Don't forget the law of large numbers!

Introduction

At first glance, you might wonder if you have to be crazy to run a test 250000 times that used to take more than 4 hours on a Pentium 3, per test! If you ask a mountaineer why he wants to climb mountains, he answers: “Because they are there!”. Accordingly, we could answer: “Because we can!”. However, the answer is not so simple. Very few scientists have a feeling for coincidences and random numbers. If you have 100 random numbers, you expect them to be nicely “random” and roughly equal to the expected average. What is often forgotten is that no random numbers stand alone, but are, due to the system, always an excerpt from an almost endless number of possibilities. Of course, it is not expected that all random numbers will be zero, although this possibility may exist with a probability of 1 in 10^-10000. With the law of large numbers always in mind, one will calculate on a sufficiently large number of cases so as not to fail at the threshold of “random” random numbers.

A small example can illustrate this idea: In the year 2006 Gil, Gonnet and Petersen published (A Repetition Test for Pseudo-Random Number Generators, Monte Carlo Methods and Appl. Vol. 12, No 5-6, pp. 385-393 / 2006) a study on the result of 100 continuous calculations on the distances of identical random numbers, 32-bit size. More details can be found in our download section under “Appendix A from VLST Test No 4”. By calculating the average of 100 cases three times, they wanted to determine whether a given RGN would meet the requirements. We later decided to perform this calculation 100 billion times in order to obtain a reliable value. You can then calculate the expected values per distance and compare the extent to which the calculated curve matches the measured curve. Calculating three times a hundred cases and drawing conclusions about the quality of the RNGs is a wasted effort and doomed to failure.

The same applies to the TestU01 Bigcrushs. Once performed, it can be an indication of gross violations of randomness. See as an example the two runs of the physical “True” Random Numbers. But we wanted to prove that our AHS-RNG families really generate true random numbers. However, we now know that this is not possible with the TestU01, even with a large number of tests. What we do know is the fact that the AHS-RNG, although its only goal is to generate the next bit with 50 percent probability for a “0” and 50 percent probability for a “1”, also works in the dimension of 1.7 trillion bits, on a par with XOshiro256** and MT19937, apart from the latter's problem with linear complexity. What we cannot prove with these 250000 tests, however, is the quality of “real” random numbers in the AH-RNG. Only by studying its structure can we rule out the possibility of finding artifacts. In the MT19937, we found artifacts at intervals of 5 billion random numbers in the 100 billion repetition test.

Test results / Book 250000 BigCrushs

For a very long time, we thought about how to process this flood of information in a profitable way. The result is our book “250000 Bigcrushs”, which presents all important information on 254 pages (Din A 3 landscape), sorted by P-Value number, and also offers some important statistics. It is best to open the book in Notepad++, for example, on a large screen, not on a cell phone or small tablet. Alternatively, you can also print it out on Din A3, with recto-verso this results in 127 sheets. We have benefited from the fact that we still have an old “Chanel-Bind” from Rank Xerox. The product was passed on to a Polish company via Leitz. Today you can find it under the name “OPUS MetalBind CLASSIC”. The good thing

about this product is the availability of covers in A3 landscape size, so that you can create yourself a real book with all the advantages. Everyone who has held it in hands has been delighted with the tactile feeling, which makes delving into the figures a real experience. Of course, you can also use classic A3 landscape folder. But working with 127 loose sheets is not satisfactory!

Structure of the book's pages

We have chosen the large format in order to have all the data of a P-Value at a glance. Spread over several A4 pages, possibly printed in recto-verso, is not expedient. The page is divided into three columns. On the right is the continuation of the list of 1/100 bins, 101 in total. The bins are numbered from 1 to 99. At the two ends, “<0.01” and >0.99" are added. This naturally causes problems with the number of expected results. The two columns “Expect.”, i.e. expectation, show the solution as to how 101 bins are to be filled with 1 percent probability each. The last four share the probability for 3 percent of the values. We have calculated that bins 0.01 and 0.99 have to cede almost half of the 500 expectations to the two at the beginning and end. The values were calculated to the best of our knowledge and belief from the rounding up and down found.

In the third column, the expectations are on the right and the bin number on the left. This makes it easier to visually search for a specific bin number. The number of cases that have ended up in the bins are shown in between, in the order of the RNGs. These are indicated in the header. The middle column starts in the upper half as a counterpart to the left column. The left-hand column starts at < 0.00001, the middle column at > 0.99999. The expectation expression acts as a separator between the two columns. In the top ten rows, 0.5 is used, as the number of 50000 is divided into 100000 bins. This is important to describe outliers on the left and right tail. After ten lines we increase to 1/10000 and the expected value now rises to 5 cases. The first line of this block is also the subtotal of the previous 10 cases, so there is no need for a separate expression. Following the logic, the lines from 21 to 30 relate to the 1/1000 bins and therefore have an expected value of 50. Again, the top line is also the subtotal of the 10 1/10000 bins. The list of 1/100 bins then begins below this, so the first row is again the subtotal of the 10 1/1000 bins. In the left column we show the first three bins. Bin 0.02 is the first normal bin, so we wanted to show the transition to normality.

