entropy

MDPI

Article

Building Test Batteries Based on Analyzing Random Number Generator Tests within the Framework of Algorithmic Information Theory

Boris Ryabko [1,2]

[1] Federal Research Center for Information and Computational Technologies, Novosibirsk 630090, Russia; boris@ryabko.net
[2] Institute of Informatics and Computer Engineering, Siberian State University of Telecommunications and Informatics, Novosibirsk 630102, Russia

Abstract: The problem of testing random number generators is considered and a new method for comparing the power of different statistical tests is proposed. It is based on the definitions of random sequence developed in the framework of algorithmic information theory and allows comparing the power of different tests in some cases when the available methods of mathematical statistics do not distinguish between tests. In particular, it is shown that tests based on data compression methods using dictionaries should be included in test batteries.

Keywords: random number generator; statistical test; algorithmic information theory; battery of tests; data compression; universal coding; Hausdorff dimension

check for updates

Citation: Ryabko, B. Building Test Batteries Based on Analyzing Random Number Generator Tests within the Framework of Algorithmic Information Theory. *Entropy* **2024**, *26*, 513. https://doi.org/10.3390/e26060513

Academic Editors: Hector Zenil, Yi-Cheng Zhang and Shimin Cai

Received: 10 April 2024
Revised: 8 June 2024
Accepted: 11 June 2024
Published: 14 June 2024

1. Introduction

Random numbers play an important role in cryptography, gambling, Monte Carlo methods and many other applications. Nowadays, random numbers are generated using so-called random number generators (RNGs), and the "quality" of the generated numbers is evaluated using special statistical tests [1]. This problem is so important for applications that there are special standards for RNGs and for so-called test batteries, that is, sets of tests. The current practice for using an RNG is to verify the sequences it generates with tests from some battery (such as those recommended by [2,3] or other standards).

Many statistical tests are designed to test some deviations from randomness described as classes of random processes (e.g., Bernoulli process with unequal probabilities 0 and 1, Markov chains with some unknown parameters, stationary ergodic processes, etc.) [1–5].

A natural question is: how do we compare different tests and, in particular, create a suitable battery of tests? Currently, this question is mostly addressed experimentally: possible candidate tests are applied to a set of known RNGs and the tests that reject more ("bad") RNGs are suitable candidates for the battery. In addition, researchers try to choose independent tests (i.e., those that reject different RNGs) and take into account other natural properties (e.g., testing speed, etc.) [1–4]. Obviously, such an approach depends significantly on the set of selected tests and RNGs pre-selected for consideration. It is worth noting that at present there are dozens of RNGs and tests, and their number is growing fast, so the recommended batteries of tests are rather unstable (see [4]).

It is clear that increasing the number of tests in a battery increases the total testing time or, conversely, if testing time is limited, increasing the number of tests causes the length of the binary sequence being examined to decrease and therefore the power of any battery test is reduced. Therefore, it is highly desirable to include in the battery powerful tests designed for different deviations from randomness.

The goal of this paper is to develop a theoretical framework for test comparison and illustrate it by comparing some popular tests. The main idea of the proposed approach is based on the definition of randomness developed in algorithmic information theory (AIT). Apparently, it is natural to use this theory, since it is the only mathematically correct theory that formally defines what a random binary sequence is, and by definition any RNG should generate such sequences. Similar to AIT, we extend the notion of "random sequence" to any statistical test T, and then compare the "size" of the set of random sequences corresponding to different tests. More precisely, let R_{T_1} and R_{T_2} be random sequences according to T_1 and T_2. Then, if $\dim(R_{T_1} \backslash R_{T_2}) > 0$, then T_1 accepts a large set of sequences as random, whereas T_2 rejects these sequences as non-random. So, in this sense, a T_1 test cannot replace T_2 in a battery of tests (here dim is the Hausdorff dimension.).

Based on this approach, we give some practical recommendations for building test batteries. In particular, we recommend including in the test batteries a test based on a dictionary data compressor, like Lempel–Ziv codes [6], grammar-based codes [7] and some others.

The rest of this paper consists is organized as follows. The next part contains definitions and preliminary information, the third part is a comparison of the test performance on Markov processes with different memories and general stationary processes, and the fourth part investigates tests based on Lempel–Ziv data compressors. The fifth part is a brief conclusion; some of the concepts used in this paper are given in the Appendix A.

2. Definitions and Preliminaries

2.1. Hypothesis Testing

In hypothesis testing, there is a main hypothesis $H_0 = \{\text{the sequence } x \text{ is random}\}$ and an alternative hypothesis $H_1 = \neg H_0$. (In the probabilistic approach, H_0 is that the sequence is generated by a Bernoulli source with equal probabilities 0 and 1.) A test is an algorithm for which the input is the prefix $x_1 \ldots x_n$ (of the infinite sequence $x_1, \ldots, x_n, \ldots$) and the output is one of two possible words: *random* or *non-random* (meaning that the sequence is random or non-random, respectively).

Let there be a hypothesis H_0, some alternative H_1, let T be a test and τ be a statistic, that is, a function on $\{0,1\}^n$ which is applied to a binary sequence $x = x_1 \ldots x_n$. Here and below $\{0,1\}^n$ is the set of all n-bit binary words, $\{0,1\}^\infty$ is the set of all infinite words $x_1 x_2 \ldots, x_i \in \{0,1\}$.

By definition, Type I error occurs if H_0 is true and H_0 is rejected; the significance level is defined as the probability of the Type I error. Denote the critical region of the test T for the significance level α by $\bar{C}_T(\alpha, n)$ and let $C_T(\alpha, n) = \{0,1\}^n \setminus \bar{C}_T(\alpha, n)$. Recall that, by definition, H_0 is rejected if and only if $x \in \bar{C}_T(\alpha, n)$ and, hence,

$$|\bar{C}_T(\alpha, n)| \leq 2^n \alpha \,, \tag{1}$$

see [8]. We also apply another natural limitation. We consider only tests T such that for all n and $\alpha_1 < \alpha_2$ $\bar{C}_T(\alpha_1, n) \subset \bar{C}_T(\alpha_2, n)$. (Here and below, $|X|$ is the number of elements X if X is a set, and the length of X, if X is a word.)

