Analyzing human random time generation behavior: A methodology and a computer program

ANDRÉ VANDIERENDONCK
Ghent University, Ghent, Belgium

In the present article, it is argued that in addition to the traditional random generation tasks discussed by Towse and Neil (1998), random time interval generation tasks should be considered as useful alternatives, because they allow a better empirical control of the executive task component in dual-task situations. First, a framework for discussing randomness over time is presented. Then, the article goes on to present three methods for the analysis of such tasks. A first method is based on the correlation between the intervals produced. The second method calculates the approximate entropy, and the third method converts the time sequences into binary sequences and estimates the statistical properties of the sequence on the basis of these binary data. A principal components analysis on 19 different measures based on 1,381 sequences produced in a number of single-task and dual-task experiments shows that the proposed measures form two general clusters, one related to output probability, perseveration, and alternation, and one related to sequential commonalities. The article also briefly describes a computer program that implements these methodologies.

In their review of the methods for analyzing human random generation behavior, Towse and Neil (1998) observed that random generation tasks are becoming increasingly important for studying cognitive performance. Indeed, the growing interest in executive functions and their interactions with cognitive tasks (see, e.g., Crawford, 1998; Rabbitt, 1997) has resulted in a growing usage of random generation as the secondary task in dual-task experiments. Even though the article by Towse and Neil is fairly comprehensive, at the end of the article, the authors admit that some researchers have used the production of random time intervals and that, for this methodology, the standard techniques they review are not well suited.

Random time interval generation requires that the participant press a key (usually the zero key of the numeric keyboard) so as to produce a sequence of keypresses that forms a completely random time pattern, an unpredictable rhythm. The method has been shown to impair memory performance in verbal short-term memory tasks, and this effect is obtained without interference at the level of the phonological loop or the visuospatial sketch pad (Vandierendonck, De Vooght, & Van der Goten, 1998). Similar results were observed when the method was used in combination with a visuospatial span task (the Corsi block task; Vandierendonck, Kemps, & Fastame, 2000). In another study, it was shown that in a span task in which items differed in content and location, performance was impaired mainly when a combined recall based on content and location was required (Martein, Kemps, & Vandierendonck, 1999). Random interval generation also affects the frequency of stimulus-independent thoughts (Stuyven & Van der Goten, 1995) and impairs antisaccadic and intentional saccadic eye movements (Stuyven, Van der Goten, Vandierendonck, Claeys, & Crevits, 2000). Finally, the method impairs verification of simple arithmetic sums (De Rammelaere, Stuyven, & Vandierendonck, 1999). The latter replicated some results obtained with another methodology by Lemaire, Abdi, and Fayol (1996).

Clearly, random time interval generation may help to design experiments in which the secondary task loads the executive component only. Probably, the task is also useful for comparing the performance of patients with that of normal controls under single-task conditions. In this vein, random time interval generation was used by Crevits, Claeys, Stuyven, Van der Goten, and Vandierendonck (1997) in a study of saccadic eye movements in prefrontal patients and normal controls and by Jimeno, Jimeno, Alonso, Segovia, and Vargas (1996) in a study with schizophrenic patients.

However, as long as there are no appropriate methods for analyzing the degree of randomness of the sequences produced, interpretation of the data remains partial, es-

pecially in comparisons of single-task and dual-task conditions. The present article first presents a framework for the conceptualization of randomness. Next, three methods for the analysis of random time intervals are described, and data from Monte Carlo simulations are presented to demonstrate the advantages and limitations of each of these methods. Finally, the merits of these methods and of random time interval generation are discussed. A short description of the computer program available for these analyses is presented in the Appendix.

RANDOM TIME INTERVALS

In this section, the notion of randomness over time is elaborated so as to provide a basis for the development of a method for analyzing the randomness of a sequence of time intervals. Consider a sequence of taps that occur randomly over time. At any one moment in time, there is a probability π with which a tap occurs. Each such moment is preceded by a number of times that a tap did not occur. The probability that there is a sequence of k moments without a tap before a tap finally occurs, $Pr(N = k)$, is given by

$$Pr(N = k) = (1 - \pi)^k \pi. \quad (1)$$

This means that, when at a particular moment a tap occurred (with a probability π), the probability that a tap will occur at the next moment is still π. This yields the property of statistical independence, which can be expressed as

$$Pr(x_{i+k} = 1 \mid x_i = 1) = Pr(x_{i+k} = 1 \mid x_i = 0). \quad (2)$$

This equation assumes a sequence of equal time intervals such that each of these intervals either contains a tap $(x_i = 1)$ or does not contain a tap $(x_i = 0)$. The expression states that the probability that a tap occurs at a particular time period is independent of whether a tap has already occurred $k = 1, 2, \ldots$ time periods earlier.

There is ample evidence that human random generation behavior deviates from randomness in a number of respects (see, e.g., Kareev, 1992; Lopes, 1982; Lopes & Oden, 1987; Neuringer, 1986; Rapoport & Budescu, 1992; Treisman & Faulkner, 1987; Wagenaar, 1970a, 1970b, 1972; Zahn, 1982). Sequences of binary events, such as heads and tails generated by humans, for example, violate the principle of statistical independence by showing too many short runs and too many alternations.

The literature documents two possible biases in random generation, a perseveration bias and an alternation bias. The perseveration bias is a tendency to repeat a previous outcome and can be defined as follows:

$$Pr(x_{i + 1} = 1) = \alpha Pr(x_i = 1) + (1 - \alpha)\pi. \quad (3)$$

This expression states that a tap is immediately repeated with a probability α and that otherwise the generation follows the geometric distribution as implied in Equation 1, with a probability π of producing a tap. The alter-

nation bias is a tendency to alternate outcomes and is defined as

$$Pr(x_{i+1} = 1) = \beta[1 - Pr(x_i = 1)] + (1 - \beta) \pi. \quad (4)$$

With a probability of β, the previous outcome is not repeated (i.e., the opposite outcome is produced), and otherwise, a tap is generated with a probability of π, as is specified in Equation 1. Note that when $\alpha = 0$ in Equation 3 or $\beta = 0$ in Equation 4, the probability of generating a tap is constant over time (π). In other words, when there is no bias, the models in Equations 3 and 4 reduce to the nonbiased model (Equation 1).

