Modelling the Lengths of Quantitative Motifs in Pseudorandom Integer Sequences

Andrew Wilson [1*]

[1] Independent Scholar, Morecambe, UK
[*] Corresponding author's email: anwilql@mailfence.com

DOI: https://doi.org/10.53482/2024_56_416

ABSTRACT

This study explores the behaviour of quantitative motifs in random numerical data, focussing on the property of length. One hundred and twenty sequences of pseudorandom integers were generated and segmented into quantitative motifs using the standard definition. The length–frequency relationship was modelled in each case using the Zipf-Alekseev function. Different combinations of sequence length, integer range, and pseudorandom number generator were trialled. The model-fitting was successful in all cases ($R_1^2 > 0.9$ in 118/120 cases and > 0.8 in 2/120), and the model parameters fell in the same range as those obtained in a previous textual study of motifs. Integer range was the main factor affecting the parameter values, with sequence length affecting the degree of spread. Further comparisons between textual and random data are needed, as well as a theoretical explanation of why motifs in random data demonstrate lawful behaviour. In future textual motif studies, particular attention needs to be paid to possible dependencies on value range and text length.

Keywords: Motifs; length; random numbers; Zipf-Alekseev function.

1 Introduction

Motifs are textual units which have become a focus of attention in quantitative linguistics over the past two decades. Along with 'hrebs' and 'Belza chains', they belong to a set of 'new' units which all have their origins in the empirical study of repetitions in texts rather than in traditional accounts of linguistic structure (Altmann, 2014). Motifs, specifically, take their inspiration from work by Moisei Boroda in the 1970s on the segmentation of music (Orlov et al., 1982). Boroda was looking for a reliable algorithmic way of breaking down musical scores into smaller phrase-like units, as an alternative to the subjectively identified units known as 'motifs' in traditional musicology. To distinguish his units from the latter, Boroda originally called them 'F-motifs' (= 'formal motifs'). His idea was to use linear patterns of note lengths to segment the music, with an F-motif being (loosely speaking) a sequence of notes of equal or increasing length.[1] Reinhard Köhler later adapted and modified Boroda's ideas for use on verbal texts, as an alternative approach to studying syntagmatic patterns (Köhler & Naumann, 2007). The aim was to

[1] The precise definition is somewhat more complicated, as it also takes into account the metrical strength of certain notes, but this description captures its essence.

establish a unit that is intermediate in size between the word and the sentence and which is not tied to a particular linguistic theory. Motifs in contemporary quantitative linguistics are therefore purely formal units, constructed on the basis of the rhythms of the particular property under study.

In practice, we distinguish two different kinds of motifs: qualitative motifs, which are maximal non-repeating sequences of elements (such as letters, words, part-of-speech categories, etc.), and quantitative motifs, which are maximal monotone non-decreasing sequences of numerical values (such as lengths, frequencies, ratings on a scale, etc.) (Altmann, 2016). The present study is concerned only with quantitative motifs. To give an example of segmentation according to quantitative motifs, consider the sequence: ⟨3, 5, 7, 9, 4, 4, 2, 4, 3, 8⟩. According to the definition just given, this can be broken down into motifs as follows: ⟨3, 5, 7, 9⟩ ⟨4, 4⟩ ⟨2, 4⟩ ⟨3, 8⟩. In other words, given a sequence $\langle x_1, x_2, \ldots x_n \rangle$, a new motif is begun when one encounters a number in the sequence that is less than the number that immediately precedes it ($x_i < x_{i-1}$); otherwise, so long as a number is the same or greater than its predecessor ($x_i \geq x_{i-1}$), it is added to the contents of the current motif. It will be appreciated from this example that quantitative motifs can be considered, formally, as partially ordered multisets, whose elements are a subset of the real numbers (and often, though by no means always, of the positive integers). The motifs themselves therefore also have measurable properties, such as frequency, length, range, mean, etc., which can be modelled (Altmann, 2016). The linear sequences of these properties can even be used to construct increasingly higher-order motifs (e.g. Beyer, 2017). In the present study, the focus will be on motif lengths, but we will not concern ourselves here with higher orders of motif structure.

Conventionally, in quantitative linguistics, it has been assumed that 'the definition of a unit may be accepted if it allows the formulation of a law' (Altmann, 2016, p. 3). In other words, if we can model the interrelationships of properties of units (e.g. frequency, length, etc.) using a common function or probability distribution, this implies that those units can be accepted as 'legal entities' (Altmann, 2016, p. 3). Thus far, this has certainly seemed to be the case with motifs, both qualitative and quantitative. However, it is clear that any numerical or symbolic sequences can be segmented, respectively, into quantitative or qualitative motifs. If such motifs in random data also turn out to behave in very similar ways to their counterparts in texts, this raises questions at least about their conceptual foundations and possibly also about their usefulness as tools for investigating language and texts.

Mačutek & Mikros (2015) used randomly generated data to compare the fitting of the Menzerath-Altmann law for word-length motifs in five Modern Greek novels. They generated five random samples, each matched for size against one of the five novels and having the same frequency distribution of length classes as that novel – i.e., effectively, a random permutation of the original data sequence. They found

that, although the Menzerath-Altmann function could be fitted successfully to the random data as well as to the texts themselves, the parameter values for the random data were always larger (in absolute terms) than those for the novels. However, this is a rather small data set from which to generalize about the possible range of parameter values: had a larger set of random permutations been generated for each novel, the distinction between the genuine texts and the random data could easily have vanished.

The present study aims to contribute a further, and somewhat different, angle on the behaviour of quantitative motifs in random data, being based entirely on pseudorandom integer sequences. It will focus specifically on the property of length, which has been extensively studied for many constructs in quantitative linguistics (Popescu et al., 2014). The basic approach will be as follows. First, a pseudorandom integer sequence of a specified length n will be generated, which can be viewed as a random counterpart of a text of length n (or, rather, of a sequence of values representing a property of that text – e.g., the lengths of successive word tokens $x_1 \ldots x_n$). The integer sequence will then be segmented into quantitative motifs, the lengths of those motifs will be identified, and the frequency of each length-class will be counted. Finally, the relationship between the length of a motif and the frequency of its length-class will be modelled using a single widely accepted function. Different lengths of integer sequence will be tried, to test whether length has any effect on the fitting and parameters of the function. Two different ranges of integer values will also be tested. If the pseudorandom sequences can be modelled successfully using the same function that has been used in textual studies, and such sequences also result in similar parameter values, this will raise further questions that need to be addressed about the role of motifs in textual research.