A finite sequence $x_1 \ldots x_n$ is considered random for a given test T and the significance level α if it belongs to $C_T(\alpha, n)$.

2.2. Batteries of Tests

Let us consider a situation where the randomness testing is performed by conducting a battery of statistical tests for randomness. Suppose that the battery $\hat{T}$ contains a finite or countable set of tests $T_1, T_2, \ldots$ and α_i is the significance level of i-th test, $i = 1, 2, \ldots$. If the battery is applied in such a way that the hypothesis H_0 is rejected when at least one test in the battery rejects it, then the significance level α of this battery satisfies the following inequality:

$$\alpha \leq \sum_{i=1}^{\infty} \alpha_i \,, \tag{2}$$

because $P(A+B) \leq P(A) + P(B)$ for any events A and B (This inequality is a simple extension of the so-called Bonferroni correction, see [9]).

It will be convenient to formulate this inequality in a different way. Suppose there is some $\alpha \in (0,1)$ and a sequence ω of non-negative ω_i such that $\sum_{i=1}^{\infty} \omega_i \leq 1$. For example, we can define the following sequence ω^*:

$$\omega_i^* = 1/(i(i+1)) \quad i = 1, 2, \ldots \,. \tag{3}$$

If the significance level T_i equals $\alpha\omega_i$, then the significance level of the battery $\hat{T}$ is not grater than α. (Indeed, from (2) we obtain $\sum_{i=1} \alpha_i = \sum_{i=1}(\alpha\omega_i) = \alpha \sum_{i=1} \omega_i \leq \alpha$.) Note that this simple observation makes it possible to treat a test battery as a single test.

2.2.1. Random and Non-Random Infinite Sequences

Kolmogorov complexity is one of the central notations of algorithmic information theory (AIT), see [10–18]. We will consider the so-called prefix-free Kolmogorov complexity $K(u)$, which is defined on finite binary words u and is closely related to the notion of randomness. More precisely, an infinite binary sequence $x = x_1x_2\ldots$ is random if there exists a constant C such that

$$n - K(x_1 \ldots x_n) < C \tag{4}$$

for all n, see [19]. Conversely, the sequence x is non-random if

$$\forall\ C > 0\ \exists\, n_C \ \ n_C - K(x_1 \ldots x_{n_C}) \geq C$$

In some sense, Kolmogorov complexity is the length of the shortest lossless prefix-free code, that is, for any (algorithmically realisable) code f there exists a constant c_f for which $K(u) \leq |f(u)| + c_f$ [10–16]. Recall that a code f is lossless if there is a mapping f^{-1} such that for any word $u\, f^{-1}(f(u)) = u$ and f is prefix-free (or unprefixed) if for any words u, v, $f(u)$ is not a prefix of $f(v)$ and $f(v)$ is not a prefix of $f(u)$.

Let f be a lossless prefix-free code defined for all finite words. Similarly to (4), we call it random with respect to f if there is a constant C_f such that

$$n - |f(x_1 \ldots x_n)| < C_f \tag{5}$$

for all n. We call this difference the statistic corresponding to f and define

$$\tau_f(x_1 \ldots x_n) = n - |f(x_1 \ldots x_n)|. \tag{6}$$

Similarly, the sequence x is non-random with respect to f if

$$\forall C > 0\ \ \exists\, n_C\ \ n_C - |f(x_1 \ldots x_{n_C}| \geq C\,. \tag{7}$$

Informally, x is random with respect to f if the statistic τ_f is bounded by some constant on all prefixes $x_1 \ldots x_n$ and, conversely, x is non-random if τ_f is unbounded when the prefix length grows.

Based on these definitions, we can reformulate the concepts of randomness and non-randomness in a manner similar to what is customary in mathematical statistics. Namely, for any $\alpha \in (0,1)$ we define the set $\{y = y_1 \ldots y_n : \tau_f(y) \geq -\log \alpha\}$. It is easy to see that (1) is valid and, therefore, this set represents the critical region $\bar{C}_T(\alpha, n)$, where the test T is as follows: $T = \{x_1 \ldots x_n\colon \tau_f(x_1 \ldots x_n) < \alpha\}$.

Based on these consideration, (6) and the definitions of randomness (4), (5) we give the following definition of randomness and non-randomness for the statistic τ_f and corresponding test T_f. An infinite sequence $x = x_1x_2\ldots$ is random according to the test T_f if there exists such $\alpha > 0$ that for any integer n and this α the word $x_1 \ldots x_n$ is random (according to the T_f test). Otherwise, the sequence x is non-random.

Note that we can use the statistic

$$\tau_f = n - |f(x_1 \dots x_n)|$$

with the critical value $t_\alpha = n - \log(1/\alpha) - 1$, $\alpha \in (0,1)$, see [20,21]. So, there is no need to use the density distribution formula and it greatly simplifies the use of the test and makes it possible to use this test for any data compressor f.

It is important to note that there are tests developed within the AIT that can be used to test RNG [22,23].

2.2.2. Test Performance Comparison

For test T, let us define the set R_T of all infinite sequences that are random for T.

We use this definition to compare the "effectiveness" of different tests as follows. The test T_1 is more efficient than T_2 if the size of the difference $R_{T_2} \backslash R_{T_1}$ is not equal to zero, where the size is measured by the Hausdorff dimension.