Both biases follow from a Markov model that has been used before (e.g., Budescu, 1985, 1987). This implies that the probability of an outcome in a given time window is independent of all previous outcomes, except the one immediately preceding.[1]

CORRELATIONS OF TIME INTERVALS

If the time intervals between taps are completely random, it may be expected that the lengths of consecutive intervals are not related. This is equivalent to a zero correlation at Lag 1. More generally, it may be expected that the length of any interval is statistically unrelated to previous intervals, and this means that a zero correlation is expected at any lag. To the extent that the generation process is biased, some of these correlations may be expected to be large and positive (expressing a perseveration tendency) or to be large and negative (corresponding to an alternation bias). Hence, a test based on the correlations at Lags 1, 2, ... could be used to detect the degree of randomness or the size and the orientation of the bias in a sequence of time intervals.

However, there is one pitfall. The standard statistical method by which to test the significance of the correlation coefficient or to calculate the confidence interval cannot be applied, because the pairs of data values are not independent. The value y_i in pair (x_i, y_i) is the same as the value x_{i+k} in pair (x_{i+k}, y_{i+k}). Usage of a permutation statistic (Edgington, 1969) may provide a way out. By taking all possible permutations of the series of values in the sample, the set of all possible samples containing the observed values is constructed. The value of the statistic (e.g., the correlation coefficient) in each of the possible samples is calculated, and this yields a sampling distribution for the statistic. A statistical test of the sample statistic is then possible, either by estimating the probability of the sample statistic in this distribution or by defining a confidence interval on the basis of the mean and the standard deviation of the sampling distribution or on the basis of an interquantile interval. In practice, it is often not practical to make all the possible permutations. Instead, a random sample of all the permutations can be used, and the characteristics of the sampling distribution can be inferred from this subset. This is also known as the bootstrap method (Efron, 1979). For its application

to correlations, a solution has been described by De Soete and Vandierendonck (1982).

A Monte Carlo study was performed to explore the possibilities of this methodology. To that end 1,000 samples of 50 time intervals were generated on the basis of the geometric distribution (Equation 1). Also, sets of 1,000 samples of 50 time intervals with either a positive or a negative bias were generated. Three levels of bias were used—namely, values of .25, .50, and .75 were used for α in Equation 3 and for β in Equation 4. In the non-biased samples, the probability of a correlation (Lags 1–10) outside the .95 confidence interval ranged between .044 and .063. The corresponding values in the positively biased sets, ranged between .027 and .059. In the negatively biased set, the range was between .039 and .062. In order to check for important differences of the biased and the nonbiased distributions, the correlations observed in the biased samples were compared with the nonbiased sampling distribution. Again, the proportions of extreme values hovered around .05 (.026–.055 in the positively biased and .033–.065 in the negatively biased samples).

It is also interesting to know whether extreme correlations tend to occur more often together in the biased than in the nonbiased samples. To that end, a count was made of the number of sequences with 0, 1, ... extreme correlations. The top half of Table 1 displays the proportions of sequences as a function of the bias parameter and of the number of extreme values. This panel shows that the number of extreme correlations in a single sequence does not depend on the bias. In fact, the figures show that two or more significant lag correlations in a single series is a rather rare phenomenon (.019–.027).

A similar study was performed with samples of 25 time intervals. In all respects, the results were very similar: The proportions of extreme correlations were in the range of .032–.068 in the no-bias distribution, .036–.071 in the negative bias distributions, and between .025 and .198 in the positive bias distribution. Actually, the ranges in the distributions with a positive bias were rather similar to the others. For bias values of .25 and .50, the proportions ranged between .025 and .056. Only when the positive bias was

high (.75) was the probability of an extreme correlation high, and it varied from .096 (at Lag 1) to .198 (at Lag 10). Clearly, these higher values were due to the presence of a strong positive bias in a short sequence. In fact, when the bias is so large, a series of 25 intervals contains many repetitions of the same interval size, and consequently the number of possible different samples becomes too small to make reliable estimations. If it is safe to generalize the tendency toward alternation observed in the generation of binary sequences (e.g., Lopes & Oden, 1987) to the generation of random time intervals, there is no need to worry about this situation, because it would be extremely unlikely to observe sequences with such a strong positive bias.

The lower half of Table 1 displays the proportions of extreme correlations observed per sequence. Overall, the figures in the lower half correspond quite well to those in the upper half, except again for the case of a strong positive bias.

Taken all together, these findings seem to suggest that by simply analyzing the intertap times, it is not easy to find out whether a sequence can be characterized as biased or unbiased. Nevertheless, the method can be used to gain some insight into the characteristics of the series.

APPROXIMATE ENTROPY

Pincus (1991) has proposed approximate entropy (ApEn) as a measure of system complexity (see also Slifkin & Newell, 1999).[2] Applied to sequences of intervals that may vary in their degree of randomness, higher levels of randomness correspond to a higher degree of complexity, whereas sequences deviating from randomness become more and more predictable and, hence, less complex.

The rationale of this measure is that entropy is approximated by estimating the probabilities of patterns of events. Let $t_i (1 \le i \le N)$ be the time intervals in a sequence of events. The standard deviation of all these values is s. Next, define a lower bound $l_i = t_i - rs$, where r is a parameter usually set at 0.2, and an upper bound $u_i = t_i + rs$, so that each t_i is the center of an interval between the lower bound (l_i) and the upper bound (u_i). Consider the vector $[t_1, t_2]$. This template vector is now compared with every possible vector in the sequence—that is, $[t_1, t_2]$, $[t_2, t_3]$, $[t_3, t_4]$, and so forth. The two vectors $[t_1, t_2]$ and $[t_i, t_{i+1}]$ are defined to be similar if $l_1 \le t_i \le u_1$ and $l_2 \le t_{i+1} \le u_2$—in other words, if the two pairs of values are rather close to each other. Within the subset of all vectors that are in this way similar to the template vector, a test is performed to check whether both vectors are continued in a similar way; this is the case if $l_3 \le t_i+2 \le u_3$. The number of vectors that pass this test is divided by the number of vectors that were found to be similar, and this yields a first proportion or probability estimate.

The procedure is then repeated for all possible template vectors $[t_2, t_3]$, $[t_3, t_4]$, and so forth. If $p_j (1 \le j \le N - 2)$ indicates the estimated probabilities, the ApEn measure can now be calculated as

Table 1
Proportion of Sequences with 0, 1, or More Extreme Correlations as a Function of Bias

Number	Negative Bias			No Bias	Positive Bias		
	.75	.50	.25	(.00)	.25	.50	.75
			Sequences of 50				
0	.787	.777	.766	.767	.777	.806	.860
1	.188	.197	.207	.207	.200	.174	.121
≥2	.025	.026	.027	.026	.023	.020	.019
			Sequences of 25				
0	.810	.780	.762	.785	.797	.844	.781
1	.161	.193	.201	.189	.179	.137	.136
≥2	.029	.027	.037	.026	.024	.019	.083

Note—In the context of this table, an extreme correlation is one with a value outside the .95 confidence interval.

$$\text{ApEn} = \frac{1}{N-2} \sum_j \ln p_j. \qquad (5)$$

The present exposition and usage of the measure is based on vectors of length 2. It is also possible to consider longer or shorter vectors. The program described in the Appendix provides a procedure to calculate the ApEn with the usual parameter settings—namely, $r = .2$ and vector length $= 2$.

