ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **18**, Number 1, 2015, 18–32

On the Interpretation of Results from the NIST Statistical Test Suite

Marek SÝS[1], Zdeněk ŘÍHA[1], Vashek MATYÁŠ[1],
Kinga MÁRTON[2], Alin SUCIU[2]

[1] Masaryk University, Brno, Czech Republic,
[2] Technical University of Cluj-Napoca, Romania

E-mail: {syso, zriha, matyas}@fi.muni.cz,
{kinga.marton, alin.suciu}@cs.utcluj.ro

Abstract. NIST Statistical Test Suite is an important testing suite for randomness analysis often used for formal certifications or approvals. Documentation of the NIST STS gives some guidance on how to interpret results of the NIST STS but interpretation is not clear enough or it uses just approximated values. Moreover NIST considers data to be random if all tests are passed yet even truly random data shows a high probability (80%) of failing at least one NIST STS test. If data fail some tests the NIST STS recommends the analysis of different samples. We analysed 819200 sequences (100 GB of data) produced by a physical source of randomness (quantum random number generator) in order to interpret results computed without analysing any additional samples. The results indicate that data can be still considered random for the significance level $\alpha = 0.01$ if they fail less than 7 NIST STS tests, 7 tests of uniformity of p-values (100 sequences) or 10 tests of proportion of passing sequences. We have also defined a more accurate interval of acceptable proportions computed with a new constant (2.6 instead of 3) for which 1000 sequences can be considered random if they fail less than 7 tests of proportion.

Key-words: NIST STS; statistical randomness testing; hypothesis testing.

1. Introduction

Randomness plays an important role in many areas of cryptography. Generating random numbers is a difficult task and so is the quality evaluation of the generated data. In practice randomness assessment relies heavily on empirical tests of randomness. Each test examines the randomness quality of data from a specific point of view, testing certain statistical features, such as the frequency of ones or m-bit blocks in the data, etc. The majority of empirical randomness tests are based on statistical hypothesis testing. Each test compares certain characteristics of data (frequency of ones, frequency of m-bit blocks, etc.) with the expected test statistic (0.5, 2^{-m}, etc.) that is precomputed for random infinite sequences. In this context randomness is a probabilistic property and it can be characterized and described in terms of probability. This is due to the fact that even a good random number generator produces sequences (for instance sequence of all ones) with characteristics significantly different from the values expected in tests. Therefore we are not able to distinguish whether a given sequence with "bad" characteristics was produced by a defective generator or the sequence was produced by a good generator by chance. In the context of empirical tests of randomness, randomness is described as the probability that a perfect random number generator would produce sequences with the same or less randomness quality than those exhibited by the analysed sequence. Since statistical randomness can be tested from many points of view, tests are usually grouped into testing suites (also called batteries) to provide a more comprehensive randomness analysis. There are three commonly used testing suites for randomness analysis: NIST Statististical Test Suite [1], Dieharder [3] (a novel version of the Diehard battery) and TestU01 [4]. The NIST STS has a special importance since it was published as a NIST standard (also used for selecting AES) and it is used for the preparation of many formal certifications or approvals.

Results of statistical tests of randomness are typically in the form of a p-value which represents the probability that a perfect random number generator would produce less random sequences than the sequence being tested. Although the p-value of a randomness test focusing on a single characteristic has a clear statistical interpretation, the interpretation of the results of testing suites (including multiple tests) is problematic. Empirical tests of randomness and their results are usually dependent and correlated. For instance, if frequencies of ones and zeroes are biased (non-equal) for a given sequence it is likely that frequencies of 2-bit blocks are biased too. For a clear statistical interpretation of results (set of p-values) we need to analyse dependency/correlation between results of tests applied on random data since randomness is expressed as the probability relative to random sequences.

In our work we focus on the interpretation of the results provided by NIST STS but the proposed approach can be used for other suites as well. Documentation of the NIST STS gives some guidance on how to interpret results of the NIST STS tests (see Section 4.2 of [1]) but "it is up to the tester to determine the correct interpretation of the test results"[1]. Moreover the interpretation of results is not clear enough and therefore "some clear guidance does need to be given in the interpretation of results" [6]. The goal of our work is to give a correct interpretation of the results of each testing procedure (proportion of passing sequences, uniformity of p-values) implemented in the NIST STS. The major contribution of the paper includes the improved formula computing an interval of acceptable proportions of passing sequences. In order to interpret the level of randomness from the perspective of the whole test suite we analysed the dependency between the results of procedures for particular NIST STS tests. We tested 100GB of data produced by a physical source of randomness [8] using a new optimized implementation of the NIST STS [5]. The results of the dependency analysis indicate that particular tests are interdependent and a random sequence fails usually more tests than it can be expected for independent tests. The obtained reference results (obtained for truly random data) can also be used for a more accurate interpretation of results computed by the whole test suite.

