Infinity Blockchain Labs
GINAR Team

PROJECT REPORT

Statistical Tests on The GINAR Random Number Generator

October 16, 2018

Thanh N.V
thanhnv@blockchainlabs.asia

Abstract

The aim of this work is to conduct statistical tests that detect non-randomness in our Random Nunber Generator, i.e. **GINAR RNG Service** and then verify that the outputs of our RNG are random. An industry-standard suite of tests (i.e. the NIST SP800-22 Statistical Test Suite) was chosen to evaluate the randomness, from a statistical viewpoint, of the numbers generated by GINAR Service. All the figures in this report are the results of our work directly conducted on the raw numbers generated by our RNG and are archived in GINAR's facilities.

Chapter 1

Introduction

It is essential to test the output bits of an RNG to check whether its statistical properties fit those of a truly random bit sequence. There are two types of test that were developed to examine the randomness of an RNG - theoretical and statistical (empirical). Theoretical tests focus on the inner structure of the RNG. This type is considered to be the most powerful test, if an RNG passes a theoretical test, it will likely pass all other statistical tests. Each test of this kind is only applied to some specific RNGs and needs great effort to construct. It requires a good knowledge of the RNG, how the RNG works and the properties of each operation in the RNG structure. On the other hand, statistical tests are conducted on bit strings generated by the RNG, examine the distribution of generated numbers to tell how good the RNG is. They require no knowledge of how the RNG works, therefore, they are widely applied to all RNGs. Until now, statistical hypothesis testing has been the most widely used method to evaluate the randomness of the numbers produced by a generator. The theory of statistics provides some quantitative measures to assess randomness in an effective way. Within the scope of this whitepaper, we only consider statistical tests.

A single statistical test is not enough to tell whether a sequence is random or not since the sequence could be in various forms of non-randomness. The main difficulty in statistical testing is that there has not been a complete method to tell whether a finite set of number is random or not. Various statistical tests can be applied to a sequence to evaluate its randomness and indeed, there are numerous statistical tests that have been developed, each considering if the sequence contains or does not contain a certain **"pattern"**, once detected, would indicate that the sequence is not random. The more tests passed, the more likely the sequence is random. Randomness is a probabilistic property, which means that how random a sequence is can be characterized in term of probability. Therefore, it is usual if sometimes, an RNG produces bit strings that do not seem to be random at all even a TNRG and some tests will fail. However, as long as the number of failed tests is small enough (i.e. within statistical limits), this does not raise any doubt about the RNGs performance.

After considering many test suites, the NIST SP 800-22 statistical test suite was chosen, mainly because it is widely recognized as the industry standard. This test suite was developed and is currently maintained by the U.S government institution NIST. It is one of the most well-known used test suites to assess the outputs of random number generators.

Chapter 2

The NIST Stastical Test Suite

The NIST Statistical Test Suite consists of 15 different statistical tests, each focuses on a particular type of non-randomness detection of arbitrary long binary sequences produced by either hardware or software-based cryptographic random or pseudorandom number generators. The test suite makes use of both existing algorithms culled from the literature and newly developed tests.