2 Material

This study is based on sequences of pseudorandom integers, generated from within the 'R' environment for statistical computing (Ihaka & Gentleman, 1996).

It is usual to distinguish between 'true' random number generators (TRNGs) and pseudorandom number generators (PRNGs) (Herrero-Collantes & Garcia-Escartin, 2017). A TRNG generates random numbers on the basis of data derived from random physical processes such as radioactive decay or atmospheric noise. A PRNG, in contrast, generates numbers using a purely computational algorithm, which is deterministic and will generate the same sequence if seeded with the same parameter settings – hence the use of the term 'pseudorandom'. In many situations, the distinction between PRNGs and TRNGs is otherwise unimportant and the interest is more in the properties of the output than the mechanism of its generation; however, in some applications (such as cybersecurity), the possibility of reconstructing a PRNG's algorithm and parameter settings from a sample of its output can be a cause for concern, whilst in other applications (such as simulation-based scientific research) the ability to reproduce results exactly

can be considered a bonus.

The formal definition of what constitutes 'randomness' has been subject to considerable debate (see, e.g., Volchan (2002) for a concise discussion in relation to random sequences). In practice, however, randomness is assessed empirically. With random number generators, a rigorous standardized battery of statistical tests is used to assess whether sequences of numbers output by the generator are sufficiently unpredictable and void of periodic patterning. This testing procedure is applied equally to TRNGs and PRNGs. Different batteries of tests have been proposed over the years; the currently preferred benchmark standard seems to be the 'Big Crush' from the TestU01 suite, which involves running 106 different tests (L'Ecuyer & Simard, 2007). Unfortunately, however, there is a lack of accumulated comparative test data for some random number generators, especially newer ones, with many comparisons being documented in casual media such as blogs.

Most statistical packages and programming languages include one or more default PRNGs. In the case of 'R', these are provided by the `RNGkind` function, whose default setting is the Mersenne Twister algorithm. Unfortunately, the Mersenne Twister has been demonstrated to have a number of issues and it does not pass all of the 'Big Crush' tests (Vigna, 2019). The `RNGkind` function does also provide some other options, but these too have failed several of the 'Big Crush' tests. For this study, therefore, sequences of pseudorandom integers were generated instead using the `dqsample.int` function from the `dqrng` add-on package for 'R' (Stubner, 2021). This offers four options for the underlying PRNG: Xoroshiro128+, Xoshiro256+ (Blackman & Vigna, 2021), pcg64 (O'Neill, 2014), and Threefry (Salmon et al., 2011). Threefry was selected as the main PRNG for the study, as its authors state that it has passed all the randomness tests in the TestU01 'Big Crush'.

Sequences were generated using two ranges of positive integers: [1..4] and [1..9]. These ranges were not intended to simulate exactly any kind of linguistic data; however, the range [1..4] would not be untypical for word lengths in some languages (when measured in syllables), whilst the range [1..9] is quite similar to the ranges of rhythmical unit lengths encountered in Wilson (2019). The aim in choosing the two ranges was to see whether the range of values has any effect on the fitting of a function and its parameter values.

Different lengths of sequence were also generated and tested. Lengths of 250, 500, 1000 and 2000 elements were used. Ten sequences were generated for each unique combination of range and length, giving 80 sequences in total for the main data set.

Finally, to ensure that the choice of PRNG was indeed not having any major effect on the results, a supplementary data set for the range [1..9] was generated using the pcg64 PRNG, which has also passed all of the 'Big Crush' tests (Lemire, 2017). A grand total of 120 integer sequences were therefore

analysed in the study.

For all 120 sequences, a chi-squared test was used to check whether the integer frequencies were approximately uniformly distributed. The goodness of fit was assessed using the discrepancy coefficient C, which is given by

$$C = \frac{\chi^2}{N} \tag{1}$$

where N is the sum of the integer frequencies in the sequence (i.e., the sequence length). A fit was considered acceptable if $C \leq 0.05$ and good if $C \leq 0.02$. Only two sequences out of 120 failed the goodness-of-fit test at $C \leq 0.05$; the other 118 passed, with 96 of them also passing at the more stringent level of $C \leq 0.02$. Those sequences with $C > 0.02$ all belonged to the shortest sequence classes ($n = 250$ and $n = 500$) and all of them bar two also belonged to the integer range $[1..9]$. Taken together, this would seem to suggest that data sparseness was the cause of the somewhat higher values of C in most of these cases. The issue affected the Threefry and pcg64 PRNGs almost equally. The full results are tabulated in the Appendix.

3 Methodology

Using the standard definition introduced earlier ('a maximal monotone non-decreasing sequence of values'), the quantitative motifs in each integer sequence were identified and the frequencies of their lengths aggregated into a table, with any missing zero frequencies inserted where necessary. The relationship between the motif lengths and their frequencies was then modelled using the Zipf-Alekseev function, which has previously been used for this purpose and as a unifying model for length-frequency data more generally (Popescu et al., 2014; Andreev et al., 2017; Wilson, 2019). The Zipf-Alekseev function is given by

$$f(x) = 1 + a\, x^{\,b + c \ln x} \tag{2}$$

where x is the motif length, $f(x)$ is the frequency of that length, and a, b, and c are parameters. The constant of 1 is added because it is not possible to have a motif with a frequency smaller than 1. Note that the naming of the parameters in this function varies considerably between papers, so it is essential to check the formula that an author has used when making comparisons between parameter values. (A discussion of various ways of deriving the Zipf-Alekseev function for textual data can be found in Koch (2014).)

The model fitting was performed using the `minpack.lm` package for 'R' (Elzhov et al., 2016). Goodness

of fit was measured using the conventional R_1^2 (Eq. 1 in Kvalseth (1985), see also Magee (1990)), which is given by:

$$R_1^2 = 1 - \frac{\sum_{i=1}^{n}(f - f_{pred})^2}{\sum_{i=1}^{n}(f - \bar{f})^2} \tag{3}$$

where f is the observed frequency, $\bar{f}$ is the mean frequency, and f_{pred} is the frequency predicted by the model. A fit was considered good if $R_1^2 \geq 0.9$ and still acceptable if $R_1^2 \geq 0.8$.

As this is an exploratory study, only descriptive statistics were used for comparisons between groups. Quantiles were calculated using Definition 5 in Hyndman & Fan (1996), which ensures that the quantile values correspond with the hinges on boxplots (as the groups are all of an even size).