This paper is organized as follows: Section 2 represents a brief introduction into the theory behind the hypothesis testing used in empirical testing of randomness. A reader familiar with the theory used in statistical randomness testing can skip this section. Section 3 describes the tests included in NIST STS, their parameters and recommended settings. The section also describes the testing strategy, *i.e.*, what tests to use, recommended lengths of sequences and how many sequences should be tested. This is followed by an illustration of the results for different testing procedures. An experienced user of the NIST STS can jump to Section 4. The main results of our work are discussed in Section 4 and Section 5. The interpretation of results provided by a single NIST STS test is discussed in Section 4, followed by the presentation of a newly proposed method that allows a more accurate interval computation for the passing sequences. Section 5 describes the interpretation of results provided by the whole test suite considering default parameters. Section 6 is dedicated to related work and it is discussing the dependency of the NIST STS tests and finally, Section 7 concludes the paper.

2. Empirical tests of randomness and hypothesis testing

The majority of empirical randomness tests, including the tests from the NIST STS, are based on the statistical hypothesis testing. Hence each test is formulated to evaluate the null hypothesis, namely that the sequence being tested is random, from the specific point of view of that test, which can be defined by a specific statistic of bits or blocks of bits. A test statistic is a function of the tested data and it compresses the measured randomness quality into a single value – the observed test statistic. In order to evaluate the test, a distribution of the test statistic must be known under the null hypothesis (when data is expected to be random). Most of the NIST STS tests have χ^2 or normal distribution as their reference distribution. An observed test statistic is usually transformed into a p-value using the reference distribution since a p-value can be interpreted more easily. The p-value represents the probability that a perfect random number generator would have produced a sequence less random than the tested sequence [1].

Remark 1. The most important property of the p-values is that for arbitrary statistical tests (and not only for randomness tests) which satisfy the null hypothesis, the p-values are uniformly distributed on the interval $[0, 1)$. [9] This means that random sequences processed by an arbitrary empirical test should be uniformly distributed on [0,1). Therefore the probability that the p-values computed for a random sequence lies within the interval [a,b) can be expressed as:

$$Pr(a \leq \text{p-value} < b) = b - a.$$

In order to evaluate a test, the resulting p-value is compared with the significance level α. If the p-value is smaller/bigger than α, the hypothesis is rejected/accepted. Since randomness is described in terms of probability we can commit two type of errors – Type I and Type II. A type I error occurs when the true hypothesis is rejected although the sequence was produced by a random number generator. The probability of a Type I error is equal to the significance level α and it is chosen by the tester. A type II error is more important for cryptographers since it represents the probability of accepting the false hypothesis (defective generator). The probability of a Type II error is denoted by β. Probability β is difficult to express but α and β are related to each other. If α is small then β is high and vice versa. In the hypothesis testing, the significance level α is set to small values (less than 0.05). In cryptography, α is commonly set to smaller values – typically 0.01. Setting $\alpha = 0.01$ means that we expect to reject the null hypothesis in less than 1% cases (for a perfect random number generator).

3. NIST Statistical Test Suite

The NIST STS battery consists of 15 empirical tests specially designed to analyze binary sequences (bitstreams). The tests examine randomness of data according to various statistics of bits or statistics of blocks of bits. All NIST STS tests examine randomness for the whole bitstream. Several tests are also able to detect local non-randomness and these tests divide the bitstream into several typically large parts and they compute a characteristic of bits for each part. All these partial characteristics are then used for the computation of the test statistic. Each NIST STS test is defined by the test statistic of one of the following three types and examines randomness of the sequence according to:

1. bits – these tests analyse various characteristics of bits like proportion of bits, frequency of bit change (runs) and cumulative sums,
2. m-bit blocks – these tests analyse distribution of m-bit blocks (m is typically smaller than 30 bits) within the sequence or its parts,
3. M-bit parts – these tests analyse complex property of M-bit (M is typically larger than 1000 bits) parts of the sequence like rank of the sequence viewed as a matrix, spectrum of the sequence or linear complexity of the bitstream.

All tests are parametrised by n which denotes the bitlength of a binary sequence to be tested. Several tests are also parametrised by the second parameter denoted by m or M. Since the reference distributions of NIST STS test statistics are approximated by asymptotic distributions (χ^2 or normal), the tests give accurate results (p-values) only for certain values of their parameters. Table 1 summarizes appropriate values of the parameters for each particular test recommended by NIST [1].

Table 1. The recommended size n of the bitstream for each particular test (Some tests are parameterised by a second parameter m, M, respectively. The table shows meaningful settings for the second parameter and the number of sub-tests executed by each particular test.)