As shown in table 2.1, the purpose of the ***Frequency Test*** is to determine whether the number of zeros and ones in a given sequence are approximately the same while the ***Frequency Test within a Block*** focuses on deciding whether the proportion of ones with M-bit blocks is approximately M/2. The next two tests focus on the number of runs with a sequence where a run is an uninterrupted sequence of identical bits. Specifically, the ***Runs Test*** determines whether the number of runs of ones and runs of zeros in a whole sequence are approximately the same while the ***Longest Run of Ones in a Block Test*** tests whether the length of the longest run of ones within a block is consistent with the one expected in a truly random sequence. The linear dependence among subsequences and periodic features of a given sequence are then tested using the ***Binary Matrix Rank Test*** and the ***Discrete Fourier Transform Test***, respectively. Template matching is tested with 2 tests in this suite, i.e. the ***Non-overlapping Template Matching Test*** and ***Overlapping Template Matching Test***. A significantly compressible sequence is considered to be non-random and this property is tested with the ***Maurers Universal Statistical Test***. To tell whether a given sequence is complex enough to be considered random, we use the ***Linear Complexity Test***. The ***Serial Test*** and the ***Approximate Entropy Test*** focus on the frequency of all possible overlapping m-bit patterns across a given sequence while the purpose of the ***Cumulative Sums Test*** uses the cumulative sum of digits (0 and 1 are mapped to -1 and 1 respectively) in a sequence to determine whether the cumulative sum of a partial sequence of the input is too large or too small relative to the expected behavior of a truly random sequence. The last two tests, i.e. the ***Random Excursions Test*** and the ***Random Excursions Variant Test***, aim to detect deviations from the expected number of visits to various states in the random walk.

All of these tests are formulated to test a specific null hypothesis (H0) [1]. All the tests we conducted used a significance level of $\alpha = 0, 01$. If the *p-value* computed is greater than the significance level then the tests accept the input bit stream as random, otherwise, the input bit stream is considered not random.

Table 2.1: The NIST SP 800-22 Statistical Test Suite

Test	**Detection**	**Properties**
Frequency	Too many zeros or ones	Equally likely (global)
Frequency within a Block	Too many zeros or ones	Equally likely (local)
Runs	Oscillation of zeroes and ones too fast or too slow	Sequential dependence (global)
Longest Run of Ones in a Block	Oscillation of zeroes and ones too fast or too slow	Sequential dependence (local)
Binary Matrix Rank	Deviation from expected rank distribution	Linear dependence
Discrete Fourier Transform	Repetitive patterns	Periodic dependence
Non-overlapping Template Matching	Irregular occurrences of a chosen template	Periodic dependence Equally likely
Overlapping Template Matching	Irregular occurrences of a chosen template	Periodic dependence Equally likely
Maurer's Universal Statistical	Compressible sequence	Dependence, Equally likely
Linear Complexity	Linear feedback shift register too short	Dependence
Serial	Non-uniformity in the joint distribution for m-bit sequences	Equally likely
Approximate Entropy	Non-uniformity in the joint distribution for m-bit sequences	Equally likely
Cumulative Sums	Too many zeros or ones at either an early or late stages	Sequential Dependence
Random Excursion	Deviation from the distribution of the number of visits of a random walk to a certain state	Sequential Dependence
Random Excursion Variants	Deviation from the distribution of the number of visits of a random walk to a certain state	Sequential Dependence

Chapter 3

Test Results and Evaluation

3.1 Test input

We have tested $m = 300$ different numbers generated by our system using the NIST STS, each number is 2,000,128 bits long.

3.2 Test Results

3.2.1 The interpretation of empirical results

The interpretation of empirical results can be conducted in many ways. As adopted by NIST [1], we focused on the proportion of numbers that pass each test. Besides, we also analyzed the distribution of *p-values* of each test.

Some of the tests, i.e. Random Excursions and Random Excursions Variant, are applied only if the input sequence satisfies certain criteria. Therefore, in this experiment, we only conducted the first 13 tests on our random outputs.

3.2.2 Proportion of Numbers Passing a Test

Table 3.1 gives an overview of the result of all the tests. The *p-value* of each test in this table is the average value of all values obtained when running the test with different inputs. The *Pass Rate* of a test is the proportion of numbers that the RNG passes a test. Follow instruction from [1], the range of acceptable proportions is determined using the confidence interval defined as: $interval = \hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$ where $\hat{p} = 1 - \alpha$. Specifically, we choose $\alpha = 0.01$ then $\hat{p} = 1 - 0.01 = 0.99$. With $m = 300$ we have: $interval = 0.99 \pm 3\sqrt{\frac{0.99(1-0.99)}{300}} = 0.99 \pm 0.01723 = [0.97277, 1]$. If the proportion falls outside of this interval (i.e *interval*), then the data is likely non-random. As we can see from table 3.1, all of the *Pass Rate* of each test is within *interval* which means that our RNG has passed the NIST STS. Additionally, the distribution of *p-value* corresponding to each test is illustrated in figures 3.1-3.7. The interval between 0 and 1 is divided into 10 sub-intervals and the *p-values* that fall inside each interval are counted and displayed.