4 Results

The full results of the model fitting, including parameter estimates and R_1^2 values, are shown in Table 1, Table 2, and Table 3. The summary descriptive statistics (quantiles) for the R_1^2 values are shown in Table 4, with a boxplot as Figure 1. The summary descriptive statistics for parameters b and c are shown in Table 5 and Table 6. (Parameter a is disregarded, as this is merely a scaling parameter that approximates to the frequency of the first length class – see especially Koch, 2014, pp. 56–58.) Boxplots of parameters b and c are displayed as Figure 2 and Figure 3.

With both of the model parameters, and with the R_1^2 values, there was considerable overlap between the ranges of values obtained for the various combinations of integer range, sequence length, and PRNG. In the majority of cases, there was also overlap between the interquartile ranges, represented by the boxes on the boxplots. Nevertheless, some provisional trends can be noted in the data, which may deserve to be followed up in future studies.

Goodness of fit: R_1^2 values

The goodness of fit for the Zipf-Alekseev function, as measured by the R_1^2 values, was excellent, with only two integer sequences out of 120 demonstrating $R_1^2 < 0.9$. Both of these were in the shortest sequence-length class ($n = 250$) and, even in these cases, $R_1^2 > 0.8$. In general, within-group variability in the R_1^2 values was greatest in the shortest sequence-length class; with longer sequences, this reduced very considerably.

Parameter *b*

Parameter b was always a positive number. Its maximal range across all data sequences was [1.292, 3.338]. The median values of parameter b tended to be somewhat greater for the integer range [1..4]

Table 1: Parameter estimates and R_1^2 values for integer range [1..9]. Threefry PRNG.

Seq. Len.	a	b	c	R_1^2
250	25.067	2.585	−2.430	0.997
	33.007	2.168	−2.349	1.000
	28.651	2.633	−2.596	0.983
	33.690	1.466	−1.747	0.988
	24.439	2.001	−1.881	0.978
	25.652	2.217	−2.076	0.985
	21.222	2.951	−2.671	0.994
	25.476	2.334	−2.161	0.975
	25.243	1.882	−1.778	0.896
	28.752	1.480	−1.579	0.911
500	59.904	1.773	−1.875	0.993
	52.943	1.883	−1.846	0.980
	55.829	2.255	−2.226	0.997
	48.225	3.046	−2.847	0.998
	44.150	2.662	−2.346	0.997
	48.033	2.471	−2.266	0.987
	64.002	2.293	−2.402	0.998
	52.583	2.082	−2.002	0.993
	53.245	2.771	−2.702	0.995
	63.211	1.605	−1.773	0.983
1000	117.797	2.182	−2.204	0.997
	106.036	2.233	−2.125	0.996
	110.369	2.416	−2.344	0.997
	107.894	2.408	−2.318	0.999
	120.224	1.726	−1.796	0.988
	113.438	2.314	−2.280	0.999
	111.858	2.184	−2.153	0.992
	122.059	1.967	−2.017	0.995
	113.645	1.864	−1.880	0.987
	103.801	2.271	−2.172	0.999
2000	221.224	2.239	−2.186	0.997
	227.339	2.325	−2.260	0.997
	227.400	2.058	−2.037	0.997
	224.411	2.031	−1.990	0.994
	239.217	1.836	−1.896	0.992
	223.275	2.068	−2.054	0.999
	240.675	1.790	−1.858	0.994
	210.393	2.178	−2.069	0.994
	207.808	2.403	−2.304	0.999
	233.353	1.959	−1.987	0.997

Table 2: Parameter estimates and R_1^2 values for integer range [1..4]. Threefry PRNG.

Seq. Len.	a	b	c	R_1^2
250	16.631	1.325	−1.145	0.879
	15.729	2.444	−2.006	0.987
	18.108	2.064	−1.760	0.946
	10.995	2.547	−1.782	0.949
	13.910	2.324	−1.781	0.936
	17.573	2.119	−1.816	0.986
	15.947	2.528	−2.062	0.976
	20.012	1.995	−1.741	0.952
	12.678	3.056	−2.379	0.995
	9.631	2.501	−1.689	0.951
500	27.113	3.144	−2.375	0.995
	29.549	2.771	−2.156	0.987
	35.374	2.187	−1.835	0.987
	33.001	2.359	−1.937	0.990
	29.416	2.577	−1.944	0.955
	33.341	2.511	−1.996	0.960
	38.512	2.052	−1.801	0.984
	24.776	3.338	−2.517	0.996
	29.890	2.479	−1.979	0.996
	27.861	2.693	−2.060	0.977
1000	65.966	2.627	−2.078	0.992
	58.951	2.394	−1.830	0.992
	52.907	3.190	−2.406	0.990
	47.967	2.687	−1.907	0.947
	66.706	2.570	−2.111	0.989
	65.171	2.330	−1.869	0.996
	65.051	2.213	−1.793	0.995
	68.701	2.466	−2.015	0.996
	67.308	2.503	−2.024	0.999
	57.179	3.141	−2.433	0.997
2000	117.602	2.645	−2.007	0.988
	148.771	2.192	−1.842	0.997
	128.121	2.404	−1.895	0.991
	125.614	2.425	−1.902	0.993
	125.079	2.155	−1.703	0.995
	120.906	2.455	−1.910	0.998
	132.513	2.144	−1.724	0.988
	120.915	2.533	−1.965	0.996
	123.027	2.803	−2.229	0.993
	121.963	2.753	−2.175	0.994

Table 3: Parameter estimates and R_1^2 values for integer range [1..9]. pcg64 PRNG.

Seq. Len.	a	b	c	R_1^2
250	21.561	2.370	−2.071	0.978
	26.121	2.409	−2.321	0.965
	24.292	2.974	−2.865	0.982
	24.903	2.358	−2.123	0.916
	26.712	1.898	−1.900	0.990
	25.345	1.501	−1.495	0.940
	18.760	2.862	−2.366	0.929
	36.993	1.292	−1.703	0.996
	28.064	1.716	−1.913	0.995
	26.794	2.526	−2.407	0.995
500	55.180	2.298	−2.267	0.994
	50.630	2.456	−2.301	0.997
	55.173	2.291	−2.300	0.998
	50.248	2.158	−2.039	0.991
	62.708	1.635	−1.779	0.992
	51.328	3.180	−3.056	0.996
	51.749	2.290	−2.205	0.996
	57.704	1.968	−2.016	0.989
	60.821	1.928	−2.026	0.995
	63.944	1.590	−1.758	0.972
1000	114.044	2.038	−2.037	0.996
	122.253	2.019	−2.088	0.990
	115.950	2.221	−2.231	1.000
	132.366	1.775	−1.931	0.997
	116.219	2.102	−2.094	0.997
	111.224	2.415	−2.388	0.999
	107.983	2.307	−2.243	0.999
	105.016	2.186	−2.083	0.996
	127.110	1.562	−1.734	0.995
	109.690	2.821	−2.748	0.994
2000	222.407	2.215	−2.201	0.999
	216.644	2.123	−2.055	0.998
	230.453	2.104	−2.134	0.999
	237.569	1.979	−2.029	0.999
	236.653	1.984	−2.037	0.999
	215.220	2.384	−2.289	0.999
	237.258	1.987	−2.023	0.997
	232.634	2.033	−2.037	0.997
	249.044	1.888	−1.963	0.992
	247.080	2.085	−2.159	0.999

Table 4: Descriptive statistics for R_1^2.

