs.

PRNGS—which always produce computable sequences—from QRNGs [19]. In the following sections we investigate more closely this question.

7. Experimentally testing for evidence of incomputability and algorithmic randomness

In this section we describe several tests based on algorithmic properties which we use to study random bits obtained from both PRNGs and the QRNG detailed in figure 1. We tested 80 sequences of 2^{26} bits[6] obtained from each of the following six sources: the initial bits of the binary representation of π (which can be seen as a form of pseudo-randomness [59]), the PRNG used by Python v3.5.4 (a Mersenne Twister algorithm) [64], Random123 v1.09 [65], PCG v0.98 [66], xoroshiro128 + [67], and the QRNG described in section 5 (see [21]).

7.1. Tests of Borel normality

As mentioned earlier, the notion of Borel normality was the first mathematical definition of algorithmic randomness [16], and although, like many standard tests of randomness, it focuses on the distribution of bits within a sequence, it is nonetheless worth looking at in its own right.

An infinite sequence $\mathbf{x} \in \{0, 1\}^{\infty}$ is (Borel) normal if every binary string appears in the sequence with the right frequency (which is 2^{-n} for a string of length n). Every Martin-Löf random infinite sequence is Borel normal [15], but the converse implication is not true: there exist computable normal sequences, such as Champernowne's sequence mentioned earlier. Normality is invariant under finite variations: adding, removing, or changing a finite number of bits in any normal sequence leaves it normal.

The notion of normality was subsequently transposed from infinite sequences to (finite) strings [15]. In doing so, one has to replace limits with inequalities, and one obtains the following definition. For any fixed integer $m > 1$, consider the alphabet $B_m = \{0, 1\}^m$ consisting of all binary strings of length m, and for every $1 \leqslant i \leqslant 2^m$ denote by N_i^m the number of occurrences of the lexicographical ith binary string of length m in the string x (considered over the alphabet B_m). By $|x|_m$ we denote the length of x over B_m; $|x|_1 = |x|$. A string $x \in B_m$ is *Borel normal* (*with accuracy* $\frac{1}{\log_2 |x|}$) if for every integer $1 \leqslant m \leqslant \log_2 \log_2 |x|$ and each $1 \leqslant j \leqslant 2^m$ we have:

$$\left| \frac{N_j^m(x)}{|x|_m} - 2^{-m} \right| \leqslant \frac{1}{\log_2 |x|}. \quad (2)$$

Almost all algorithmic random strings are Borel normal with accuracy $\frac{1}{\log_2 |x|}$ [15]; in particular, they have approximately the same number of 0 and 1 s. Furthermore, if all prefixes of a sequence are Borel normal, then the sequence itself is also Borel normal.

[6] The sequences were obtained from 10 longer sequences of 2^{29} bits, each obtained during separate experimental runs. We split them further into smaller sequences in order to provide a more detailed statistical analysis.

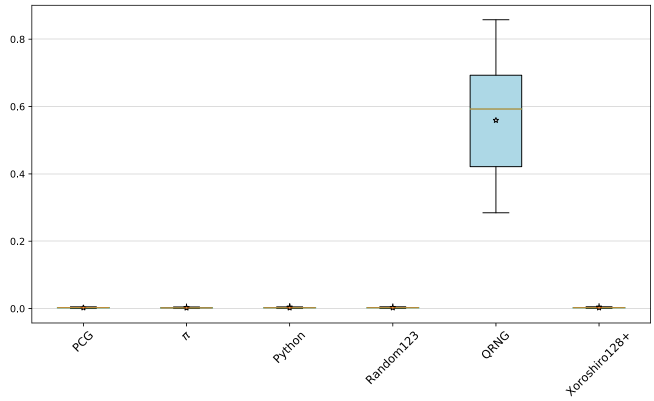


Figure 2. Borel normality test: box-plot showing the distribution of the quantity $\max\left(\left| \frac{N_j^m(x)}{|x|_m} - 2^{-m} \right|\right)\log_2|x|$ for the 80 strings of length $|x| = 2^{26}$ bits produced by each the six RNGs tested.

The fact that Borel normality for finite strings is only defined up to the accuracy function arises from the fact that the definition is well behaved (and converges to the definition for sequences in the limit) if the right-hand-side of equation (2) is replaced by any decreasing computable real function in $|x|$ converging to 0. Fixing a specific accuracy function allows one to test explicitly the normality of finite sequences (and such tests have previously been performed on strings produced by QRNGs [19, 68, 69]), but such a choice of accuracy function is necessarily somewhat arbitrary. However, the relative normality of strings can be tested by comparing the values of a metric based on (2); a reasonable choice of such a metric is the quantity $\max\left(\left| \frac{N_j^m(x)}{|x|_m} - 2^{-m} \right|\right)\log_2|x|$ over the values $m = 1,\ldots,\lfloor \log_2 \log_2|x| \rfloor$ and each $1 \leqslant j \leqslant 2^m$. We recorded this metric for the six sources of random bits under consideration, and the resulting distributions are shown in figure 2.

The results show clearly that the bits produced by the QRNG are significantly less normal than those produced by the other sources. This is, however, not surprising, since the experiment implementing the QRNG was known to exhibit bias due to experimental imperfections [21]. Although it is possible to use normalisation procedures to unbias a source, simple techniques significantly reduce the length of the output strings (and thus the obtainable statistical power) and can alter various computability theoretic properties of a sequences [62]; conversely, the consequences of more complicated techniques on such properties of the output are poorly understood, so we opted against performing any such normalisation. Nonetheless, as discussed at the end of section 6, a sufficiently small bias may be less problematic in practical applications for QRNGs than for traditional PRNGs.

While examining the normality of sequences produced by any RNG is important, this algorithmic property fails to test properties of algorithmic randomness or incomputability in the way we aim to do. The example of Champernowne's sequence again testifies to this. To probe such behaviour of

QRNGs we thus need to delve further into algorithmic properties of randomness.

7.2. A Martin-Löf test of incomputability

Is it possible to give formally a test which rejects every computable sequence as non-random? Martin-Löf randomness is an important, if not the most important, form of algorithmic randomness and is based on the notion of Martin-Löf test of randomness. A test of randomness is defined by a uniformly computably enumerable shrinking sequence of constructive open sets in Cantor space (the components of the test) whose intersection is a constructive null set (with respect to Lebesgue measure); see [15] for more details. A sequence passes the test if it is not contained in this null set. A sequence is Martin-Löf random if it passes all Martin-Löf tests. There exist countably many such tests: some test normality, others test the law of large numbers, etc. The answer to the question above is affirmative: such a Martin-Löf test exists.