Table 3.1: Overview of the result of the NIST test suite

Test Name	p-value	Pass Rate
Frequency	0.488394371094	99.333333
Frequency with a Block	0.49842979240357305	99.666667
Runs Test	0.515203452388	99.666667
Longest Run of Ones in a Block	0.518494675342847	99.0
Binary Matrix Rank	0.532005888747	99.333333
Discrete Fourier Transform	0.467606302775	98.666667
Non-overlapping Template Matching	0.4940106489959625	99.333333
Overlapping Template Matching	0.5083653373857795	98.333333
Maurer's Universal Statistical	0.99931026847	100.0
Linear Complexity	0.5137538547552889	98.0
Serial	0.508506895646323	99.0
Approximate Entropy	0.499358590316	98.666667
Cumulative Sums	0.501233465124	99.0

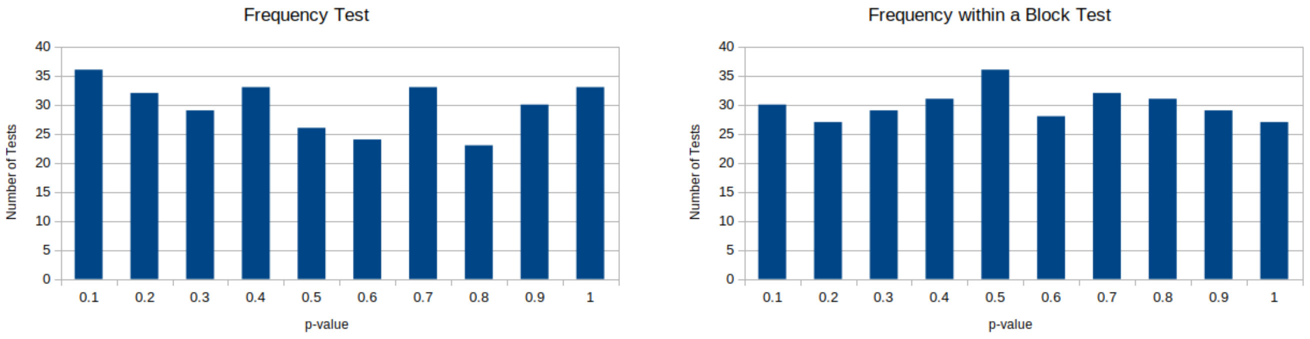

Figure 3.1: P-values distribution

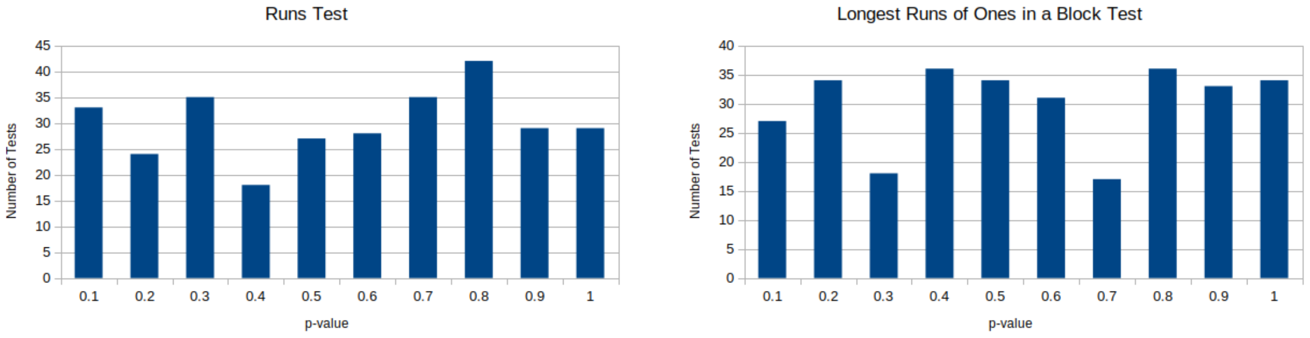

Figure 3.2: P-values distribution

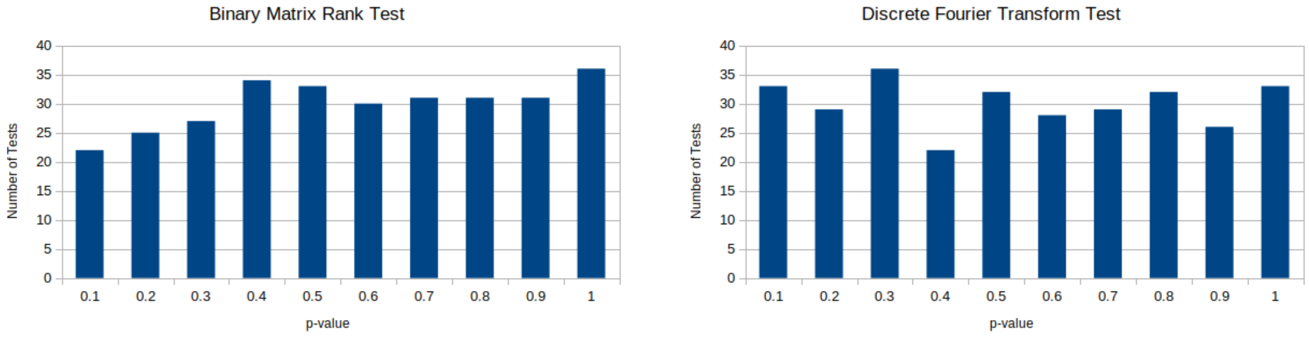

Figure 3.3: P-values distribution

