Subjective randomness as statistical inference

Thomas L. Griffiths

Department of Psychology, University of California, Berkeley

Dylan Daniels

Department of Statistics, University of California, Berkeley

Joseph L. Austerweil

Department of Psychology, University of Wisconsin-Madison

Joshua B. Tenenbaum

Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology

Author Note

Preliminary results from Experiment 2 and the analysis of Falk and Konold (1997) and Lopes and Oden (1982) were presented at the Annual Conference of the Cognitive Science Society and the Neural Information Processing Systems conference (Griffiths & Tenenbaum, 2003; 2004).

Abstract

Some events seem more random than others. For example, when tossing a coin, a sequence of eight heads in a row does not seem very random. Where do these intuitions about randomness come from? We argue that subjective randomness can be understood as the result of a statistical inference assessing the evidence that an event provides for having been produced by a random generating process. We show how this account provides a link to previous work relating randomness to algorithmic complexity, in which random events are those that cannot be described by short computer programs. Algorithmic complexity is both incomputable and too general to capture the regularities that people can recognize, but viewing randomness as statistical inference provides two paths to addressing these problems: considering regularities generated by simpler computing machines, and restricting the set of probability distributions that characterize regularity. Building on previous work exploring these different routes to a more restricted notion of randomness, we define strong quantitative models of human randomness judgments that apply not just to binary sequences – which have been the focus of much of the previous work on subjective randomness – but also to binary matrices and spatial clustering.

Keywords: randomness; Bayesian inference; algorithmic complexity

Subjective randomness as statistical inference

Imagine you are driving and approach a stoplight. You see eight cars in front of you, all of which are black, and begin to wonder why all of these vehicles are the same color. At the next stoplight you see eight cars all of varying colors and perceive the situation as happenstance, thinking nothing of it. These two events strike us differently because of our intuitions about randomness. The second event seems clearly a result of chance, but the first event seems to suggest another explanation. But where do these intuitions about randomness come from?

One naïve explanation purports that non-random events are simply low-probability events, but multiple studies have confirmed that humans do not judge equally likely events as equally random (Kahneman & Tversky, 1972; Tversky & Kahneman 1974). One of the classic examples concerns flips of an fair coin: Asked to choose which of the following coin flip sequences of length eight is more likely to occur, `HHHHHHHH` or `HTHTTHTT`, most people will choose the latter, despite the fact that each sequence has the same probability of occurring, $\left(\frac{1}{2}\right)^8$. This intuition is surprisingly strong. Indeed, even trained statisticians would likely be surprised to see a coin turn up heads eight times in a row. Despite knowing that the two coin flip sequences are both equally probable, we still question whether we are actually witnessing random flips of fair coin; it takes cognitive effort to believe the coin is truly unbiased.[1]

For this reason, randomness has emerged as a central and persistent topic in the cognitive sciences. Many studies (e.g., Lopes & Oden, 1987; Falk & Konold, 1997) suggest that what people usually mean by randomness is the absence of certain detectable patterns. Because this notion is at odds with ideas of randomness used in probability and

[1] A statistically-minded reader might be concerned that getting a sequence of the kind "eight heads in a row" *is* less probable than getting a sequence of the kind "three heads and five tails." However, randomness isn't just about the probability of different kinds of events. Griffiths and Tenenbaum (2007) discuss this issue in detail, and provide counter-examples. For instance, `HHHHTHTTHHHTHTHHTHTTHHH` is an instance of the kind "fifteen heads, eight tails" which is a less probable outcome of tossing a fair coin 23 times than `HHHH` – "four heads in a row" – is for tossing a fair coin 4 times, even though the former would presumably be considered more random.

statistics, people seem to reason poorly about chance. As Nickerson (2002) puts it, "The general conclusion that the results of these experiments in the aggregate seem to support is that people are not very good at these tasks – that they find it hard to generate random sets on request and to distinguish between those that have been produced by random processes and those that have not" (p. 71). Bar-Hillel and Wagenaar (1993) conclude "People either acquire an erroneous concept of randomness, or fail to unlearn it" (p. 388). Cohen (1960) draws the strong conclusion that "nothing is so alien to the human mind as the idea of randomness" (p. 42).

But are people really bad at reasoning about chance? Or are they solving a different problem from merely judging the probability of different outcomes under a random process? If subjective randomness is about detecting patterns, then it is intimately related to intelligent action: "random" stimuli provide no help in predicting outcomes (e.g., what information do eight cars of all different colors confer?), whereas "non-random" stimuli aid our thinking (e.g., if you see eight black cars in a row you might infer that a foreign dignitary is visiting your town). Consistent with this idea, several papers have connected subjective randomness to formal frameworks for characterizing the amount of structure in a stimulus, such as algorithmic complexity (Falk & Konold, 1997; Feldman, 2004; Gauvrit, Zenil, Delahaye, & Soler-Toscano, 2014; Gauvrit, Singmann, Soler-Toscano, & Zenil, 2016; Griffiths & Tenenbaum, 2003; 2004) and Bayesian inference (Griffiths & Tenenbaum, 2001; Hsu, Griffiths, & Tenenbaum, 2010; Williams & Griffiths, 2013).