	Seq. Len.	Min.	1st Qu.	Median	3rd Qu.	Max.
[1..4], Threefry	250	0.879	0.946	0.952	0.986	0.995
	500	0.955	0.977	0.987	0.995	0.996
	1000	0.947	0.990	0.994	0.996	0.999
	2000	0.988	0.991	0.994	0.996	0.998
[1..9], Threefry	250	0.896	0.975	0.984	0.994	1.000
	500	0.980	0.987	0.994	0.997	0.998
	1000	0.987	0.992	0.996	0.999	0.999
	2000	0.992	0.994	0.997	0.997	0.999
[1..9], pcg64	250	0.916	0.940	0.980	0.995	0.996
	500	0.972	0.991	0.994	0.996	0.998
	1000	0.990	0.995	0.996	0.999	1.000
	2000	0.992	0.997	0.999	0.999	0.999

Table 5: Descriptive statistics for parameter b.

	Seq. Len.	Min.	1st Qu.	Median	3rd Qu.	Max.
[1..4], Threefry	250	1.325	2.064	2.384	2.528	3.056
	500	2.052	2.359	2.544	2.771	3.338
	1000	2.213	2.394	2.537	2.687	3.190
	2000	2.144	2.192	2.440	2.645	2.803
[1..9], Threefry	250	1.466	1.882	2.192	2.585	2.951
	500	1.605	1.883	2.274	2.662	3.046
	1000	1.726	1.967	2.208	2.314	2.416
	2000	1.790	1.959	2.063	2.239	2.403
[1..9], pcg64	250	1.292	1.716	2.364	2.526	2.974
	500	1.590	1.928	2.224	2.298	3.180
	1000	1.562	2.019	2.144	2.307	2.821
	2000	1.888	1.984	2.059	2.123	2.384

Table 6: Descriptive statistics for parameter c.

	Seq. Len.	Min.	1st Qu.	Median	3rd Qu.	Max.
[1..4], Threefry	250	−2.379	−2.006	−1.781	−1.741	−1.145
	500	−2.517	−2.156	−1.988	−1.937	−1.801
	1000	−2.433	−2.111	−2.019	−1.869	−1.793
	2000	−2.229	−2.007	−1.906	−1.842	−1.703
[1..9], Threefry	250	−2.671	−2.430	−2.119	−1.778	−1.579
	500	−2.847	−2.402	−2.246	−1.875	−1.773
	1000	−2.344	−2.280	−2.163	−2.017	−1.796
	2000	−2.304	−2.186	−2.045	−1.987	−1.858
[1..9], pcg64	250	−2.865	−2.366	−2.097	−1.900	−1.495
	500	−3.056	−2.300	−2.122	−2.016	−1.758
	1000	−2.748	−2.243	−2.091	−2.037	−1.734
	2000	−2.289	−2.159	−2.046	−2.029	−1.963

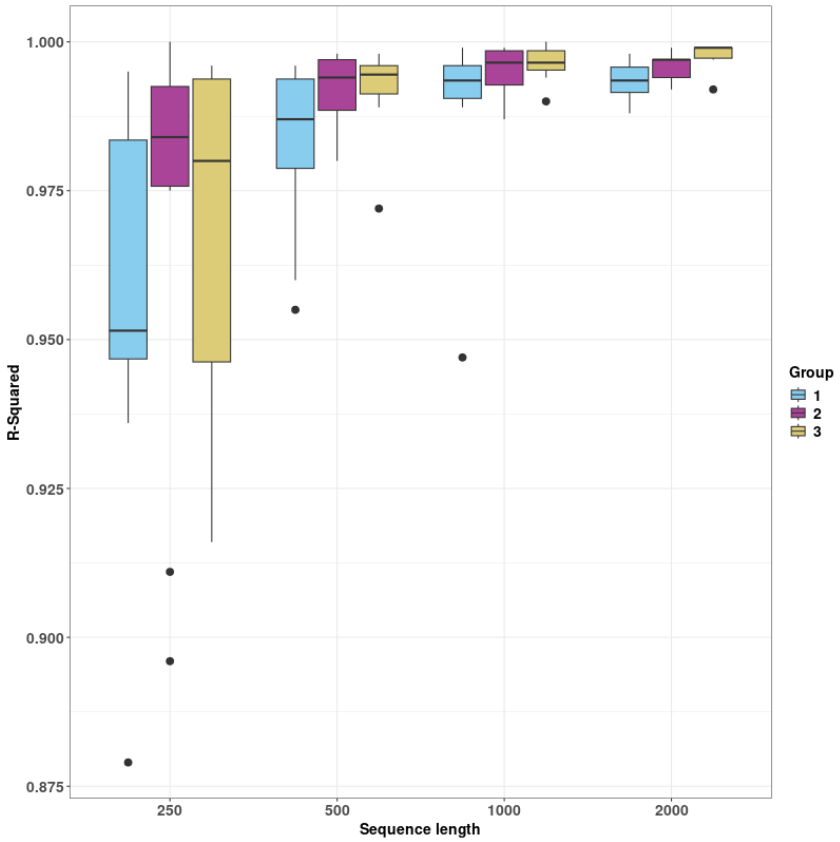

Figure 1: Boxplots of R_1^2 values. Group 1 = [1..4], Threefry. Group 2 = [1..9], Threefry. Group 3 = [1..9], pcg64.