The same Monte Carlo samples used for the analysis of the lag correlations were used to calculate distributions for the ApEn measure. Table 2 displays the averages and the confidence intervals for series of 50 and 25 intervals.

Table 2 shows that in sequences of both lengths, the ApEn measure tends to yield a larger value the more the sequence is positively biased; the larger the negative bias, the closer the value is to zero. The .95 confidence intervals around the mean also show that the more negative the bias, the narrower the distribution is, but that could be a floor effect.

Nevertheless, the data in this table suggest that the ApEn measure allows a distinction between negatively and positively biased sequences. However, a distinction between truly random and biased sequences does not seem to be completely feasible. This suggestion was tested by means of the d' measure of the theory of signal detection (Swets, 1986; Swets, Tanner, & Birdsall, 1961; Van der Goten & Vandierendonck, 1997). For each distribution, the proportion of measures falling above the mean of the unbiased distribution was calculated and used to estimate d'. Table 3 displays the results of this calculation for both sequence lengths.

Certainly for the shorter sequences, which is the most frequent kind of sequence encountered in dual-task research, d' is rather small, except when the sequence has an outspoken negative bias. In the sequences of length 50, discrimination is clear for moderately and strongly biased negative sequences. Weakly biased sequences and positively biased sequences cannot easily be distinguished from random ones.

It is clear from this Monte Carlo study that the ApEn measure is more useful than the calculation of lag correlations, at least for strongly negatively biased sequences. A more sensitive measure remains desirable.

Table 2
Mean and .95 Confidence Interval
of the Approximate Entropy Measures for
Biased and Unbiased Sequences of Lengths 50 and 25

Confidence Interval	Negative Bias			No Bias	Positive Bias		
	.75	.50	.25	(.00)	.25	.50	.75
Sequences of 50							
Mean	0.041	0.070	0.114	0.117	0.145	0.170	0.156
Lower bound	0.013	0.030	0.058	0.056	0.077	0.085	0.050
Upper bound	0.069	0.110	0.169	0.178	0.213	0.256	0.261
Sequences of 25							
Mean	0.060	0.144	0.193	0.191	0.210	0.199	0.142
Lower bound	0.000	0.035	0.079	0.079	0.087	0.058	0.000
Upper bound	0.151	0.252	0.307	0.303	0.334	0.340	0.304

Table 3
Estimates of d' on Approximate Entropy Distributions,
Displayed as a Function of Negative and Positive
Sequence Bias for Sequences of Lengths 50 and 25

Length	Negative Bias			Positive Bias		
	.75	.50	.25	.25	.50	.75
50	−3.29	−2.05	−0.18	0.76	1.23	0.67
25	−2.29	−0.81	0.04	0.29	0.04	−0.62

ANALYSIS ON THE BASIS OF BIAS MODELS

A quite different approach is directly based on the bias models described in the introduction. This approach assumes that a sequence of constant time intervals underlies the tapping behavior in a random time interval generation task, so that each of these intervals either does or does not contain a tap. If it is possible to estimate the length of this unit interval, this information can be used to estimate the parameters (π, α, and β) that correspond to the generated sequence and to infer which generation model (no bias, positive bias, or negative bias) shows the closest fit to the sequence. Furthermore, the extent and the orientation of the bias result in deviations from statistical independence, which can be measured by a number of statistics, such as the runs statistics, the autocorrelation statistic, and the alternation index. The runs statistic for a run length of k (Sternberg, 1959) is defined as:

$$r_k = \frac{1}{N-k+1} \sum_{i=1}^{N-k+1} x_i x_{i+1} \cdots x_{i+k-1} \qquad (6)$$

For length 3, for example, this statistic expresses the proportion of consecutive triples in the sequence in which all three events have an outcome of 1. The autocorrelation at Lag k (Sternberg, 1959) is defined as

$$c_k = \frac{1}{N-2} \sum_{i=1}^{N-k} x_i x_{i+k}. \qquad (7)$$

An autocorrelation of Lag 2, for example, expresses, within the pairs that are separated by one position, the proportion in which both members of the pair have an outcome of 1. Finally, the alternation index simply looks at consecutive events and takes the proportion of event pairs with alternating outcomes (1 and 0 or 0 and 1) to the total number of pairs:

$$P_a = \frac{1}{N-1} \sum_{i=1}^{N-1} [x_i(1-x_{i+1}) + (1-x_i)x_{i+1}]. \qquad (8)$$

In a statistically independent sequence, the value of P_a is expected to be close to .5; when there is a strong positive bias, there are few alternations, and P_a is small. If, on the contrary, there is a negative bias, the proportion of alternations is high, and the stronger this bias, the closer Pa comes to 1.

In the remainder of this section, a method will be described to convert a sequence of taps over time into a binary sequence. On the basis of this sequence, the values

of the statistics defined in Equations 6–8 are calculated; the parameters of the three models are estimated and used to predict the values of the statistics. By comparing predicted and observed values, it is possible to estimate the degree of fit of each of the three models.

The first step in this methodology consists of the estimation of the length of the unit time interval and the conversion of the time intervals to a binary sequence. This is achieved by subjecting the intervals to a hierarchical clustering analysis based on a nearest neighbor procedure and then fine-tuning the obtained result. The purpose of this step is to find a unit interval that preserves the structure in the series as much as possible. For example, if taps were produced at 200, 500, 700, 1,000, ... msec, the unit interval recovered would be 100 msec, and the corresponding binary sequence is 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, ... If, the interval size were 200 msec, the structure would not be so nicely captured, as appears from the resulting binary sequence 1, 0, 1, 1, 1, ...

This conversion procedure works as follows. Initially, the n intertap intervals are the clusters. A first reduction is achieved by merging all the equal intervals. Next, in an iterative loop, the two nearest clusters are merged until the shortest distance between two clusters exceeds 100 msec or until all the clusters have been merged. This result is then further refined by maximization of a function of several variables—namely, the correlation between the original sequence and the recovered one, the deviation between the original and the recovered intervals, and the proportion of keypresses recovered. All three variables measure a particular aspect of the degree of correspondence between the recovered and the observed sequence. This search is performed in the range between $.25u$ and $1.5u$, where u is the interval size obtained in the clustering procedure.

