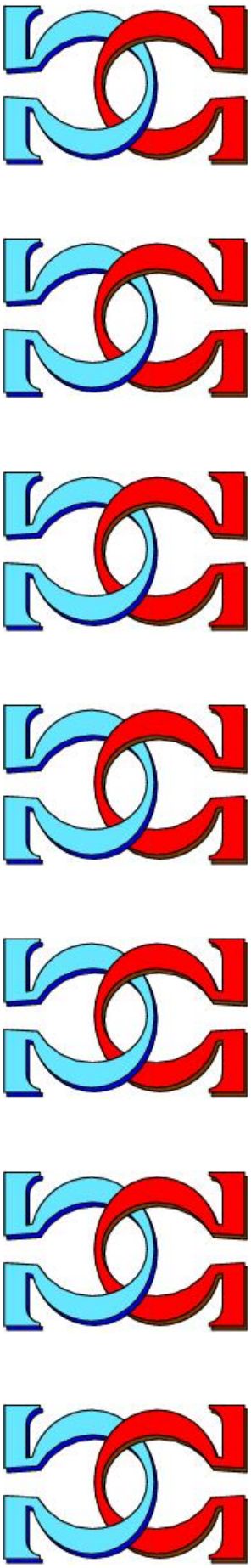


CDMTCS
Research
Report
Series

Experimentally Probing the Incomputability of Quantum Randomness

Alastair A. Abbott
University of Grenoble Alpes, CNRS,
Grenoble, France

Cristian S. Calude
Michael J. Dinneen
Nan Huang
Department of Computer Science,
University of Auckland,
Auckland, New Zealand.

CDMTCS-515
June 2018 (Version 2)

Centre for Discrete Mathematics and
Theoretical Computer Science

Experimentally Probing the Incomputability of Quantum Randomness

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

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, certified by the Kochen-Specker Theorem.

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.

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.[1] This has driven a recent surge of interest in RNGs exploiting physical phenomena, and more particularly in quantum RNGs (QRNGs) that utilise the purportedly inherent randomness in quantum mechanics [16, 52, 60, 62]. QRNGs are generally

[*]Univ. Grenoble Alpes, CNRS, Grenoble INP, Institut Néel, 38000 Grenoble, France.

[†]Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand.

[1]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 [42]. As of 2018, Java still relies on a linear congruential generator, a low quality PRNG.

considered to be, by their very nature, better than classical RNGs (such as PRNGs), but how (or can) one test this in practice?

RNGs are usually tested by conducting batteries of tests on (finite) sequences they have produced [46, 53]. Traditionally, such tests have focused on intuitive aspects of randomness, such as the frequencies of certain (strings of) bits, but human intuition about randomness is notoriously poor [14, 32] 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 [20]. With standard randomness tests designed with PRNGs in mind, it is reasonable to wonder whether there are tests more appropriate for analysing QRNGs and perceiving the advantage they can provide. Indeed, QRNGs should excel precisely on properties of randomness where algorithmic PRNGs are doomed to fail: incomputability and their inherent unpredictability [4, 7, 9, 23]. Although incomputability is not directly testable (not least because it is a property of infinite sequences and thus holds only in the limit), one may ask whether there are tests than can reveal related advantages in practice.

With this goal, we study several possible tests of randomness based on algorithmic information theory. In particular, we consider tests based on Borel normality [18, 19] as well as novel tests based on the Solovay-Strassen probabilistic primality test [22, 59]—an algorithm which can be made deterministic when given access to algorithmic randomness [25]. 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 [41] using these tests. 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 incomputability and algorithmic complexity. 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 [18]. 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* $0\,1\,00\,01\,10\,11\,000\,001\,011\,100\ldots$ [26] 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 [24]. The incompressibility of strings depends on the choice of universal Turing machine; this shortcoming disappears when the definition is extended to infinite sequences [20, 29]. 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 [20] 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 $K_U(x) \leq K_M(x) + c$, 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 $n + \mathrm{O}(1)$. 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 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 [47]; 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 "pure" or "absolute" randomness exists, only degrees of randomness [20, 21, 35]. 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 [1]. Indeed, as von Neumann noted, "there is no such thing as a random number—there are only methods to produce random numbers" [63]. 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* [1, 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 [8, 30].