In this paper, we present a formal framework that unifies these previous approaches, demonstrating that subjective randomness can be explained as a form of statistical inference about the process that generated a stimulus. The key challenge for this approach is characterizing the kinds of structure that people might identify in a stimulus – a problem that algorithmic complexity theory solves by considering all regularities generated by short computer programs (e.g., Li & Vitányi, 2008). This notion is too general to capture human subjective randomness judgments, but by reformulating algorithmic complexity as a

statistical inference we identify two ways in which it can be adapted: simplifying the kinds of computing machines considered and restricting the set of possible regularities. We use these two approaches to develop models of human randomness judgments for three different kinds of stimuli: sequences of coin flips, binary matrices, and dot patterns in spatial arrays.

Phenomena and theories of subjective randomness

Much of the research regarding humans' perceptions of randomness has concerned sequences of numbers or binary outcomes. Reichenbach (1934/1949) is credited with having made the original suggestion that mathematical novices will be unable to produce random sequences of numbers, instead showing a tendency to overestimate the frequency with which numbers alternate. Subsequent research has provided strong support for this claim (reviewed in Bar-Hillel & Wagenaar, 1991; Tune, 1964; Wagenaar, 1972), with both sequences of numbers (Rabinowitz, Dunlap, Grant, & Campione, 1989; Budescu, 1987; Kareev, 1995a) and two-dimensional black and white grids (Falk, 1981). Most strikingly, people believe that subsequent coinflips are more likely to alternate than to stay the same. Basic probability tells us that we should expect the probability of alternation from `H` to `T` (and vice versa) to be about 0.5, yet Falk and Konold (1997) provide a vast body of evidence which shows that people perceive sequences with a probability of alternation around 0.6 or 0.7 as most random. In addition to alternation, people are sensitive to symmetries (e.g., `HHHHTTTT`) and duplications (e.g., `HHTTHHTT`) within sequences, and deem sequences with these properties less likely to have been created by random processes (see, e.g., Lopes & Oden, 1987; Hahn & Warren, 2009).

A number of theories have been presented in an attempt to account for the accuracy of Reichenbach's conjecture. One of the earliest such theories can be traced to Skinner (1942), involving the suggestion that people develop a concept of "randomness" that differs from the true definition of the term. More recently, a theory of this kind was endorsed by Falk (1981). The claim that individuals hold a concept of randomness is supported by the

finding that people consistently apply the same criteria in their judgments of randomness (Budescu, 1987; Falk, 1981). However, some studies have failed to find this kind of consistency (e.g. Wiegesma, 1982), and the difficulty of finding a useful definition of randomness to which this subjective concept might be compared is viewed as a deficiency of the theory (Lopes, 1982).

Other explanations have focused on the limitations that the human information processing system places upon the generation of random numbers. Baddeley (1966) and Wiegesma (1982) suggested that restrictions in short term memory span might account for the high levels of alternation present in generated random sequences. This claim is attractive, but fails to provide an account of why similar effects are obtained in grids and other stimulus displays (Falk, 1981). Furthermore, Neuringer (1986) showed that biases in the judgment of randomness could be removed by educating participants about the statistical properties of random sequences. This result suggests that memory alone may not provide a full explanation of the observed phenomena.

One account that has had a strong influence upon the wider literature of cognitive psychology is Kahneman and Tversky's (1972) suggestion that subjects may be attempting to produce sequences that are "representative" of the output of the generating process. For sequences, representativeness means that the number of elements of each type appearing in the sequence will correspond directly to the overall probability with which these elements occur. Kahneman and Tversky suggested that random sequences are also subject to the constraint of "local representativeness," such that small subsequences also maintain the appropriate probabilities. This implies that a high degree of alternation is to be expected, as it is the only means of guaranteeing the local representativeness of subsequences. This claim receives support from the observation that people tend to generate random sequences that maintain a good level of local representativeness (Budescu, 1987), but the idea of "representativeness" is sufficiently loosely defined to make it difficult to apply to other domains in which randomness may occur. For example, it is not immediately apparent how

the number 7 could be representative of a process that selects random numbers between 0 and 9.