Test #	Test name	n	m or M	# sub-tests
1.	Frequency	$n \geq 100$	-	1
2.	Frequency within a Block	$n \geq 100$	$20 \leq M \leq n/100$	1
3.	Runs	$n \geq 100$	-	1
4.	Longest run of ones	$n \geq 128$		1
5.	Rank	$n > 38\,912$	-	1
6.	Spectral	$n \geq 1000$	-	1
7.	Non-overlapping T. M.	$n \geq 8m - 8$	$2 \leq m \leq 21$	148^*
8.	Overlapping T.M.	$n \geq 10^6$		1
9.	Maurer's Universal	$n > 387\,840$		1
10.	Linear complexity	$n \geq 10^6$	$500 \leq M \leq 5000$	1
11.	Serial		$2 < m < \lfloor \log_2 n \rfloor - 2$	2
12.	Approximate Entropy		$m < \lfloor \log_2 n \rfloor - 5$	1
13.	Cumulative sums	$n \geq 100$		2
14.	Random Excursions	$n \geq 10^6$		8
15.	Random Excursions Variant	$n \geq 10^6$		18

Several of the NIST STS tests are performed in more variants, i.e., they execute several sub-tests and examine more properties of the sequence of the same type. For instance, the Cumulative sum test examines a sequence according to forward and backward cumulative sum. Table 1 also summarizes the number of sub-tests performed by each particular test. The Non-overlapping template matching test is marked by an asterisk since the number of its sub-tests is not fixed and depends on the value chosen for the parameter m (the number 148 mentioned in the Table 1 corresponds to the default value of the parameter $m = 9$).

3.1. Testing

NIST STS allows the analysis of an input file as one block (sequence) or to divide it into sequences of a fixed length n, where n is set using the command line. The user has to choose parameters that are listed here in their order of appearance in the text-based user interface:

1. file for the analysis – user can choose his own file or data can be generated by one of the predefined pseudorandom number generators (Blum-Blum-Shub, several congruential generators, modular exponentiation and others);
2. tests – what test/tests should be applied to data;
3. values for the second parameter (m or M) for several tests – Block frequency (128), Non-overlapping template matching (9), Overlapping template matching

(9), Approximate entrophy (10), Serial (16), Linear Complexity (500) (default values are listed in brackets behind each test);

4. number of bitstreams to be processed;
5. file format – ASCII (sequence of ASCII 0's and 1's) or binary format (each byte of the file contains 8 bits of the sequence).

Which tests should be chosen for randomness analysis is a difficult question. It depends on the considered generator (data), its application domain and the defects in randomness which are not acceptable. Without any information about the data to be analysed, all NIST STS tests should be used for the randomness analysis. In order to apply all tests, the parameter n (bitlength of the sequences) should be greater than 100000 (see Table 1). The NIST STS documentation recommends that at least $k = \alpha^{-1} = 100$ sequences should be tested. This is also an appropriate value for the uniformity test of p-values (at least 55 sequences must be processed). Since p-values are processed by the NIST STS using some approximation the more sequences are tested the more accurate results will be obtained. NIST suggests that a number of 1000 or more sequences should be tested [1].

3.2. Results

All tests when applied to one sequence result in one or more p-values (the exact number depends on the number of sub-tests, see Table 1). It should be noted that some tests – Runs, Random Excursions and Random Excursions Variant, are not always applicable. These tests are applied only if the sequence meets certain criteria (Frequency test is passed, number of cycles is greater than 500). If a test is not applicable the resulting p-value is set to 0.

All NIST STS tests produce several files with results. Each test produces its own `result.txt` file that stores all resulted p-values computed by the test (or sub-tests) for all tested sequences. When a test executes sub-tests it also produces files `datai.txt` that store p-values computed by the i^{th} sub-test. The file `result.txt` stores p-values in the natural order, *i.e.*, the first p-value from `data1.txt`, the first p-value from the `data2.txt`, etc.

The NIST STS processes all results (p-values) from all `result.txt` files into the final file `finalAnalysisReport.txt`. This file stores the "final" table that summarizes all results of all chosen tests. The following Table 2 illustrates the table from the `finalAnalysisReport.txt` file that was obtained after processing 1000 binary sequences each consisting of 10^6 bits.

Each row of Table 2 corresponds to one test (or a sub-test). Values in the columns $C1, C2, \cdots, C10$ represent number of p-values that fall within intervals $[0.0, 0.1)$, $[0.1, 0.2), \cdots, [0.9, 1.0)$, *i.e.*, 108 of the p-values computed by the Frequency test fall within interval $[0.1, 0.2)$. Values in the P-value column represent the results for uniformity testing of p-values computed for a given test (see Section 4.2). Value in the column Proportion represents proportion of sequences that pass a given test. In the first row (Frequency) we can see that the proportion of sequences that pass the Frequency test is 0.991, *i.e.*, 991 out of 1000 sequences passed. Results which NIST interprets as non-randomness of the data are marked by an asterisk. However these marked results just indicate potential problems with data. Statistical interpretation of all result is discussed in the next two sections.