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 $\mathbf{x} = x_1 x_2 \ldots$ 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 [20]. 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 [36] (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 [8].

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 [33]. 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 [34], since 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., [17]) 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 [11]. 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

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

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) [43], 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 [11]. 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 [15] 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 [12] between entangled particles. The Kochen-Specker Theorem [39], although perhaps lesser known, pinpoints this breakdown in determinism in a more precise way: it shows that, for any quantum system with more than 2 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 [6, 10]. 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 2 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 [9], 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 [55], but simpler systems based, for example, on measuring the polarisation [38, 57, 60] or detection times [61] of photons, have become the norm due to the practical advantages they provide. Such approaches have led to the development of commercial QRNGs, such as ID Quantique's Quantis [37].

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 3-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.

More recently there has been significant interest in implementing QRNGs that violate Bell's inequalities in order to provide a stronger certification [27, 52]. Specifically, such devices allow the indeterminism of a QRNG to be certified in a device-independent way—i.e., without assuming knowledge of how the device works—which is particularly important in cryptographic settings, where one perhaps does not wish to trust the workings of a given RNG. Such certification, however, comes at a cost, since not only does it still require an initial small random seed, but it also relies on the QRNG being separated 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 [52].

An alternative approach outlined in [4, 7] is to use 3-dimensional systems exhibiting value indefiniteness (via the Kochen-Specker Theorem) to certify a QRNG. While such a certification is device dependent (i.e., one relies on knowledge of the functioning of the QRNG), it allows the practical advantages of standard QRNGs to be maintained while providing stronger theoretical certification. Although this QRNG was originally proposed specifically for spin-1 particles, the principle is applicable to any 3-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 below). 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.[2] However, by the results of [4, 10], 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 [41], 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 [4, 7] based on the implementation used by Kulikov et al. [41]. 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 focus on observing differences between such sequences and pseudo-random sequences arising from algorithmic properties of the sequences.

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

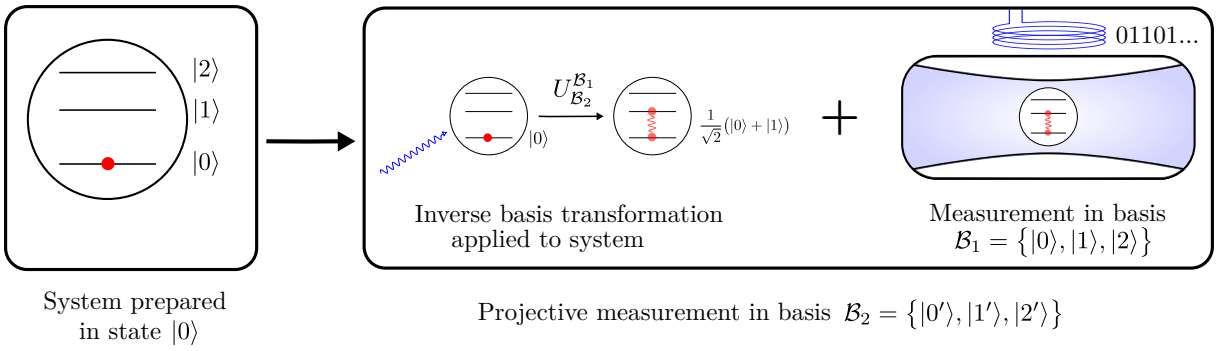


Figure 1: Schematic showing the QRNG based on the Kochen-Specker Theorem as implemented in [41]. A transmon qutrit system is initially prepared in the state $|0\rangle$ (with respect to the computational basis $\mathcal{B}_1 = \{|0\rangle, |1\rangle, |2\rangle\}$ by thermal cooling. The system is then 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.