Recent theories have attempted to synthesize these positions. Kareev (1992; 1995b) argued that the phenomena of random sequence generation could be explained through the interaction of an attempt to produce typical sequences and the limitations imposed by a restricted working memory capacity. Thus, local representativeness is enforced by the fact that individuals can only maintain small subsequences in memory at any one time. Falk and Konold (1997) suggested that the concept of "randomness" can be connected to the subjective complexity of a sequence. Complexity here refers to the difficulty of defining a rule by which the sequence could be generated, a definition consistent with the approach taken to randomness in computer science (e.g., Wolfram, 1985). We will use this notion of "algorithmic complexity" as the starting point for our formal investigation of subjective randomness.

Algorithmic complexity and randomness

Research on subjective randomness is made complicated by the fact that clear definitions of randomness are difficult to achieve (Falk, 1981; Kac, 1983). In particular, attempting to apply the label of randomness to single events involves a potential confusion between the randomness of a process and the randomness of its outcome (Spencer-Brown, 1957) – outcomes that appear non-random, such as `HHHHHHHH`, will still result from random processes, such as flipping a fair coin.

Information theory provides a way to address these issues, resulting in a coherent formal definition of randomness that can be applied to individual outcomes. The key ideas come from a branch of information theory known as algorithmic complexity (see, e.g., Li & Vitányi, 2008). The algorithmic complexity of an outcome, such as a binary sequence, is defined to be the length of the shortest computer program required to produce it (see Figure 1).[2] If the length of the shortest program approaches the length of the sequence, the

[2]Technically, it is the shortest input to a universal Turing machine that will produce the sequence. A

sequence is said to be random – the sequence has no more efficient description than itself. The randomness of a sequence can thus be measured in terms of its algorithmic complexity relative to its length. These ideas were independently developed by Kolmogorov (1965), Solomonoff (1964), and Chaitin (1969).

Falk and Konold (1997) drew inspiration from this idea, arguing that subjective complexity – the difficulty that people have in encoding a sequence – might account for subjective randomness. Consistent with this hypothesis, they showed that sequences that people find harder to memorize are considered more random. In an attempt to quantify their theory, they found that traditional encoding difficulty measures such as second-order entropy (Shannon, 1948), a measure of randomness from information theory based on transition probabilities, failed to explain the randomness judgments of participants. Instead, they argued that algorithmic complexity better describes human randomness judgments.

One problem with the practical use of algoritmic complexity is that it is not computable – finding the shortest program that generates a sequence is a computational problem that is impossible to solve, an "incomputable" task (Li & Vitányi, 2008). Given its incomputability, Falk and Konold (1997) proposed an approximation to algorithmic complexity called the difficulty predictor (DP), intended to capture the difficulty of memorizing a particular binary sequence. The DP value of a sequence assigns one point to each subsequence which is a run (all heads or tails), and two points to each subsequence which alternates. For example, the DP value of the sequence `HHTTHTHTH` is equal to 4,

---

Turing machine is an abstract machine defined by Turing (1936) in order to formally characterize the notion of computation. Its input and output appear on a tape that is processed by the machine, which follows an internal set of instructions indicating whether to move forward or backward along the tape and what symbols to write in each position based on the symbols that it reads. A universal Turing machine is a machine that can simulate the operation of any other Turing machine, with the instructions for that Turing machine encoded in the input that is provided on the tape. These instructions constitute the "program" followed by the universal Turing machine. The particular universal Turing machine does not need to be specified, because we can always write a program to translate the input for one universal machine into input for another universal machine, which would only add a constant to the length of the input. Likewise, we can describe programs in an arbitrary programming language as it is always possible to write a program to translate from one programming language to another. For simplicity we will simply talk about computer programs and illustrate those programs using intuitive programming languages.

because it can be parsed as two run subsequences (HH and TT), followed by an alternating subsequence (HTHTH). Falk and Konold (1997) showed that the DP could explain why people's randomness judgments peaked when the probability of alternation of a sequence reached about 0.6 – this was the value that maximized the mean DP of the sequences.

Formal measures of complexity, such as algorithmic complexity, have also been advocated as accounts of other aspects of human cognition. Chater (1999) argued that simplicity is the key principle that may unite all of cognitive science, with algorithmic complexity providing a way to quantify this notion. Feldman (2000; 2003) expressed a similar argument, using measures of the complexity of a concept to explain difficulty of learning, and Feldman (2004) argued that people's sense of surprise – which is related to randomness – might be accounted for in terms of simplicity.