Testing incomputability rather than randomness directly is an important initial step: indeed, all algorithmically random sequences are incomputable and it is this property of randomness that PRNGs fail most starkly, since they necessarily produce computable sequences. Moreover, the robustness of incomputability to bias in a sequences makes such tests potentially more robust in practice. To specify a Martin-Löf test for computability, we must define the sequences contained in its nth component for all integers $n > 0$. To do so, one can take the nth component to be the union of all $\sigma\{0, 1\}^*\{0, 1\}^\infty$ for which there is an e such that $\sigma(0) = \varphi_e(0)$, ..., $\sigma(e + n + 1) = \varphi_e(e + n + 1)$ and $\sigma \in \{0, 1\}^*$. This is an open computably enumerable class that contains all computable sets, as each computable set has a computable characteristic function φ_e. Furthermore, the measure of the nth component is bounded from above by $\sum_e 2^{-n-e-2}$, which in turn is bounded from above by 2^{-n-1}, as the string σ derived from φ_e has length $e + n + 2$ and is a prefix of the set for which φ_e computes the characteristic function.

It is not difficult to see that the above test for computability depends on the enumeration (φ_e), and there is no obvious 'natural' choice. Furthermore, invariance under finite variations renders the test unsuitable for finite experiments. As a result, it is necessary to consider more indirect methods to test the incomputability of sequences produced by RNGs.

7.3. Chaitin–Schwartz-Solovay–Strassen tests

In this section we propose and carry out several related tests based on a rather different property of random sequences: their ability to de-randomise the Solovay–Strassen probabilistic test of primality [18]. In contrast with most standard tests of randomness which check specific properties of strings of bits, these tests are based on the behaviour of the strings with respect to certain 'secondary' tasks. We first briefly describe the Solovay–Strassen primality test and the advantage offered in this task by random strings, before presenting the tests themselves.

The Solovay–Strassen test checks the primality of a positive integer n: take k natural numbers uniformly distributed between 1 and n–1, inclusive, and, for each $i(= i_1, \ldots, i_k)$, check whether a certain, easy to compute, predicate $W(i, n)$ holds (W is called the Solovay–Strassen predicate). If $W(i, n)$ is true then 'i is a witness of n's compositeness', hence n is composite. If $W(i, n)$ holds for at least one i then n is composite; otherwise, the test is inconclusive, but in this case the probability that n is prime is greater than $1 - 2^{-k}$. This is due to the fact that *at least half* the i's between 1 and $n - 1$ satisfy $W(i, n)$ if n is composite, and *none* of them satisfy $W(i, n)$ if n is prime [18].

Chaitin and Schwartz [20] proved that, if c is a large enough positive integer and s is a long enough c-Kolmogorov random binary string, then n is prime if and only if $Z(s, n)$ is true, where Z is a predicate constructed directly from $O(\log n)$ conjunctions of negations of W predicates (see section 7.3.3 below for more details). The crucial fact is that the set of c-Kolmogorov random strings is highly incomputable: technically the set is immune, that is, it contains no infinite computably enumerable subset [15]. As a consequence, derandomisation is thus non-constructive, and thus without practical value.

Drawing on this result, we propose several tests that operationalise it in order to test the randomness of a sequence based on whether certain numbers obtained from RNGs succeed in witnessing the compositeness of well chosen targets. We will make particular use of Carmichael numbers as these target composites. A Carmichael number is a composite positive integer n satisfying the congruence $b^{n-1} \equiv 1 (\mathrm{mod}\ n)$ for all integers b relatively prime to n. Although Carmichael numbers are composite, they are difficult to factorise and thus are 'very similar' to primes; they are sometimes called pseudoprimes. Many Carmichael numbers can pass Fermat's primality test, but less of them pass the Solovay–Strassen test. Increasingly Carmichael numbers become 'rare'[7].

In what follows we thus present four different tests based on the Chaitin–Schwartz theorem and the Solovay–Strassen test. Since the proposed tests rely directly on the algorithmic randomness of a string, they can potentially give direct empirical evidence of incomputability, in stark contrast to most tests of randomness. For example, the Borel normality test discussed previously is unable to do so: the normality of Champernowne's sequence mentioned earlier is evidence of this.

Since the Chaitin–Schwartz theorem relies on the Kolmogorov randomness of the sequences it involves, these tests also go beyond the previous one in not only probing incomputability, but also algorithmic complexity more generally. Indeed, an ideal QRNG should produce c-Kolmogorov random strings with very high probability, while PRNGs produces strings of very low Kolmogorov complexity (since, in the limit, they are computable). Nonetheless, we focus on probing the incomputability of strings from QRNGs rather than their Kolmogorov complexity or randomness, a doubly motivated choice. Firstly, the fact that incomputability is a

[7] There are 1401 644 Carmichael numbers in the interval $[1, 10^{18}]$.

weaker property than Kolmogorov randomness and less affected by bias means that any difference between pseudo and quantum randomness will potentially be easier to observe. Secondly, as mentioned earlier, subject to an additional physical assumption, QRNGs can be shown to produce incomputable sequences with certainty, and not just probability one [10].

As in [19], we conduct various statistical tests to determine whether any observed difference is statistically significant or not. If a difference is found to be significant, we then look at whether this really provides evidence of incomputability or not. As it is not *a priori* clear what distribution the various test metrics we employ should follow, we utilise the non-parametric and distribution free Kolmogorov–Smirnov test for two samples [70] to determine whether two datasets differ significantly. This test returns a p-value[8] indicating the probability, given the observed test statistic, that the observed distributions were indeed drawn from the same distribution. We conclude that 'the difference between the two datasets is statistically significant' if the p-value is less than 0.005. We choose this relatively strict p-value to lower the chance of false positives arising from the fact that we will perform several tests between several different data sources: the probability of observing a spurious difference (simply by chance) on at least one of the many tests is much higher than the critical p-value of 0.005 of obtaining such a spurious result on any single test. A higher critical p-value (such as the commonly used 0.05) would mean such false positives would be highly probable.

When no significant difference is found by the Kolmogorov–Smirnov test, we additionally check whether the test metric distribution is consistent with a normal distribution by performing a Shapiro–Wilk test [71][9]; if it is[10], we then use the (parametric) Welch t-test [72], which is a version of Student's test, to determine whether there is a significant difference between the means of the test statistics for the different RNGs under the assumption of normally distributed test metrics.

7.3.1. First Chaitin–Schwartz-Solovay–Strassen test. The first test we look at, which was previously used in [19], probes directly the efficacy of a set of random bits in simulations (in our case for checking primality).