This approach to certifying a QRNG via value indefiniteness leads to some interesting additional consequences 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 [4] 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" [29]) 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.

6 Testing RNGs

While it is crucial to have a good theoretical understanding of any RNG, there are several reasons why testing experimentally their performance is nonetheless crucial. Firstly, one can never be sure that the implementation of a RNG matches tits 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 RNGs can be illustrated pointedly with an example. Imagine a device which deterministically outputs the digits of the binary expansion of $\pi = \pi_1\pi_2\pi_3\ldots$ 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 [45] despite the fact that it is not even known to be Borel normal [13, 64]. Nevertheless, the sequence produced by this box would be computable and thus not random at all.

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 [42]. QRNGs have generally been tested against similar tests, such as the NIST [53] and DIEHARD [46] batteries, and generally perform well. For example, Quantis is officially certified as passing these tests on 1000 samples of 1 million bits [37]. 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 [2].

Such statistical non-uniformity is, however, to be expected in RNGs exploiting physical phenomena due to experimental imperfections and instability [7]. 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 [9]. Moreover, bias can be reduced by careful post-processing [3, 50, 63], 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 [40], ostensibly as a proxy for direct tests of Kolmogorov complexity. In practice, however, just like the aforementioned tests, this approach also tests simple statistical properties and suffers from the same 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 PRNGS—which always produce computable sequences—from QRNGs [22]. 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[3] 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 [13]), the PRNG used by Python v3.5.4 (a Mersenne Twister algorithm) [48], Random123 v1.09 [54], PCG v0.98 [49], xoroshiro128+ [44], and the QRNG described in Section 5 (see [41]).

[3]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.

7.1 Tests of Borel normality

As mentioned earlier, the notion of Borel normality was the first mathematical definition of algorithmic randomness [18], 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 [20], 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 [20]. 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 $1 \leq i \leq 2^m$ 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* $\frac{1}{\log_2}$) if for every integer $1 \leq m \leq \log_2 \log_2 |x|$ and each $1 \leq j \leq 2^m$ we have:

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

Almost all algorithmic random strings are Borel normal with accuracy $\frac{1}{\log_2}$ [20]; in particular, they have approximately the same number of 0s and 1s. Furthermore, if all prefixes of a sequence are Borel normal, then the sequence itself is also Borel normal.

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 Eq. (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, but this choice 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 $1 \leq j \leq 2^m$. 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 [41] and, 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 the incomputability 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 nonrandom? 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

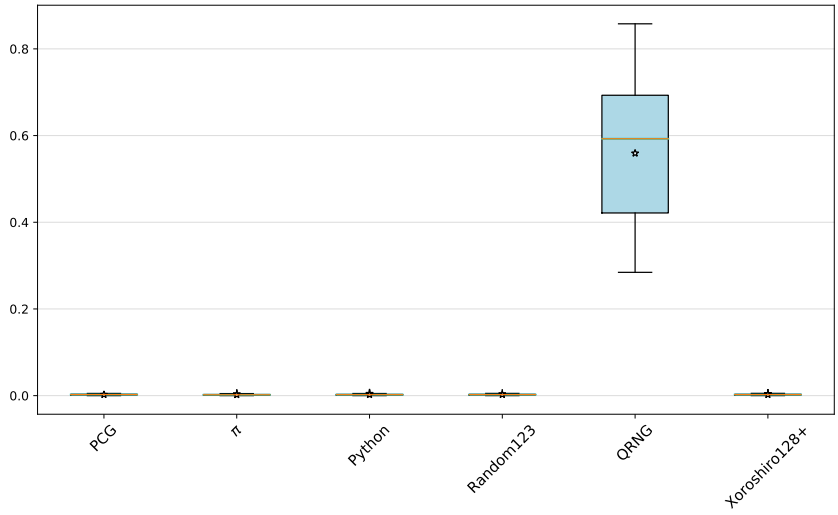


Figure 2: Borel normality test: Box-plot showing the distribution of the quantity $\max\left(\left|\frac{N_j^m(x)}{|x|_m} - 2^{-m}\right|\right)\log_2|x|$ for the 80 strings of length $|x| = 2^{26}$ bits produced by each the six RNGs tested.