Despite the appeal of algorithmic complexity as a framework for understanding human cognition, it has a major problem as the foundation of a theory of human subjective randomness in addition to its incomputability: the set of all computer programs captures many regularities that people simply do not detect. For example, a binary sequence corresponding to the digits of π (HTHHHTHHTHTTTTTTHHHTTHHTTHTTHHTH...), can be generated by a short program, but would presumably not be considered either simple or non-random by a human observer. Thus, in order to adapt the idea of algorithmic complexity to the structure of the human mind, we need to restrict the set of regularities that might be expressed (and detected) in sequences. To do so, we first need to gain deeper insight into the link between algorithmic complexity and statistical inference.[3]

Algorithmic complexity and statistical inference

Consider the problem of deciding whether a given binary sequence was generated by a random process, such as flipping a fair coin. This is fundamentally a problem of statistical

[3]The connection between algorithmic complexity and statistical inference that we present here is one of the core results of algorithmic information theory, as discussed in detail in Li and Vitanyi (2008). In psychology, this connection has been explored by Griffiths and Tenenbaum (2003; 2004), Gauvrit et al. (2014; 2016).

inference: based on the available data, we seek to determine the process that generated it. Accordingly, a rational solution to this problem is provided by Bayesian inference. Establishing two hypotheses – the sequence was generated from a random process, or a regular process – we seek the probability of each of these hypotheses given the observed data. The answer is given by Bayes' rule, which we express here in its log-odds form:

$$\log \frac{P(\text{random}|x)}{P(\text{regular}|x)} = \log \frac{P(x|\text{random})}{P(x|\text{regular})} + \log \frac{P(\text{random})}{P(\text{regular})}, \tag{1}$$

where x is the observed sequence, $P(x|\text{random})$ is its probability under a random process, $P(x|\text{regular})$ its probability under a regular process, $P(\text{random})$ and $P(\text{regular})$ are the prior probabilities of these processes, and $P(\text{random}|x)$ and $P(\text{regular}|x)$ are their posterior probabilities.

The only term in Equation 1 that depends on x is the log-likelihood ratio, so we use the following for a definition of the randomness of the stimulus x:

$$\text{randomness}(x) = \log \frac{P(x|\text{random})}{P(x|\text{regular})} = \log P(x|\text{random}) - \log P(x|\text{regular}). \tag{2}$$

This is the amount of evidence that the sequence x provides for having been generated from a random process. Consequently, we have an elegant way of addressing the potential confusion between random processes and random outcomes that concerned Spencer-Brown (1957): the randomness of an outcome is the evidence it provides in favor of having been generated by a random source.

$P(x|\text{random})$ is usually straightforward to define. In the case where x is a sequence of coin flips from a fair coin with length $\ell(x)$, $P(x|\text{random}) = (\frac{1}{2})^{\ell(x)} = 2^{-\ell(x)}$. The more difficult challenge is finding a way to express $P(x|\text{regular})$, the distribution over sequences generated by regular outcomes. Since $P(x|\text{random})$ is fixed, this is what will determine the randomness of a sequence.

Following the ideas in the previous section, we can choose to define $P(x|\text{regular})$

using algorithmic complexity. There are actually two ways to do this. The first is to use a probability distribution over sequences defined by Solomonoff (1964). We choose a universal Turing machine, and then provide it with a (potentially infinite) random binary sequence as input.[4] The probability that this machine generates the binary sequence x as output is given by

$$m(x) = \sum_{p:U(p)=x} 2^{-\ell(p)} \tag{3}$$

where p is a program (ie. a binary input sequence), $U(p)$ is the consequence of applying the Turing machine to that sequence, and $\ell(p)$ is the length of the program.

A second way to define a distribution that assigns higher probability to more regular sequences is to start with the set of computable distributions. A computable distribution is a probability distribution over the natural numbers in which the probability $P(x)$ can be computed for each input x.[5] Each of these distributions can be thought of as expressing a hypothesis about a regular generating process – something other than purely random generation. If we want to define a distribution that captures all computable regularities, then we can take a mixture of these computable distributions, defining

$$\mathbf{m}(x) = \sum_{n \geq 1} P_n(x)\alpha(n) \tag{4}$$

[4]Technically, we require this to be a universal "prefix" Turing machine, for which no program is the prefix of any other program. A prefix Turing machine moves monotonically through its input, always going in one direction from the first symbol on its tape to the last, as it follows its internal instructions. For some inputs it processes the entire input and halts after producing meaningful output. For others it halts partway through or never halts. Since it processes its input monotonically, the set of inputs which a prefix Turing machine processes before halting has a special property: no such input is the prefix of any other. For example, if it halted after processing the input 011 then it would halt partway through 0110 or 0111 or any other sequence starting with 011. A universal prefix Turing machine is a prefix Turing machine that can simulate any other prefix Turing machine, with the relevant instructions appearing on its input tape as a program. This means that the programs themselves have the same property, with no program being a prefix of any other program. This matters in defining Solomonoff's probability distribution, as it means that the sum given in Equation 3 is bounded and thus defines a valid probability distribution. For further details see Li and Vitányi (2008).