Informally, the "smallest" set of random sequences corresponds to a test based on Kolmogorov complexity (4) (corresponding set R_K contains "truly" random sequences). For a given test T_1 we cannot calculate the difference $R_{T_1} \backslash R_K$ because the statistic (4) is noncomputabele, but in the case of two tests T_1 and T_2, where $\dim(R_{T_2} \backslash R_{T_1}) > 0$, we can say that the set of sequences random according to T_2 contains clearly non-random sequences. So, in some sense, T_1 is more efficient than T_2. (Recall that we only consider computable tests.)

The definition of the Hausdorff dimension is given in the Appendix A, but here we briefly note that we use the Hausdorff dimension for it as follows: for any binary sequence $x_1 x_2 \dots$ we define a real number $\sigma(x) = 0.\ x_1 x_2 \dots$ and for any set of infinite binary sequences S we denote the Hausdorff dimension of $\sigma(S)$ by $\dim S$. So, a test T_1 is more efficient than T_2 (formally $T_1 \succeq T_2$) if $\dim(R_{T_2} \backslash R_{T_1}) > 0$. Obviously, information about a test's effectiveness can be useful to developers of the test's batteries.

Also note that the Hausdorff dimension is widely used in information theory. Perhaps the first such use was due to Eggleston [24] (see also [25,26]), and later the Hausdorff dimension found numerous applications in AIT [27–29].

2.2.3. Shannon Entropy

In RNG testing, one of the popular alternative hypotheses (H_1) is that the considered sequence generated by Markov process of memory (or connectivity) $m, m > 0$, (S_m), but the transition probabilities are unknown. (S_0, i.e., $m = 0$, corresponds to the Bernoulli process). Another popular and perhaps the most general H_1 is that the sequence is generated by a stationary ergodic process (S_∞) (excluding H_0).

Let us consider the Bernoulli process $\mu \in S_0$ for which $\mu(0) = p, \mu(1) = q, (p + q = 1)$. By definition, the Shannon entropy $h(\mu)$ of this process is defined as $h(\mu) = -(p \log p + q \log q)$ [30]. For any stationary ergodic process $\nu \in S$ the entropy of order k is defined as follows:

$$h_k(\nu) = E_\nu(-\sum_{u \in \{0,1\}^k} (\nu(0/u)\log(0/u) + \nu(1/u)\log \nu(1/u))),$$

where E_ν is the mathematical expectation according to ν, $\nu(z/u)$ is the conditional probability $\nu(x_{i+1} = z | x_{i-k} \dots x_i = u)$, it does not depend on i due to stationarity [30].

It is known in Information Theory that for stationary ergodic processes (including S_∞ and $S_m, m \geq 0$) $h_k \geq h_{k+1}$ for $k \geq 0$ and there exists the limit Shannon entropy $h_\infty(\nu) = \lim h_k(\nu)$. Besides, for $\nu \in S_m$ $h_\infty = h_m$ [30].

Shannon entropy plays an important role in data compression because for any lossless and prefix-free code, the average codeword length (per letter) is at least as large as the entropy, and this limit can be reached. More precisely, let ϕ be a lossless, prefix-free code defined on $\{0,1\}^n, n > 0$, and let $\nu \in S$. Then, for any ϕ, ν, and codewords of average length

$$E_n(\phi, \nu) = \frac{1}{n} \sum_{u \in \{0,1\}^n} \nu(u) |\phi(u)| \tag{8}$$

$E_n(\phi, \nu) \geq h(\nu)$. In addition, there are codes $\phi_1, \phi_2, \ldots$ such that $\lim_{n\to\infty} E_n(\phi_n, \nu) = h(\nu)$ [30].

2.2.4. Typical Sequences and Universal Codes

The sequence $x_1 x_2 \ldots$ is typical for the measure $\mu \in S_\infty$ if for any word $y_1 \ldots y_r$ $\lim_{t\to\infty} N_{x_1\ldots x_t}(y_1 \ldots y_r)/t = \mu(u)$, where $N_{x_1\ldots x_t}(y_1 \ldots y_r)$ is the number of occurrences of a word $y_1 \ldots y_r$ in a word $x_1 \ldots x_t$.

Let us denote the set of all typical sequences as $\mathbf{T}_\mu$ and note that $\mu(\mathbf{T}_\mu) = 1$ [30]. This notion is deeply related to information theory. Thus, Eggleston proved the equality $\dim \mathbf{T}_\mu = h(\mu)$ for Bernoulli processes ($\mu \in S_0$) [24], and later this was generalized for $\mu \in S_\infty$ [26,28].

By definition, a code ϕ is universal for a set of processes S if for any $\mu \in S$ and any $x \in \mathbf{T}_\mu$

$$\lim_{n\to\infty} |\phi(x_1 \ldots x_n)|/n = h_\infty(\mu). \tag{9}$$

In 1968, R. Krichevsky [31] proposed a code $\kappa_m^t(x_1 \ldots x_t)$, $m \geq 0, t$ is an integer, whose redundancy, i.e., the average difference between the code length and Shannon entropy, is asymptotically minimal. This code and its generalisations are described in the Appendix A, but here we note the following main property. For any stationary ergodic process μ, that is, $\mu \in S_\infty$ and typical $x \in \mathbf{T}_\mu$,

$$\lim_{t\to\infty} |\kappa_m^t(x_1 \ldots x_t)|/t = h_m(\mu)\,, \tag{10}$$

see [32].

Currently there are many universal codes which are based on different ideas and approaches, among which we note the PPM universal code [33], the arithmetic code [34], the Burrows–Wheeler transform [35], which is used along with the book-stack (or MTF) code [36–38], and some others [39–41].

The most interesting for us is the class of grammar-based codes suggested by Kieffer and Yang [7,42] which includes the Lempel–Ziv (LZ) codes [6] (note that perhaps the first grammar-based code was described in [43]).