We performed this test on all of the 246 683 Carmichael numbers n with at most 16 digits as computed in [73], using strings of bits from each random source to specify the numbers tested as potential witnesses of compositeness. More precisely, for a fixed k (see below) and each Carmichael number n we take k strings of $\lceil \log_2 n \rceil$ bits from the source string and reject and resample those which specify the binary representation of a number greater than $n - 1$. These k strings, interpreted as the binary representation of k numbers $i_1, \ldots, i_k$, serve as the witnesses to test the primality of n (i.e. the i in $W(i, n)$). Initially we take $k = 1$ and increase k until all the Carmichael numbers are correctly determined to be composite.

The metric for the test is taken to be the smallest k such that at most k witness numbers were required to obtain a verdict of non-primality for all of the Carmichael numbers. For each k, new bits are read from the sample string for each Carmichael number to be tested; we only restart reading from the start of the string (and thus recycling bits) when there was a need to try a larger value of k to pass this test.

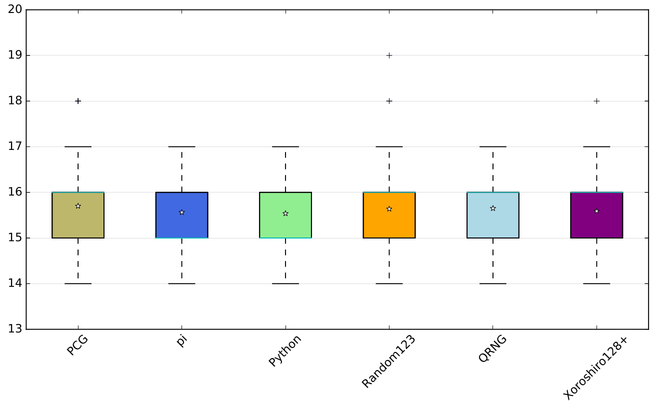


Figure 3. First Chaitin–Schwartz-Solovay–Strassen test on 80 samples: box-plot showing the distribution in the minimum number of witnesses needed to verify the compositeness of all Carmichael numbers of at most 16 digits.

Figure 3 shows the performance of the 80 bit strings from each RNG (i.e. the same ones as tested for Borel normality in section 7.1) using the metric described above.

The full results of the statistical analysis of this test (as well as the following) are given in the appendix. The Kolmogorov–Smirnov tests found no statistical significant difference between any of the sources of randomness (see table A1). The Shapiro–Wilk tests showed that the distribution of test statistics were not normally distributed (see table A2), so further parametric tests were not performed. This test therefore did not provide any evidence of significant differences between the RNGs, let alone evidence of incomputability of the QRNG.

7.3.2. Second Chaitin–Schwartz-Solovay–Strassen test. We next consider a closely related (and similarly motivated) test with a slightly different metric. For each Carmichael number n, we repeatedly obtain a witness from the string being tested (in the same manner as in the first test and using new bits for each Carmichael number) until the compositeness of n is successfully witnessed. For this test metric we take the total number of bits used (for a given string to test) to confirm the compositeness of all 16-digit Carmichael numbers. We calculate this as the sum, over all such Carmichael numbers n, of $\lceil \log_2 n \rceil$ times the number of Solovay–Strassen trials needed to witness the compositeness of n. (In this way, bits

[8] Exact p-values are only available for the two-sided two-sample tests with no ties.

[9] More precisely, the Shapiro–Wilk test examines the null hypothesis that the samples $z_1, \ldots, z_n$ come from a normally distributed population. This test is appropriate for small samples, since it is not an asymptotic test.

[10] Here we consider evidence for non-normality to be a p-value below 0.05.

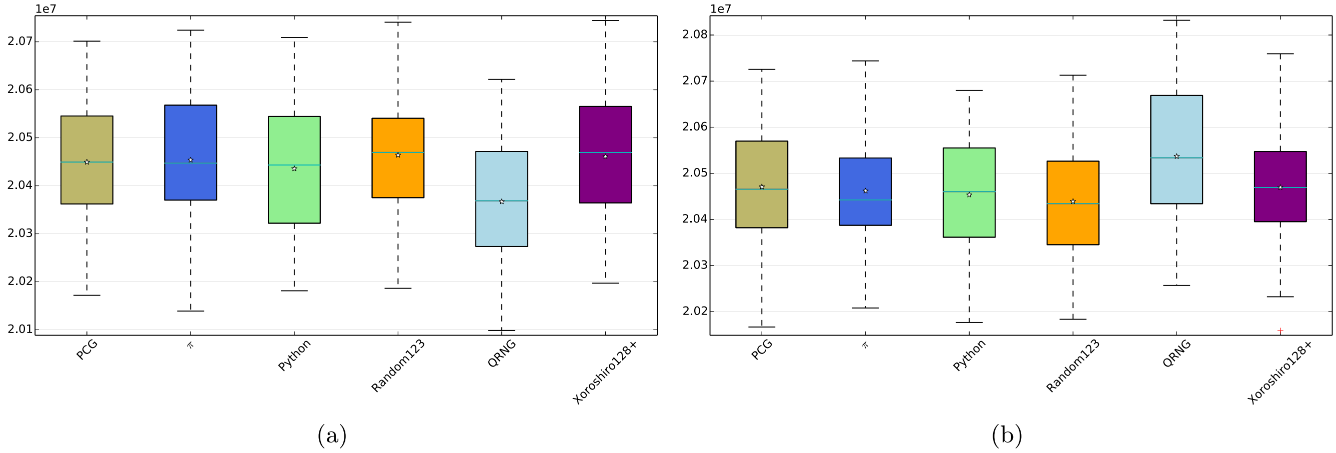


Figure 4. Second Chaitin–Schwartz-Solovay–Strassen test: total number of bits required to verify the compositeness of all Carmichael numbers of at most 16 digits using (a) the 80 strings from each RNG, and (b) the complement of these strings.

that are read but then rejected because they give a witness larger than n do not contribute to the metric.)

Figure 4(a) shows a boxplot of the results for the 80 strings from each RNG being tested. The visible difference between the QRNG and the other sources is confirmed by the Kolmogorov–Smirnov tests (see table A3), which showed a statistically significant difference between the QRNG and π, Random123 and xoroshiro128+. There is not, however, a general trend of normality for the test metric across all sources (in particular, there is weak evidence to reject normality of the distribution for the Python strings; see table A4), so it is not appropriate to use Welch's t-test to look for a difference between the QRNG and Python.