In the second step, the three models presented in Equations 1–4 can be estimated, and the independence statistics (Equations 6–8) can be calculated. The obtained parameter estimates can then be used to generate predictions for the values of the statistics and to estimate the degree of correspondence between the model and the data. To that end, the model parameters must be expressed in terms of observed proportions. Considering any pair of consecutive intervals, there are four possibilities: no tap followed by no tap with a probability p_{00}, no tap followed by a tap (p_{01}), a tap followed by no tap (p_{10}), and a tap followed by a tap (p_{11}). On the basis of these four transition probabilities, two proportions are defined as follows:

$$p_0 = \frac{p_{00}}{p_{00} + p_{10}} \quad (9)$$

and

$$p_1 = \frac{p_{11}}{p_{11} + p_{01}}. \quad (10)$$

The model parameters can then be expressed as follows:

$$\hat{\pi} = \begin{cases} \dfrac{1 - p_0}{2 - p_0 - p_1} & \text{if } 2 - p_0 - p_1 > 0 \\ .5 & \text{otherwise.} \end{cases} \quad (11)$$

It can be shown that the π-parameter of the perseveration (positive bias) model can be estimated in the same way; α can then be estimated by

$$\hat{\alpha} = \max(0, p_0 + p_1 - 1). \quad (12)$$

In the context of the alternation (negative bias) model, the parameter β is estimated in a similar way:

$$\hat{\beta} = \max(0, 1 - p_0 - p_1). \quad (13)$$

The other parameter, π, is then estimated by

$$\hat{\pi} = \begin{cases} \dfrac{p_1}{p_0 + p_1} & \text{if } p_0 + p_1 > 0 \\ .5 & \text{otherwise.} \end{cases} \quad (14)$$

Comparing Equations 12 and 13, it is transparent that α and β are each other's opposite, but when one of them is larger than 0, the other one is 0. Expressions have been derived for each of the statistics in terms of the parameters of the three models. These expressions can be used to derive the predictions of the models given the best-fitting parameters (as displayed in Equations 11–14). This yields 21 values (10 run lengths, 10 autocorrelations, and the alternation index). Per statistic, or overall, a degree of fit can be calculated on the basis of the root-mean squared deviations (RMSD). In general, the model yielding the smallest RMSD is closest to the data. Yet, comparison of the models is not so straightforward. Since the bias models have two free parameters, as compared with only one free parameter for the no-bias model, it is easier for the bias models to fit the data.

To find out whether the procedure is indeed biased toward the two-parameter models, a Monte Carlo study was performed, which also tested the validity of the procedure. First, binary sequences of 50 events were generated on the basis of the no-bias model (50 sequences with π of .25, .50, and .75), the perseveration model (50 sequences with the same π levels combined with three levels of α—namely, .25, .50, and .75), and the alternation model (50 sequences with the same π levels combined with three levels of β: .25, .50, and .75). In addition, 50 sequences based on fixed repetitions of five, seven, or nine units were also produced (e.g., by repetition of the five-unit sequence 1, 1, 0, 0, 1 over and over again). Thus, simple repetitive rhythms were obtained. In all, 1,200 sequences were generated and the values of the 21 statistics were calculated, the model parameters were estimated, and the best fits were inferred. Next, these 1,200 sequences were converted to time intervals, by taking a unit interval of 200 msec. These times were then fed to the program Rig-Anal described in the Appendix.

The central question was whether and to what extent the recovered binary sequences differed from the original binary sequences in terms of their statistical properties. To that end, the model parameters and the goodness of fit of the three models were pair-wise compared in the two sets of sequences. Table 4 shows the average deviation between characteristics of the initial and the recovered binary sequences. The differences are in general rather small, and they tend to be somewhat larger for the systematic series. But even in this category, the deviations are rather small.

The recovered binary sequences showed the no-bias model to be the best-fitting one in 108 (72%) of the 150 sequences generated by that model; of 450 perseveration sequences, 90% were attributed to the perseveration model, and of the 450 alternation sequences, 98% were assigned to the alternation model. In the 150 rhythm sequences, 54% conformed to the alternation model, and in 40% the no-bias model was the best-fitting one. The procedure used to make these decisions was based on the following principles: (1) The no-bias model was taken as the best-fitting model whenever the two other models did not apply; (2) the perseveration model was taken to be correct if $\hat{\alpha} > .005$ and the RMSD for this model was smaller by at least .005 than the RMSD of the other models; and (3) with $\hat{\beta} > .005$ and an RMSD smaller by at least .005, the alternation model was taken to be the correct one.

This Monte Carlo study shows that the procedure recovers 91% of the 1,050 test sequences based on one of the models. In general, as Table 4 shows, the deviations were rather small. There is indeed a tendency for the two-parameter models to be the best-fitting ones more often than the no-bias model. This is, to a large extent, due to the fact that small deviations from the no-bias model are easily accounted for by one of the other two models.

It is clear from this overview that this model-based approach distinguishes quite well between nonbiased and biased sequences. For sake of completeness, d' measures

Table 4
Average Differences Between Generated and Inferred Binary Sequences as a Function of Sequence Type

Variable	No bias ($n = 150$)	Perseveration ($n = 450$)	Alternation ($n = 450$)	Systematic ($n = 150$)
Interval	0.000	0.833	0.000	1.333
Recovery	0.000	0.000	0.000	0.000
π	0.004	0.013	0.003	0.011
α	0.006	0.018	0.000	0.006
π_a	0.010	0.008	0.018	0.011
β	0.010	0.003	0.012	0.031
$RMSD_n$	0.002	0.006	0.001	0.004
$RMSD_p$	0.002	0.006	0.001	0.004
$RMSD_a$	0.002	0.003	0.001	0.003

Note—The estimate of π is different in the context of the alternation model than in the context of the other two models; π_a is the estimate under the alternation model. The letters n, p, and a indexing the root-mean squared deviation (RMSD) values refer, respectively, to the no-bias, the perseveration, and the alternation models.

were calculated on the proportions of sequences of each type that were recovered as being not biased, positively biased, or negatively biased. For example, 108 of the 150 nonbiased sequences were recovered as not biased, whereas only 107 of the 1,050 other sequences were classified as not biased; this corresponds to a d' of 1.85. Similarly, the contrasts of positively biased versus other sequences, negatively biased versus other sequences, positively versus negatively biased sequences, and nonbiased versus systematic sequences yielded d' values of 3.33, 3.70, 5.39, and 0.84, respectively.