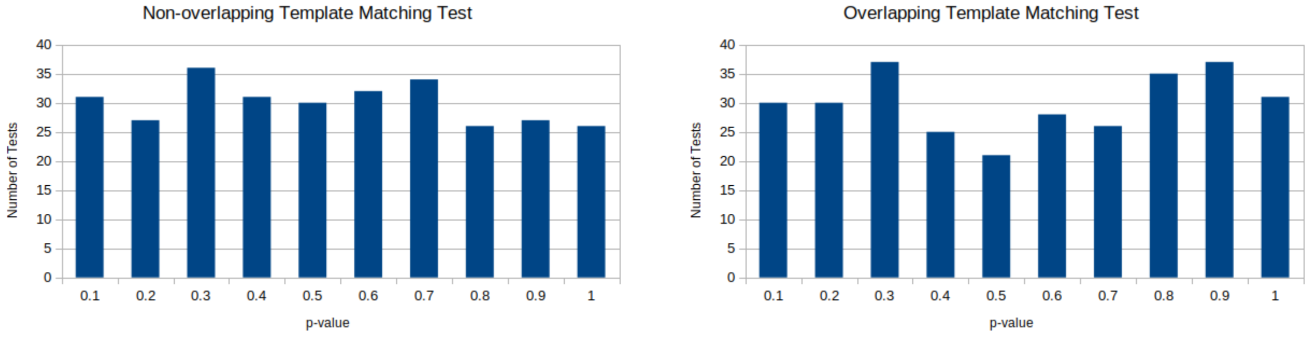

Figure 3.4: P-values distribution

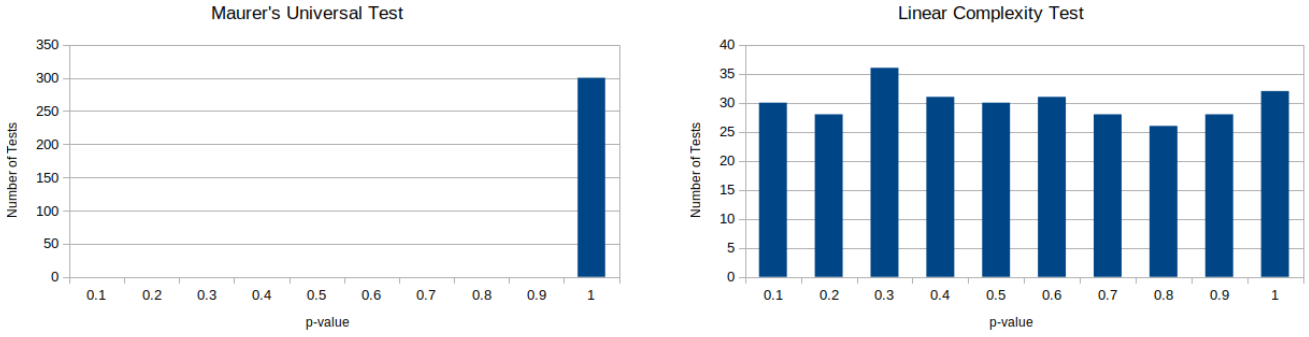

Figure 3.5: P-values distribution

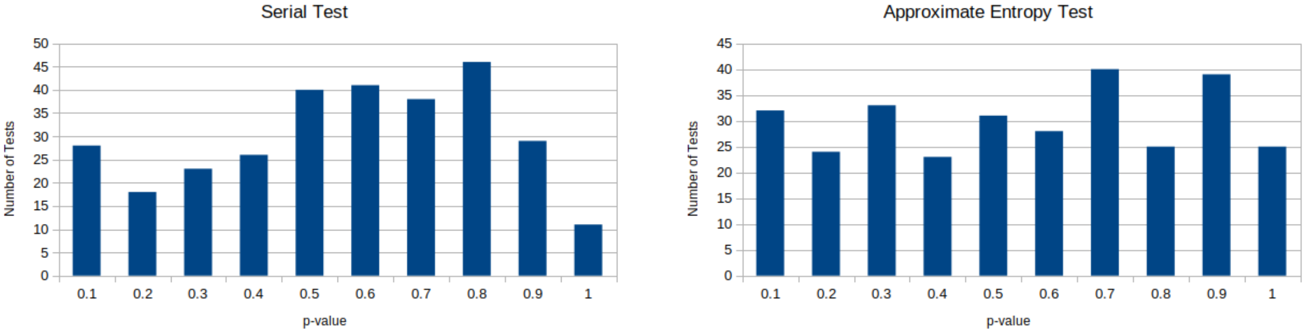

Figure 3.6: P-values distribution

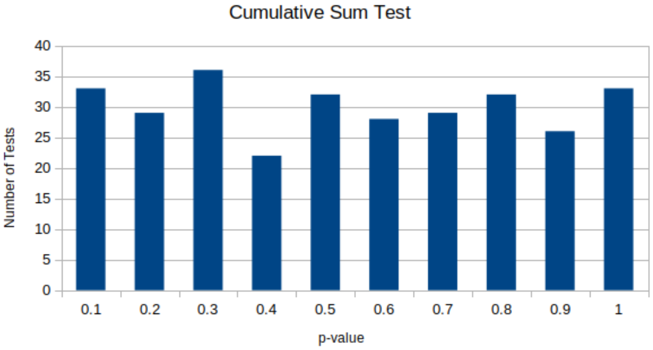

Figure 3.7: P-values distribution

Bibliography

[1] NIST SP 800-22r1a. *A Statistical Test Suite for Random and Pseudo-random Number Generators for Cryptographic Applications* . April, 2010.