Although a significant difference was found between the QRNG and most the other sources, this is not necessarily a result of the incomputability we wish to test. Indeed, we have already seen from the Borel normality test that the QRNG has a small statistical bias, so we should thus verify that the difference seen here is not also a result of this bias. As mentioned earlier, we opted against trying to normalise the data to see if the bias is indeed to origin of the effect, not only because, with the amount of data available to us, this would markedly reduce our statistical power, but also because the effect of normalisation on the algorithmic complexity of the sequence is not entirely understood. Instead, a simple way to test whether bias is behind the observed differences is to perform the same test on the complement of the strings we have tested (i.e. exchanging 0 and 1). Since this transformation preserves randomness and incomputability, if the difference observed is evidence of such properties it should not be affected by such a transformation.

Figure 4(b) shows the result of the test on the complemented sequences. Here we see that again there is an apparent difference between the QRNG and some of the other sources. This is confirmed by the Kolmogorov–Smirnov tests (see table A5) to be the case between the QRNG and π, Python and Random123. In this case, the test metric is consistent with being normally distributed (see table A6), so it is reasonable to use Welch's t-test to try and confirm this difference further under an assumption of normality. Doing so (see table A7) shows that there is indeed a statistically significant difference between the QRNG and all the other sources on the complemented strings.

However, as is clear from figure 4(b), this difference is in the opposite direction to (and of the same magnitude as) that in figure 4(a): in the latter the QRNG appears to perform better, while in the former, it performs worse. It thus appears that this difference was indeed due to the bias of the QRNG rather than incomputability. Nonetheless, we note that it is strange that biased sequences (in particular, biased towards having more zeroes) perform better in proving the compositeness of Carmichael numbers; we are not aware of any number theoretic explanation for this.

To conclude, this test shows that the QRNG behaves significantly differently from almost all the other sources on this test (whether we use either the original bits or the complemented bits), but that this difference is likely due to the bias of the QRNG. Understanding better why this bias makes such a difference would nonetheless be interesting.

7.3.3. Third Chaitin–Schwartz-Solovay–Strassen test. While the above tests are inspired by the Chaitin–Schwartz theorem [20], they do not directly test the predicate $Z(s, n)$ appearing therein that we mentioned earlier. A key difference between these tests and the previous ones is the method they use to convert strings of random bits into potential witnesses to test.

Consider $s = s_0 \dots s_{m-1}$ a binary string (of length m) and n an integer greater than 2. Let k be the smallest integer such that $(n-1)^{k+1} > 2^m - 1$; we can thus rewrite the number whose binary representation is s into base $n-1$ and obtain the unique string $d_k d_{k-1} \dots d_0$ over the alphabet $\{0, 1, \dots, n-2\}$, that is,

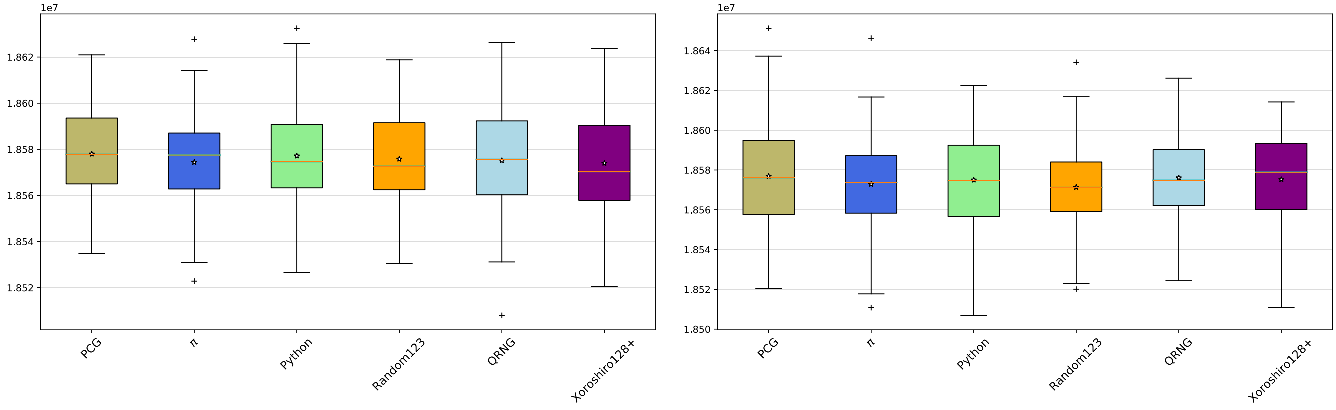
$$\sum_{i=0}^{k} d_i (n-1)^i = \sum_{t=0}^{m-1} s_t 2^t.$$

The predicate $Z(s, n)$ is defined by

$$Z(s, J) = \neg W(1 + d_0, n) \wedge \dots \wedge \neg W(1 + d_{k-1}, n), \quad (3)$$

where W is the Solovay–Strassen predicate from section 7.3. The digits of s (rewritten in base $n-1$) define the witnesses used to test the primality of n.

The main result from [20] is:



(a) (b)

Figure 5. Third Chaitin–Schwartz-Solovay–Strassen test: box-plot showing the distribution of total number of bits used to identify all 16-digit Carmichael numbers as composite by (a) the 80 strings from each RNG, and (b) the complement of these strings.

Theorem 1. *For all sufficiently large c, if s is a c-Kolmogorov random string of length* $\ell(\ell + 2c)$ *and n is an integer whose binary representation is* ℓ *bits long, then Z(s, n) is true if and only if n is prime.*

In order to carry out these tests we first fix c. For each Carmichael number n (with an ℓ-bit binary representation) we take[11] $c = \ell - 1$.