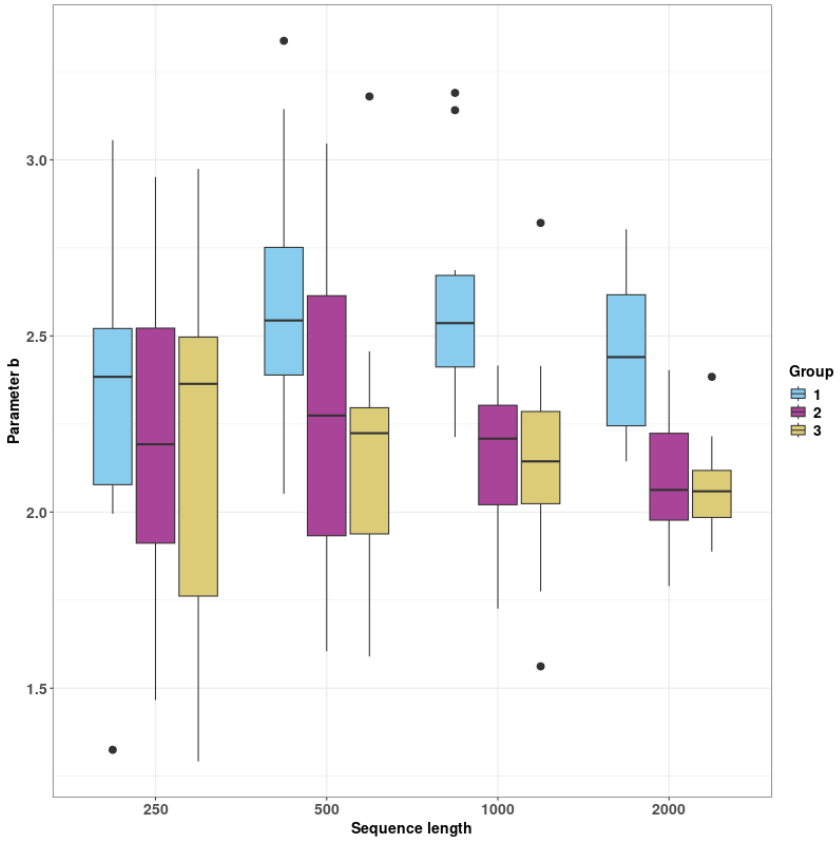

Figure 2: Boxplots of Parameter b. Group 1 = [1..4], Threefry. Group 2 = [1..9], Threefry. Group 3 = [1..9], pcg64.

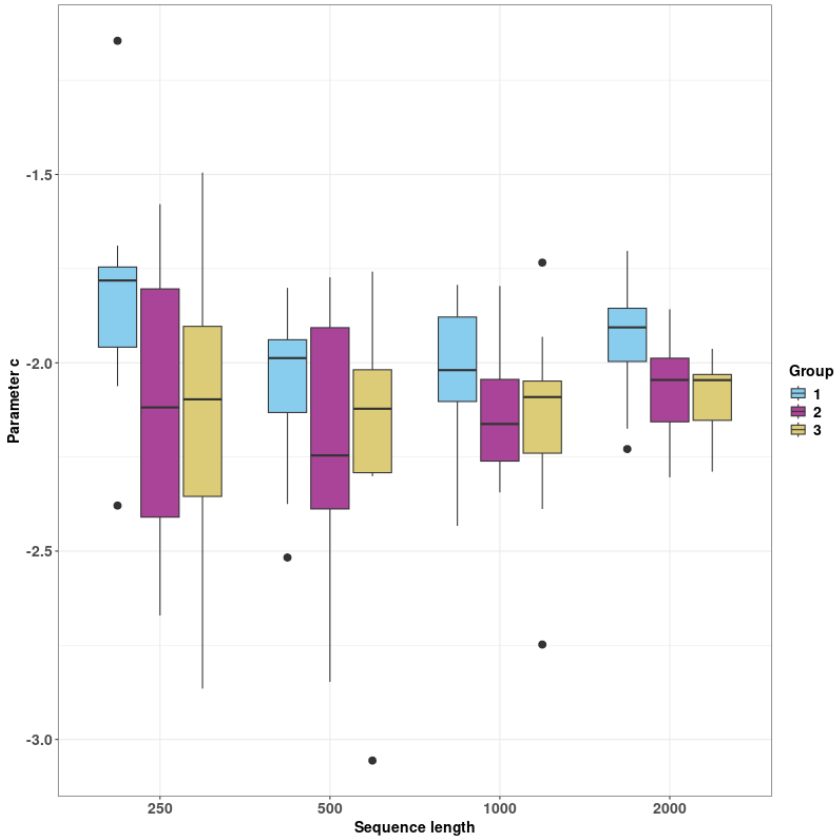

Figure 3: Boxplots of Parameter c. Group 1 = [1..4], Threefry. Group 2 = [1..9], Threefry. Group 3 = [1..9], pcg64.

than for the range [1..9], this being most marked for the longer sequences ($n > 500$), where the boxes on the boxplot also do not overlap. With the integer range [1..9], there was more within-group variability in the values of b for the shorter sequence lengths ($n = 250$ and $n = 500$), with the ranges and interquartile ranges becoming narrower for the longer sequence lengths ($n = 1000$ and $n = 2000$). With the exception of the shortest sequence length ($n = 250$), the median values of b for the Threefry and pcg64 PRNGs were very similar; however, the ranges and interquartile ranges for pcg64 tended to be somewhat narrower (apart from the occasional outlier).

Parameter *c*

Parameter c was always negative. Its maximal range across all data sequences was $[-3.056, -1.145]$. It exhibited smaller absolute median values for the integer range [1..4] than for the range [1..9]. As with parameter b, the amount of within-group variability reduced with sequence length; however, the amount of variability for parameter c was, in general, smaller than for parameter b. In the case of parameter c, there was slightly more difference in medians between the Threefry and pcg64 PRNGs; this affected the sequence lengths $n = 500$ and $n = 1000$, where the absolute median for Threefry was greater than for pcg64, but not the shortest or longest sequences ($n = 250$ and $n = 2000$), which had very similar medians.

5 Conclusion

This study set out to investigate whether the lengths of quantitative motifs in pseudorandom integer sequences exhibit lawful behaviour. The analysed data have shown that they do indeed exhibit such behaviour, which can be modelled successfully using the Zipf-Alekseev function. The goodness of fit is nearly always excellent, and seems even to improve marginally in the case of longer sequences and more extensive ranges of integer values.

The Zipf-Alekseev function was adopted by Popescu et al. (2014) as a general model for length–frequency relationships, and theoretically derived within the wider 'unified approach' to modelling properties of language and texts (Wimmer & Altmann, 2007). However, other functions have also been used to model the frequencies of motif-length classes, including the simple exponential function and the Lorentzian function (Andreev et al., 2018; Zörnig et al., 2019); the choice of function depends partly on the shape of the distribution. Future work might consider comparing the suitability of these functions, where they seem appropriate, alongside the Zipf-Alekseev function. Like the present study, this should be done for random data, as well as for actual texts, to see whether there are differences in the goodness of fit or in parameter values.