The program RigAnal (see the Appendix), was developed to calculate the values of the statistics, estimate the parameters, calculate the model predictions, and estimate the degree of fit of each of the three models.[3]

A COMPARISON OF THE MEASURES

The three methods discussed yield a multiplicity of measures that can be used as indices of randomness. Some of these measures may be better indices than others, but the question remains as to what extent they estimate a common concept. In order to achieve a better insight into the commonalities underlying these measures, a principal components analysis was performed on 19 measures taken over 1,381 random sequences collected in single- and dual-task conditions in a series of experiments. The measures used were the following: (1) ApEn with vectors of length 1 and length 2 calculated on the raw intervals; (2) the four parameters of the three models; (3) Runs 1–3, Autocorrelations 2–5, and alternation index (note that runs of length 2 and autocorrelations of Lag 1 are the same); (4) lag correlations of Lags 1–5.

Table 5 displays the component loadings after applying an equamax rotation. The structure is interesting.[4] A first component attracts high loadings on the variables π, α, π_a, r_1–r_3, and c_2–c_5. All these indices are sensitive to the probability of a tap and/or the tendency to perseverate. The second component shows high loadings of ApEn-2, π, β, r_1, c_2–c_5, and P_a. Because of the very high positive loadings of the β parameter and of the alternation index, it may be said that this component indicates the tendency toward alternation. Indeed, a strong alternation tendency is associated with short runs (so only u_1) and strong autocorrelations at longer lags (hence c_2–c_5). The rather high loading of the π parameter can be understood from the fact that when there is alternation, α is zero, and consequently π should correlate with the tendency to alternate. Finally, the ApEn-2 measure also seems to capture part of the alternation tendency, which is expressed in a moderate positive loading on this "alternation" component. The remaining three components essentially capture variance associated with the five lag correlations and with the ApEn-1 measure. The third component exhibits high loadings of lag correlations 2–4, whereas lag correlations 4–5 show a high loading on the fourth component and lag correlations 1–2 and ApEn-1 load mainly on the fifth component.

Table 5
Rotated Principal Components of 19 Measures
Based on a Sample of 1,381 Random and Fixed Sequences
Obtained in Single-Task and Dual-Task Conditions

Measure	Components				
	1	2	3	4	5
ApEn (1)	−.05	.19	.06	.18	**−.63**
ApEn (2)	.15	**.47**	−.17	−.20	.06
π	**.64**	**.76**	.04	−.03	−.02
α	**.40**	−.29	−.05	−.15	.12
π_a	**.87**	.29	−.03	−.08	.01
β	−.19	**.92**	.13	.06	−.05
u_1	**.63**	**.76**	.04	−.03	−.02
u_2	**.98**	.09	−.03	−.07	.03
u_3	**.93**	−.08	−.03	−.09	.07
c_2	**.59**	**.72**	−.05	−.08	.04
c_3	**.72**	**.47**	.08	−.03	−.03
c_4	**.63**	**.64**	.05	−.01	.06
c_5	**.78**	**.50**	.03	−.02	.01
P_a	.05	**.97**	.09	.03	−.05
Lag_1	−.06	.16	.09	.27	**.70**
Lag_2	.01	.13	**.60**	−.11	**.45**
Lag_3	.00	−.04	**.78**	.01	−.02
Lag_4	.02	−.06	**.40**	**.66**	−.26
Lag_5	.02	−.06	−.30	**.80**	.20
Percent variance	29	25	7	7	6

Note—Component loadings ≥ .4 are shown in boldface to elucidate the structure.

The first two components explain 54% of the variance and essentially capture the variability associated with the model parameters, the dependency statistics, and the ApEn-2 measure. The remaining three components explain 20% of the variance and are related to the lag correlations and the ApEn-1 measure. Interestingly, Table 5 illustrates that the indices that have important loadings on the first two components do not load on the last three components and vice versa.

Given these interpretations, the following conclusions can be formulated. (1) The three groups of measures (lag correlations, model-based statistics and parameters, and entropy) constitute two sets of performance indicators— namely, output probability, perseveration, and alternation (Components 1–2), on the one hand, and correlational structure (Components 3–5), on the other. (2) The probability of a tap in the sequence of intervals, the tendency toward alternation, the tendency toward perseveration, and the degree of sequence complexity (ApEn-2) are all clearly present in the component structure (Components 1 and 2). (3) It is not at all clear which aspects of randomness are tapped by the lag correlations, but whatever aspects are measured, the lag correlations do not seem to be related to any of the other measures (correlations between −.12 and .13). This indicates that the lag correlations are not sensitive to the same aspects of randomness as the model-based and the ApEn-2 indices. It may be suspected therefore that, at best, the lag correlations are measures sensitive both to variations in randomness and to other kinds of temporal structure that do not seem to be related to event probability, repetition proba-

bility, or alternation probability. At worst, lag correlations are not measures of randomness at all.

DISCUSSION

Three methods were described for the analysis of the randomness of time intervals. The findings of the principal components analysis highlight the differences between the three methods. On the basis of theoretical considerations combined with the results of this analysis, it would seem that the parameters α and β and the alternation index are the best measures to indicate deviations from randomness.

The principal components analysis has revealed that the measures derived from the three methods can be segregated in two sets. ApEn-2 and the model-based measures together constitute two components that explain more than half of the variance. The lag correlations and ApEn-1 explain another 20%, distributed over three components.

The two most important components in the obtained component structure seem to be sensitive to alternations and to perseverations each time in combination with the global probability of a tap. A third final component is related to lag correlations. It is quite isolated from the rest of the measures. In fact, the lag correlations do not at all correlate with the other measurements (most correlations are around zero, and the highest one observed is −.20). It seems clear, therefore, that lag correlations have nothing to do with measurement of regularity or of randomness. The last component, which corresponds to the entropy measure, is at the low end sensitive to restricted regularities, but at the high end it does not differentiate between randomness and regularities of another form.

Taken altogether, the present data clearly show that the model-based measures come closest to measuring randomness in time sequences. The method is quite laborious, but the program presented in the Appendix may help to circumvent this difficulty. Even then, the method yields a variety of measures (4 parameters, 21 statistics, 63 deviations between the observed and the predicted statistics, and a number of goodness-of-fit measures). Although the best way to characterize the randomness of a sequence is on the basis of the parameters of the best-fitting model, the alternation index seems to provide a valuable shortcut. If its value is smaller than .5, it expresses a tendency towards perseveration; if its value is larger than .5, it shows a tendency toward alternation, and values near to .5 indicate absence of bias.