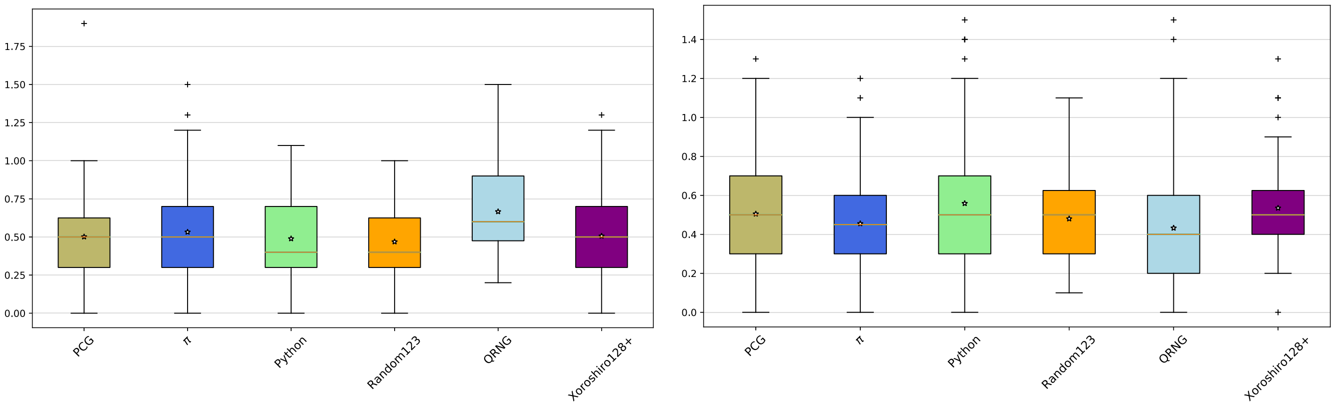
The metric of the third test has some similarities to that used in the second test. For each such n we take $\ell(\ell + 2c)$ bits. Rewriting s in base $n - 1$ as described above, we then compute $W(1 + d_j, n)$ for $0 \leqslant j \leqslant k$ until the first j is found such that $W(1 + d_j, n)$ holds (and the compositeness of n is thus witnessed). The metric itself is then taken as the sum (over all 16 digit Carmichael numbers n tested) of $j \times \lceil \log_2(n - 1) \rceil$. Note that, if no first $j \leqslant k$ is found such that $W(1 + d_j, n)$ holds (which occurs very rarely), then we simply count all the bits used when testing that Carmichael number, i.e. $\ell(\ell + 2c)$. Figure 5(a) shows the performance of the 80 strings from each of the six sources according to this metric. In order to be able to do decouple any potential difference between the QRNG and the other sources due to algorithmic randomness from those resulting from the bias of the QRNG, we similarly perform the same test on the complement of each of the strings, the results of which are shown in figure 5(b).

The results of the Kolmogorov–Smirnov tests on the data shown in figures 5(a) and (b) are given in tables A8 and A11, respectively. No statistically significant differences between any of the sources were found, reinforcing the impression given by figure 5 that the RNGs all give similar results. The Shapiro–Wilk test shows (see tables A9 and A12) that there is no strong evidence against the normality of test metric for the non-complemented strings (but there was weak evidence against it for the complemented ones), so we were able to use Welch's t-test to look for any further evidence of differences between the sources on these strings (see table A10). No significant differences between the sources were found by these tests either. We therefore conclude that the third Chaitin–Schwartz–Solovay–Strassen test with this metric, which counts the total number of bits required to verify the compositeness of all Carmichael numbers of at most 16 digits, failed to find significant differences between the QRNG and the PRNGs tested.

[11] This is somewhat arbitrary; other choices could of course be made, but would make little difference to our test.

7.3.4. Fourth Chaitin–Schwartz–Solovay–Strassen test. The final test is the most closely based on the Chaitin–Schwartz theorem out of the tests we consider. Rather than looking at how many witnesses need to be tested until a Carmichael number's compositeness is verified, we look directly at the ability of the entire *set* of witnesses evaluated in (3) to verify the compositeness of a number. In other words, we look for direct violations of the Chaitin–Schwartz theorem: a violation appears when for all $j = 0,\ldots,k - 1$, $W(1 + d_j, n)$ are false; that is, all tests wrongly conclude that n is 'probably prime'.

However, as the Solovay–Strassen test guarantees that $W(1 + d_j, n)$ is true with probability at least one half when n is a composite number, it quickly becomes difficult, in practice, to observe such violations for even the smallest Carmichael numbers used in the previous tests. In order to observe some violations with the length of random strings (and time) we have access to, we have to severely restrict ourselves and be content with testing the performance of the strings on only the odd composite numbers less than 50: 9, 15, 21, 25, 27, 33, 35, 39, 45, 49. For these numbers, we compute $Z(s, n)$ by reading $\ell(\ell + 2c)$ bits and following the same procedure as in the third test. When $Z(s, n) = 1$, a violation of the Chaitin–Schwartz theorem is thus observed. Since testing this predicate a single time on the ten numbers above would give insufficient statistics to observe any difference between the sources, we then repeated the above procedure reading from then 2nd bit of each string, then the 3rd, etc, until all the random bits have been used. The metric is thereby taken as the average number of violations observed for the 10 composites tested (where the average is taken over all the repetitions). Figures 6(a) and (b) show the results of this test for the 80 strings of each of the six sources used in the previous tests: again, the tests in the former figure use the



(a) (b)

Figure 6. Fourth Chaitin–Schwartz–Solovay–Strassen test: box-plot showing the distribution of the average count of violations of the Chaitin–Schwartz theorem for all odd composite numbers less than 50 by (a) the 80 strings from each RNG, and (b) the complement of these strings.

original strings from each source while the tests in the latter use the complemented strings.

We apply the same statistical tests to determine whether there are any statistically significant differences in performance between the different RNGs. The results of the Kolmogorov–Smirnov tests for the data in figures 6(a) and (b) are given in tables A13 and A15, respectively. Unlike the results for the previous metrics, the QRNG exhibits significantly different behaviour on the original (i.e. non-complemented) strings from the PCG, Python and Random123 PRNGs. However, no significant difference is found on any of the complemented strings. The Shapiro–Wilk tests (see tables A14 and A16) find strong evidence against the normality of the distribution of the test metric, so Welch's *t*-test was not applied to see if further evidence of significant differences was present.

Again, the reason for the apparently significant differences in performance between the QRNG and some of the sources (at least for the non-complemented strings) is unclear, and further investigation is required. The fact that only very small composite numbers were able to be tested means that, in the absence of strong evidence of differences between the sources, the results should be interpreted cautiously. Indeed, the Chaitin–Schwartz theorem is an asymptotic result, and a significant difference on larger composites (ideally Carmichael numbers), would be preferable. We thus cautiously conclude that the fourth Chaitin–Schwartz–Solovay–Strassen test with the violation-count metric potentially identifies differences between QRNGs and the other sources, but that further testing and study is needed to confirm the robustness of the initial results observed here.

8. Conclusions

In this paper we looked at the ability to formulate tests that probed the incomputability and algorithmic randomness of strings produced by QRNGs. Standard tests used to assess the quality of strings produced by PRNGs and QRNGs alike focus on simple statistical properties of the sequences, yet the most marked differences between QRNGs and PRNGs are the algorithmic properties of strings produced by such devices. Such tests thus provide an important and novel approach to evaluating the performance of QRNGs. This type of test, which probes the randomness of *outputs* of QRNGs, is complementary to the certification of a QRNG as exploiting random *processes*, either via theoretical analysis of the device [9, 10] or the use of device-independent randomness expansion schemes [4, 6].

The properties of incomputability (and, consequently, of algorithmic randomness too) mean that one must resort to indirect tests of incomputability in practice, and we discussed several such approaches. We considered testing the Borel normality of sequences—a necessary property of algorithmic randomness—which probes the bias of a sequence rather than its incomputability *per se*. This served as a useful preliminary probe for the analysis of later tests. We then focused on a different approach based around the Chaitin–Schwartz theorem, which shows a practical consequence of algorithmic randomness in probabilistic primality testing algorithms. We proposed four different tests based on this result which, in principle, could exhibit advantages due to the incomputability—as well as the algorithmic randomness—of sequences from QRNGs over PRNGs.