The Zipf-Alekseev parameter values that were obtained here for the pseudorandom sequences were not very much different from those seen in some comparable text-based studies. For instance, in a study of rhythmical unit motifs in twelve British English texts, Wilson (2019) obtained median values of $b = 2.636$ (IQR 2.52, 2.902) and $c = -1.991$ (IQR -2.261, -1.907) for ranges of length-classes varying from [1..8] to [1..11]. These figures all fall within the overall ranges of the results reported here for the comparable integer range [1..9].

In the case of the pseudorandom sequences, the parameter values seem to be affected mainly by the range of values in the integer sequence, with the data for range [1..4] tending to show somewhat larger absolute values of b and smaller absolute values of c than the data for range [1..9]. It would be interesting to check whether this behaviour holds true in data from textual investigations. The present data also suggest that shorter sequences have a greater variability in parameter values than longer ones. Again, it would be interesting to see whether this is also true when modelling motifs in texts of different lengths.

The choice of PRNG did not appear to have had very much effect on the results, with the parameter values for the sequences generated by Threefry and those generated by pcg64 being, for the most part, quite similar. This would seem to suggest that the results obtained here are not due to any undesirable artifact of the chosen PRNG and are indeed a genuine feature of random integer sequences. However, it would be useful to confirm this using other PRNGs, as well as TRNGs.

The findings of this study have implications for research in quantitative linguistics. In particular, they raise questions about the theoretical status of motifs as elements of text study. This does not automatically mean that motifs can be of no further use in text comparisons, but it does suggest that the successful fitting of a function or distribution to motif data cannot, in itself, be used to distinguish between random and non-random sequences. The fact that the ranges of parameter values may also differ relatively little between random data and textual data adds to the difficulties. Going forwards, we need to know, in particular, whether the range of values and/or the text length might be the only factors affecting the parameter values in text-based motif studies. Even if they are not, we still need to know whether there could be a dependency on one or both of those factors, as has been shown to exist between text length and some measures of vocabulary richness (Wilson & Popescu, 2022). One possibility for any future textual studies using motifs might be to generate some sets of random data with a similar length and range of values. The fitting statistics and parameter estimates from the random data could then be compared with those from the textual data. This is similar to what was done already by Mačutek & Mikros (2015); however, a greater number of random data sets will give a better impression of how far the textual data are distinct from the randomly generated data.

A further possibility is that motifs are indeed 'legal entities', but do not exist only in texts. It could be that the motif-length analysis is detecting a certain kind of regularity in supposedly random integer sequences – in other words, such random sequences may not be quite as random as we would like to think they are. The stringent and multiple statistical testing that is applied to random number generators would seem to argue against this view, but it cannot be rejected entirely without further experimentation. At the very least, we need a theoretical explanation of why such apparently lawful patterns of motif-length distribution can be found in virtually all these (pseudo) random integer sequences. Zörnig (2010) has made a start on explaining gap formation in binary random strings, but the case of quantitative motifs will be somewhat more complex, as their definition involves ordered relations between members, not merely the repetition of an element.

The present study has focussed only on the lengths of the quantitative motifs. However, as mentioned earlier, motifs have other measurable properties too, such as their frequency, their range, and the mean or median of their elements. Similar studies to the present one could be carried out for these properties too, which might tell a different story about the similarities or differences between textual and random data. Furthermore, although many measurements in linguistics are integer-valued, not all of them are: for instance, vowel duration, in principle, can take on any positive real value. There is therefore scope to study motifs in random sequences that are not comprised of integers. There are then, of course, also the qualitative motifs to be considered. The book on motifs in quantitative linguistics is therefore not closed by the present study; on the contrary, much more research needs to be done if we are to understand this

unit fully.

References

Altmann, G. (2014). Supra-sentence levels. *Glottotheory*, *5*(1), 25–39. doi: 10.1515/glot-2014-0002

Altmann, G. (2016). On Köhlerian motifs. In E. Kelih, R. Knight, J. Mačutek, & A. Wilson (Eds.), *Issues in quantitative linguistics 4, dedicated to Reinhard Köhler on the occasion of his 65th birthday* (pp. 2–8). Lüdenscheid: RAM-Verlag.

Andreev, S., Místecký, M., & Altmann, G. (2018). *Sonnets: Quantitative inquiries*. Lüdenscheid: RAM-Verlag.

Andreev, S., Popescu, I.-I., & Altmann, G. (2017). Some properties of adnominals in Russian texts. *Glottometrics*, *38*, 77–106. https://www.ram-verlag.eu/wp-content/uploads/2018/08/g38zeit.pdf

Beyer, A. P. (2017). Persistency of higher order motifs. In H. Liu & J. Liang (Eds.), *Motifs in language and text* (pp. 1–12). Berlin: De Gruyter. doi: 10.1515/9783110476637-002

Blackman, D., & Vigna, S. (2021). Scrambled linear pseudorandom number generators. *ACM Transactions on Mathematical Software*, *47*(4), 1–32. doi: 10.1145/3460772

Elzhov, T. V., Mullen, K. M., Spiess, A.-N., & Bolker, B. (2016). minpack.lm: R interface to the Levenberg-Marquardt nonlinear least-squares algorithm found in MINPACK, plus support for bounds [Computer software manual]. https://CRAN.R-project.org/package=minpack.lm (R package version 1.2-1)

Herrero-Collantes, M., & Garcia-Escartin, J. C. (2017). Quantum random number generators. *Reviews of Modern Physics*, *89*, 015004. doi: 10.1103/RevModPhys.89.015004

Hyndman, R. J., & Fan, Y. (1996). Sample quantiles in statistical packages. *The American Statistician*, *50*(4), 361–365. doi: 10.2307/2684934

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299–314. doi: 10.1080/10618600.1996.10474713

Koch, V. (2014). *Quantitative film studies: Regularities and interrelations exemplified by shot lengths in Soviet feature films* (Unpublished doctoral dissertation). Karl-Franzens-Universität, Graz.

