Statistical Measurement of Unpredictability in Pseudo and True Random Number Generators

Sepehr Akbari Andrew Gard

Steven Galovich Memorial Student Symposium, *4/7/2026*

Department of Math & CS, Lake Forest College

Identifying Randomness

Consider two sets of 5 numbers between 0 and 10.

$$\{2, 3, 4, 5, 6\} \quad \text{and} \quad \{7, 1, 0, 4, 1\}$$

Which set seems randomly generated? Why?

Observe the probability of generating each set from a random process that selects numbers uniformly at random from the set $\{0, 1, 2, \ldots, 10\}$.

$$\Big\{\mathbb{P}(2),\ \mathbb{P}(3),\ \mathbb{P}(4),\ \mathbb{P}(5),\ \mathbb{P}(6)\Big\} \quad \text{and} \quad \Big\{\mathbb{P}(7),\ \mathbb{P}(1),\ \mathbb{P}(0),\ \mathbb{P}(4),\ \mathbb{P}(1)\Big\}$$

$$\left\{\frac{1}{11}, \frac{1}{11}, \frac{1}{11}, \frac{1}{11}, \frac{1}{11}\right\} \quad \text{and} \quad \left\{\frac{1}{11}, \frac{1}{11}, \frac{1}{11}, \frac{1}{11}, \frac{1}{11}\right\}$$

Both sets are equally likely to be generated by a random process.

Predicting Randomness

Randomness is not a property of a result, but a property of the process.

To move beyond human intuition, we define two types of uncertainty:

- **Epistemic Uncertainty.** We cannot predict the outcome because we lack information about the deterministic system.
- **Ontological Uncertainty.** The outcome is fundamentally not determined by the prior state of the universe.

Goal. Quantify the gap using statistical measurement.

Pseudo-Random Number Generators (PRNGs)

A PRNG is defined by a deterministic algorithm that produces a sequence of values $\{x_n\}$ from an initial "seed" s_0, in the form

$$x_n = g\left(f^{(n)}(s_0)\right) \quad \text{for } n = 0, 1, 2, \ldots$$

where

- S is a finite set of states.
- $s_0 \in S$ is the initial state (Seed).
- $f : S \to S$ maps the current state to the next state.
- $g : S \to \mathcal{U}$ maps the state to a value in the output space.

The sequence $\{x_n\}$ is entirely defined by s_0. If s_0 is known, x_{n+k} is predictable for all $k \geq 0$.

Periodicity of PRNGs

If a generator produces more than $|S|$ values, by the pigeonhole principle, there must exist i, j such that $s_i = s_j$. Since $s_{i+1} = f(s_i)$, the sequence will repeat every $T = j - i$ steps:

$$x_{n+T} = x_n, \quad \forall n \geq i, \text{ where } T \leq |S|$$

Therefore, because S is a finite set, a PRNG must eventually repeat (it's periodic).

PRNGs in the industry use massive periods (e.g. $2^{19937} - 1$), but they remain mathematically bounded.

True Random Number Generators (TRNGs)

A TRNG is ideally a high-entropy source where

$$\mathbb{P}(X_n|X_{n-1},\ldots,X_0) = \mathbb{P}(X_n), \quad \forall n$$

We measure this "information density" using *Shannon Entropy* $H(X)$:

$$H(X) = -\sum_i P(x_i)\log_2 P(x_i)$$

For a sequence of bits, a TRNG strives for $H(X) = 1$ bit of information per digit, which occurs when $\mathbb{P}(0) = \mathbb{P}(1) = 0.5$.

Any value $H(X) < 1$ indicates some internal structure or predictability.

How do we generate TRNs?

Quantum Source (QRNG)

QRNGs achieve ontological randomness via quantum state collapse.