Table 2. Partial results from the `finalAnalysisReport.txt` file produced by the NIST STS after processing of 1000 binary sequences produced by a biased random number generator

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-value	Proportion	Test
99	108	91	105	109	104	92	101	93	98	0.920383	0.9910	Frequency
90	89	103	101	111	105	100	94	108	99	0.853049	0.9970	BlockFrequency
90	114	93	114	96	90	102	96	101	104	0.643366	0.9910	CumulativeSums
103	91	101	99	113	97	87	88	114	107	0.506194	0.9930	CumulativeSums
41	44	44	45	50	568	51	48	51	58	0.000000*	0.9970	NonOverlapping
41	44	49	46	47	589	54	41	51	38	0.000000*	1.0000*	NonOverlapping
99	107	99	113	94	100	110	87	91	100	0.733899	0.9940	Serial
104	116	103	96	94	95	101	102	84	105	0.695200	0.9890	Serial
97	107	101	111	115	90	100	94	98	87	0.622546	0.9900	LinearComplexity

4. Interpretation of a single test result

There are several ways to interpret a set of p-values computed by an empirical test of randomness. NIST adopted the following two ways:

1. The examination of the proportion of sequences that pass a certain statistical test – relative number of sequences passing the test should lie within a certain interval.

2. The uniformity testing of p-values – p-values computed for random sequences should be uniformly distributed on the interval $[0, 1)$. Uniformity of p-values can be tested again using statistical tests (uniformity of p-values forms a hypothesis).

The NIST STS also includes analytical routines that analyse the uniformity and the proportion of the computed p-values for each particular test (sub-test).
The following types of results should be interpreted:

1. set of p-values,

2. proportions of sequences passing a given test (p-values greater than significance level $\alpha = 0.01$),

3. p-values resulted from the uniformity test of p-values computed for a test.

The NIST STS documentation describes a way to interpret the results of a single empirical test and includes the computed values into the final results (marked values indicate non-randomness). However, several improvements and corrections can be introduced. When a single sequence is tested, the computed p-value can be interpreted simply as: "the probability that a perfect random number generator would have produced a sequence less random than the sequence that was tested" [1]. Values in the columns P-value and Proportion are meaningful only for an appropriate (NIST recommends at least 100) number of tested sequences.

4.1. Proportion of sequences passing a test

The probability that a random sequence passes a given test is equal to the complement of the significance level $1-\alpha$. For multiple random sequences, the proportion of sequences that pass a given test is usually different but close to $(1-\alpha)$. The proportion of passing sequences should fall into a certain interval around $(1-\alpha)$ with a high probability. The NIST STS computes the interval of acceptable proportions and saves it into the `finalAnalysisReport.txt` file. The interval is computed using the significance level α and the number of tested sequences k as:

$$(1-\alpha) \pm 3\sqrt{\frac{\alpha(1-\alpha)}{k}}, \tag{1}$$

where k is the number of tested sequences. The acceptable proportion of passing sequences should fall within the interval 0.99 ± 0.0094392 for the significance level $\alpha = 0.01$ and the number of tested sequences $k = 1000$. In Table 2 the value in the Proportion column for the Frequency test (0.991) lies within the interval of acceptable proportions. This means that 991 out of 1000 tested sequences passed the Frequency test (991 of p-values are greater than the significance level $\alpha = 0.01$) and therefore the data can be considered random according to the Frequency test. On the other hand, the proportion (1.0) for two considered Non-overlapping sub-tests is outside the acceptable region [0.9805607,0.9994392], therefore the data can be considered non-random – and values are marked by an asterisk.

Improvements to the computation of the acceptable region of passing sequences: Formula (1) is based on the approximation of the binomial distribution [1] which is reasonably accurate for many tested sequences ($k \geq 1000$). The probability that the proportion of passing random sequences falls into the computed interval is 99.73%. It corresponds to the probability 0.27% of the Type I error (see [11]). Hence we will get the probability of the Type I error closer to 1% if the interval of acceptable proportion is computed by the following formula:

$$0.99 \pm 2.6\sqrt{\frac{0.01 * 0.99}{k}}. \tag{2}$$

The above formula gives accurate results for large k [11]. For a small number of tested sequences (k) the tester should use the formula (3) based on the binomial distribution which is exact, not an approximation as formulas (1 or 2). The probability that a random sequence passes given test is equal to $1-\alpha = 0.99$. The probability that k_1 out of k random sequences pass a test has a binomial distribution and it can be computed[1] as:

$$\binom{k}{k_1}(1-\alpha)^{k_1}\alpha^{k-k_1} = \binom{k}{k_1}0.99^{k_1}0.01^{k-k_1}. \tag{3}$$

[1] When using the online Wolfram Alfa Engine [10] type *sum Binomial[k,i]0.99 ^i*0.01^(k-i), k= k_1 to k_2.*