The point is that all of them are universal codes and hence they "compress" stationary processes asymptotically to entropy and therefore cannot be distinguishable at S_∞. On the other hand, we show that grammar-based codes can distinguish "large" sets of sequences as non-random beyond S_∞.

2.2.5. Two-Faced Processes

The so-called two-faced processes are described in [20,21] and their definitions will be given in Appendix A. Here, we note some of their properties: the set of two-faced processes $\Lambda_s(p)$ of order $s, s \geq 1$, and probability $p, p \in (0,1)$, contains the measures λ from S_s such that

$$\begin{aligned} h_0(\lambda) = h_1(\lambda) = \cdots = h_{s-1}(\lambda) = 1, \\ h_s(\lambda) = h_\infty(\lambda) = -(p \log p + (1-p) \log(1-p)). \end{aligned} \tag{11}$$

Note that they are called two-faced because they appear to be truly random if we look at word frequencies whose length is less than s, but are "completely" non-random if the word length is equal to or greater than s (and p is far from $1/2$).

3. Comparison of the Efficiency of Tests for Markov Processes with Different Memories and General Stationary Processes

We now describe the statistical tests for Markov processes and stationary ergodic processes as follows. By (6), statistical definitions are as follows:

$$\tau_{K^t_m}(x_1 \dots x_n) = n - |\hat{\kappa}^t_m(x_1 \dots x_n)|,$$

$$\tau_{R^t}(x_1 \dots x_n) = n - |\hat{\rho}^t(x_1 \dots x_n)|$$

where $\hat{\kappa}^t_m$ and $\hat{\rho}^t$ are universal codes for S_m and S_∞ defined in the Appendix A, see (A4) and (A5). We also denote the corresponding tests by $T^t_{K_m}$ and T^t_R. The following statement compares the performance of these tests.

Theorem 1. *For any integers m, s and $t = ms$*

$$T^t_{K_m} \preceq T^t_{K_{m+1}}, \; T^t_{K_m} \preceq T^t_{K_R}.$$

Moreover, $\dim(T^t_{K_m} \setminus T^t_{K_{m+1}}) = 1$.

Proof. First, let us say a few words about the scheme of the proof. If we apply the $T^t_{K_m}$ test to typical sequences of a two-faced process $\lambda \in \mathbf{T}_{\Lambda_{m+1(p)}}$, $p \neq 1/2$, they will appear random since $h_m(\lambda) = 1$. So, the set of random sequences $R_{T^t_{K_m}}$ (i.e., random sequences according to $T^t_{K_m}$ test) contains the set of the typical sequences $\mathbf{T}_{\Lambda_{m+1(p)}}$ for which $\dim(\mathbf{T}_{\Lambda_{m+1(p)}})$ equals the limit Shannon entropy $-(p \log p + (1-p)\log(1-p))$. Hence, $\dim(R_{T^t_{K_m}}) \geq \dim(\mathbf{T}_{\Lambda_{m+1(p)}}) = -(p \log p + (1-p)\log(1-p))$.

On the other hand, typical sequences of a two-faced process $\lambda \in \mathbf{T}_{\Lambda_{m+1(p)}}, p \neq 1/2$ are not random according to $T^t_{K_{m+1}}$ since $h_{m+1}(\lambda) = -(p \log p + (1-p)\log(1-p)) \; < \; 1$ (11) and the test. $T^t_{K_m}$ "compresses" them till the Shannon entropy $-(p \log p + (1-p)\log(1-p))$. So, $\dim(R_{T^t_{K_m}} \setminus R_{T^t_{K_{m+1}}}) \geq \dim(R_{T^t_{K_m}}) \geq -(p \log p + (1-p)\log(1-p))$. Then $\sup_{p \in (0,1/2)}$ $\dim(T^t_{K_m} \setminus T^t_{K_{m+1}}) = 1$.

More formally, consider a typical sequence x of $\mathbf{T}_{\Lambda_{m+1(p)}}$, $p \neq 1/2$. So, $\lim_{t \to \infty} - \sum_{u \in \{0,1\}^{m+1}} (N_{x_1 \dots x_t}(u)/t) \log(N_{x_1 \dots x_t}(u)/t) = h_\lambda(m) = 1$, see (11), where the first equality is due to typicality, and the second to the property of two-faced processes (11).

From here and (A1), (A4) we obtain $E_\lambda(1/n)|\hat{\kappa}^t_m(x_1 \dots x_n)| = 1 + \epsilon$, where $\epsilon > 0$. From this and typicality we can see that $\lim_{n \to \infty} |\hat{\kappa}^t_m(x_1 \dots x_n)|/n = 1 + \epsilon$. Hence, there exists such n_δ that $1 + \epsilon - \delta < |\hat{\kappa}^t_m(x_1 \dots x_n)|/n < 1 + \epsilon + \delta$, if $n > n_\delta$. So $n - |\hat{\kappa}^t_m(x_1 \dots x_n)|$ $\leq n - (n + \epsilon - \delta)$. So, if we take $\delta = \epsilon/2$, we can see that for $n > n_\delta$ $n - |\hat{\kappa}^t_m(x_1 \dots x_n)|$ is negative. From this and the definition of randomness (5), we can see that typical sequences from $\mathbf{T}_{\Lambda_{m+1(p)}}$ are random according to $\hat{\kappa}^t_m(x_1 \dots x_n)$, i.e., $T^t_{K_m}$. From this and (A6), we obtain $T^t_{K_{m+1}} \preceq T^t_R$. □

4. Effectiveness of Tests Based on Lempel-Ziv Data Compressors

In this part we will describe a test that is more effective than T^t_R and $T^t_{K_m}$ for any m.