Köhler, R., & Naumann, S. (2007). Quantitative text analysis using L-, F- and T-segments. In C. Preisach, H. Burkhardt, L. Schmidt-Thieme, & R. Decker (Eds.), *Data analysis, machine learning and applications.*

Proceedings of the 31st annual conference of the Gesellschaft für Klassifikation e.V., Albert-Ludwigs-Universität Freiburg, March 7-9, 2007 (pp. 637–645). Springer. doi: 10.1007/978-3-540-78246-9_75

Kvalseth, T. O. (1985). Cautionary note about R^2. *The American Statistician*, *39*(4), 279–285. doi: 10.2307/2683704

L'Ecuyer, P., & Simard, R. (2007). TestU01: A C library for empirical testing of random number generators. *ACM Transactions on Mathematical Software*, *33*(4). doi: 10.1145/1268776.1268777

Lemire, D. (2017). *Testing non-cryptographic random number generators: My results.* (https://lemire.me/blog/2017/08/22/ [Accessed: 9 Feb 2024])

Mačutek, J., & Mikros, G. (2015). Menzerath-Altmann law for word length motifs. In J. Mačutek & G. Mikros (Eds.), *Sequences in language and text* (pp. 125–131). Berlin: De Gruyter. doi: 10.1515/9783110362879

Magee, L. (1990). R^2 measures based on Wald and Likelihood Ratio joint significance tests. *The American Statistician*, *44*(3), 250–253. doi: 10.2307/2683704

O'Neill, M. E. (2014, Sep). *PCG: A family of simple fast space-efficient statistically good algorithms for random number generation* (Tech. Rep. No. HMC-CS-2014-0905). Claremont, CA: Harvey Mudd College.

Orlov, J. K., Boroda, M. G., & Nadarejšvili, I. S. (1982). *Sprache, Text, Kunst. Quantitative Analysen*. Bochum: Brockmeyer.

Popescu, I.-I., Best, K.-H., & Altmann, G. (2014). *Unified modeling of length in language*. Lüdenscheid: RAM-Verlag.

Salmon, J. K., Moraes, M. A., Dror, R. O., & Shaw, D. E. (2011). Parallel random numbers: As easy as 1, 2, 3. In S. A. Lathrop, J. Costa, & W. Kramer (Eds.), *Conference on high performance computing networking, storage and analysis, SC 2011, Seattle, WA, USA, November 12-18, 2011* (pp. 16:1–16:12). ACM. doi: 10.1145/2063384.2063405

Stubner, R. (2021). dqrng: Fast pseudo random number generators [Computer software manual]. https://CRAN.R-project.org/package=dqrng (R package version 0.3.0)

Vigna, S. (2019). *It is high time we let go of the Mersenne Twister*. https://arxiv.org/abs/1910.06437

Volchan, S. B. (2002). What is a random sequence? *The American Mathematical Monthly*, *109*(1), 46–63. doi: 10.1080/00029890.2002.11919838

Wilson, A. (2019). Lengths and L-motifs of rhythmical units in formal British speech. *Glottometrics*, *48*, 37–51. https://www.ram-verlag.eu/wp-content/uploads/2019/12/g48zeit.pdf

Wilson, A., & Popescu, I.-I. (2022). *Vocabulary richness in English poetry: The Lambda indicator and beyond*. Bucharest: Contemporary Literature Press. https://editura.mttlc.ro/iovitz-wilson-lambda.html

Wimmer, G., & Altmann, G. (2007). Towards a unified derivation of some linguistic laws. In P. Grzybek (Ed.), *Contributions to the science of text and language: Word length studies and related issues* (pp. 329–337). Dordrecht: Springer. doi: 10.1007/978-1-4020-4068-9_17

Zörnig, P. (2010). Statistical simulation and the distribution of distances between identical elements in a random sequence. *Computational Statistics and Data Analysis*, *54*(10), 2317–2327. doi: 10.1016/J.CSDA.2010.01.005

Zörnig, P., Stachowski, K., Rácová, A., Qu, Y., Místecký, M., Ma, K., ... Altmann, G. (2019). *Quantitative insights into syllabic structures*. Lüdenscheid: RAM-Verlag.

Appendix

The results of the χ^2 tests for uniformity of distribution are presented below, in descending order of C. ('Samp. Nr.' serves merely to identify unique sequences within the twelve groups defined by PRNG, Range, and Sequence Length.)

PRNG	Range	Seq. Len.	Samp. Nr.	χ^2	df	$P(\chi^2)$	C
Threefry	[1..4]	250	4	17.36	3	0.001	0.069
pcg64	[1..9]	250	3	13.52	8	0.095	0.054
Threefry	[1..9]	250	5	11.216	8	0.19	0.045
pcg64	[1..9]	250	6	10.712	8	0.219	0.043
pcg64	[1..9]	250	10	9.416	8	0.308	0.038
Threefry	[1..4]	250	6	9.232	3	0.026	0.037
pcg64	[1..9]	250	9	9.056	8	0.338	0.036
pcg64	[1..9]	250	2	8.48	8	0.388	0.034
Threefry	[1..9]	250	3	8.552	8	0.381	0.034
Threefry	[1..9]	250	2	7.976	8	0.436	0.032
pcg64	[1..9]	500	2	14.98	8	0.06	0.03
Threefry	[1..9]	250	7	7.4	8	0.494	0.03
pcg64	[1..9]	500	8	14.584	8	0.068	0.029
Threefry	[1..9]	250	6	7.184	8	0.517	0.029
pcg64	[1..9]	250	7	6.896	8	0.548	0.028
Threefry	[1..9]	500	7	13.972	8	0.082	0.028
Threefry	[1..9]	250	1	6.464	8	0.595	0.026
Threefry	[1..4]	250	3	5.936	3	0.115	0.024
pcg64	[1..9]	250	4	5.672	8	0.684	0.023
Threefry	[1..9]	500	3	11.668	8	0.167	0.023