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 [20] 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.

To specify such a 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 $\sigma(0) = \varphi_e(0), \ldots, \sigma(e+n+1) = \varphi_e(e+n+1)$ 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 [59]. 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 [59].

Chaitin and Schwartz [25] 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 snd 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 [20]. As a consequence, de-randomisation 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 $b^{n-1} \equiv 1 \pmod{n}$ 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 pseudo-primes. Many Carmichael numbers can pass Fermat's primality test, but less of them pass the Solovay-Strassen test. Increasingly Carmichael numbers become "rare".[4]

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.

Moreover, while our primary objective in formulating these tests is to probe indirectly the incomputability of quantum randomness, the fact that the Chaitin-Schwartz Theorem relies on the stronger property of Kolmogorov randomness means that these tests also probe this property. 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 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 [4].

As in [22], 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 [28] to determine whether two datasets differ significantly. This test returns a p-value[5] 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

[4] There are 1,401,644 Carmichael numbers in the interval $[1, 10^{18}]$.

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

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 [56];[6] if it is,[7] we then use the (parametric) Welch t-test [65], 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 [22], 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 [51], 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 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

[6] 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.

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

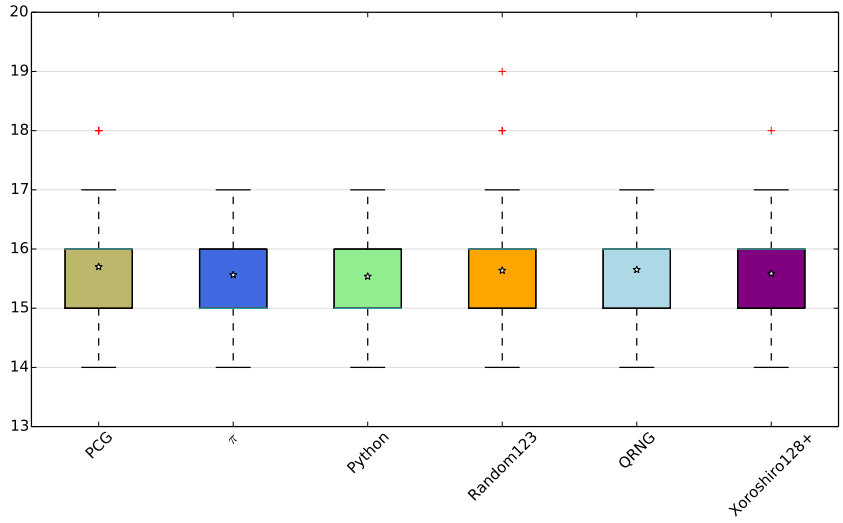
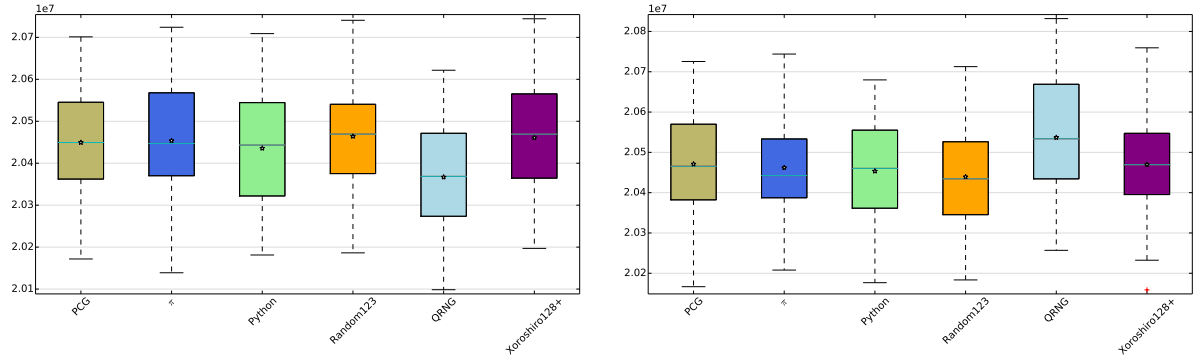


Figure 3: First Chaitin-Schwartz-Solovay-Strassen test on 80 samples: Box-plot showing the distribution in the minimum number of witnesses needed to verify the compositeness of all Carmichael numbers of at most 16 digits.