First, we will briefly describe the LZ77 code based on the definition in [44]. Suppose there is a binary string σ^* that is encoded using the code LZ77. This string is represented by a list of pairs $(p_1; l_1) \dots (p_s; l_s)$. Each pair $(p_i; l_i)$ represents a string, and the concatenation of these strings is σ^*. In particular, if $p_i = 0$, then the pair represents the string l_i, which is a single terminal. If $p_i \neq 0$, then the pair represents a portion of the prefix of σ^* that is represented by the preceding $i - 1$ pairs; namely, the l_i terminals beginning at position p_i in σ^*; see ([44] part 3.1). The length of the codeword depends on the encoding of the sub-words p_i, l_i which are integers. For this purpose we will use a prefix code C for integers, for which for any integer m

$$|C(m)| = \log m + 2 \log\log(m+1) + O(1). \tag{12}$$

Such codes are known in information theory; see, for example, ([30] part 7.2). Note that C is the prefix code and, hence, for any $r \geq 1$ the codeword $C(p_1)C(l_1) \dots C(p_r)C(l_r)$ can be decoded to $(p_1; l_1) \dots (p_r; l_r)$. There is the following upper bound for the length of the LZ77 code [30,44]: for any word $w_1 w_2 \dots . w_m$

$$|code_{LZ}(w_1 w_2 \dots w_m)| \leq m\,(1 + o(1)), \tag{13}$$

if $m \to \infty$.

We will now describe such sequences that, on the one hand, are not typical for any stationary ergodic measure and, on the other hand, are not random and will be rejected by the suggested test. Thus, the proposed model allows us to detect non-random sequences that are not typical for for any stationary processes.On the other hand, those sequences are recognized tests based on LZ77 as non-random. To do this, we take any random sequence $x = x_1 x_2 \dots$ (that is, for which (4) is valid) and define a new sequence $y(x) = y_1 y_2 \dots$ as follows. Let for $k = 0, 1, 2, \dots$

$$u_k = x_{2^{2^k}-1} x_{2^{2^k}} x_{2^{2^k}+1} \dots x_{2^{2^{k+1}}-2}$$

$$y(x) = u_0 u_0 u_1 u_1 u_2 u_2 u_3 u_3 \dots \tag{14}$$

For example, $u_0 = x_1 x_2$, $u_1 = x_3\ x_4 \dots\ x_{14}$, $u_2 = x_{15} \dots x_{254}$, $y(x) = x_1 x_2\ x_1 x_2 x_3 x_4 \dots x_{14}$ $x_3 x_4 \dots x_{14}\ x_{15} \dots x_{254}\ x_{15} \dots x_{254} \dots .$

The idea behind this sequence is quite clear. Firstly, it is obvious that the word y cannot be typical for a stationary ergodic source and, secondly, when $u_0 u_0 u_1 u_1 \dots u_k u_k$ is encoded the second subword u_k will be encoded by a very short word (about $O(\log |u_k|)$), since it coincides with the previous word u_k. So, for large k the length of the encoded word $LZ(u_0 u_0 u_1 u_1 \dots u_k u_k)$ will be about $|u_0 u_0 u_1 u_1 \dots u_k u_k|\,(1/2 + o(1))$. So, $\liminf_{n \to \infty} |LZ(y_1 y_2 \dots y_n)|/n = 1/2$. Hence, it follows that

$$\dim(\{y(x) : x\, is\, random\}) \,=\, 1/2. \tag{15}$$

Here, we took into account that x is random and, $\dim\{x : x \ is \ random\} = 1$, see [28].) So, having taken into account the definitions of non-randomness (6) and (7), we can see that $y(x)$ is non random according to statistics $\tau = n - |LZ(y_1 \dots y_n)|$. Denote this test by T_{LZ}.

Let us consider the test $T^t_{K_m}$, m, t are integers. Having taken into account that the sequence x is random, we can see that $\lim_{t \to \infty} |\kappa^t_m(x_i x_{i+1} \dots x_{i+t}|/t = 1$. So, from from (A4) we can see that for any n $|\hat\kappa^t_m(x_1 \dots x_n|/t = 1 + o(1)$. The same reasoning is true for the code $\hat\rho^t$.

We can now compare the size of random sequence sets across different tests as follows:

$$R_{T^t_{K_m}} \backslash R_{T_{LZ}} \supset \{y(x) : x\, is\, random\}\,.$$

Taking into account (15), we can see that

$$\dim(R_{T^t_{K_m}} \backslash R_{T_{LZ}}) \geq 1/2\,.$$

Likewise, the same is true for the T_R test. From the latest inequality we obtain the following

Theorem 2. *For any random (according to (4)) sequence x the sequence $y(x)$ is non-random for the test T_{LZ}, whereas this sequence is random for tests T^t_R and $T^t_{K_m}$. Moreover, $T^t_R \preceq T_{LZ}$ and $T^t_{K_m} \preceq T_{LZ}$ for any m, t.*

Comment. The sequence $y(x)$ is constructed by duplicating parts of x. This construction can be slightly modified as follows: instead of duplication (as $u_i u_i$), we can use $u_i u_i^\gamma$, where u_i^γ contains $\gamma|u|$ the first letters of u, $\gamma < 1/2$. In this case, $\dim(R_{T^t_{Km}} \backslash R_{T_{LZ}}) \geq 1 - \gamma$ and, therefore,

$$\sup_{\gamma \in (0,1/2)} \dim(R_{T^t_{Km}} \backslash R_{T_{LZ}}) = 1\,.$$

5. Conclusions

Here, we describe some recommendations for the practical testing of RNGs, based on the method of comparing the power of different statistical tests. Based on Theorem 1, we can recommend to use several tests $T_{K^t_s}$, based on the analysis of occurrence frequencies of words of different length s. In addition, we recommend using tests for which s depends on the length n of the sequence under consideration. For example, $s_1 = O(\log\log n)$), $s_2 = O(\sqrt{\log n})$, etc. They can be included in the test battery directly or as the "mixture" T_R with several non-zero β coefficients, see (A2) in the Appendix A.