It should be pointed out, though, that estimation of the randomness of a sequence of events is only a tool to elucidate the statistical characteristics of the observed sequence. Inferences about the underlying generation process are just speculative. This means that the proper context for usage of the tool is always in a comparative situation—for example, to compare the statistical qualities of the sequences generated under single-task and dual-task conditions, to compare the statistical proper-

ties of sequences generated under instructions of random tapping and fixed or systematic tapping (for examples of such applications, see Vandierendonck, 2000), or to find out whether there is a tradeoff between the primary and the secondary tasks.

Applications of the method to a series of such comparisons has shown that, in general, human random time interval generation is biased toward alternation and that this tendency increases if the primary task load is larger (Vandierendonck, 2000). Under instructions of fixed tapping (generate taps about once a second), the alternation tendency is even bigger than under random tapping with a heavy primary task load.

It is quite likely that other useful methodologies for the analysis of random time intervals are possible. The present article, therefore, may be considered a first step toward an improvement of the methodology.

REFERENCES

BUDESCU, D. V. (1985). Analysis of dichotomous variables in the presence of serial dependence. *Psychological Bulletin*, **97**, 547-561.

BUDESCU, D. V. (1987). A Markov model for generation of random binary sequences. *Journal of Experimental Psychology: Human Perception & Performance*, **13**, 25-39.

CRAWFORD, J. R. (ED.) (1998). *Assessment of attention and executive functions*. Hove, U.K.: Psychology Press.

CREVITS, L., CLAEYS, K., STUYVEN, E., VAN DER GOTEN, K., & VANDIERENDONCK, A. (1997). Saccadic behaviour and working memory in normal subjects and prefrontal patients. *Neuro-Ophthalmology Supplement*, **150**, S126.

DE RAMMELAERE, S., STUYVEN, E., & VANDIERENDONCK, A. (1999). The contribution of working memory resources in the verification of simple mental arithmetic sums. *Psychological Research*, **62**, 72-77.

DE SOETE, G., & VANDIERENDONCK, A. (1982). On the use of the jackknife and the bootstrap for estimating a confidence interval for the product-moment correlation coefficient. *Psychologica Belgica*, **22**, 87-97.

EDGINGTON, E. S. (1969). *Statistical inference: The distribution-free approach*. New York: McGraw-Hill.

EFRON, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, **7**, 1-26.

HORNERO, R., ALONSO, A., JIMENO, N., JIMENO, A., LÓPEZ, M. (1999). Nonlinear analysis of time series generated by schizophrenic patients. *IEEE Engineering in Medicine & Biology*, **18**, 84-90.

JIMENO, N., JIMENO, A., ALONSO, A., SEGOVIA, E., & VARGAS, M. L. (1996). Basic symptoms and cognitive dynamic disorders in schizophrenic patients. *Neurology, Psychiatry & Brain Research*, **4**, 171-178.

KAREEV, Y. (1992). Not that bad after all: Generation of random sequences. *Journal of Experimental Psychology: Human Perception & Performance*, **18**, 1189-1194.

LEMAIRE, P., ABDI, H., & FAYOL, M. (1996). The role of working memory resources in simple cognitive arithmetic. *European Journal of Cognitive Psychology*, **8**, 73-103.

LOPES, L. L. (1982). Doing the impossible: A note on induction and the experience of randomness. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **8**, 626-636.

LOPES, L. L., & ODEN, G. C. (1987). Distinguishing between random and nonrandom events. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **13**, 392-400.

MARTEIN, R., KEMPS, E., & VANDIERENDONCK, A. (1999). The role of working memory in a content-location span task. *Psychologica Belgica*, **39**, 15-27.

NEURINGER, A. (1986). Can people behave "randomly?": The role of feedback. *Journal of Experimental Psychology: General*, **115**, 62-75.

PINCUS, S. M. (1991). Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, **88**, 2297-2301.

RABBITT, P. (1997). *Methodology of frontal and executive function*. Hove, U.K.: Psychology Press.

RAPOPORT, A., & BUDESCU, D. V. (1992). Generation of random series in two-person strictly competitive games. *Journal of Experimental Psychology: General*, **121**, 352-363.

SLIFKIN, A. B., & NEWELL, K. M. (1999). Noise, information transmission, and force variability. *Journal of Experimental Psychology: Human Perception & Performance*, **25**, 837-851.

STERNBERG, S. H. (1959). A path-dependent linear model. In R. R. Bush & W. K. Estes (Eds.), *Studies in mathematical learning theory* (pp. 308-339). Stanford: Stanford University Press.

STUYVEN, E., & VAN DER GOTEN, K. (1995). Stimulus independent thoughts and working memory: The role of the central executive. *Psychologica Belgica*, **35**, 241-251.

STUYVEN, E., VAN DER GOTEN, K., VANDIERENDONCK, A., CLAEYS, K., & CREVITS, L. (2000). Saccadic eye movements under conditions of cognitive load. *Acta Psychologica*, **104**, 69-85.

SWETS, J. A. (1986). Indices of discrimination or diagnostic accuracy: Their ROCs and implied models. *Psychological Bulletin*, **99**, 100-117.

SWETS, J. A., TANNER, W. P., JR., & BIRDSALL, T. G. (1961). Decision processes in perception. *Psychological Review*, **68**, 301-340.

TOWSE, J. N., & NEIL, D. (1998). Analyzing human random generation behavior: A review of methods used and a computer program for describing performance. *Behavior Research Methods, Instruments, & Computers*, **30**, 583-591.

TREISMAN, M., & FAULKNER, A. (1987). Generation of random sequences by human subjects: Cognitive operations or psychophysical process? *Journal of Experimental Psychology: General*, **116**, 337-355.

VAN DER GOTEN, K., & VANDIERENDONCK, A. (1997). Software to calculate measures of sensitivity and response bias based on detection theory and threshold theory. *Behavior Research Methods, Instruments, & Computers*, **29**, 461-463.

VANDIERENDONCK, A. (2000). Bias and processing capacity in generation of random time intervals. *Cognitive Science Quarterly*, **1**, 205-233.

VANDIERENDONCK, A., DE VOOGHT, G., & VAN DER GOTEN, K. (1998). Does random time interval generation interfere with working memory executive functions? *European Journal of Cognitive Psychology*, **10**, 413-442.