The distribution of probabilities (for random data) is "symmetric" about the mean $1-\alpha$ and therefore the interval has $1-\alpha$ as its center. The probability that the number of passing sequences fall within the interval $[k_1, k_2]$ can be computed as

$$Pr(k_1/k \leq proportion \leq k_2/k) = \sum_{i=k_1}^{k_2} \binom{k}{i} 0.99^i 0.01^{k-i}.$$

We get $Pr(\frac{981}{1000} \leq proportion \leq \frac{999}{1000}) = 0.9966$ (corresponds to 0.34% Type I error) for the interval computed by formula (1) used in the NIST STS. For the interval [0.9805,0.9994] computed by the newly proposed formula (2) we get the probability $Pr(\frac{982}{1000} \leq proportion \leq \frac{998}{1000}) = 0.9926$ (Type I error equal to 0.74%). In order to obtain a probability of the Type I error closer to 1% it usually suffices to change one of the bounds k_1, k_2 by 1. For $k = 1000$ the most appropriate bounds are $k_1 = 982, k_2 = 997$ for which we get $Pr(\frac{982}{1000} \leq proportion \leq \frac{997}{1000}) = 0.9904$ with the corresponding 0.96% probability of the Type I error which is quite close to 1% given by the significance level α.

4.2. Uniformity of p-values

The p-values computed by a singe test should be uniformly distributed on the interval [0,1). Hence, the uniformity of p-values forms a hypothesis and it can be tested by a statistical test. The NIST uses one sample χ^2 test to assess the uniformity of p-values. χ^2 test measures whether the observed discrete distribution (histogram) of some feature follows the expected distribution. In the NIST STS, the interval [0,1) is divided into 10 sub-intervals $[i/10, (i+1)/10)$ and χ^2 test checks whether the number of p-values for each sub-interval (C_i columns of Table 2) is close to $k/10$ (where k denotes number of p-values/tested sequences). For the first considered Non-overlapping sub-test, a number of 568 (C6 column of Table 2) p-values fall into the interval [0.5,0.6). The observed number 568 is quite different from the expected 100 and therefore the uniformity test fails.

Remark 2. The χ^2 test works well only for $k/10$ greater than 5.5. Therefore the number of tested sequences should be at least 55 ($k \geq 55$) to get a meaningful result for the uniformity test.

The value in the P-value column of the final table (Table 2) represents the results (p-value) of the uniformity test of p-values. A computed small p-value indicates a problem of the generator, but it is hard to identify a concrete weakness. The NIST STS documentation recommends a very small value for the significance level $\alpha = 0.0001$ for the uniformity test, *i.e.*, p-values are considered as non-uniform if a p-value from the P-value column is smaller than 0.0001. The p-values computed by the first considered Non-overlapping template matching test are non-uniform on [0,1) since the resulted uniformity p-value 0 is smaller than $\alpha = 0.0001$. A very small value of the significance level α recommended by NIST implies a large probability of the Type II error (β - acceptance of bad generator). From the practical point of view a small β is more important than a small α. On the other hand the non-uniformity of p-values usually indicates that there can be a problem. We believe a less conservative (0.001 or 0.01) value of the significance level α would be more appropriate for testing the uniformity of p-values.

5. Interpretation of multiple tests

NIST suggests to consider data to be random if and only if the sequence/sequences pass all testing procedures (uniformity test of p-values, test of the proportion of passing sequences). This is slightly misleading since the probability that a sequence fails at least one of the used procedures increases with the increasing number of used procedures/tests (sub-tests). At the same time, the NIST STS recommends the analysis of different samples produced by the same generator.

Remark 3. If the tested sequence(s) fail one or more randomness testing procedures "additional numerical experiments should be conducted on different samples of the generator to determine whether the phenomenon was a statistical anomaly or a clear evidence of non-randomness"[1].

Overall a number of 188 tests (sub-tests) are applied to a given sequence in the default settings of the NIST STS. The probability that a given sequence passes exactly k_1 out of k tests (NIST STS test, uniformity test, test of proportion) all with $\alpha = 1\%$ has again the binomial distribution and can be computed using the above mentioned formula (3). For instance, the probability that a given sequence passes 188 independent tests is equal to $0.99^{188} = 0.15 = 15\%$. However, due to interdependency of the NIST STS tests the probability that given sequence passes all NIST STS tests is higher than the expected 15%.

In order to evaluate the randomness of data we have measured the probability that a random sequence fails i or more tests for each testing procedure (proportion of passing sequences, uniformity test). These computed probabilities can be used to compute a p-value for the whole test suite – probability that a perfect random number generator would produce sequences with results (set of p-values) worse than the results computed for the given sequence. We analysed the results of all 188 NIST STS tests/sub-tests with their default settings. We have analysed 100 GB of data (downloaded from [8]) produced by a physical source of randomness. The randomness analysis was performed by an optimised implementation of the NIST STS [5] which is overall 30x faster than the original implementation. In order to use all the tests the bitlength of each sequence n was set to 100000 bits which corresponds to 819200 analysed sequences.

