Random Number Generation & Testing

Juan Soto

soto@nist.gov

301/975-4641

The RNG Team

- *Statistics Department*, **UMBC**
  - Andrew Rukhin
- *Computer Security Division*, **NIST**
  - Miles Smid, Elaine Barker, Jim Nechvatal, Jim Dray, San Vo, and Juan Soto
- *Statistical Engineering Division*, **NIST**
  - Stefan Leigh, Mark Vangel, David Banks, Mark Levenson, Allen Heckert

Outline

- Introduction
- Overview of the NIST test suite
- Empirical Testing
- Future work
- Summary

Introduction

- **Random Number Generation**
  - Von Neumann is often quoted as having stated: “Anyone who considers arithmetic methods of producing random digits is, of course, in a state of sin.”
- **Testing RNGs**
  - He also stated, “…that in his experience it was more trouble to test random sequences than to manufacture them.”

NIST Goals

- A set of statistical tests suitable in the assessment of the randomness of (P)RNGs.
- Provide supporting documentation.
- Inclusion of the tests in the **Cryptographic Module Validation Program?**
- Development of a **Special Publication**?

Work In Progress

- **The development of several documents:**
  - ***"A Statistical Test Suite for the Validation of Cryptographic RNGs"*** including test strategy and test interpretation.
  - ***"The NIST Statistical Test Suite User's Guide Version 1.0"***
- **A reference implementation in ANSI C.**

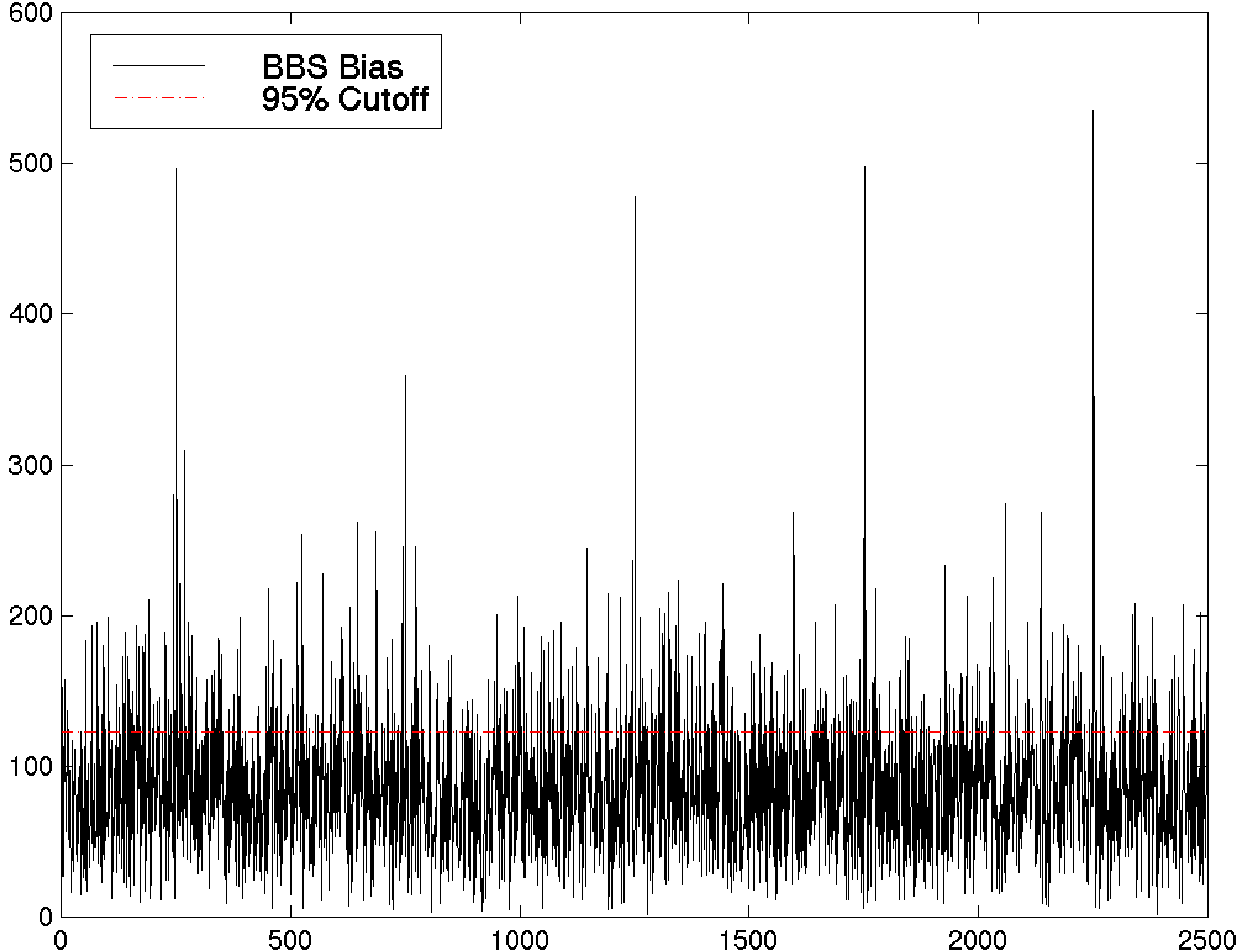
Example: A Finite Length Binary Sequence