Theorem 2 shows that it is useful to include tests based on dictionary data compressors such as the Lempel–Ziv code. In such a case we can use the statistic

$$\tau_{LZ} = n - |LZ(y_1 \dots y_n)|$$

with the critical value $t_\alpha = n - \log(1/\alpha) - 1$, $\alpha \in (0,1)$, see [20,21]. Note that in this case, there is no need to use the density distribution formula, which greatly simplifies the use of the test and makes it possible to use a similar test for any grammar-based data compressor.

Funding: This study was completed with the support of the state assignment of SibSUTIS (reg. no. 071-03-2024-008.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The author declare no conflicts of interest.

Appendix A

Appendix A.1. Hausdorff Dimension

Let $A \subset [0,1], \rho > 0$. A family of sets S is called a ρ-covering A if

(i) S is finite or countable, (ii) any $\sigma \subset [0,1]$ and its length is not greater than ρ and (iii) $\cup_{\sigma \in S} \sigma \supset A$. Let

$$l(\alpha, A, \rho) = \inf \sum_{\sigma \in S} diam(\sigma)^\alpha,$$

where the infimum is taken over all ρ-coverings. Then, Hausdorff dimension $\dim(A)$ is determined by the equality

$$\dim(a) = \inf_\alpha \lim_{\rho \to 0} l(\alpha, A, \rho) = 0 = \sup_\alpha \lim_{\rho \to 0} l(\alpha, A, \rho) = \infty.$$

Appendix A.2. Krichevsky Universal Code and Twice-Universal Code

Krichevsky in [31] described the following measure K_0 and universal code κ_0 for Bernoulli processes, which in the case of the binary alphabet looks like

$$K_0^t(x_1 x_2 \dots x_t) = \prod_{i=0}^{t-1} \frac{N_{x_1 \dots x_i}(x_{i+1}) + 1/2}{i+1},$$

$$\kappa_0^t(x_1 x_2 \dots x_t) = \lceil -\log K_0(x_1 x_2 \dots x_t) \rceil\,.$$

Then, he generalized them for Markov chains of memory m, $m > 0$ [32], as follows:

$$K_m^t(x_1 \dots x_t) = \begin{cases} \frac{1}{2^t} & \text{if } t \leq \ m \\ \frac{1}{2^t} \prod_{i=m}^{t-1} \frac{N_{x_1 \dots x_i}(x_{i+1-m} \dots x_{i+1}) + 1/2}{N_{x_1 \dots x_{i-1}}(x_{i+1-m} \dots x_i) + 1} & \text{if } t > m, \end{cases}$$

$\kappa_m^t(x_1 \dots x_t) = \lceil -\log K_m^t(x_1 \dots x_t) \rceil$, see [32]. For example,

$$K_0^5(01010) = \frac{1/2}{1} \frac{1/2}{2} \frac{132}{3} \frac{3/2}{4} \frac{5/2}{5},$$

$$K_1^5(01010) = \frac{1}{2} \frac{1/2}{1} \frac{1/2}{1} \frac{3/2}{2} \frac{3/2}{2}.$$

The code κ_m^t is universal for a set of processes S_m, and , for any $\nu \in S_m$

$$h_m(\nu) < E_\nu(\kappa_m^t, \nu) \leq h_m(\nu) + 2^m \log t/(2t) + O(1/t) \tag{A1}$$

Refs. [31,32]. (This code is optimal in the sense that the redundancy, that is $2^m \log t/(2t) + (1/t)$, is asymptotically minimal [31,32].)

One of the first universal codes for the set of all stationary ergodic processes S_∞ was proposed in [45]. For this code, the measure ρ and the code length R are defined as follows:

$$R^t(x_1 \dots x_t) = \sum_{i=0}^{\infty} \beta_i K_i^t(x_1 x_2 \dots x_t), \tag{A2}$$

$$\rho^t(x_1 \dots x_t) = \lceil -\log R^t(x_1 x_2 \dots x_t) \rceil,$$

where $\sum_{i=0}^{\infty} \beta_i = 1$ and $\forall i : \beta_i > 0$. Obviously, for any j

$$-\log \sum_{i=0}^{\infty} \beta_i K_i^t(x_1 x_2 \dots x_t) = -\log \beta_j K_j^t(x_1 x_2 \dots x_t) +$$

$$-\log(1 + \sum_{i=0, i \neq j}^{\infty} \beta_i K_i^t(x_1 x_2 \dots x_t)/(\beta_j K_j^t(x_1 x_2 \dots x_t))$$

$$\leq -\log \beta_j K_j^t(x_1 x_2 \dots x_t).$$

Hence,

$$\rho^t(x_1 \dots x_t) \leq \lceil -\log \beta_j - \log K_j^t(x_1 x_2 \dots x_t) \rceil \leq \lceil -\log \beta_j \rceil$$

$$+ \lceil -\log K_j^t(x_1 x_2 \dots x_t) \rceil = \lceil -\log \beta_j \rceil + |\kappa_j^t(x_1 x_2 \dots x_t)|. \tag{A3}$$

This code is called twice universal [45] because it can be used to compress data when both the process memory and the probability distribution are unknown.