Using a Hadamard gate (H), we prepare a qubit in a uniform superposition

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Upon measurement in the computational basis, the state collapses. By the Born Rule, the probability of measuring bit i is

$$\mathbb{P}(i) = |\langle i|\psi\rangle|^2 = \frac{1}{2}, \quad i \in \{0, 1\}$$

Unlike PRNGs, this outcome is not a function of some s_0 but a probabilistic physical limit.

How do we measure randomness?

Paradigms of Randomness

Our study categorizes randomness into three measurable features.

- **Uniformity:** Are all possible values equally likely?

$$\mathbb{P}(x_i) = \frac{1}{k}, \quad k = |\mathcal{U}|, \forall i$$

- **Patterns:** Are there local dependencies or "clusters"?

$$\mathbb{P}(X_n | X_{n-1}, \ldots, X_0) \stackrel{?}{=} \mathbb{P}(X_n)$$

- **Periodicity:** Does the sequence exhibit cyclic behavior?

$$x_{n+T} = x_n, \quad \forall n \geq i, \text{ where } T \leq |S|$$

Fourier Transform

Periodicity ends up being the key to distinguishing between PRNGs and TRNGs. We measured this using the Discrete Fourier Transform (DFT).

Fourier Transform in one sentence

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{i 2\pi k \frac{n}{N}}$$

To find the energy at a particular frequency, spin your signal around a circle at that frequency and average a bunch of points along the path.[1]

[1] By Stuart Riffle.

Spectral Test

The DFT decomposes a sequence into its frequency components and detects hidden periodic structures by treating the sequence as a signal.

For a sequence $x_0, \ldots, x_{n-1}$, the transform is defined as:

$$f_k = \sum_{j=0}^{n-1} x_j e^{-2\pi i k j/n}, \quad k = 0, 1, \ldots, n-1$$

- We measure the *magnitude* $|f_k|$ to identify dominant frequencies.
- A TRNG should look like "white noise" in the frequency domain.
- If any $|f_k|$ exceeds a critical threshold $T = \sqrt{n \log(1/0.05)}$, it indicates a non-random periodic pattern.

Spectral Signatures

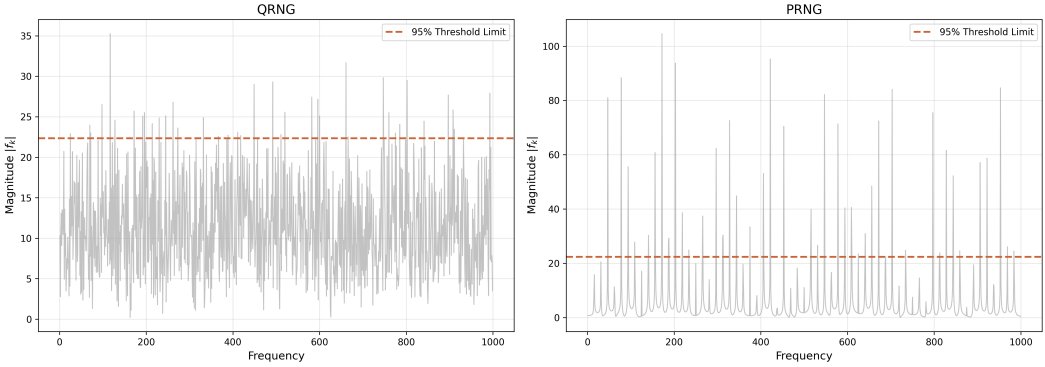


Figure 1: Left: QRNG showing a flat spectrum, Right: PRNG showing distinct spikes.

Test Suite

To capture a broader range of features, we applied tests such as (not limited to):

- **Uniformity.**
  - *Chi-Square Test.* Compares observed to expected frequencies.
  - *Kolmogorov-Smirnov Test.* Max distance between the empirical and the uniform distribution.
  - *Frequency Test.* Measures the proportion of 0s and 1s in a binary sequence.
- **Patterns.**
  - *Gap Test.* Analyzes the distribution of gaps between occurrences of a value.
  - *Serial Autocorrelation Test.* Measures the correlation between values at different positions.
- **Periodicity.**
  - *Shannon Entropy.* Quantifies the average information per symbol.
  - *Fast Fourier Transform (FFT).* Efficient DFT to identify periodic components.