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



(a) (b)

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.

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. A simple way to test this

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 effected 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 [25], 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

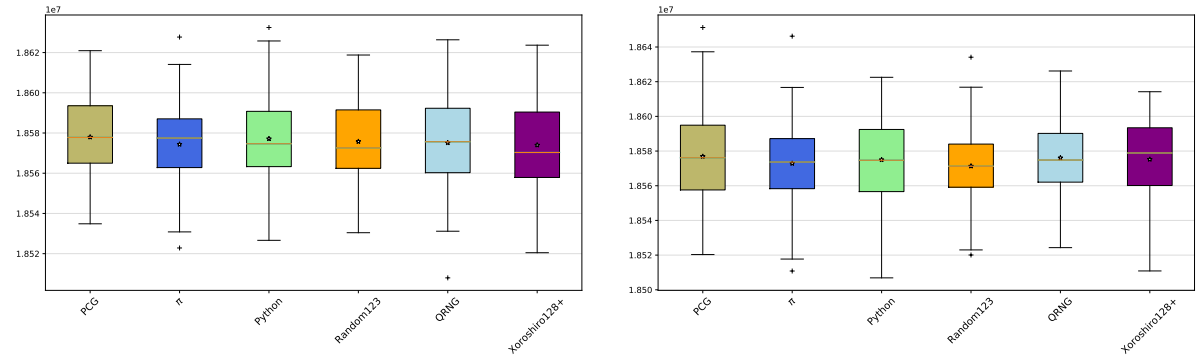
$$Z(s, J) = \neg W(1 + d_0, n) \wedge \dots \wedge \neg W(1 + d_{k-1}, n), \tag{3}$$

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 [25] is:

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 $c = \ell - 1$.[8]



(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.

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 $0 \leq j \leq k$ 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 $j \leq k$ 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., $\ell\,(\ell+2c)$. Figures 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 5(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 the 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.

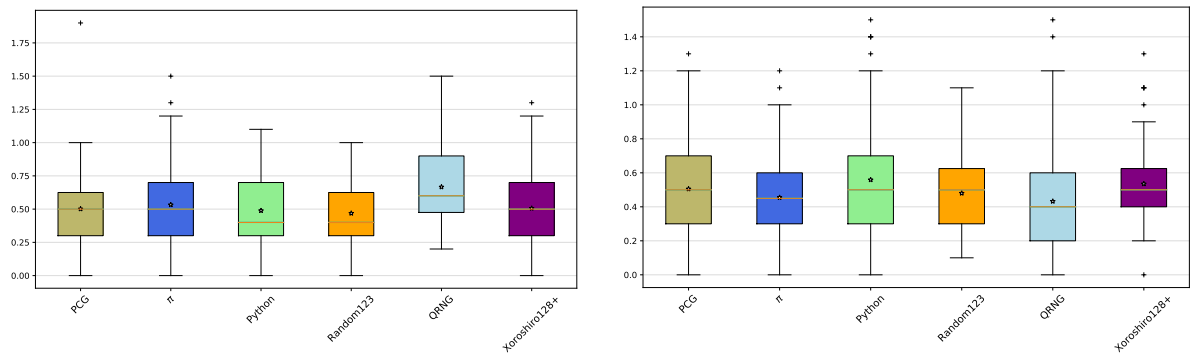
7.3.4 Fourth Chaitin-Schwartz-Solovay-Strassen test

The final test is based more closely 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

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

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



(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.