01010110111010100110101001101011101001010010100101010

- **Bits** — **0s** = 26 — **1s** = 27
- **Templates** — **00s** = 5 — **01s** = 20 — **10s** = 20 — **11s** = 7
- **Runs** = 41 — 0, 1, 0, 1, 0, 11, 0, 111, 0, 1, ...
- **Cycles** = 5 — 01, 01, 01, 10, 111010…..010
- **Words** = 18 — 0, 1, 01, 011, 0111, 010, …
- **Linear Complexity** — < **27**, $1+D^{3}+D^{8}+D^{10}+D^{17}+D^{19}+D^{22}+D^{23}+D^{24}+D^{25}+D^{26}$ >

Overview of the NIST Test Suite

- **Frequency (Monobits)** Test
  - Assess the distribution of 0s and 1s.
- **Block Frequency** Test
  - Assess the distribution of m-bit blocks.
- **Spectral (DFT)** Test
  - Assess the spectral frequency of a bitstring.


BBS Bias
95% Cutoff
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500

Overview of the NIST Test Suite

- **Runs** Test
  - Assess the expected total number of runs.
- **Long Runs** Test
  - Assess the distribution of runs of ones; runs should not exceed $\log_2 n$.
- **Marsaglia's Rank** Test
  - Assess the distribution of the rank for 32x32 binary matrices.

Rank of Binary Matrices

$$rank\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = 4 \qquad rank\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = 3$$

The Rank of 32x32 Binary Matrices

28.88 % of binary matrices have rank = 32

57.76 % of binary matrices have rank = 31

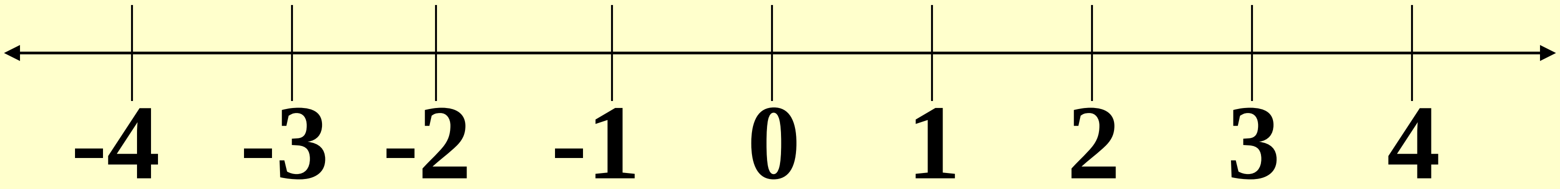
13.36 % of binary matrices have rank $\leq$ 30

Overview of the NIST Test Suite

- **NonOverlapping Template Matching** Test
  - Assess the frequency of m-bit nonperiodic patterns.
- **Cumulative Sums** Test
  - Assess that the sum of partial sequences isn't too large or too small; indicative of too many 0s or 1s.
- **Random Excursions** Test
  - Assess the distribution of states within a cycle of a random walk.