5.1. Considering only one sequence

In this section we focus on the probability $Pr(i)$ that a random sequence fails (the p-value of the NIST STS tests is smaller than $\alpha = 1\%$) i tests for a given procedure, and illustrate how to use these probabilities to evaluate the randomness of the sequence. The Random Excursions test and the Random Excursions Variant test and their sub-tests are not always applicable. Random Excursions test and Random

Excursions Variant are either both (including all the sub-tests) applicable or non-applicable. Therefore 188 tests or 162(= 188 − 8 − 18) tests are applicable. We computed two probabilities:

1. The probability $Pr(i, 188)$ that a random sequence fails exactly i NIST STS tests. Probabilities are computed for 505557 (out of all 819200) sequences for which all NIST STS are applicable.
2. The probability $Pr(i, 162)$ that a random sequence fails exactly i NIST STS tests (except the Random Excursions test and the Random Excursions Variant test). Probabilities are computed for the remaining 313643 (= 819200 − 505557) sequences for which the Random Excursions (Variant) tests are not applicable.

Table 3 shows the probabilities (expected theoretical and observed) that a random sequence fails exactly i NIST STS tests. Expected probabilities are computed for $k = 188$ independent tests using the following formula: $\binom{k}{i}0.99^{(k-i)}0.01^{i}$. For practical reasons the table also shows the cumulative probability that a random sequence fails i or more NIST STS tests.

Table 3. Percentage probability that a random sequence fails exactly i out of all 188 NIST STS tests (default setting) or out of 162 NIST STS tests (when the Random Excursions (Variant) tests are not applicable) ($Pr(i, 188)$ is computed for sequences for which all NIST STS tests are applicable. $Pr(i, 162)$ is computed for sequences for which Random Excursions (Variant) tests are not applicable. Cumulative probability represents the corresponding probability that a random sequence fails i or more empirical tests.)

number of failed tests (i)	0	1	2	3	4	5	6	7	8	9	10	11
expected	15.1	28.7	27.1	17.0	7.9	2.9	0.9	0.2	0.1	0.0	0.0	0.0
$Pr(i, 188)$	19.4	28.9	23.8	14.3	7.2	3.3	1.5	0.7	0.3	0.2	0.1	0.1
cumulative	100	80.6	51.7	28.0	13.6	6.4	3.0	1.6	0.8	0.5	0.3	0.1
$Pr(i, 162)$	20.0	29.5	23.8	14.1	6.9	3.1	1.3	0.6	0.3	0.1	0.1	0
cumulative	100	80.0	50.4	26.6	12.5	5.6	2.5	1.2	0.6	0.3	0.2	0.1

The computed cumulative probabilities from Table 3 can be simply used for assessing randomness of the given sequence S. If the sequence S is tested using the NIST STS with the default settings, S fails 5 tests and the Random Excursions (Variant) tests are applicable (*i.e*, all NIST STS tests are applicable) then 6.4% of random sequences are “equally or less random” than the sequence. Since the corresponding value 6.4% is greater than the significance level $\alpha = 1\%$ the sequence S can be considered random. From the Table 3 a sequence can be considered non-random for $\alpha = 1\%$ if it fails 8 or more NIST STS tests.

5.2. Considering multiple sequences

In this section we discuss the most common situation in randomness testing where many sequences are tested by all NIST STS tests. An interpretation of all computed p-values, 188 p-values of uniformity tests and 188 proportions of passing sequences is discussed.

Set of p-values: We can use probabilities computed for each particular sequence (Table 3 from the previous section 5.1) to interpret a set of p-values computed by multiple tests for multiple sequences. In order to describe the randomness for k tested sequences $S_1, \cdots, S_k$ it is sufficient to compute the product of probabilities p_i from Table 3 corresponding to each particular sequence S_i. Let us consider the randomness assessment of data consisting of two sequences S_1, S_2. If the sequence S_1 fails 2 tests and the Random Excursions (Variant) tests are not applicable, then for the corresponding probability we have $p_1 = 0.504$. If the sequence S_2 fails 1 test and all tests are applicable then the corresponding probability is equal to $p_2 = 0.806$. The probability (p-value for the whole test suite) that two random sequences are less random than S_1, S_2 can be computed as $p = p_1.p_2 = 0.504 * 0.806 = 0.4$. Therefore data (sequences S_1, S_2) can be considered random for $\alpha = 0.01$ since $p > \alpha$.