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 6(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 of incomputability for QRNGs. As we argued, such tests are important since they probe key advantages of QRNGs over PRNGs that are not addressed by standard statistical testing of RNGs. The properties of incomputability 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 effected 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 effected 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.

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) 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 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 conclude by noting that all the test data (i.e., random strings), programs and results are available online in [5].

Acknowledgments

The authors acknowledge fruitful discussions with Arkady Fedorov, Anatoly Kulikov, Frank Stephan and Karl Svozil.

References

[1] A. A. Abbott. *Value Indefiniteness, Randomness and Unpredictability in Quantum Foundations*. PhD thesis, University of Auckland; École Normale Supérieure de Paris, 2015.

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

[3] A. A. Abbott and C. S. Calude. Von Neumann normalisation of a quantum random number generator. *Computability*, 1(1):59–83, 2012.

[4] A. A. Abbott, C. S. Calude, J. Conder, and K. Svozil. Strong Kochen-Specker theorem and incomputability of quantum randomness. *Physical Review A*, 86:062109, 2012.

[5] A. A. Abbott, C. S. Calude, M. J. Dinneen, and N. Huang. Experimental probing of the incomputability of quantum randomness. *CDMTCS Research Report Series 515v2*, Centre for Discrete Mathematics and Theoretical Computer Science, University of Auckland, 2018.

[6] A. A. Abbott, C. S. Calude, and K. Svozil. Value-indefinite observables are almost everywhere. *Physical Review A*, 89:032109, 2013.

[7] A. A. Abbott, C. S. Calude, and K. Svozil. A quantum random number generator certified by value indefiniteness. *Mathematical Structures in Computer Science*, 24(3):e240303, 2014.

[8] A. A. Abbott, C. S. Calude, and K. Svozil. A non-probabilistic model of relativised predictability in physics. *Information*, 6(4):773–789, 2015.

[9] A. A. Abbott, C. S. Calude, and K. Svozil. On the unpredictability of individual quantum measurement outcomes. In L. D. Beklemishev, A. Blass, N. Dershowitz, B. Finkbeiner, and W. Schulte, editors, *Fields of Logic and Computation II – Essays Dedicated to Yuri Gurevich on the Occasion of His 75th Birthday*, volume 9300 of *Lecture Notes in Computer Science*, pages 69–86. Springer International, Switzerland, 2015.

[10] A. A. Abbott, C. S. Calude, and K. Svozil. A variant of the Kochen-Specker theorem localising value indefiniteness. *Journal of Mathematical Physics*, 56:102201, 2015.

[11] A. Acín. True quantum randomness. In A. Suarez and P. Adams, editors, *Is Science Compatible with Free Will?: Exploring Free Will and Consciousness in the Light of Quantum Physics and Neuroscience*, chapter 2, pages 7–22. Springer-Verlag, New York, 2013.

[12] A. Aspect, P. Grangier, and G. Roger. Experimental realization of Einstein-Podolsky-Rosen-Bohm *Gedankenexperiment*: A new violation of Bell's inequalities. *Physical Review Letters*, 49(2):91–94, 1982.

[13] D. H. Bailey, J. M. Borwein, C. S. Calude, M. J. Dinneen, M. Dumitrescu, and A. Yee. An empirical approach to the normality of π. *Experimental Mathematics*, 21(4):375–384, 2012.

[14] M. Bar-Hillel and W. A. Wagenaar. The perception of randomness. *Advances in Applied Mathematics*, 12(4):428–454, 1991.

[15] J. S. Bell. On the Eistein-Podolsky-Rosen paradox. *Physics*, 1(3):195–200, 1964.

[16] M. N. Bera, A. Acín, M. Kuś, M. Mitchell, and M. Lewenstein. Randomness in quantum mechanics: Philosophy, physics and technology. *Reports on Progress in Physics*, 80:124001, 2017.

[17] D. J. Bernstein, Y.-A. Chang, C.-M. Cheng, L.-P. Chou, N. Heninger, T. Lange, and N. van Someren. Factoring RSA keys from certified smart cards: Coppersmith in the wild. In K. Sako and P. Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 341–360, Berling, 2013. Springer.