PRNG	Range	Seq. Len.	Samp. Nr.	χ^2	df	$P(\chi^2)$	C
pcg64	[1..9]	500	5	10.804	8	0.213	0.022
pcg64	[1..9]	250	1	5.168	8	0.739	0.021
Threefry	[1..9]	250	8	5.168	8	0.739	0.021
Threefry	[1..9]	500	6	10.552	8	0.228	0.021
pcg64	[1..9]	1000	8	18.8	8	0.016	0.019
pcg64	[1..9]	500	3	9.652	8	0.29	0.019
Threefry	[1..9]	250	10	4.808	8	0.778	0.019
Threefry	[1..4]	500	1	9.136	3	0.028	0.018
Threefry	[1..9]	500	1	8.428	8	0.393	0.017
pcg64	[1..9]	500	1	7.852	8	0.448	0.016
pcg64	[1..9]	500	9	7.528	8	0.481	0.015
pcg64	[1..9]	1000	4	13.652	8	0.091	0.014
pcg64	[1..9]	500	7	6.808	8	0.557	0.014
Threefry	[1..4]	250	9	3.504	3	0.32	0.014
Threefry	[1..9]	1000	2	14.084	8	0.08	0.014
Threefry	[1..9]	500	4	7.204	8	0.515	0.014
Threefry	[1..9]	500	9	6.88	8	0.55	0.014
pcg64	[1..9]	1000	9	13.04	8	0.11	0.013
pcg64	[1..9]	250	5	2.936	8	0.938	0.012
pcg64	[1..9]	500	4	6.052	8	0.641	0.012
Threefry	[1..4]	250	10	2.896	3	0.408	0.012
Threefry	[1..4]	250	5	2.64	3	0.451	0.011
Threefry	[1..4]	250	7	2.768	3	0.429	0.011
Threefry	[1..4]	500	6	5.68	3	0.128	0.011
Threefry	[1..9]	250	9	2.72	8	0.951	0.011
Threefry	[1..9]	500	5	5.512	8	0.702	0.011
pcg64	[1..9]	1000	6	9.656	8	0.29	0.01
Threefry	[1..4]	250	1	2.448	3	0.485	0.01
Threefry	[1..9]	1000	5	9.98	8	0.266	0.01
Threefry	[1..9]	500	2	4.972	8	0.761	0.01
pcg64	[1..9]	1000	1	8.864	8	0.354	0.009
pcg64	[1..9]	1000	5	8.576	8	0.379	0.009
pcg64	[1..9]	250	8	2.36	8	0.968	0.009
pcg64	[1..9]	500	6	4.252	8	0.834	0.009
Threefry	[1..9]	500	8	4.576	8	0.802	0.009
pcg64	[1..9]	1000	10	7.892	8	0.444	0.008
Threefry	[1..4]	1000	3	7.928	3	0.048	0.008
Threefry	[1..4]	250	8	1.936	3	0.586	0.008
Threefry	[1..9]	1000	8	8.486	8	0.387	0.008
Threefry	[1..9]	1000	9	7.622	8	0.471	0.008
pcg64	[1..9]	1000	7	6.506	8	0.591	0.007
Threefry	[1..4]	250	2	1.808	3	0.613	0.007
Threefry	[1..9]	1000	3	6.686	8	0.571	0.007
Threefry	[1..9]	2000	1	13.327	8	0.101	0.007
Threefry	[1..9]	2000	9	14.218	8	0.076	0.007
Threefry	[1..9]	500	10	3.64	8	0.888	0.007
Threefry	[1..9]	1000	6	5.75	8	0.675	0.006
Threefry	[1..9]	2000	2	11.536	8	0.173	0.006
pcg64	[1..9]	1000	2	4.67	8	0.792	0.005
pcg64	[1..9]	2000	7	10.6	8	0.225	0.005
pcg64	[1..9]	2000	9	10.69	8	0.22	0.005
Threefry	[1..4]	1000	5	5.064	3	0.167	0.005
Threefry	[1..4]	1000	7	5.008	3	0.171	0.005

PRNG	Range	Seq. Len.	Samp. Nr.	χ^2	df	$P(\chi^2)$	C
Threefry	[1..4]	500	3	2.352	3	0.503	0.005
Threefry	[1..4]	500	5	2.368	3	0.5	0.005
Threefry	[1..9]	1000	10	5.156	8	0.741	0.005
Threefry	[1..9]	1000	4	5.462	8	0.707	0.005
Threefry	[1..9]	2000	5	9.115	8	0.333	0.005
Threefry	[1..9]	2000	6	10.6	8	0.225	0.005
Threefry	[1..9]	2000	7	10.654	8	0.222	0.005
pcg64	[1..9]	1000	3	3.536	8	0.896	0.004
pcg64	[1..9]	2000	2	8.494	8	0.387	0.004
pcg64	[1..9]	2000	4	7.558	8	0.478	0.004
pcg64	[1..9]	2000	5	8.791	8	0.36	0.004
pcg64	[1..9]	500	10	2.02	8	0.98	0.004
Threefry	[1..4]	500	7	1.776	3	0.62	0.004
Threefry	[1..4]	500	8	1.888	3	0.596	0.004
Threefry	[1..9]	1000	1	4.004	8	0.857	0.004
Threefry	[1..9]	1000	7	3.896	8	0.866	0.004
Threefry	[1..9]	2000	3	7.081	8	0.528	0.004
pcg64	[1..9]	2000	10	5.713	8	0.679	0.003
pcg64	[1..9]	2000	6	5.542	8	0.698	0.003
pcg64	[1..9]	2000	8	5.263	8	0.729	0.003
Threefry	[1..4]	1000	8	3.416	3	0.332	0.003
Threefry	[1..4]	2000	5	6.348	3	0.096	0.003
Threefry	[1..4]	500	2	1.424	3	0.7	0.003
Threefry	[1..4]	500	4	1.472	3	0.689	0.003
Threefry	[1..4]	500	9	1.744	3	0.627	0.003
Threefry	[1..9]	2000	8	5.605	8	0.691	0.003
Threefry	[1..9]	250	4	0.848	8	0.999	0.003
pcg64	[1..9]	2000	1	4.138	8	0.844	0.002
Threefry	[1..4]	1000	10	2.104	3	0.551	0.002
Threefry	[1..4]	1000	6	2.2	3	0.532	0.002
Threefry	[1..4]	2000	3	4.652	3	0.199	0.002
Threefry	[1..4]	500	10	0.816	3	0.846	0.002
Threefry	[1..9]	2000	10	4.012	8	0.856	0.002
pcg64	[1..9]	2000	3	2.338	8	0.969	0.001
Threefry	[1..4]	1000	4	1.128	3	0.77	0.001
Threefry	[1..4]	2000	10	2.2	3	0.532	0.001
Threefry	[1..4]	2000	4	2.988	3	0.393	0.001
Threefry	[1..4]	2000	7	1.252	3	0.741	0.001
Threefry	[1..9]	2000	4	2.878	8	0.942	0.001
Threefry	[1..4]	1000	1	0.072	3	0.995	0
Threefry	[1..4]	1000	2	0.488	3	0.922	0
Threefry	[1..4]	1000	9	0.28	3	0.964	0
Threefry	[1..4]	2000	1	0.456	3	0.928	0
Threefry	[1..4]	2000	2	0.684	3	0.877	0
Threefry	[1..4]	2000	6	0.132	3	0.988	0
Threefry	[1..4]	2000	8	0.948	3	0.814	0
Threefry	[1..4]	2000	9	0.432	3	0.934	0