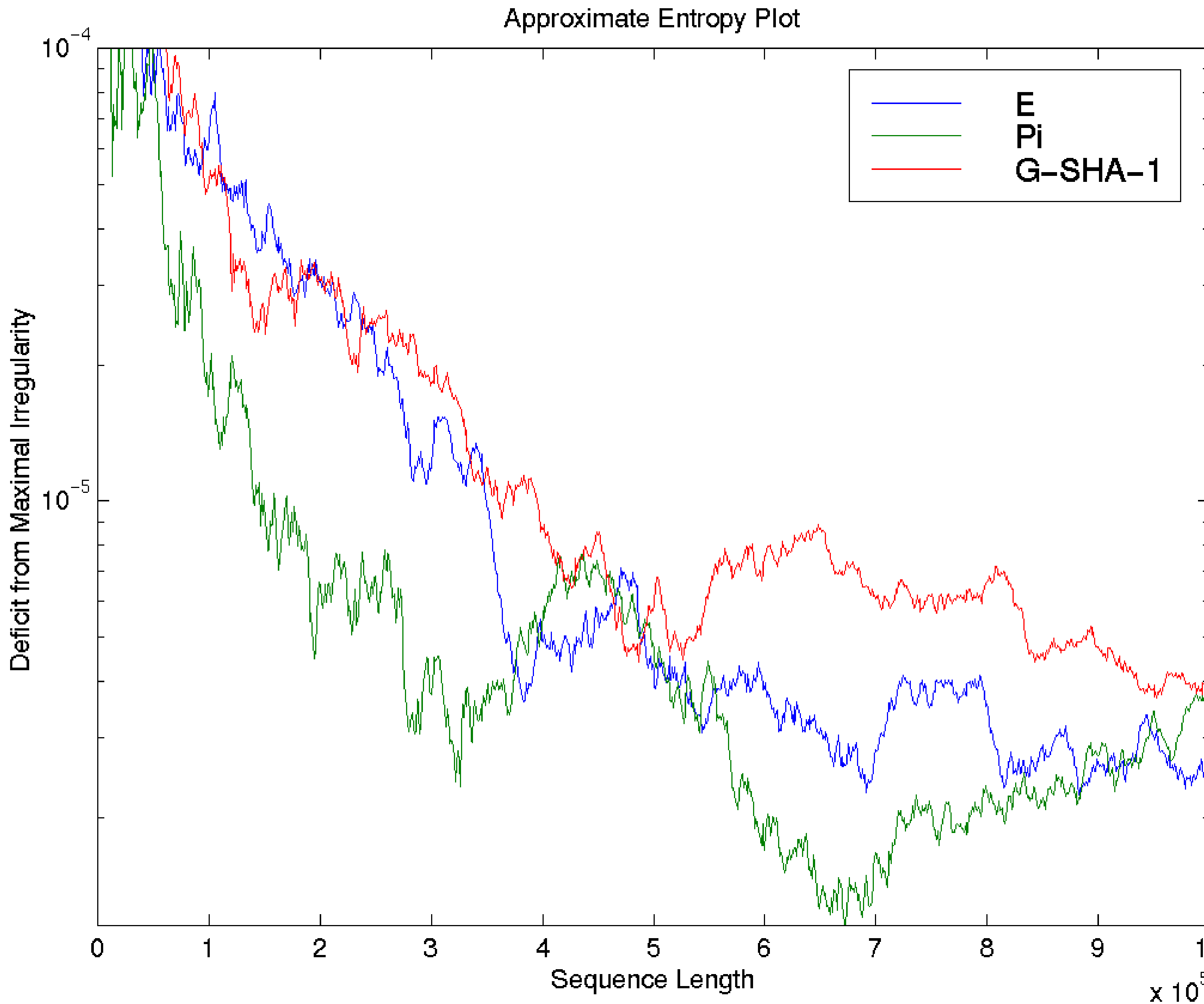
Random Walk (1D)