Usually, when using universal codes, the sequence $x_1 \dots x_n$ is encoded in parts as follows:

$$\hat{\kappa}_m^t(x_1 \dots x_n) = \kappa_m^t(x_1 \dots x_t) \kappa_m^t(x_{t+1} \dots x_{2t}) \dots . \kappa_m^t(x_{n-t+1} \dots x_n) \tag{A4}$$

(for brevity, we assume that n/t is an integer). Let us similarly define

$$\hat{\rho}^t(x_1 \dots x_n) = \rho^t(x_1 \dots x_t) \rho^t(x_{t+1} \dots x_{2t}) \dots . \rho^t(x_{n-t+1} \dots x_n) \tag{A5}$$

Taking into account the definition of $\kappa_j^t(x_1 x_2 \dots x_t)$ and Equations (4), (A4) and (A5) we obtan that for any integer j

$$\hat{\rho}^t(x_1 \dots x_n) \leq \hat{\kappa}_j^t(x_1 \dots x_n) + O(n/t). \tag{A6}$$

Appendix A.3. Two-Faced Processes

Let us first consider several examples of two-faced Markov chains. Let a matrix of transition probabilities T_1 be as follows:

$$T_1 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & \nu & 1-\nu \\ 1 & 1-\nu & \nu \end{array} \quad ,$$

where $\nu \in (0,1)$ (i.e., $P\{x_{i+1} = 0|x_i = 0\} = \nu$, $P\{x_{i+1} = 0|x_i = 1\} = 1-\nu, \dots$). The "typical" sequences for $\nu = 0.9$ and $\nu = 0.1$ can be as follows:

$$0000000000 \;\; 111111111 \;\; 0000000000 \;\; 1111111 \;\; 0 \dots ,$$

$$01010101 \;\; 1010101010 \;\; 010101010101010101 \;\; 1010 \dots \; .$$

(Here, the gaps correspond to state transitions.) Of course, these sequences are not truly random. On the other hand, the frequencies of 1s and 0s go to 1/2 due to the symmetry of the matrix T_1.

Define

$$\hat{T}_1 = \begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1-\nu & \nu \\ 1 & \nu & 1-\nu \end{array}$$

$$T_2 = (T_1\hat{T}_1) = \begin{array}{c|cccc} & 00 & 01 & 10 & 11 \\ \hline 0 & \nu & 1-\nu & 1-\nu & \nu \\ 1 & 1-\nu & \nu & \nu & 1-\nu \end{array}$$

(Here $P\{x_{i+1} = 0|x_i = 0, x_{i-1} = 0\} = \nu$, $P\{x_{i+1} = 0|x_i = 0, x_{i-1} = 1\} = 1-\nu, \dots .$)

Now, we can define a transition matrix with two-faced Markov chains with different memory as follows.

The $k+1$-order transition matrix $T_{k+1} = T_k\hat{T}_k$, $\hat{T}_{k+1} = \hat{T}_k T_k$, $k = 2, 3, \dots .$ T In order to define the process $x_1x_2\dots$ the initial probability distribution needs to be specified. We define the initial distribution of the processes T_k and $\bar{T}_k$, $k = 1, 2, \dots,$ to be uniform on $\{0,1\}^k$, i.e., $P\{x_1 \dots x_k = u\} = 2^{-k}$ for any $u \in \{0,1\}^k$.

The following statement from [20,21] describes the main properties of the processes defined above.

Claim. Let a sequence $x_1x_2\dots$ be generated by the process T_k (or $\bar{T}_k$), $k \geq 1$ and u be a binary word of length k. Then, if the initial state obeys the uniform distribution over $\{0,1\}^k$, then

(i) For any $j \geq 0$

$$P(x_{j+1} \dots x_{j+k} = u) = 2^{-|u|}. \tag{A7}$$

(ii) For each $\nu \in (0,1)$, the k-order Shannon entropy (h_k) of the processes T_k and $\bar{T}_k$, equals 1 bit per letter, whereas the limit Shannon entropy (h_∞) equals $-(\nu \log_2 \nu + (1-\nu)\log_2(1-\nu))$.

References

1. L'Ecuyer, P. *Random Number Generation*; Springer: Berlin/Heidelberg, Germany, 2012.
2. L'Ecuyer, P.; Simard, R. TestU01: AC library for empirical testing of random number generators. *ACM Trans. Math. Softw.* **2007**, *33*, 22. Available online: http://simul.iro.umontreal.ca/testu01/tu01.html (accessed on 10 June 2024) [CrossRef]
3. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E.; Leigh, S.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; National Institute of Standards and Technology: Gaithersburg, MD, USA , 2010.

