**User’s Guide to Running the Draft NIST SP 800-90B Section 9 Entropy Estimation Tests**

30 September 2014

T. A. Hall

This is a brief introduction on how to run the Python command-line programs that implement the statistical entropy estimation tests found in Section 9 of the Draft NIST SP 800-90B (August 2012). It is not a description or explanation of the tests themselves. Please refer to the standard itself for definitions and descriptions of the tests and their rationales.

**Python files to implement the tests:**

***iidmain.py***

* Contains main routine to give the IID entropy estimate, if IID assumption holds
* Run six shuffle tests to determine if IID
* Run chi-square independence and goodness of fit tests to determine if IID
* Estimate min entropy if passes above tests
* Run sanity check tests

***noniid\_main.py***

* Contains main routine to give the non-IID entropy estimate
* Run five tests to estimate min-entropy
* min-entropy as lowest of the five
* Run sanity check tests

***shuffle\_tests.py***

* Contains six shuffle tests to determine if dataset is IID from Section 9.1.2

***chi\_square\_tests.py***

* Contains the chi square independence and goodness of fit for binary and non-binary data from Section 9.1.3

***iid\_tests.py***

* Contains the min entropy calculation for IID data from Section 9.2

***sanity\_checks.py***

* Contains the two sanity check tests from Section 9.4

***noniid\_collision.py***

* the non-IID collision test from Section 9.3.3
* Test may not be valid for all datasets.

***partial\_collection.py***

* The non-IID partial collection test from Section 9.3.4
* Test may not be valid for all datasets.

***markov.py***

* Contains the non-IID Markov test from Section 9.3.5
* Per SP 800-90B, only up to 6 bits per symbol used for Markov test

***maurer.py***

* Contains the non-IID compression test (Maurer Universal Statistic) from Section 9.3.6
* Test may not be valid for all datasets

***frequency.py***

* Contains the non-IID frequency test from Section 9.3.7

***util90b.py***

* Contains utility functions

**Sample dataset files:**

**Three files generated with TrueRand that should pass determine iid tests:**

* 1 000 000 data samples
  + 1 bit per sample (***truerand\_1bit.bin***)
  + 4 bits per sample (***truerand\_4bit.bin***)
  + 8 bits per sample (***truerand\_8bit.bin***)

**One file generated with TrueRand that should pass shuffle tests but fail chi square tests:**

* 1 000 000 data samples, 9 bits per sample (***truerand\_9bit.bin***)

**One file containing binary digits of pi that fails iid tests:**

* ***data.pi.bin***

**This User Guide**

* ***user\_guide.pdf***

The code has been tested and run successfully on the following:

* Python 2.6 on Linux
* Python 2.7 on Mac OS X
* Python 3.3 on Windows 7

It should run on any OS with Python 2.6+ or Python 3.

Note that this tool does not come with a Python installation. If you do not already have Python installed on your system, go to [https://www.python.org](https://www.python.org/) and select “Download.” No additional modules or packages are required to run the code. However, some routines will run faster if you have the **numpy** package installed. You can get **numpy** at <http://www.scipy.org> . If you are running a Windows OS, you can also find it here: <http://www.lfd.uci.edu/~gohlke/pythonlibs>. Alternatively, you can download the entire **scipy-stack**, which includes **numpy**.

**Running the tests**

The help message for the IID tests is:

C:\est>python iidmain.py -h

usage: iid\_main.py [-h] [-v] datafile bits\_per\_symbol number\_of\_shuffles

Run the Draft NIST SP 800-90B (August 2012) IID Tests

positional arguments:

datafile dataset on which to run tests

bits\_per\_symbol number of bits used to represent sample output values

number\_of\_shuffles number of shuffles per data subset for shuffle tests

optional arguments:

-h, --help show this help message and exit

-v, --verbose verbose mode: show detailed test results

Examples of running the IID tests follow.

Run the IID tests on the included truerand\_8bit.bin dataset, which contains 8 bits per sample. Use 1,000 shuffles of the data subsets and append the verbose flag for detailed output:

C:\est>python iid\_main.py truerand\_8bit.bin 8 1000 –v

Read in file truerand\_8bit.bin, 1000000 bytes long.

Dataset: 1000000 8-bit symbols.

Output symbol values: min = 0, max = 255

Compression Test:

Scores Ranks

106842 670

106886 856

106858 673

106718 106

106845 564

106899 867

106752 224

106867 741

106936 981\*

106849 695

---

1

Passed Compression Test

...