[5]While we describe these as computable distributions, technically they are enumerable discrete semimeasures, ie. computable functions from the natural numbers $\mathbf{N}$ to the real numbers $\mathbf{R}$ such that $\sum_{x \in \mathbf{N}} P(x) \leq 1$ (as opposed to summing to 1, which would make this a true probability distribution or enumerable discrete measure). The semimeasure defined in Equation 4 is known as the maximal enumerable discrete semimeasure, having the property that for any enumerable discrete semimeasure P there exists a constant c_P such that $c_P\mathbf{m}(x) \geq P(x)$ (Li & Vitányi, 2008).

where P_n is the nth computable distribution, and $\alpha(n)$ is the weight assigned to that distribution. The values of $\alpha(n)$ indicate how likely we think it is that each regularity will be encountered – in the absence of any further information, our best guess as to what the output of a regular generating process will be is obtained by averaging over all of the hypothetical regularities weighted by their probability.

One of the most celebrated results in algorithmic information theory is that *both* of these ways of defining a distribution yield the same result: the probability of a sequence can be approximated (up to a multiplicative constant) by $2^{-K(x)}$, where $K(x)$ is the length of the shortest program that generates x – its algorithmic complexity (Levin, 1974).[6] We can thus explore the consequences of using these distributions to define $P(x|\text{regular})$ by taking

$$P(x|\text{regular}) \approx 2^{-K(x)}, \tag{5}$$

so that more complex sequences have lower probabilities of being produced by a regular process.

Substituting our definitions of $P(x|\text{random})$ and $P(x|\text{regular})$ into Equation 2, we obtain

$$\text{randomness}(x) = \log 2^{-\ell(x)} - \log 2^{-K(x)} = K(x) - \ell(x) \tag{6}$$

which yields precisely the definition of randomness used in algorithmic information theory: the randomness of a sequence is the difference between its shortest description and its length. Formulating randomness as a statistical inference is thus – under appropriate assumptions about the kind of regular processes involved – equivalent to formulating it as a judgment of complexity.

This reformulation still faces the problem that $K(x)$ is both incomputable and unrealistic as a characterization of the regularities that people will detect. However, thinking in terms of statistical inference makes it clear that the key to defining an

[6] To get an intuition for why this might be the case for Solomonoff's distribution, note that $2^{-K(x)}$ is the largest summand in Equation 3.

appropriate metric of subjective randomness is in characterizing what comprises a regular process. In the next section we explore two paths to restricting algorithmic complexity to better characterize the regularities that people can detect, inspired by the two ways in which we defined $P(x|\text{regular})$ in this section.

Adapting algorithmic complexity to model human minds

Algorithmic information theory suggests two strategies for defining $P(x|\text{regular})$: specify the distribution in terms of computing machines, and specify it as a mixture of distributions that capture different regularities. Using either universal Turing machines or all computable distributions yields algorithmic complexity as a measure of randomness. But each of these strategies can be applied to a more restricted set of possibilities. If the possibilities are restricted in a manner reflecting prior knowledge and cognitive limitations, the resulting measure should capture human randomness judgments (assuming people judge randomness as a statistical inference problem). In the remainder of the paper we explore this possibility, focusing first on binary sequences and then widening our lens to consider other kinds of stimuli for which people find it easy to assess subjective randomness.

Gauvrit et al. (2014; 2016) investigated one method for defining a more restrictive form for $P(x|\text{regular})$: using simple Turing machines. Building on previous work by Delahaye and Zenil (2012), Soler-Toscano, Zenil, Delahaye and Gauvrit (2014) were able to run all Turing machines with five states and two symbols to determine whether a given machine halts and, if so, what output it produces. This makes it possible to calculate the probability that a randomly selected (halting) machine produces a particular sequence as output – an approximation to the algorithmic probability of that sequence. The result is an extremely elegant constrained version of algorithmic complexity. Gauvrit et al. (2014) used this complexity measure to analyze random sequence production by children, showing that they produced sequences that were more complex on average than those produced by flipping a coin – a result that is consistent with the human sense of randomness being

related to this measure of complexity. Gauvrit et al. (2016) performed a similar analysis with adults, and also showed that using a "local" measure of complexity (ie. the complexity of substrings) could account for some of the variance in randomness judgment for sequences of length 21.

An alternative way to pursue the strategy of defining distributions using computing machines is to focus on computing machines that are simpler than a Turing machine. Chomsky (1959) defined a hierarchy of computing machines based on their ability to recognize different classes of languages. The Turing machine can recognize any computable language, but successive restrictions to the machine limit the set of languages it can be used to recognize. This provides a framework that allows us to flexibly explore different forms for $P(x|\text{regular})$.