Interim remark for the mathematicians

As a burnt child, I shy away from fire, and before the mathematicians grill me again, I would like to explain the following mathematical convention:

It is based on practical reasoning that we do not give the correct name of the rows. Mathematically correct, of course, it should read:

>= 0.00000 < 0.00001
>= 0.00001 < 0.00002
>= 0.00002 < 0.00003

In our opinion, this would be a useless waste of space. Therefore, let's agree that the number in the 30 detail lines within the three blocks only represents the increase compared to the previous lines! This makes it understandable that <0.001 in the 20th line means the difference between the total number and the number of all < 0.0009, while the 21st line, as the first of

this block, now indicates the effective number of all < 0.001. This seems easy to understand for all of us normal people, and if the mathematicians still have a stomach ache, then they would have to grit their teeth in the interests of the rest of humanity. This morning, for example, we saw a table of the standard normal distribution in a mathematics professor's textbook, where all values in the last line, 3.90 to 3.99, were rounded up to 1, as the calculation only involved 4 decimal places. This also caused me a lot of stomach ache!

Lower part of the right column

The five lines in the header give details about the test, including the parameters used here. Below that are the statistics. The first block at the top gives the calculated SDBernoulli-process per RNG, first column for the 101 percent bins, then the left tail and the right tail only in the 1/1000 scale. With an expected value of 50, an SD is already meaningful, above that it is no longer meaningful with an expected value of 5.

As “changes” we consider the change of neighboring bins from smaller/equal expected value to larger expected value, or vice versa, as a change.

In the block below, the first two columns are “average” and “median”. The average refers only to the percentage bins, the median to the middle bin, i.e. 101 of the 51st values of an ordered set of bin values. Both are important indicators of an anomaly, here exceptionally to be understood as an anomaly of the test program: Incidentally, it was the average in test no. 27 that prompted us to take a closer look at the program (or rather the microscope). There, the average is around 0.34 for all five RNGs.

The block > 500 indicates the number of bins with a value above the expected value in steps of ten. The bin .50 is not taken into account. If the distribution is uneven, the causes can be investigated in the bin list.

The lowest block is an evaluation of the SDBern of the percentage bins. The middle of the 9 columns shows the number of bins that have an SD of > -1 to < +1. Then to the left and to the right 4 levels of one counter each, i.e. > -2 but <= -1 and so on, all easy to understand, but very meaningful. I don't want to pillory individual tests now, but everyone will find a lot of candidates when browsing through.

Downloadable data

In order to facilitate the processing of the “mathematical derailments”, we have made all the data from our tests and their preparation available. The only request is that, if a researcher publishes a paper with this data, a reference to the source (sicap.lu) should be made in accordance with customary practice. Sending the paper to sicapas@pt.lu would be a nice gesture.

The original reports are available in three different versions. The gross version is intended for forensic examinations. Here, the reports output by the test program are copied to the exact byte. There are five downloads per RNG, from test number 10000 to 19999, 20000 to 29999, 30000 to 39999, 40000 to 49999 and 50000 to 59999. Based on our experience, we have started the numbering at 10000, which often makes handling easier, as there are always five digits.

If the zip files are unpacked, 10 files of around 94 MB each are created.

The second version is better suited for active work. In this version, we have inserted a line with the Run No in the heading. In addition, the p-values have been numbered from 1 to 254. We have combined all 50 files with 1000 reports as one file with cat, just under 5 GB. If you have enough RAM available, you can load this large file and view, for example, run no 14888, p-v no 199 in a matter of seconds.

If you want to browse through the reports manually, there is also the last version, called color and with extension ".log". These also contain ASCII color codes for red. When opened with Notepad++, for example, the run no and p-v lines are then displayed in red, which makes it easier to visually skim through the reports.

We have prepared files with 12,700,000 p-values (12,600,000 for MT19937, as all 197 and 199 were sorted out because of 100,000 times "eps-1") in 7 different variants. The first is the "forensic" version, which takes over the information from the report. This cannot be adopted in bc (scientific notation with e as exponent. We then created an intermediate version by conversion, which replaces the "e" with "*10^". To read this into bc, you either have to start bc -l or first read in a file with "scale=12". Furthermore, we then have a file with all values in decimal representation. For a better overview, all right-sided zeros have been hidden. But be careful: These two files work with arrays of a size of 25459999. If you can only use arrays of 16 million, then you can use the last .bcx file (.bcx is used to indicate that it is a file that is read into bc as a table of values).

This table consists of three arrays, which have a common sorting characteristic, namely the p-values in ascending order. The array m1 gits the p.v test-no at the same position, m2 the run no and m3 the pv value. The other files each use the code letter of the RGN. This limits the array to 12700000, namely the number of p-values.

The other three files always contain the same information in text form, but sorted differently. If someone wants to experiment with excell, these are perhaps the best basis. In any case, we have not tried it yet, just to be clear.

On the main page "Download" there is also the "All SDBern" if someone wants to work with the SD values of the individual bins. The index of the array is formed from the pv number times 1000 and the consecutive number of the bin, from 1 to 101.

Conclusion

The presentation of our research results is a trial balloon. AI believes that our format could actually become a new standard for RNGs research. We will be surprised. Therefore, remarks and comments to sicapas@pt.lu, addressed to Alain Schumacher, are very welcome, as well as well-founded criticism. We hope that hate comments and insults among scientists will not yet occur.