The full program output is not listed for space considerations. The first three lines of output are information about the dataset: its name, total size in bytes, how the raw bytes are interpreted (1,000,000 8-bit symbols as opposed to 500,000 16-bit symbols, for example) and the range of sample values in the dataset.

Following this is detailed information about the individual shuffle tests. The test name is followed by the scores and ranks of the 10 original (unshuffled) data subsets. If the rank of the score an original (unshuffled) data subset in a ranked ordering of the scores of all 1,000 shuffled data subsets is in the top or bottom 5%, then the rank is marked with an asterisk. For 1,000 shuffles, this works out to ranks of greater than or equal 950 or less than or equal to 50. If eight or more of the data subsets fall in this range, then the test fails, indicating that the IID assumption does not hold. Please see the Draft NIST SP 800-90B (August 2012) for an explanation and more details on this. A similar display of scores, ranks and Passed/Failed verdict is output for the other five shuffle tests.

If the dataset passes all of the shuffle tests, as is the case for truerand\_8bit.bin, then the program output indicates this and prints out details and results of the Chi-square tests and the overall determination of the IID assumption. If the determination is that the IID assumption holds, as is true for our example, the min-entropy estimate is output, followed by the details and results of the two sanity checks.

\*\* Passed iid shuffle tests

Chi square independence

score = 65212.5, degrees of freedom = 65280, cut-off = 66402.2

\*\* Passed chi-square independence test

Chi square stability

score = 2449.48, degrees of freedom = 2313 cut-off = 2528.88

\*\* Passed chi-square stability test

IID = True

min-entropy = 7.87108

Compression sanity check...

dataset 1 compressed length = 854736, cutoff = 787108...Pass

dataset 2 compressed length = 855088, cutoff = 787108...Pass

dataset 3 compressed length = 854864, cutoff = 787108...Pass

dataset 4 compressed length = 853744, cutoff = 787108...Pass

dataset 5 compressed length = 854760, cutoff = 787108...Pass

dataset 6 compressed length = 855192, cutoff = 787108...Pass

dataset 7 compressed length = 854016, cutoff = 787108...Pass

dataset 8 compressed length = 854936, cutoff = 787108...Pass

dataset 9 compressed length = 855488, cutoff = 787108...Pass

dataset 10 compressed length = 854792, cutoff = 787108...Pass

Collision sanity check...

Dividing dataset into 4-tuples

Check rule 1 - do three or more 4-tuples have the same value?...Pass

Check rule 2 - probability of number of collisions below cutoff

number of collisions = 6, cutoff = 10.4023...Pass

sanity check = PASS

If the same dataset and test parameters are run without the verbose flag, only the results of the IID determination, min-entropy estimate and sanity check are output:

C:\est>python iid\_main.py truerand\_8bit.bin 8 1000

IID = True

min-entropy = 7.87108

sanity check = PASS

Note that 1,000 is used as the number of shuffles for the examples above. 1,000 shuffles is specified in Draft NIST SP 800-90B (August 2012) and thus what should be used in order to run the tests in conformance with the standard. However, you may use a different number if you choose.

The help message for the non-IID tests is:

C:\est>python noniid\_main.py -h

usage: noniid\_main.py [-h] [-u use\_bits] [-v] datafile bits\_per\_symbol

Run the Draft NIST SP 800-90B (August 2012) non-IID Tests

positional arguments:

datafile dataset on which to run tests

bits\_per\_symbol number of bits used to represent sample output values

optional arguments:

-h, --help show this help message and exit

-u use\_bits, --usebits use\_bits

use only the N lowest order bits per sample

-v, --verbose verbose mode: show detailed test results

Next are some examples of running the non-IID tests.

First example: run the non-IID tests on the included truerand\_4bit.bin dataset, which has 4 bits per sample. The verbose flag is set in order to obtain detailed test results. Note that the results of each individual non-IID test are shown. The output is below:

C:\est>python noniid\_main.py truerand\_4bit.bin 4 -v

Read in file truerand\_4bit.bin, 1000000 bytes long.

Dataset: 1000000 4-bit symbols.