Figure 2 shows an augmented version of the Chomsky hierarchy with the simplest machines on the left. As we move from left to right, the set of regularities the machine is capable of recognizing increases in size. A finite state automaton takes the most basic elements of a Turing machine – the ability to read symbols from an input sequence and to change its internal state according to a set of instructions about how to process those symbols – but discards the ability to move back and forth and modify the input sequence, hence losing the capacity for memory. It can thus only recognize simple regularities, such as a repeated pattern. There are two typical ways to extend a final state automaton. Augmenting the finite state automaton with a queue, a first-in-first-out memory system that can be consulted and modified, results in a queue automaton. Symbols can be written to the queue, and read in the order they were written. The queue memory allows the machine to remember a series of events in order, thereby permitting the machine to recognize the regularity of duplication. Adding a pushdown stack, a last-in-first-out memory system, to the finite state automaton results in a pushdown automaton. Symbols can be written to the stack ("pushed" onto the stack), but only the most recent addition to the stack can be read. The pushdown stack can recall a series of events in reverse order,

enabling the pushdown automaton to recognize symmetrical patterns. Both symmetry and duplication can be recognized by an automaton with a readable stack (henceforth "stack automaton"), which has a special type of stack in which every element in the stack can be read (items are still added to or removed from the first position in the stack).

The Chomsky hierarchy, with its familiar restrictions on the power of computing machines and corresponding restrictions on the kinds of regularities – or languages – that those machines can identify provides a natural starting point for defining distributions based on simpler computing machines than the Turing machine. This first strategy is most usefully applied to modeling the subjective randomness of binary sequences, since these are the standard input and output of computing machines. We pursue the potential of this approach in the next section.

The literature on human subjective randomness has explored a variety of different kinds of stimuli that go beyond binary sequences. For these stimuli, following the second strategy for defining $P(x|\text{regular})$ and taking a mixture of distributions that express specific regularities provides a straightforward way of defining models of subjective randomness. Rather than taking a mixture of all computable distributions, we take a mixture of a set of distributions that we believe capture the kinds of regularities that people might be sensitive to. This strategy is straightforward to apply in a wide range of circumstances, and we use it to model human randomness judgments for binary matrices and spatial arrays later in the paper.

Probabilistic machines and the Chomsky hierarchy

To define an appropriate approximation of $P(x|\text{regular})$ when x is a sequence of coin flips, we can use probabilistic machines as a formal description of probabilistic rules and patterns for generating sequences. A probabilistic machine defines a distribution $P(x|\text{regular})$ based on the probability of x being generated by the machine. The structure and fitted parameters of the probabilistic machine provides a formal description of the

kinds of regularities that people identify in binary sequences.

We begin our analysis with a finite state automaton, the simplest machine in the hierarchy shown in Figure 2, and evaluate how well it can reproduce people's randomness judgments. We use a specific type of probabilistic finite state automaton known as a hidden Markov model (HMM). A finite state automaton processes an input sequence by moving from one internal state to another as it reads each symbol from the sequence, with the transitions between states resulting from a deterministic set of instructions. A hidden Markov model allows these relationships to be probabilistic. Rather than starting with a set of instructions for processing a sequence, it provides a procedure for generating a sequence. Each state is associated with a probability distribution over symbols and a probability distribution over other states. An initial state is selected randomly and a symbol is generated from the probability distribution associated with that state. The next state is then sampled from the probability distribution over states associated with the current state and the process continues. Probabilistic inference can be used to infer the sequence of states used to generate a sequence, and this inference process is a stochastic generalization of the deterministic procedure followed by a finite state automaton. For further details see Manning and Schütze (1999). .

More formally, assume that each symbol $x_i \in \{\texttt{H},\texttt{T}\}$ in a sequence of length n, $x = x_1x_2...x_n$, can be seen as being produced by some latent state z_i. This can be interpreted as the current pattern being produced by the machine (e.g., repeating `HT`). Making a *first-order Markov assumption* about the latent variables $z = z_1z_2...z_n$ forces the probability distribution of z_i to depend only on the state of its immediate predecessor, z_{i-1}.[7] The conditional independence assumption allows inference for the parameters of a HMM to be computationally tractable.

[7]We assume that sequences are produced from left-to-right. As our behavioral data comes from English speakers, a left-to-right reading of stimuli is natural. However, it should be noted that the probability of producing a sequence and its reverse differs by very little in the particular hidden Markov model we use as what matters most is the number and length of particular subsequences, which does not change if the order of the sequence is reversed.

Under these assumptions, the joint probability distribution of x and z is given by

$$P(x, z) = P(z_1) \prod_{i=2}^{n} P(z_i|z_{i-1}) \prod_{i=1}^{n} P(x_i|z_i). \tag{7}$$

From this, the probability of generating a given sequence x can be found by marginalizing out all possible latent states z that could have produced the observed sequence:

$$P(x) = \sum_{z} P(x, z). \tag{8}$$

The model is fully specified through the choice of states and the distributions $P(x_i|z_i)$ and $P(z_i|z_{i-1})$.