VANDIERENDONCK, A., KEMPS, E., & FASTAME, C. (2000). *Working memory components involved in the Corsi block test*. Unpublished manuscript.

WAGENAAR, W. A. (1970a). Appreciation of conditional probabilities in binary sequences. *Acta Psychologica*, **34**, 348-356.

WAGENAAR, W. A. (1970b). Subjective randomness and the capacity to generate information. *Acta Psychologica*, **33**, 233-242.

WAGENAAR, W. A. (1972). Generation of random sequences by human subjects: A critical review of the literature. *Psychological Bulletin*, **77**, 65-72.

WIEGERSMA, S. (1984). High-speed sequential vocal response production. *Perceptual & Motor Skills*, **59**, 43-50.

ZAHN, H.-E. (1982). Compulsion towards order: Man as random generator. *Studia Psychologica*, **24**, 241-249.

NOTES

1. This assumption may seem rather restrictive. However, it should be noted that it is an assumption about the generation process. Even if the present event directly depends on the previous outcome only, it should be kept in mind that the previous outcome was directly affected by the one before that. Even with a rather moderate alternation probability ($\pi = \beta = .5$), autocorrelations deviate from the neutral baseline up to six lags. To the extent that a comparison is warranted, this is consistent with the observation by Wiegersma (1984) that alternation extends beyond the directly succeeding item in random digit production. There seems to be no reason yet to trade the Markov assumption for a more complex one.

2. A related technique based on chaos theory has been described by Hornero, Alonso, Jimeno, Jimeno, and López (1999) and was used to show that rhythmic random pattern tapping in schizophrenic patients is less variable than that of matched controls. This measure was not included in the present study because the boundary conditions of the tech-

nique (generation of a series consisting of a fixed number of taps) seem to be too limited to be used in dual-task conditions. More information about the utility of the measure under conditions of variable duration is, however, most welcome.

3. The procedure described here has been improved in comparison with the variant used in an earlier publication (Vandierendonck et al., 1998). Essentially, the conversion to binary values has been improved. As a consequence, values for runs and autocorrelations tend to take lower values. The published data were reanalyzed with the method as described here. Although not every individual finding could be con-

firmed, the general line of findings remains the same. An extension of these findings can be found in Vandierendonck (2000).

4. One may wonder whether the component structure was affected by mixing data obtained under single- and dual-task conditions. Of the 1,381 sequences included in the analysis, only 196 were obtained under single-task conditions. Separate analyses for single-task and dual-task conditions do not alter the picture presented here. However, when sequences are analyzed that were obtained under fixed tapping instructions, the picture is slightly different, probably owing to a lower degree of complexity in the latter sequences.

APPENDIX
RigAnal: A Program to Calculate Randomness Measures

The program implements the three methods discussed in the present article. It reads the file with time sequences and produces an output file with calculated measures per sequence. The program is developed for two different platforms—namely, the Unix/Linux platform and the MS Windows platform. Only the interface is different between these two forms.

This appendix presents a brief description of the program and its usage. First, the two interfaces are briefly described. Next, input and output file formats are described. And finally, an example of input and output is presented.

The Windows Interface

The interface is Windows based. Upon start, an empty window is shown with a bar menu at the top, showing the choices "Initialize," "Finalize," and "Information." The leftmost menu (Initialize) allows one to specify the procedure (lag, entropy, or model based), the options, the input file, and the output file by means of appropriate dialog boxes. The input and output file dialog boxes provide a dialog to enter the name of the file. Radio buttons are used in the procedure dialog box to select the major application (lag correlation, approximate entropy, or model-based statistics) and, in the options dialog box, to specify the model-based output (beautified or regular output) and check buttons to indicate the specific selections required.

The second main menu allows two choices. One is to run the program, and this will only work after the initializations (specification of options and files) have been done. The other choice is for quitting the program.

Version and other information about the program can be probed by selecting the Information menu.

The Unix/Linux Interface

The program is called from the command line, as follows:

```
riganal -l infile outfile
riganal -e infile outfile
riganal -a|b|i|r|s [-cuxz123gpm] infile outfile
riganal -v
```

The last form (-v) just reports the revision information of the program. The first form (-l) calculates lag correlations at Lags 1–10 and estimates the probability and the confidence intervals on the basis of 1,000 random permutations on each sequence given in the input file. The results are printed in the output file according to the format displayed in Table A1.

The second form (-e) calculates the ApEn measure (Pincus, 1991) for vector lengths of 1 and 2 on each sequence given in the input file. The output file consists of one line per sequence.

The third form calculates the model-based measures. The options refer to specific applications: Print all statistics in a line per sequence (-a), produce beautified output as shown in Table A2 (-b; default), produce only the binary conversion of the time intervals (-i), assume that the input is binary and calculate the measures specified in the second set of options (-r), and calculate selected measures as specified in the other options (-s).

The other options allow one to select only specific measures: the model parameters (-p), the overall model fits (-m), all the statistics (-z), the runs (-u), the autocorrelations (-c), the alternation index (-x), the deviations between the predictions of the first model and the statistics (-1), the same for the second model (-2), the same for the third model (-3), and the detailed fits of all

Table A1
Output of the Lagcor Option of the Program
as Applied to the Example Mentioned in the Appendix

Permutation Statistic on Lag Correlation (Revision: 1.4)

Lags from 1 to 10
Number of replications per test = 1000

Subject Observation		Lag	r	P = .025	Confidence interval P = .500	P = .975	Probability	Deviation
15	1	1	0.3005	-0.4513	-0.0768	0.4150	0.9430	0.1423
15	1	2	-0.0597	-0.4725	-0.0734	0.4377	0.5180	0.0002
15	1	3	-0.4139	-0.4973	-0.0699	0.4612	0.0630	0.1184
15	1	4	-0.5975	-0.4958	-0.0723	0.4592	0.0040	0.2758
15	1	5	-0.3188	-0.4993	-0.0716	0.5469	0.1590	0.0611
15	1	6	-0.0813	-0.5343	-0.0454	0.5703	0.4620	0.0013
15	1	7	0.5792	-0.5231	-0.0392	0.5728	0.9770	0.3825
15	1	8	0.5812	-0.5493	-0.0516	0.6613	0.9580	0.4004
15	1	9	0.4823	-0.5831	-0.0426	0.6597	0.9170	0.2755
15	1	10	-0.0015	-0.6163	-0.0084	0.7093	0.5060	0.0000

Note—The program was instructed to calculate correlations and confidence intervals for Lags 1–10.

the models (-g). Several of these options can be specified as a concatenated string. For example, the option -pmx asks for the parameters, the overall fits, and the alternation index.