4. Hurley-Smith, D.; Patsakis, C.; Hernez-Castro, J. On the unbearable lightness of FIPS 140–2 randomness tests. *IEEE Trans. Inf. Forensics Secur.* **2020**, *17*, 3946–3958. [CrossRef]
5. Ryabko, B. Asymptotically most powerful tests for random number generators. *J. Stat. Plan. Inference* **2022**, *217*, 1–7. [CrossRef]
6. Ziv, J.; Lempel, A. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* **1977**, *23*, 337–343. [CrossRef]
7. Yang, E.-H.; Kieffer, J.C. Efficient universal lossless data compression algorithms based on a greedy sequential grammar transform. i. without context models. *IEEE Trans. Inf. Theory* **2000**, *46*, 755–777. [CrossRef]
8. Kendall, M.; Stuart, A. *The Advanced Theory of Statistics*; Volume 2: Inference and Relationship; Hafner Publishing Company: New York, NY, USA, 1961.
9. Mittelhammer, R.C.; Judge, G.G.; Miller, D.J. *Econometric Foundations*; Cambridge University Press: Cambridge, UK, 2000; pp. 73–74.
10. Hutter, M. Algorithmic information theory. *Scholarpedia* **2007**, *2*, 2519. [CrossRef]
11. Li, M.; Vitányi, P. *An Introduction to Kolmogorov Complexity and Its Applications*; Springer: New York, NY, USA, 2008.
12. Calude, C.S. Information: The algorithmic paradigm. In *Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 79–94.
13. Downey, R.; Hirschfeldt, D.R.; Nies, A.; Terwijn, S.A. Calibrating randomness. *Bull. Symb. Log.* **2006**, *12*, 411–491. [CrossRef]
14. Merkle, W.; Miller, J.S.; Nies, A.; Reimann, J.; Stephan, F. Kolmogorov–loveland randomness and stochasticity. *Ann. Pure Appl. Log.* **2006**, *138*, 183–210. [CrossRef]
15. V'yugin, V. On Nonstochastic Objects. *Probl. Peredachi Inf.* **1985**, *21*, 3–9.
16. Vereshchagin, N. Algorithmic Minimal Sufficient Statistics: A New Approach. *Theory Comput. Syst.* **2015**, *58*, 463–481. [CrossRef]
17. Zenil H. A review of methods for estimating algorithmic complexity: Options, challenges, and new directions. *Entropy* **2020**, *22*, 612. [CrossRef]
18. Zenil, H.; Kiani, N.A.; Tegnér, J. Low-algorithmic-complexity entropy-deceiving graphs. *Phys. Rev.* **2017**, *96*, 012308.
19. Downey, R.G.; Reimann, J. Algorithmic randomness. *Scholarpedia* **2007**, *2*, 2574. [CrossRef]
20. Ryabko, B.Y.; Monarev, V.A. Using information theory approach to randomness testing. *J. Stat. Plan. Inference* **2005**, *133*, 95–110. [CrossRef]
21. Ryabko, B.; Fionov, A. *Cryptography in the Information Society*; World Scientific Publishing: Hackensack, NJ, USA, 2020; p. 280.
22. Soler-Toscano, F.; Zenil, H.; Delahaye, J.P.; Gauvrit, N. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PLoS ONE* **2014**, *9*, e96223. [CrossRef]
23. Zenil H, Hernández-Orozco S, Kiani NA, Soler-Toscano F, Rueda-Toicen A, Tegnér J. A decomposition method for global evaluation of Shannon entropy and local estimations of algorithmic complexity. *Entropy* **2018**, *20*, 605. [CrossRef]
24. Eggleston, H.G. The fractional dimension of a set defined by decimal properties. *Q. J. Math.* **1949**, *1*, 31–36. [CrossRef]
25. Billingsley, P. Hausdorff dimension in probability theory. *Ill. J. Math.* **1960**, *4*, 187–209. [CrossRef]
26. Billingsley, P. *Ergodic Theory and Information*; Wiley: New York, NY, USA, 1965.
27. Lutz, J.H. The dimensions of individual strings and sequences. *Inform. Comput.* **2003**, *187*, 49–79. [CrossRef]
28. Reimann, J. Information vs. Dimension: An Algorithmic Perspective. In *Structure and Randomness in Computability and Set Theory*; World Scientific: Singapore, 2020; pp. 111–151.
29. Tadaki, K. Partial randomness and dimension of recursively enumerable reals. In *International Symposium on Mathematical Foundations of Computer Science*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 687–699.
30. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*; Wiley-Interscience: New York, NY, USA, 2006.
31. Krichevsky, R. A relation between the plausibility of information about a source and encoding redundancy. *Probl. Inform. Transm.* **1968**, *4*, 48–57.
32. Krichevsky, R. *Universal Compression and Retrival*; Kluwer Academic Publishers: Norwell, MA, USA, 1993.
33. Cleary, J.; Witten, I. Data compression using adaptive coding and partial string matching. *IEEE Trans. Commun.* **1984**, *32*, 396–402. [CrossRef]
34. Rissanen, J.; Langdon, G.G. Arithmetic coding. *IBM J. Res. Dev.* **1979**, *23*, 149–162. [CrossRef]
35. Burrows, M.; Wheeler, D.J. A Block-Sorting Lossless Data Compression Algorithm. 1994. Available online: http://www.eecs.harvard.edu/~michaelm/CS222/burrows-wheeler.pdf (accessed on 10 June 2024).
36. Ryabko, B.Y. Data compression by means of a "book stack". *Probl. Inf. Transm.* **1980**, *16*, 265–269.
37. Bentley, J.; Sleator, D.; Tarjan, R.; Wei, V. A locally adaptive data compression scheme. *Commun. ACM* **1986**, *29*, 320–330. [CrossRef]
38. Ryabko, B.; Horspool, N.R.; Cormack, G.V.; Sekar, S.; Ahuja, S.B. Technical correspondence. *Commun. ACM* **1987**, *30*, 792–797.
39. Louchard, G.; Szpankowski, W. Average profile and limiting distribution for a phrase size in the Lempel-Ziv parsing algorithm. *IEEE Trans. Inf. Theory* **1995**, *41*, 478–488. [CrossRef]
40. Drmota, M.; Reznik, Y.; Szpankowski, W. Tunstall code, Khodak variations, and random walks. *IEEE Trans. Inf. Theory* **2010**, *56*, 2928–2937. [CrossRef]
41. Reznik, Y.A. Coding of Sets of Words. In Proceedings of the 2011 Data Compression Conference, Snowbird, UT, USA, 29–31 March 2011; IEEE: Piscataway, NJ, USA, 2011.
42. Kieffer, J.C.; Yang, E.-H. Grammar-based codes: A new class of universal lossless source codes. *IEEE Trans. Inf. Theory* **2000**, *46*, 737–754. [CrossRef]

43. Kurapova, E.V.; Ryabko, B.Y. Application of Formal Grammars for Encoding Information Sources. *Probl. Inform. Transm.* **1995**, *31*, 23–26
44. Charikar, M.; Lehman, E.; Liu, D.; Panigrahy, R.; Prabhakaran, M.; Rasala, A.; Sahai, A.; Shelat, A. Approximating the smallest grammar: Kolmogorov complexity in natural models. In Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing, Montreal, QC, Canada, 19–21 May 2002; pp. 792–801.
45. Ryabko, B. Twice-universal coding. *Probl. Inf. Transm.* **1984**, *3*, 173–177.

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