For reasons that will become clear shortly, we define an HMM with six hidden states and organize the transitions between these states into four motifs (or subpatterns): repeating heads, repeating tails, repeating heads-tails, and repeating tails-heads (see Figure 3). In any state, we define the probability of continuing its corresponding motif to be equal to $\delta \in [0, 1]$, and the probability of switching to a different motif to be proportional to $\alpha^k \in [0, 1]$, where k is the number of states within the motif. Thus, the probability of entering a new motif decreases with the length of the motif, thereby assigning less probability to more complex motifs. The matrix expressing these unnormalized transition probabilities is thus

$$P(z_i|z_{i-1}) = \begin{pmatrix} \delta & C\alpha & C\alpha^2 & 0 & 0 & C\alpha^2 \\ C\alpha & \delta & C\alpha^2 & 0 & 0 & C\alpha^2 \\ C\alpha & C\alpha & 0 & \delta & 0 & C\alpha^2 \\ C\alpha & C\alpha & \delta & 0 & 0 & C\alpha^2 \\ C\alpha & C\alpha & C\alpha^2 & 0 & 0 & \delta \\ C\alpha & C\alpha & C\alpha^2 & 0 & \delta & 0 \end{pmatrix}, \tag{9}$$

where each row is a vector of unnormalized transition probabilities and $C = \frac{1-\delta}{2\alpha+2\alpha^2}$.[8] For example, the probability of transitioning to $z_i = 3$ given that we are in state $z_{i-1} = 5$ is located in the fifth row and third column of the matrix (it is $C\alpha^2$). The initial hidden states are given by unnormalized probability vector $P(z_1) \propto \begin{pmatrix} C\alpha & C\alpha & C\alpha^2 & 0 & 0 & C\alpha^2 \end{pmatrix}$.

The Difficulty Predictor as a special case

Recall Falk and Konold's (1997) Difficulty Predictor (DP) for binary sequences, which assigned one point for each repeating subsequence and two points for each alternating subsequence. This model provides an intuitive way to measure the complexity of a sequence, but it has some limitations. For one, DP does not account for sequence length adequately: `HHTHT` and `HHHHHHHHHHHHHHHHHHTHT` both have DP = 3 but the former seems more random. In addition, DP fails to account for symmetry and duplication within a sequence. If we can identify the assumptions behind DP, we can generalize those assumptions to yield a more flexible model.

Using the HMM defined above to specify $P(x|\text{regular})$, we can show that DP is just a special case of considering subjective randomness as statistical inference (Griffiths & Tenenbaum, 2003). The HMM allows two classes of motifs – corresponding to repetition and alternation. Switching between motifs is done with probability proportional to δ, repetition is chosen with probability proportional to α, and alternation with probability proportional to α^2. For a choice of z indicating n_1 runs and n_2 alternating subsequences, $P(x, z) \propto \delta^{n-n_1-n_2}(\frac{1-\delta}{2\alpha+2\alpha^2})^{n_1+n_2}\alpha^{n_1+2n_2}$. Taking $P(x|\text{regular})$ to be $\max_z P(x, z)$, it is straightforward to show that $\text{random}(X) = -\text{DP} \log \alpha$ when $\delta = 0.5$ and $\alpha = \frac{\sqrt{3}-1}{2}$. Having identified this special case, we can see whether it is possible to get better results by exploring other values for these parameters.

In Experiment 1 of Falk and Konold (1997), 97 participants gave apparent

[8] The choice of C is motivated by the concern that the sum of each row ought to be less than 1. It should be noted that we do not require the rows to sum to 1 for now (this is relevant to establishing equivalence with previous work). Instead, each state should be thought of carrying some positive probability of transitioning to a state that has never been observed previously, i.e., a "null state."

randomness ratings on a scale of 1 to 10 for ten heads-tail sequences, each of length 21. Each sequence contained between 2 and 20 alternations between heads and tails. On average, a fair coin should produce a probability of alternation of about 0.5, meaning that participants should have seen sequences with about 10 alternations as most random. However, as shown in Figure 4, participants rated those sequences that had about 14 alternations as optimally random. To explain their results, Falk and Konold (1997) applied their DP complexity measure to the data, yielding a correlation of $r = 0.93$.

We can find a better fit than DP for Falk and Konold's (1997) data by optimizing our general model: $\delta = 0.525$ and $\alpha = 0.107$ yields a correlation of $r = 0.99$ (see Figure 4).[9] These new parameters also address some of the counter-intuitive predictions of DP. For example, if $\delta > 0.5$, increasing the length of a sequence but not changing the number of runs or alternating subsequences reduces its randomness, because $P(X|\text{regular})$ decreases more slowly than $P(X|\text{random})$. When we solve for these new unrestricted values of δ and α, random(`HHTHT`) = 3.33, while random(`HHHHHHHHHHHHHHHHHHTHT`) = 2.61.