Uniformity of p-values: In order to evaluate randomness of many sequences for all the NIST STS tests we computed probabilities that random sequences fail i (or more) uniformity tests for p-values computed by all 188 NIST STS tests. The probabilities are computed from 8192= 819200/100 sets of sequences each consisting of $k = 100$ sequences. It should be noted that the non-applicable Random Excursions (Variant) tests are not an issue since uniformity of p-values computed by the Random Excursions (Variant) tests is examined for a smaller set (60 in average) of p-values. Table 4 shows the observed probability that a set of 100 random sequences fail exactly i uniformity tests. Table 4 also shows the cumulative probability that 100 sequences fail i or more uniformity tests for different significance levels $\alpha = 1\%, 0.1\%$ and 0.01% recommended by NIST.

Table 4. Percentage probability that 100 random sequences fail exactly i out of 188 uniformity tests used for each particular NIST STS test ($\alpha = 1\%$) (Cumulative probabilities represent probability that 100 random sequences fail i or more uniformity tests for different significance levels $\alpha = 1\%, 0.1\%, 0.01\%$.)

number of failed tests (i)	0	1	2	3	4	5	6	7	8	9
expected	15.1	28.7	27.1	17.0	7.9	2.9	0.9	0.2	0.1	0
observed	11.3	24.8	26.6	19.0	10.8	4.8	1.8	0.5	0.2	0.1
cumulative ($\alpha = 1\%$)	100	88.7	63.9	37.3	18.3	7.5	2.7	0.85	0.33	0.09
cumulative ($\alpha = 0.1\%$)	100	21.8	3.1	0	0	0	0	0	0	0
cumulative ($\alpha = 0.01\%$)	100	3.3	0.12	0	0	0	0	0	0	0

The computed probabilities can be used for assessing randomness of 100 sequences as follows: Let us assume that 100 sequences are analysed by the NIST STS in the default settings and sequences fail 5 uniformity tests (5 p-values in the P-value column of the final table are marked by asterisk). For the $\alpha = 0.01\%$ recommended by NIST, the corresponding probability value from Table 4 is 0%. Since the probability is smaller than $\alpha = 0.01\%$ the data can be considered non-random according to the uniformity test procedure. On the other hand, the data can be considered random for the significance level $\alpha = 1\%$ since $7.5\% > \alpha$.

Proportion of sequences passing a test: We have also computed the probability that a set of k sequences fails the test of proportion of passing sequences. We have considered a set of 100 or 1000 sequences ($k = 100$ or $k = 1000$). In order to get more accurate results we have also tested $k = 1000$ with the interval of acceptable proportions computed using the new formula (2) (constant 3 is replaced by a more accurate value 2.6). Table 5 shows the expected and observed probability that 100 respectively 1000 random sequences fail i (or more) tests of proportion of passing sequences. The results were analysed for all 188 NIST STS tests. The non-applicable Random Excursions (Variant) tests are not an issue, since a proportion of passing sequences is computed from a smaller set (60 or 600 in average) of p-values computed for sequences for which the Random Excursions (Variant) tests are applicable.

Table 5. Percentage probability that random sequence fails exactly i out of 188 uniformity tests used for each particular NIST STS test ($\alpha = 1\%$) (Cumulative probability represents percentage probability that a random sequence fails i or more NIST STS tests for different number of tested sequences $k = 100, 1000$ and interval of acceptable proportions given by constants $const = 3$ or $const = 2.6$.)

number of failed tests (i)	0	1	2	3	4	5	6	7	8	9	10	11	12	13
expected	15.1	28.7	27.1	17.0	7.9	2.9	0.9	0.2	0.1	0	0	0	0	0
observed	46.9	33.7	14.7	4.3	0.5	0	0	0	0	0	0	0	0	0
cumulative $k = 10^3$, $const = 3$	100	53.1	19.4	4.8	0.5	0	0	0	0	0	0	0	0	0
cumulative $k = 10^2$, $const = 3$	100	95.9	84.3	66.3	45.6	28.1	15.6	8.2	4.1	1.8	0.8	0.3	0.2	0.1
cumulative $k = 10^3$, $const = 2.6$	100	79.2	48.6	25.2	9.4	3.2	1.2	0.6	0.1	0	0	0	0	0

Computed cumulative probabilities can be used for assessing the randomness of 1000 sequences as follows: Let us assume that k=1000 sequences fail 4 tests of proportions (4 marked values in the Proportion column of a final table). The corresponding cumulative probability from Table 5 is equal to 0.5%. The probability is smaller than the significance level $\alpha = 1\%$ and therefore sequences can be considered non-random. On the other hand, computing the interval more accurately, using the 2.6 value for the constant, sequences can be considered random since the probability that 1000 random sequences fail 4 or more proportion tests is $9.4\% > 1\%$.

6. Related work

In order to eliminate redundant tests NIST analysed the dependency of p-values using principal component analysis. The results were "that there is no large redundancy among tests". However, analyzing p-values, which are not linear, using principal component analysis assuming linearity, seems awkward, as authors in the paper [6] have also observed.