[18] É. Borel. Les probabilités dénombrables et leurs applications arithmétiques. *Rendiconti del Circolo Matematico di Palermo*, 27:247–271, 1909.

[19] C. S. Calude. Borel normality and algorithmic randomness. In G. Rozenberg and A. Salomaa, editors, *Developments in Language Theory*, pages 113–129. World Scientific, Singapore, 1994.

[20] C. S. Calude. *Information and Randomness: An Algorithmic Perspective*. Springer-Verlag, Berlin, second edition, 2002.

[21] C. S. Calude. Quantum randomness: From practice to theory and back. In S. B. Cooper and M. Soskova, editors, *The Incomputable: Journeys Beyond the Turing Barrier*, pages 169–181. Springer, 2017.

[22] C. S. Calude, M. J. Dinneen, M. Dumitrescu, and K. Svozil. Experimental evidence of quantum randomness incomputability. *Physical Review A*, 82:022102, 2010.

[23] C. S. Calude and K. Svozil. Quantum randomness and value indefiniteness. *Advanced Science Letters*, 1(2):165–168, 2008.

[24] G. J. Chaitin. Algorithmic information theory. *IBM Journal of Research and Development*, 21(4):350–359, 1977.

[25] G. J. Chaitin and J. T. Schwartz. A note on Monte Carlo primality tests and algorithmic information theory. *Communications on Pure and Applied Mathematics*, 31(4):521–527, 1978.

[26] D. G. Champernowne. The construction of decimals normal in the scale of ten. *Journal of the London Mathematical Society*, 8:254–260, 1933.

[27] R. Colbeck and A. Kent. Private randomness expansion with untrusted devices. *Journal of Physics A: Mathematical and General*, 44:095305, 2011.

[28] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, New York, 1999.

[29] R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, Berlin, 2010.

[30] A. Eagle. Randomness is unpredictability. *The British Journal for the Philosophy of Science*, 56(4):749–790, 2005.

[31] A. Eagle. Chance versus randomness. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, Spring 2014 edition, 2014.

[32] M. Figurska, M. Stańczyk, and K. Kulesza. Humans cannot consciously generate random numbers sequences: Polemic study. *Medical Hypotheses*, 70(1):182–185, 2008.

[33] J. E. Gentle. *Random Number Generations and Monte Carlo Methods*. Springer-Verlag, New York, 2 edition, 2003.

[34] O. Goldreich. *Foundations of cryptography I: Basic Tools*. Cambridge University Press, Cambridge, 2001.

[35] R. Graham and J. H. Spencer. Ramsey theory. *Scientific American*, 262:112–117, Sept. 1990.

[36] A. Hájek. Interpretations of probability. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Stanford University, Winter 2012 edition, 2014.

[37] ID Quantique. Quantis QRNG. https://www.idquantique.com/random-number-generation/.

[38] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger. A fast and compact quantum random number generator. *Review of Scientific Instruments*, 71:1675–1680, 2000.

[39] S. B. Kochen and E. Specker. The problem of hidden variables in quantum mechanics. *Journal of Mathematics and Mechanics (now Indiana University Mathematics Journal)*, 17(1):59–87, 1967.

[40] M. G. Kovalsky, A. A. Hnilo, and M. B. Agüero. Kolmogorov complexity of sequences of random numbers generated in Bell's experiments. *arXiv:1805.07161 [quant-ph]*, 2018.

[41] A. Kulikov, M. Jerger, A. Potočnik, A. Wallraff, and A. Fedorov. Realization of a quantum random generator certified with the Kochen-Specker theorem. *Physical Review Letters*, 119:240501, 2017.

[42] A. K. Lenstra, J. P. Hughes, M. Augier, J. W. Bos, T. Kleinjung, and C. Wachter. Ron was wrong, Whit is right. Santa Barbara: IACR: 17, https://eprint.iacr.org/2012/064.pdf, 2012.