To assess the practical utility of these tests, we applied them to long sequences generated by various RNGs: a QRNG (described in section 5), and several different PRNGs. Two of the tests (the first and the third) failed to find any significant differences between the QRNG and the PRNGs. A significant difference was, on the other hand, observed, for the second test. However, we were able to show that the difference was due to a small bias present in the strings produced by the QRNG rather than a result of any incomputability. Indeed, this highlighted a key challenge: the need to decouple the incomputability from the bias within the test results, since the tests can in general be affected by both these elements. To this end, we examined the performance of tests on the complement of the strings as well as the strings themselves, but conclude that care should be taken to formulate tests that are not affected by the bias of a sequence. This task is complicated, however, by the fact that the effect of using a biased distribution in probabilistic primality testing is not well understood theoretically. For future studies, it would thus be desirable to have sufficiently long strings to analyse from a certified QRNG for which the bias is sufficiently small so as to not be a limiting factor for the tests. Conversely, one should also study further the effect of normalisation procedures [29, 62] on such metrics, so that such tests can be properly analysed when applied to normalised, rather than raw, sequences of bits.

Our fourth test, which was designed to follow more faithfully the Chaitin–Schwartz theorem and to be potentially more robust to bias (but, unfortunately, more demanding to apply in practice), produced ambiguous results. In particular, significant differences were found only on the non-complemented strings, but it was not clear whether these differences were entirely due to bias, as one would expect the complemented strings to show a similar difference in the opposite direction, which was not observed. Due to the practical limitations of this test and small numbers tested, further testing (and, probably, refinements of the test itself) on more data are needed to understand this effect better.

While our tests failed to find any conclusive experimental evidence of incomputability of quantum randomness, they provide an important study for the development of new types of tests aimed at probing algorithmic properties of quantum randomness. Indeed, being based on the Chaitin–Schwartz theorem, the tests in fact probe the stronger property of c-Kolmogorov randomness, and this fact potentially contributes to the difficulty in observing indirect effects of incomputability. The development of further tests to this end, as well as additional experimental studies, are therefore merited.

We finish by reiterating that tests of the output of QRNGs, such as we describe here, complement, rather than replace, the certification by value indefiniteness of a QRNG. Indeed, just as with standard statistical tests of RNGs, even if no difference between QRNGs and PRNGs is found on the tests, it is important that QRNGs are verified to pass such tests. QRNGs certified by the Kochen–Specker theorem, such as the one used to provide the data for this paper [21], can also be used to perform device-independent randomness expansion [54–56], and it would be interesting to combine algorithmic tests like we develop here with such an approach, although producing sufficient amounts of data for this to be possible remain a challenge.

Lastly, we note that all the test data (i.e. random strings), programs and results are available online in [74].

Acknowledgments

The authors acknowledge fruitful discussions with Arkady Fedorov, Anatoly Kulikov, Frank Stephan and Karl Svozil. A A Abbott acknowledges financial support from the ‘Retour Post-Doctorants’ (ANR-13-PDOC-0026) programme of the French National Research Agency.

Appendix. Chaitin–Schwartz–Solovay–Strassen test analysis tables

Table A1. Kolmogorov–Smirnov tests for the first Chaitin–Schwartz–Solovay–Strassen test with the metric that records the minimum number of witnesses needed to verify the compositeness of all Carmichael numbers of at most 16 digits.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.8186	0.8186	1	1	1
π		0.9976	0.9976	0.5596	1
Python			0.9976	0.5596	0.9976
Random123				0.9976	1
QRNG					0.9780

Table A2. Shapiro–Wilk tests of normality for the first Chaitin–Schwartz–Solovay–Strassen test with the metric that records the minimum number of witnesses needed to verify the compositeness of all Carmichael numbers of at most 16 digits.

	PCG	π	Python	Random123	QRNG	xoroshiro128+
p-value	$<\mathbf{10^{-4}}$	$<\mathbf{10^{-4}}$	$<\mathbf{10^{-4}}$	$<\mathbf{10^{-4}}$	$<\mathbf{10^{-4}}$	$<\mathbf{10^{-4}}$

Table A3. Kolmogorov–Smirnov tests for the second Chaitin–Schwartz–Solovay–Strassen test with the ‘bit counting’ metric on the non-complemented (i.e. original) bits.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.6953	0.4383	0.922	0.0132	0.6953
π		0.4383	0.8219	**0.0045**	0.9794
Python			0.0814	0.0537	0.5625
Random123				**0.0014**	0.5625
QRNG					**0.0026**

Table A4. Shapiro–Wilk tests of normality for the second Chaitin–Schwartz–Solovay–Strassen test with the 'bit counting' metric on the non-complemented (i.e. original) bits.

	PCG	π	Python	Random123	QRNG	xoroshiro128+
p-value	0.4892	0.2003	**0.048 67**	0.5951	0.1669	0.0808

Table A5. Kolmogorov–Smirnov tests for the second Chaitin–Schwartz–Solovay–Strassen test with the 'bit counting' metric on the complemented bits.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.4383	0.3307	0.2424	0.053 72	0.5625
π		0.4383	0.1202	**0.0045**	0.5625
Python			0.5625	**0.0026**	0.8219
Random123				**0.0014**	0.2424
QRNG					0.0132

Table A6. Shapiro–Wilk tests of normality for the second Chaitin–Schwartz–Solovay–Strassen test with the 'bit counting' metric on the complemented bits.

	PCG	π	Python	Random123	QRNG	xoroshiro128+
p-value	0.199	0.2433	0.0754	0.4401	0.0518	0.9673

Table A7. Welch t-tests for the second Chaitin–Schwartz–Solovay–Strassen test with the 'bit counting' metric on the complemented bits.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.6422	0.3796	0.1265	**0.0034**	0.9454
π		0.6343	0.2287	**0.0004**	0.6795
Python			0.4683	**0.0001**	0.3964
Random123				$\mathbf{<10^{-4}}$	0.1271
QRNG					**0.0020**

Table A8. Kolmogorov–Smirnov tests for the third Chaitin–Schwartz–Solovay–Strassen test with the 'bit-counting' metric for the non-complemented (i.e. original) bits for all Carmichael numbers of at most 16 digits.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.269 4	0.4821	0.2988	0.4013	0.1054
π		0.6953	0.4383	0.3307	0.4383
Python			0.8186	0.5625	0.5625
Random123				0.9794	0.8219
QRNG					0.8219

Table A9. Shapiro–Wilk tests of normality for the third Chaitin–Schwartz–Solovay–Strassen test with the 'bit-counting' metric for the non-complemented (i.e. original) bits for all Carmichael numbers of at most 16 digits.