According to our best knowledge, there is only one paper [7] focused on dependency of NIST STS tests. The authors of [7] measured only the interdependency of the NIST STS tests (probability that a random sequence fails two tests simultaneously). Moreover, quality of random data is essential for the measurement of the interdependency of tests, but authors did not specify how data were generated. They also skipped several NIST STS tests. In our approach we are focusing on different aspect of the dependency. In order to evaluate randomness of a given sequence we computed the probability that a random sequence fails more than a given number of tests (empirical test, uniformity tests, test of proportion).

7. Conclusion

In this paper we focused on the interpretation of results of the NIST STS in its default settings. The NIST STS suggests to consider data to be random if all tests are passed – yet even truly random data shows a high probability (80%) of failing at least one NIST STS test. If data fail some tests the NIST STS recommends the analysis of additional samples. We analysed 819200 sequences of 1000000 bits produced by a physical source of randomness (quantum random number generator) in order to interpret results computed without analysing any additional samples.

We have computed the reference probabilities that random sequences fail i or more tests for each particular testing procedure (NIST STS tests, uniformity test, proportion of passing sequences). Computed probabilities reflect the dependency between p-values computed by the NIST STS tests, p-values of uniformity tests and proportions of passing sequences. We also improved the formula computing the interval of acceptable proportions of passing sequences.

Computed reference cumulative probabilities indicate that a single sequence can be considered non-random if it fails (p-values are smaller than $\alpha = 1\%$) 7 or more NIST STS tests. According to the uniformity test, 100 sequences can be considered non-random if they fail 7 uniformity tests ($\alpha = 1\%$) or 3 uniformity tests (for $\alpha = 0.1\%$ or $\alpha = 0.01\%$). According to the test of proportions, 1000 (100) sequences can be considered non-random if they fail 4 (10) tests of proportion. We have also redefined a more accurate interval of acceptable proportions computed with a more accurate constant (2.6 instead of 3). Using this interval, 1000 sequences can be considered non-random if they fail 7 or more test of proportions. For this interval, a tester has to check by hand whether the proportion of passing sequences is out of the new interval of acceptable proportions since results in the final table are marked only for the old one. All previous results indicate that even random sequences often fail one or more tests. However, it is still necessary to examine additional samples in order to evaluate whether failed tests show a statistical anomaly or a clear evidence of non-randomness.

Acknowledgement. The first author was supported by the Ministry of Education, Youth, and Sport project CZ.1.07/2.3.00/30.0037 – Employment of Best Young Scientists for International Cooperation Empowerment. Second and third author were supported by the Czech Science Foundation project GBP202/12/G061. Fourth author was supported by the Post-Doctoral Programme POSDRU/159/1.5/S/137516, project co-funded from European Social Fund through the Human Resources Sectorial Operational Program 2007-2013.

References

[1] RUKHIN A., SOTO J., NECHVATAL J., SMID M., BARKER E., LEIGH S., LEVENSON M., VANGEL M., BANKS D., HECKERT A., DRAY J., VO S., *A Statistical Test Suite for the Validation of Random Number Generators and Pseudo Random Number Generators for Cryptographic Applications, Version STS-2.1*, NIST Special Publication 800-22rev1a, April, 2010. `http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf`.

[2] MARSAGLIA G., *The Marsaglia random number CDROM including the DIEHARD battery of tests of randomness.* See `http://stat.fsu.edu/pub/diehard`, 1996.

[3] BROWN R. G., *Dieharder: A Random Number Test Suite*, Version 3.31.1, 2004.

[4] L'ECUYER P., SIMARD R., *TestU01: A C library for empirical testing of random number generators*, ACM Trans. Math. Softw., vol. **33**, 2007.

[5] SÝS M., ŘÍHA Z., *Faster Randomness Testing with the NIST Statistical Test Suite*, Security, Privacy, and Applied Cryptography Engineering, LNCS 8804, pp. 272–284, 2014.

[6] KENNY CH., MOSURSKI K., *Random Number Generators: An Evaluation and Comparison of Random.org and Some Commonly Used Generators.* 2005. `https://www.random.org/analysis/Analysis2005.pdf`.

[7] DOGANAKSOY A., EGE B., MUS K., *Extended Results for Independence and Sensitivity of NIST Randomness Tests*, (3rd) ISCTurkey 2008.

[8] Nano-Optics groups at the Department of Physics of Humboldt University and PicoQuant GmbH: QRNG Service. `https://qrng.physik.hu-berlin.de`

[9] MURDOCH D., TSAI Y., ADCOCK J., *P-Values are Random Variables*, The American Statistician, **62**, 2008, pp. 242–245.

[10] WolframAlpha: computational knowledge engine. `http://www.wolframalpha.com`

[11] NIST: Cumulative Distribution Function of the Standard Normal Distribution. `http://www.itl.nist.gov/div898/handbook/eda/section3/eda3671.htm`