Output symbol values: min = 0, max = 15

- Collision test : p(max) = 0.0715332, min-entropy = 3.80524

- Partial collection test : p(max) = 0.074295, min-entropy = 3.75059

- Markov test : p(max) = 3.02369e-153, min-entropy = 3.95827

- Compression test : p(max) = 0.0789795, min-entropy = 3.66238

- Frequency test : p(max) = 0.063134, min-entropy = 3.95774

min-entropy = 3.66238

Compression sanity check...

dataset 1 compressed length = 435232, cutoff = 366238...Pass

dataset 2 compressed length = 435720, cutoff = 366238...Pass

dataset 3 compressed length = 435904, cutoff = 366238...Pass

dataset 4 compressed length = 435336, cutoff = 366238...Pass

dataset 5 compressed length = 435768, cutoff = 366238...Pass

dataset 6 compressed length = 435480, cutoff = 366238...Pass

dataset 7 compressed length = 435288, cutoff = 366238...Pass

dataset 8 compressed length = 435632, cutoff = 366238...Pass

dataset 9 compressed length = 435648, cutoff = 366238...Pass

dataset 10 compressed length = 435936, cutoff = 366238...Pass

Collision sanity check...

Dividing dataset into 9-tuples

Check rule 1 - do three or more 9-tuples have the same value?...Pass

Check rule 2 - probability of number of collisions below cutoff

number of collisions = 0, cutoff = 0.738097...Pass

sanity check = PASS

As with the IID tests, not setting the verbose flag produces compact results output:

C:\est>python noniid\_main.py truerand\_4bit.bin 4

min-entropy = 3.66238

sanity check = PASS

The usebits option allows you to instruct the program to consider only a lower order subset of the bits per sample. This is useful when almost all of the entropy is in these low order bits. Below is the case where only the lowest order two bits of the four bit samples are used.

C:\est>python noniid\_main.py truerand\_4bit.bin 4 –-usebits 2 -v

Read in file truerand\_4bit.bin, 1000000 bytes long.

Dataset: 1000000 4-bit symbols.

Output symbol values: min = 0, max = 15

\* Using only low 2 bits out of 4.

\* Using output symbol values: min = 0, max = 3

- Collision test : p(max) = 0.277344, min-entropy = 1.85025

- Partial collection test : p(max) = 0.276367, min-entropy = 1.85534

- Markov test : p(max) = 1.48356e-77, min-entropy = 1.9939

- Compression test : p(max) = 0.276001, min-entropy = 1.85725

- Frequency test : p(max) = 0.250906, min-entropy = 1.98776

min-entropy = 1.85025

Compression sanity check...

dataset 1 compressed length = 215816, cutoff = 185025...Pass

...<output deleted to save space>...

dataset 10 compressed length = 215928, cutoff = 185025...Pass

Collision sanity check...

Dividing dataset into 17-tuples

Check rule 1 - do three or more 17-tuples have the same value?...Pass

Check rule 2 - probability of number of collisions below cutoff

number of collisions = 0, cutoff = 0.587999...Pass

sanity check = PASS

**Disclaimers**

* This code is made available without any assertion or guarantee, implied or otherwise, of correctness or completeness.
* No support is provided for this code.
* The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

**Appendix: Selected Runs of IID and Non-IID Tests**

**Non-IID Tests on 1 bit TrueRand data:**

C:\est>python noniid\_main.py truerand\_1bit.bin 1 –v

Read in file truerand\_1bit.bin, 1000000 bytes long.

Dataset: 1000000 1-bit symbols.

Output symbol values: min = 0, max = 1

- Collision test : p(max) = 0.53125, min-entropy = 0.912537

- Partial collection test : p(max) = 0.526367, min-entropy = 0.925859

- Markov test : p(max) = 3.39255e-39, min-entropy = 0.998381

- Compression test : p(max) = 0.529541, min-entropy = 0.917186

- Frequency test : p(max) = 0.500433, min-entropy = 0.995227

min-entropy = 0.912537

Compression sanity check...

dataset 1 compressed length = 125008, cutoff = 91253.7...Pass

dataset 2 compressed length = 124792, cutoff = 91253.7...Pass

dataset 3 compressed length = 124920, cutoff = 91253.7...Pass

dataset 4 compressed length = 124896, cutoff = 91253.7...Pass

dataset 5 compressed length = 124904, cutoff = 91253.7...Pass

dataset 6 compressed length = 124824, cutoff = 91253.7...Pass

dataset 7 compressed length = 124912, cutoff = 91253.7...Pass

dataset 8 compressed length = 124920, cutoff = 91253.7...Pass

dataset 9 compressed length = 125032, cutoff = 91253.7...Pass

dataset 10 compressed length = 125240, cutoff = 91253.7...Pass

Collision sanity check...