	PCG	π	Python	Random123	QRNG	xoroshiro128+
p-value	0.2076	0.4921	0.3337	0.195 6	0.7608	0.1347

Table A10. Welch t-tests for the third Chaitin–Schwartz–Solovay–Strassen test with the 'bit-counting' metric for the non-complemented (i.e. original) bits for all Carmichael numbers of at most 16 digits.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.2838	0.81	0.5227	0.4335	0.2437
π		0.4186	0.6833	0.8401	0.911
Python			0.6956	0.584	0.3653
Random123				0.8585	0.6096
QRNG					0.7629

Table A11. Kolmogorov–Smirnov tests for the third Chaitin–Schwartz–Solovay–Strassen test with the 'bit-counting' metric for the complemented bits for all Carmichael numbers of at most 16 digits.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.5596	0.9794	0.173	0.9794	0.3307
π		0.922	0.8219	0.8219	0.6953
Python			0.5625	0.9194	0.6953
Random123				0.4383	0.1201
QRNG					0.8219

Table A12. Shapiro–Wilk tests of normality for the third Chaitin–Schwartz–Solovay–Strassen test with the 'bit-counting' metric for the complemented bits for all Carmichael numbers of at most 16 digits.

	PCG	π	Python	Random123	QRNG	xoroshiro128+
p-value	0.4616	0.6708	0.6067	0.94	0.9355	**0.0239**

Table A13. Kolmogorov–Smirnov tests for the fourth Chaitin–Schwartz–Solovay–Strassen test with the 'violation-count' metric for non-complemented (i.e. original) bits for all odd composite numbers that are less than 50.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.318	0.2414	0.692	**0.0027**	0.9976
π		0.692	0.8186	0.053 97	0.9976
Python			0.9194	**0.0004**	0.8186
Random123				**0.0047**	0.8186
QRNG					0.0348

Table A14. Shapiro–Wilk tests of normality for the fourth Chaitin–Schwartz–Solovay–Strassen test with the 'violation-count' metric for non-complemented (i.e. original) bits for all odd composite numbers that are less than 50.

	PCG	π	Python	Random123	QRNG	xoroshiro128+
p-value	$\mathbf{<10^{-4}}$	**0.0040**	**0.0002**	**0.0056**	**0.0115**	**0.0148**

Table A15. Kolmogorov–Smirnov tests for the fourth Chaitin–Schwartz–Solovay–Strassen test with the 'violation-count' metric for the complemented bits for all odd composite numbers that are less than 50.

p-values	π	Python	Random123	QRNG	xoroshiro128+
PCG	0.692	0.9194	0.9194	0.1725	0.5596
π		0.5596	0.9976	0.692	0.2414
Python			0.692	0.1725	0.8186
Random123				0.5596	0.5596
QRNG					0.0135

Table A16. Shapiro–Wilk tests of normality for the fourth Chaitin–Schwartz–Solovay–Strassen test with the 'violation-count' metric for the complemented bits for all odd composite numbers that are less than 50.

	PCG	π	Python	Random123	QRNG	xoroshiro128+
p-value	0.066 01	**0.029 57**	$<\mathbf{10^{-4}}$	**0.0080**	$<\mathbf{10^{-4}}$	**0.001 7**

ORCID iDs

Alastair A Abbott https://orcid.org/0000-0002-2759-633X
Cristian S Calude https://orcid.org/0000-0002-8711-6799
Michael J Dinneen https://orcid.org/0000-0001-9977-525X