Bitstring	0	0	1	0	1	1
Transformed	-1	-1	1	-1	1	1
Summation	-1	-2	-1	-2	-1	0



Overview of the NIST Test Suite

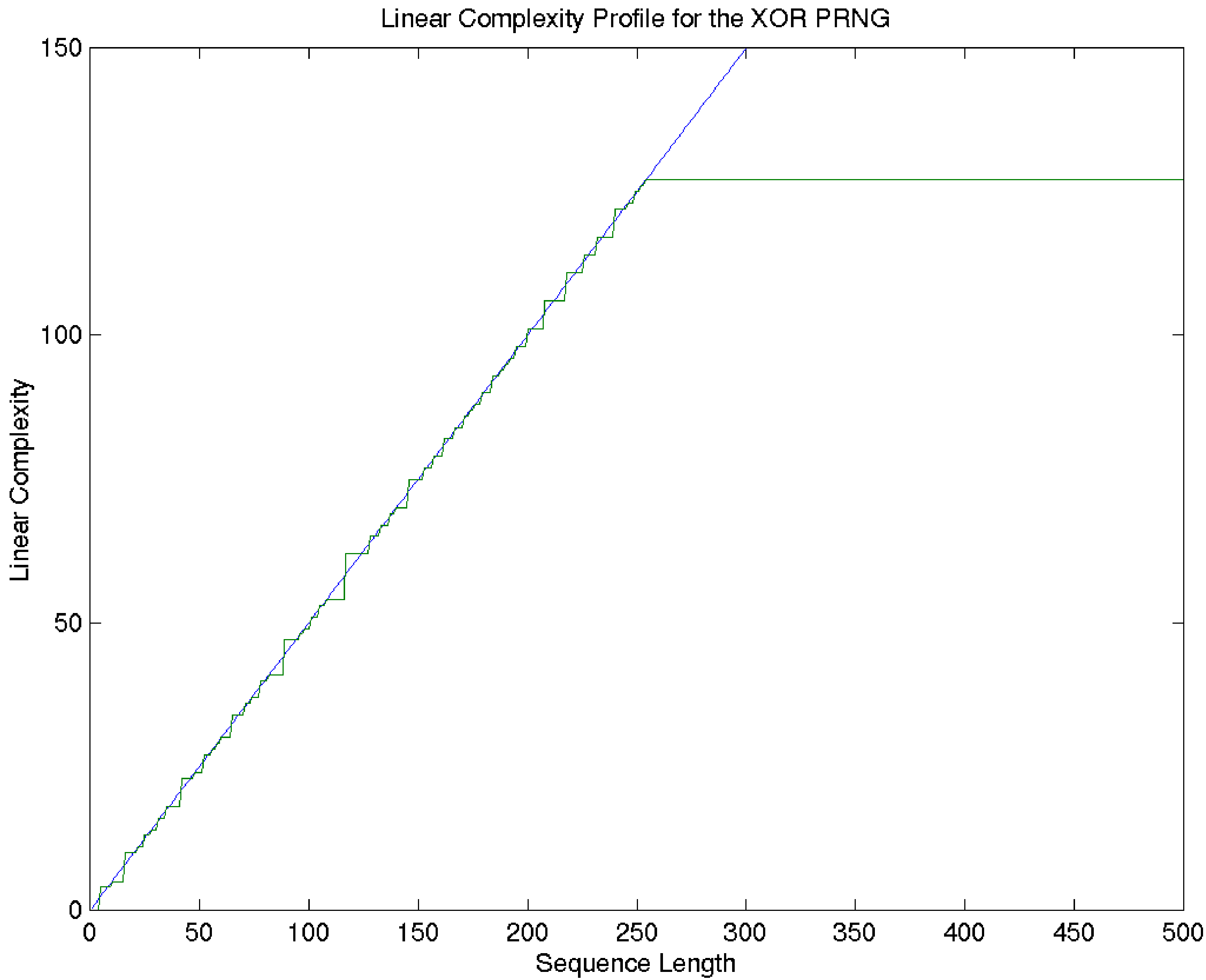
- **Overlapping Template Matching** Test
  - Assess the frequency of *m-bit* periodic templates.
- **Serial** Test
  - Assess the distribution of all 2^m *m-bit* blocks.
- **Approximate Entropy** Test
  - Assess the entropy (regularity) of a bitstring; compares the frequency of all *m-bit* patterns against all *(m+1)-bit* patterns.



Approximate Entropy Plot
Deficit from Maximal Irregularity
Sequence Length
x 10^5
E
Pi
G-SHA-1

Overview of The NIST Test Suite

- **Maurer's Universal Statistical** Test
  - Assess the compressibility of a bitstring.
- **Lempel-Ziv Complexity** Test
  - Assess the compressibility of a bitstring.
- **Linear Complexity** Test
  - Assess the linear complexity of a bitstring; the shortest LFSR that can generate the bitstring.



Linear Complexity Profile for the XOR PRNG
150
100
50
0
0
50
100
150
200
250
300
350
400
450
500
Sequence Length
Linear Complexity

Empirical Testing

- **Good PRNGs**
  - ANSI X9.17, G-SHA-1, G-DES
  - Blum-Blum-Shub
- **Block Cipher Algorithms** (AES)
  - Correlation, CBC Mode
  - Key (Plaintext) Avalanche
  - Special Key (Plaintext) Inputs

Poor PRNGs

- **XOR PRNG**
  - Fails the linear complexity test, rank test and several other tests. Failure due to the simplicity of the scheme.
- **HPC Key Avalanche**
  - Fails the monobits test, approximate entropy test and several others. Failure due to the existence of equivalent keys.

Our Efforts

- Tests developed for cryptographic use.
- Full scientific documentation provided (each algorithm based on rigorous math).
- Sixteen statistical tests fully developed to date; over 200 if one considers alternate input parameters.

Future Work

- Peer Review Process
- Testing Hardware RNG data
- Development of Additional Statistical Tests
  - Moving Averages & Generalized OPSO test
  - Block Cipher tests
- Inclusion of Assessment Tools
  - Graphical Utilities & Goodness-of-Fit tests

Summary

- Statistical tests are very important in ensuring good quality (P)RNGs.
- Statistical tests are necessary but not sufficient to recommend a (P)RNG.
- A statistical test suite must be diverse.
- In the last two years, **NIST** has developed over 200 tests.