Jitter Entropy v3.4.0-3.6.0 Opportunistically Essentially IID Assessment Procedure

Joshua E. Hill, PhD

KeyPair CONSULTING

CMUF Entropy WG
20241112

Goals

- SP 800-90B requires “a technical argument” to justify the entropy rate claim made in the EAR.
- Making a stochastic model for this system is generally infeasible.
- We wanted to create a heuristic analysis approach that:
    - Tries to find and use parameter settings that make JEnt library behavior very simple to make statistical assessment more meaningful.
    - Allows for a graceful transition to another approach where necessary.
- Conceptually, we try to decimate (throw away data) until the source acts like an IID source.

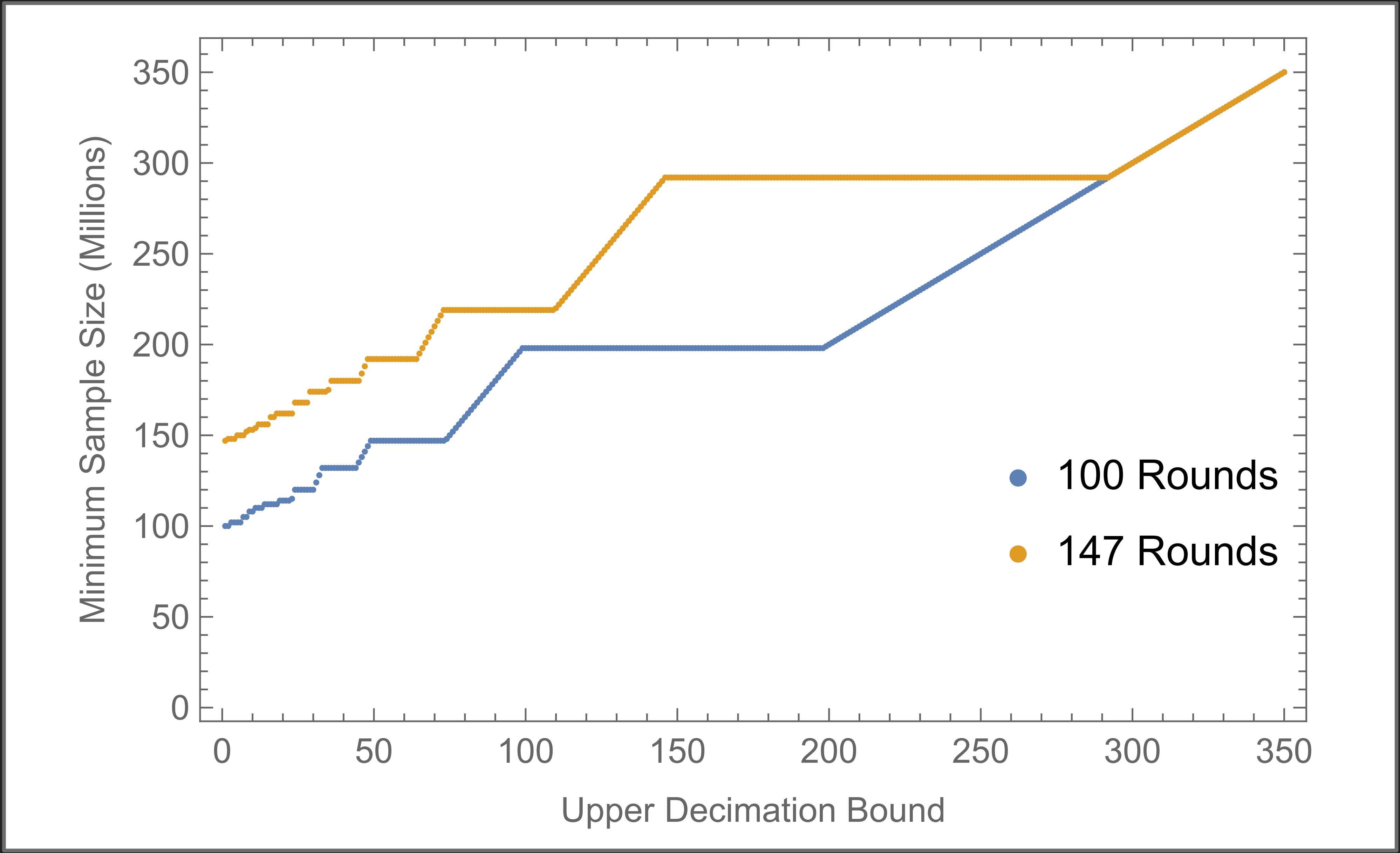
Important Parameters

- Often the system behavior is better with a larger memory size.
    - On the order of 2-10x the cache size often works well, though less sometimes works.
    - Note that it can be practically hard to force a particular memory size.
- For the particular system and use, determine the upper decimation bound ($\overline{D}$).
    - This is dependent on how fast the system needs to produce data.
    - This would commonly be based on some latency requirement for seeding.
    - This bound establishes the range of testing.
    - This bound must be thoroughly justified within the EAR.

Data Requirements

- Get a large amount of data.
  - The amount is dependent on the upper decimation bound.
  - Would like at least 147 blocks of 1 million samples of decimated data.

Data Requirements



Decimation Testing

- Perform large-scale IID testing at various decimation factors
  - We want all the tests to pass adequately often.
    - (There is clearly a technical definition for this.)
  - There are tools provided that automate this search.
- If a decimation factors are found that induce “essentially IID” behavior, use the lowest such factor as the selected decimation factor (D).
- If all tested decimation factors result in non-IID behavior, then use the upper decimation bound as the decimation factor ($D = \overline{D}$).
- This testing is hyper-hardware dependent.
- This testing can take a while.

Identification of Sub-Distributions

- If the (undecimated) data contains multiple sub-distributions (e.g., as with 3.4.0-3.5.0) then testing of the decimated data needs to occur for each identified sub-distribution.
- The minimum assessed rate for any sub-distribution is used as $H_{\text{decimated}}$.

$$H_{\text{heuristic}} = \frac{H_{\text{decimated}}}{D}$$

Results!

Select osr and $H_{\text{submitter}}$ so that

$$H_{\text{submitter}} = \frac{1}{osr} \leq H_{\text{heuristic}}$$

Goodies!

- Several worked examples demonstrating various expected conditions and testing approaches.
- There are links to software tools that can be quite helpful in the document. This includes:
  - A Python testing package produced by Yvonne Cliff (Teron Labs).
  - A histogram tool produced by Michael Bauknecht (UL).
  - An automated testing tool produced by Joshua Hill (KeyPair)
- A detailed investigation of the observed bimodal distributions produced by JEnt 3.4.0-3.5.0.

Now See Here!

Available on the CMUF Workspace under Documents / Entropy Working Group / CPU Jitter Guide.

https://cmuf-workspace.org/Products/Files/DocEditor.aspx?fileid=6240