References

[1] Lenstra A K, Hughes J P, Augier M, Bos J W, Kleinjung T and Wachter C 2012 *Ron was Wrong, Whit is Right* (Santa Barbara: IACR) p 17 https://eprint.iacr.org/2012/064.pdf
[2] Svozil K 1990 *Phys. Lett.* A **143** 433
[3] Stefanov A, Gisin N, Guinnard O, Guinnard L and Zbinden H 2000 *J. Mod. Opt.* **47** 595
[4] Pironio S *et al* 2010 *Nature* **464** 09008
[5] Bera M N, Acín A, Kuś M, Mitchell M and Lewenstein M 2017 *Rep. Prog. Phys.* **80** 124001
[6] Colbeck R and Kent A 2011 *J. Phys. A: Math. Gen.* **44** 095305
[7] Marsaglia G and Zaman A 1990 *Stat. Probab. Lett.* **9** 35
[8] Rukhin A 2010 *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* NIST 800-22
[9] Abbott A A, Calude C S and Svozil K 2014 *Math. Struct. Comput. Sci.* **24** e240303
[10] Abbott A A, Calude C S, Conder J and Svozil K 2012 *Phys. Rev.* A **86** 062109
[11] Calude C S and Svozil K 2008 *Adv. Sci. Lett.* **1** 165
[12] Abbott A A, Calude C S and Svozil K 2015 *Fields of Logic and Computation II—Essays Dedicated to Yuri Gurevich on the Occasion of His 75th Birthday* (*Lecture Notes in Computer Science* vol 9300) ed L D Beklemishev *et al* (Switzerland: Springer International) pp 69–86
[13] Bar-Hillel M and Wagenaar W A 1991 *Adv. Appl. Math.* **12** 428
[14] Figurska M, Stańczyk M and Kulesza K 2008 *Med. Hypotheses* **70** 182
[15] Calude C S 2002 *Information and Randomness: An Algorithmic Perspective* 2nd edn (Berlin: Springer)
[16] Borel É 1909 *Rend. Circolo Mat. Palermo* **27** 247
[17] Calude C S 1994 *Developments in Language Theory* ed G Rozenberg and A Salomaa (Singapore: World Scientific) pp 113–29
[18] Solovay R and Strassen V 1977 *SIAM J. Comput.* **6** 84 Corrigendum in Ref. [75]
[19] Calude C S, Dinneen M J, Dumitrescu M and Svozil K 2010 *Phys. Rev.* A **82** 022102
[20] Chaitin G J and Schwartz J T 1978 *Commun. Pure Appl. Math.* **31** 521
[21] Kulikov A, Jerger M, Potočnik A, Wallraff A and Fedorov A 2017 *Phys. Rev. Lett.* **119** 240501
[22] Champernowne D G 1933 *J. London Math. Soc.* **8** 254
[23] Chaitin G J 1977 *IBM J. Res. Dev.* **21** 350
[24] Downey R and Hirschfeldt D 2010 *Algorithmic Randomness and Complexity* (Berlin: Springer)
[25] Martin-Löf P 1966 *Inf. Control* **9** 602
[26] Calude C S 2017 *The Incomputable: Journeys Beyond the Turing Barrier* ed S B Cooper and M Soskova (Berlin: Springer) pp 169–81
[27] Graham R and Spencer J H 1990 *Sci. Am.* **262** 112
[28] Abbott A A 2015 Value indefiniteness, randomness and unpredictability in quantum foundations *Ph.D. Thesis* University of Auckland; École Normale Supérieure de Paris
[29] Von Neumann J 1963 *John von Neumann, Collected Works* ed A H Traub (New York: MacMillan) pp 768–70
[30] Eagle A 2014 *The Stanford Encyclopedia of Philosophy* ed E N Zalta (Standford, CA: Stanford University) Spring, 2014 ed
[31] Solis A and Hirsch J G 2015 *J. Phys.: Conf. Ser.* **624** 012001
[32] Eagle A 2005 *Br. J. Phil. Sci.* **56** 749
[33] Abbott A A, Calude C S and Svozil K 2015 *Information* **6** 773
[34] Hájek A 2014 *The Stanford Encyclopedia of Philosophy* ed E N Zalta (Standford, CA: Stanford University) Winter, 2012 ed.
[35] Gentle J E 2003 *Random Number Generations and Monte Carlo Methods* 2nd edn (New York: Springer)
[36] Goldreich O 2001 *Foundations of Cryptography I: Basic Tools* (Cambridge: Cambridge University Press)
[37] Bernstein D J, Chang Y-A, Cheng C-M, Chou L-P, Heninger N, Lange T and van Someren N 2013 *Advances in Cryptology—ASIACRYPT 2013* ed K Sako and P Sarkar (Berlin: Springer) pp 341–60
[38] Acín A 2013 *Is Science Compatible with Free Will?: Exploring Free Will and Consciousness in the Light of Quantum Physics and Neuroscience* ed A Suarez and P Adams (New York: Springer) ch 2 pp 7–22
[39] Longo G and Paul T 2008 *Computability in Context: Computation and Logic in the Real World* ed S B Cooper and A Sorbi (London: Imperial College Press/ World Scientific) ch 7 pp 243–74
[40] Bell J S 1964 *Physics* **1** 195
[41] Aspect A, Grangier P and Roger G 1982 *Phys. Rev. Lett.* **49** 91
[42] Kochen S B and Specker E 1967 *J. Math. Mech. (now Indiana University Math. J.)* **17** 59
[43] Abbott A A, Calude C S and Svozil K 2013 *Phys. Rev.* A **89** 032109
[44] Abbott A A, Calude C S and Svozil K 2015 *J. Math. Phys.* **56** 102201
[45] Schmidt H 1970 *J. Appl. Phys.* **41** 462
[46] Jennewein T, Achleitner U, Weihs G, Weinfurter H and Zeilinger A 2000 *Rev. Sci. Instrum.* **71** 1675
[47] Shen Y, Tian L and Zou H 2010 *Phys. Rev.* A **81**
[48] Stipčević M and Rogina B M 2007 *Rev. Sci. Instrum.* **78** 045104
[49] Quantis QRNG 2010 *ID Quantique White Paper* https://www.idquantique.com/random-number-generation/
[50] Ma X, Yuan X, Cao Z, Qi B and Zhang Z 2016 *NPJ Quantum Inf.* **2** 16021
[51] Himbeeck T V, Woodhead E, Cerf N J, García-Patón R and Pironio S 2017 *Quantum* **1** 33
[52] Klyachko A A, Can M A, Binicioğly S and Shumovsky A S 2008 *Phys. Rev. Lett.* **101** 020403
[53] Cabello A 2008 *Phys. Rev. Lett.* **101** 210401
[54] Um M, Zhang X, Zhang J, Wang Y, Yangchao S, Deng D-L, Duan L-M and Kim K 2013 *Sci. Rep.* **3** 1627

[55] Deng D-L, Zu C, Chang X-Y, Hou P-Y, Yang H-X, Wang Y-X and Duan L-M 2013 arXiv:1301.5364 [quant-ph]

[56] Miller C A and Shi Y 2017 *SIAM J. Comput.* **46** 1304

[57] Marsaglia G 2005 *Interstat.* **10** 1

[58] Wagon S 2004 *Pi: A Source Book* ed J L Berggren, J M Borwein and P B Borwein (New York: Springer) 557–9

[59] Bailey D H, Borwein J M, Calude C S, Dinneen M J, Dumitrescu M and Yee A 2012 *Exp. Math.* **21** 375

[60] Abbott A A, Bienvenu L and Senno G 2014 Non-uniformity in the quantis random number generator *CDMTCS Research Report Series* 472 Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland

[61] Peres Y 1992 *Ann. Stat.* **20** 590

[62] Abbott A A and Calude C S 2012 *Computability* **1** 59

[63] Kovalsky M G, Hnilo A A and Agüero M B 2018 *Phys. Rev.* A **98** 042131

[64] Matsumoto M and Nishimura T 1998 *ACM Trans. Model. Comput. Simul.* **8** 3

[65] Salmon J K, Moraes M A, Dror R O and Shaw D E 2011 *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC11) (ACM, New York)*

[66] O'Neill M E 2014 PCG: a family of simple fast space-efficient statistically good algorithms for random number generation *Tech. Rep.* HMC-CS-2014-0905 Harvey Mudd College, Claremont, CA

[67] Marsaglia G 2003 *J. Stat. Softw.* **8**

[68] Solis A, Martínez A M A, Alarcón R R, Ramírez H C, U'Ren A B and Hirsch J G 2015 *Phys. Scr.* **90** 074034

[69] Martínez A C, Solís A, Rojas R D H, U'Ren A B, Hirsch J G and Castillo I P 2018 arXiv:1810.08718 [quant-ph]

[70] Conover W J 1999 *Practical Nonparametric Statistics* (New York: Wiley) p 584

[71] Shapiro S S and Wilk M B 2005 *Biometrika* **52** 591

[72] Welch B L 1947 *Biometrika* **34** 28

[73] Pinch R G E 2007 *Proceedings of Conference on Algorithmic Number Theory 2007* **vol 46** ed A-M Ernvall-Hytönen *et al* pp 129–31

[74] Abbott A A, Calude C S, Dinneen M J and Huang N 2018 Experimental probing of the incomputability of quantum randomness *CDMTCS Research Report Series 515v2* Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland https://cs.auckland.ac.nz/research/groups/CDMTCS/export/80_random_seqs/

[75] Solovay R and Strassen V 1977 *SIAM J. Comput.* **7** 118