[43] G. Longo and T. Paul. The mathematics of computing between logic and physics. In S. B. Cooper and A. Sorbi, editors, *Computability in Context: Computation and Logic in the Real World*, chapter 7, pages 243–274. Imperial College Press/World Scientific, London, 2008.

[44] G. Marsaglia. Xorshift rngs. *Journal of Statistical Software*, 8(14), 2003.

[45] G. Marsaglia. On the randomness of Pi and other decimal expansions. *Interstat*, 10(5):1–17, 2005.

[46] G. Marsaglia and A. Zaman. Towards a universal random number generator. *Statistics & Probability Letters*, 9(1):35–39, 1990. http://www.stat.fsu.edu/pub/diehard/.

[47] P. Martin-Löf. The definition of random sequences. *Information and Control*, 9(6):602–619, 1966.

[48] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, 1998.

[49] M. E. O'Neill. Pcg: A family of simple fast space-efficient statistically good algorithms for random number generation. Technical Report HMC-CS-2014-0905, Harvey Mudd College, Claremont, CA, Sep 2014.

[50] Y. Peres. Iterating von Neumann's procedure for extracting random bits. *The Annals of Statistics*, 20(1):590–597, 1992.

[51] R. G. E. Pinch. The Carmichael numbers up to 10^{21}. In A.-M. Ernvall-Hytönen, M. Jutila, Juhani, Karhumäki, and A. Lepistö, editors, *Proceedings of Conference on Algorithmic Number Theory 2007*, volume 46, pages 129–131, 2007.

[52] S. Pironio, A. Acín, S. Massar, A. B. de la Giroday, D. N. Matsukevich, P. Maunz, S. Olmschenk, D. Hayes, L. Luo, T. A. Manning, and C. Monroe. Random numbers certified by Bell's Theorem. *Nature*, 464(09008), 2010.

[53] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Special Publication 800-22, NIST, 2010.

[54] J. K. Salmon, M. A. Moraes, R. O. Dror, and D. E. Shaw. Parallel random numbers: As easy as 1, 2, 3. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC11)*, New York, 2011. ACM.

[55] H. Schmidt. Quantum-mechanical random-number generator. *Journal of Applied Physics*, 41(2):462–468, 1970.

[56] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 2005.

[57] Y. Shen, L. Tian, and H. Zou. Practical quantum random nubmer generator based on measuring the shot noise of vacuum states. *Physical Review A*, 81(063814), 2010.

[58] R. Solovay and V. Strassen. Erratum: A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 7(1):118, 1977.

[59] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6(1):84–85, 1977. Corrigendum in Ref. [58].

[60] A. Stefanov, N. Gisin, O. Guinnard, L. Guinnard, and H. Zbinden. Optical quantum random number generator. *Journal of Modern Optics*, 47(4):595–598, 2000.

[61] M. Stipčević and B. M. Rogina. Quantum random number generator based on photonic emission in semiconductors. *Review of Scientific Instruments*, 78(4):045104, 2007.

[62] K. Svozil. The quantum coin toss – testing microphysical undecidability. *Physics Letters A*, 143(9):433–437, 1990.

[63] J. von Neumann. Various techniques used in connection with random digits. *National Bureau of Standards Applied Math Series*, **12** (1951), 36–38. In A. H. Traub, editor, *John von Neumann, Collected Works*, pages 768–770. MacMillan, New York, 1963.

[64] S. Wagon. Is π normal? In J. L. Berggren, J. M. Borwein, and P. B. Borwein, editors, *Pi: A Source Book*, pages 557–559. Springer-Verlag, New York, 2004.

[65] B. L. Welch. The generalization of "Student's" problem when several different population variances are involved. *Biometrika*, 34(1-2), 1947.

A 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	**0.04867**	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	0.05372	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	0.2694	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	0.1956	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	0.05397	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	0.06601	**0.02957**	$\mathbf{< 10^{-4}}$	**0.0080**	$\mathbf{< 10^{-4}}$	**0.0017**