Files

Except for usage with the -r option, the input file in all cases consists of any number of sequences presented one sequence per line. The first three values of each line specify the participant identification, the condition identification, and the task identification. These values always return in the output. The fourth value is the number of intervals in the sequences—say, *n*—and after that value, *n* other cumulative time values follow.

The input for the -r option is a file with binary sequences. The first three values are the same. Then follows the size of the interval (integer), the number of binary events (*m*), and the value of the recovery index (floating point) followed by *m* binary values (0 or 1). When the option -i is specified, the output file produced follows these same rules.

Application with options -a, -r, or -s produce output with all the requested values on a single line. The first three values in that output repeat the participant, condition, and task identification. Next, the interval size (for the binary conversion) and the recovery index are printed. After that, the statistics are printed in a standard order (conform to what was requested): autocorrelations (-c or -z), runs (-u or -z), alternation index (-x or -z), deviations between the statistics and the values predicted by the first (unbiased) model in the same order as the statistics, deviations between the statistics and the values predicted by the second (perseveration) model in the same order as the statistics, deviations between the statistics and the values predicted by the third (alternation) model in the same order as the statistics, the fits per group of statistics and per model (-g), the model parameters (-p), and finally the overall model fits (-m). This print out format is useful for further statistical analysis.

An Example

The following short series of time intervals (in milliseconds) is used as an example: 218, 418, 985, 852, 1065, 262, 106, 113, 256, 748, 322, 1397, 846, 168, 169, 231, 301, 386, 612. Note that these values are not suited as input for the program, since cumulative values (0, 218, 636, etc.) are required. Table A1 displays the result obtained by application of the lag correlations method. The first part of the table displays the general information for this application. The main part of the table reports the correlations at Lags 1–10. The first two values in the row identify the participant and the sequence in the analysis. Each row also displays the confidence interval ($P = .025$ and $P = .975$), as well as the median of the sampling distribution ($P = .500$). Under the header "Probability," the probability of the correlation in the sampling distribution is given. The last column shows the value of the squared deviation between the observed correlation and the mean of the distribution. It appears that in this series, only the correlation at Lag 4 falls outside the .95 confidence interval; the other nine values lie inside their respective confidence intervals, although the values at Lags 3, 7, and 8 are quite close to the edges of the confidence interval. Given these results, one may be tempted to conclude that the series of intervals is not at all random. However, as appeared from the Monte Carlo study, the probability of observing one extreme probability in one sequence is about the same in biased and unbiased sequences. On the other hand, the sequence used here as an example is rather short, so the correlations have generally, and certainly at the larger lags, wide confidence intervals because they are calculated on small samples.

Application of the approximate entropy option of the program yields two values—namely, .298 for vectors of length 1 and .043 for vectors of length 2. Both values are quite close to zero, so that it may be concluded that this sequence has a rather simple structure that could indicate a substantial deviation from randomness.

Table A2
Output of the Rigstat Program as Applied to the
Example Mentioned in the Appendix

Subject 15—Condition 2—Task 24—Interval 142—Recuperation 1.000
Model 1. Estimated pi = 0.287
Model 2. Estimated pi = 0.287—Estimated alpha = 0.000
Model 3. Estimated pi = 0.244—Estimated beta = 0.090

Autocorrelations		Model 1		Model 2		Model 3	
Lag	Obs	Exp	Dev	Exp	Dev	Exp	Dev
1	0.061	0.082	−0.022	0.082	−0.022	0.063	−0.002
2	0.108	0.082	0.026	0.082	0.026	0.083	0.025
3	0.078	0.082	−0.004	0.082	−0.004	0.081	−0.003
RMSD C			0.022		0.022		0.021

Runs		Model 1		Model 2		Model 3	
Lag	Obs	Exp	Dev	Exp	Dev	Exp	Dev
1	0.284	0.287	−0.003	0.287	−0.003	0.284	0.000
2	0.061	0.082	−0.022	0.082	−0.022	0.063	−0.002
3	0.031	0.024	0.007	0.024	0.007	0.014	0.017
RMSD U			0.013		0.013		0.010

A-index	0.439	0.409	0.030	0.409	0.030	0.426	0.013
RMSD T			0.017		0.017		0.015

Note—The output contains a general section and a section for each group of statistics. The predictions and the fits of the models are also included. The program normally displays 10 autocorrelation lags, but to keep the size of the table manageable, only 3 lags are displayed here. The root-mean squared deviations (RMSDs) are based on all the data used by the program.

APPENDIX (Continued)

The model-based method was also applied to the example. The beautified output is displayed in Table A2. The top panel of the table displays general information, such as the identification of participant, condition, and task. This section also shows the size of the unit interval (142 msec in this case) and the degree of recuperation (1 here, which means that no interval contained more than one tap). Also shown in this panel are the values of the estimated parameters. Under the no-bias and the positive bias models, $\pi = .29$ and $\alpha = 0$, which means that there is no positive bias in this sequence. On the contrary, there is a small negative bias ($\beta = .09$) combined with a π of .24.

The second part of the table presents the values for the autocorrelations at Lags 1–4 (the program calculates and prints Lags 1–10). In the first column, the lag is indicated, and the second column displays the observed autocorrelation. In the following columns, for each of the three models, the expected autocorrelation (given the estimated parameters) is shown, as well as the difference with the observed value. This section concludes with the RMSD calculated for each model over the 10 autocorrelation lags. A small difference in favor of the alternation model is observed here.

The third part of the table displays the results for the runs statistic. Only the nonzero values of the runs statistic are presented—here, runs of lengths 1–3. The columns in the table have the same meaning as the columns in the part with the autocorrelation statistics. Again, the section ends with an RMSD calculated over the 10 run lengths. With respect to this set of statistics, the difference between the alternation model and the other two models is more pronounced.

The last part of the table displays the results for the alternation index. Again, the best fit is observed for the alternation model. The final line, then, presents the overall RMSD calculated over all 21 statistics. As this overall RMSD is somewhat smaller for the alternation model than for the other models and the alternation parameter is clearly different from zero, it would seem appropriate to conclude that the sequence in this analysis is slightly biased toward alternation.

If the program was run under the -s (selected output) option with -p as selection, the output would consist of just one line,

15 2 24 142 1.000 0.287 0.000 0.244 0.090,

where the first three values are identifications, the fourth value is the unit interval, the fifth is the recuperation index, and the last four values are the four model parameters.