Dividing dataset into 32-tuples

Check rule 1 - do three or more 32-tuples have the same value?...Pass

Check rule 2 - probability of number of collisions below cutoff

number of collisions = 0, cutoff = 0.791109...Pass

sanity check = PASS

**Non-IID Tests on 9-bit TrueRand data:**

C:\est>python noniid\_main.py truerand\_9bit.bin 9 –v

Read in file truerand\_9bit.bin, 2000000 bytes long.

Dataset: 1000000 9-bit symbols.

Output symbol values: min = 0, max = 511

- Collision test : p(max) = 0.00601578, min-entropy = 7.37703

- Partial collection test : p(max) = 0.00437081, min-entropy = 7.83788

- Markov test (map 6 bits): p(max) = 5.70333e-223, min-entropy = 5.7678

- Compression test : p(max) = 0.00553894, min-entropy = 7.49617

- Frequency test : p(max) = 0.002091, min-entropy = 8.23683

min-entropy = 5.7678

Compression sanity check...

dataset 1 compressed length = 948384, cutoff = 576780...Pass

dataset 2 compressed length = 947576, cutoff = 576780...Pass

dataset 3 compressed length = 946792, cutoff = 576780...Pass

dataset 4 compressed length = 949304, cutoff = 576780...Pass

dataset 5 compressed length = 947480, cutoff = 576780...Pass

dataset 6 compressed length = 947896, cutoff = 576780...Pass

dataset 7 compressed length = 950720, cutoff = 576780...Pass

dataset 8 compressed length = 947952, cutoff = 576780...Pass

dataset 9 compressed length = 946872, cutoff = 576780...Pass

dataset 10 compressed length = 947384, cutoff = 576780...Pass

Collision sanity check...

Dividing dataset into 6-tuples

Check rule 1 - do three or more 6-tuples have the same value?...Pass

Check rule 2 - probability of number of collisions below cutoff

number of collisions = 0, cutoff = 0.530863...Pass

sanity check = PASS

**IID Tests run on binary digits of pi**

Note that since the reordering of samples in the shuffle tests is random, the ranks may change from run to run of the IID tests for the same dataset.

C:\est>python iid\_main.py data.pi.bin 1 1000 –v

Read in file data.pi.bin, 1165666 bytes long.

Dataset: 1165666 1-bit symbols.

Output symbol values: min = 0, max = 1

Compression Test:

Scores Ranks

17867 1\*

17865 1\*

17874 1\*

17904 1\*

17851 1\*

17846 1\*

17894 1\*

17888 1\*

17877 1\*

17910 1\*

---

10

Failed Compression Test

Over/under Test:

Scores Ranks

22 52034 826 1\*

19 52410 501 1\*

19 52318 501 1\*

20 52577 549 1\*

20 52347 501 1\*

18 52090 328 1\*

20 52132 547 1\*

19 52169 501 1\*

19 52136 501 1\*

19 52131 501 1\*

--- ---

0 10

Failed Over/under Test

Excursion Test:

Scores Ranks

156.625 642

125.619 364

160.774 699

138.821 507

84.1666 42\*

85.0692 54

114.445 229

93.5417 80

102.259 123

109.691 238

---

1

Passed Excursion Test

Directional runs Test:

Scores Ranks

8735 8 6102 1000\* 187 1000\*

8721 8 6043 1000\* 202 1000\*

8710 9 6090 1000\* 501 1000\*

8734 9 6072 1000\* 501 1000\*

8779 8 6096 1000\* 182 1000\*

8810 9 6088 1000\* 501 1000\*

8711 8 6046 1000\* 162 1000\*

8736 8 6064 1000\* 190 1000\*

8842 9 6158 1000\* 501 1000\*

8636 9 6086 979\* 501 1000\*

--- --- ---

10 0 10

Failed Directional runs Test

Covariance Test:

Scores Ranks

0.0219853 1000\*

0.0203951 1000\*

0.0210892 1000\*

0.0200727 1000\*

0.0205968 1000\*

0.021612 1000\*

0.0218409 1000\*

0.0212882 1000\*

0.0217061 1000\*

0.0216055 1000\*

---

10

Failed Covariance Test

Collision Test:

Scores Ranks

1 18.4866 56 501 626 635

1 18.9863 49 501 964\* 124

1 18.051 46 501 146 13\*

1 17.9182 56 501 59 622

1 18.2407 49 501 390 132

1 17.9208 47 501 116 38\*

1 18.5787 52 501 662 370

1 17.9169 51 501 124 276

2 17.9608 56 971\* 102 634

1 18.0984 49 501 235 135

--- --- ---

1 1 2

Passed Collision Test

\*\* Failed iid shuffle tests

IID = False