Distinguishing PRNGs and TRNGs

We applied these tests to a dataset with sequences from a QRNG and three well-known PRNGs.

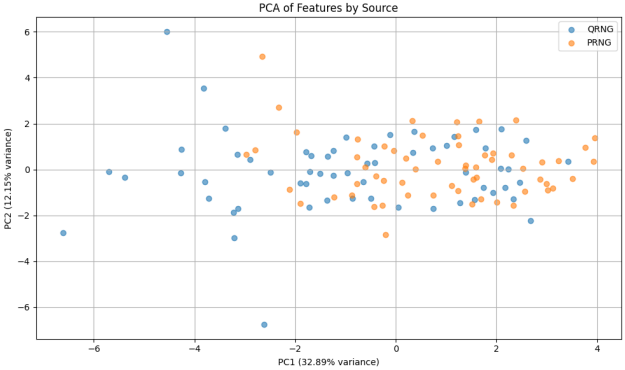
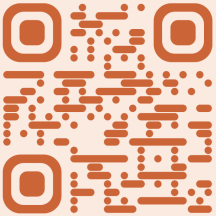


Figure 2: PCA feature space showing clear separation of QRNG and PRNG samples.

randomity



Randomity

`randomity` is an open-source Python package that automates the testing pipeline:

- **Extraction:** Takes an arbitrary sequence of numbers.
- **Testing:** Runs the full suite of statistical tests (Uniformity, Patterns, Periodicity).
- **Normalization:** Maps disparate metrics onto a standardized $[0, 1]$ scale.

You can also generated random sequences using `randomity`, both pseudo and true, for testing and benchmarking.

Randomness Score

`randomity` synthesizes a single composite score $S \in [0, 1]$, where

$$S = \frac{1}{3}\left(\overline{U} + \overline{P} + \overline{F}\right)$$

Where:

- $\overline{U}$ is the mean uniformity score.
- $\overline{P}$ is the mean pattern score.
- $\overline{F}$ is the mean periodicity score.

With an adjustable threshold (e.g. $S > 0.6$), we can flag sequences as sufficiently random for high-entropy applications.

Thank You!

Questions?

Sepehr Akbari
akbaris79@lakeforest.edu
Department of Math & CS, Lake Forest College

Feature Selection

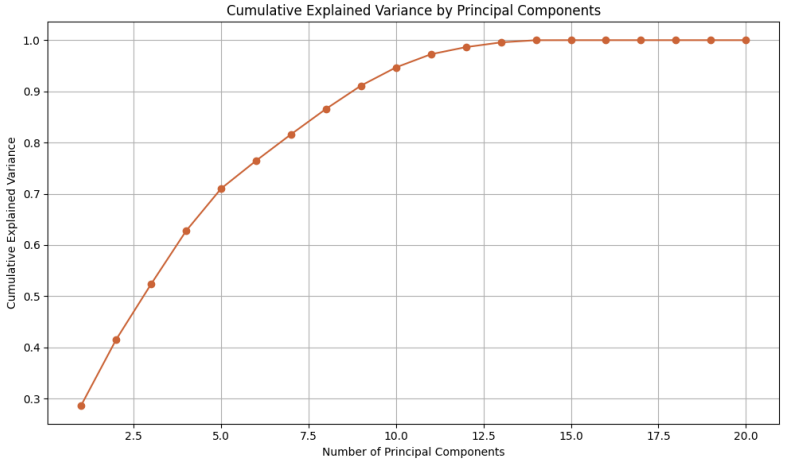


Figure 3: Cumulative Explained Variance from PCA (PC1-PC3 capture about 52%).

Feature Importance

Figure 4: Feature importance for PC1-PC3.