Generalizing the model and predicting the classification of binary sequences

The finite state automaton effectively captures the bias people show towards avoiding streaks when generating random sequences. But other regularities, such as symmetry, cannot be encoded by the finite state automaton unless we include an unreasonable number of states and motifs. Instead, symmetry can be incorporated by ascending the Chomsky hierarchy and expanding the set of languages our machine can recognize from regular languages to context-free languages. Context-free languages can be recognized by a pushdown automaton, which is a finite state automaton with a stack. Similarly, we can augment the finite state automaton with a queue to allow the machine to recognize duplication. Comparing the performances of the pushdown and queue automata will allow us to determine how sensitive people are to symmetry and duplication pattern grammars

[9] All parameter optimization is done using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.

when evaluating randomness.

For the pushdown automaton, we model two types of symmetry: mirror symmetry (e.g., HHHTTHHH) and complement symmetry (e.g., TTTTHHHH). The model is augmented so that $P(x|\text{regular}) = \max_{z,M} P(x, z|M)P(M)$, where M is a method of production. The first half of the sequence is generated by the finite state automaton defined above, and the latter half is generated either by repetition (i.e. the finite state automaton continues to produce the remaining half of hidden states) or by affixing the symmetric or complement-symmetric counterpart, according to probabilities $P(M)$. The queue automaton uses a similar scheme, where M can be either repetition or duplication. Further generalizing our machines along the Chomsky hierarchy leads us to formulate the stack automaton, which can use methods of production corresponding to symmetry, complement symmetry, and duplication.

In the first experiment of Lopes and Oden (1987), participants were split into two groups.[10] Both groups were given the task of classifying whether binary sequences of length eight were generated from a random source or a non-random source whose nature was unspecified. Half of the sequences that each group saw were created by an unbiased coin flipping process ($p = 0.5$). The other half of the sequences were produced in a manner that depended on which group the participant was placed in. For one group the other half of the sequences was generated with a repetition bias. For another group the other half of sequences was generated with an alternation bias. The top two panels of Figure 5 show the performance of participants in the experiment, separating out sequences that were symmetric and asymmetric. To evaluate the performance of different models on this task we transform Equation 2 into a logistic regression model, with

$$P(\text{random}|x) = \frac{1}{1 + \exp(-\lambda\, \text{randomness}(x) - \psi)}, \tag{10}$$

where λ weights the importance of randomness(x) and $\psi = \log \frac{P(\text{random})}{P(\text{regular})}$ is the log prior odds. Note that setting $\lambda = 1$ corresponds to an appropriately-calibrated inference.

[10] We only analyze results from the *uninformed* condition of the experiment here.

Using the finite state automaton as our definition for $P(x|\text{regular})$ and the model in Equation 10, we fit the data from Lopes and Oden (1987) to yield $\delta = 0.638$, $\alpha = 0.659$, $\lambda = 0.128$, and $\psi = -2.75$, with correlation $r = 0.90$. However, we can improve upon this result by using the pushdown automaton to account for the fact that participants in their experiment were likely to view symmetric sequences as non-random. Using the pushdown automaton, we reach a correlation of $r = 0.98$ using parameters $\delta = 0.688$, $\alpha = 0.756$, $P(M = \text{repetition}) = 0.437, P(M = \text{symmetry}) = 0.491, P(M = \text{complement symmetry}) = 0.072$, $\lambda = 0.125$ and $\psi = -2.99$. The high value for $P(M = \text{symmetry})$ is notable, indicating that people clearly use symmetry as a factor to determine non-randomness. Additionally, the value for $\psi = -2.99$ implies that $P(\text{regular}) > P(\text{random})$, suggesting that people are biased towards classifying a sequence as non-random a priori. When we impose the constraint $\lambda = 1$ so that the model is forced to use correct Bayesian inference, the correlation only drops to $r = 0.97$, with $\delta = 0.656$, $\alpha = 0.314$, $P(M = \text{repetition}) = 0.421, P(M = \text{symmetry}) = 0.504, P(M = \text{complement symmetry}) = 0.074$, and $\phi = -1.64$. Unfortunately, Lopes and Oden (1987) did not study the effects of duplication, so we cannot determine the efficacy of the queue automaton with these data.

While the data of Falk and Konold (1997) and Lopes and Oden (1987) provide a way to evaluate some of the basic predictions of our models, they do not allow us to explore all of the regularities that people might be sensitive to or to test which of the components of the models are critical for fitting human data. These models are relatively complicated, with multiple free parameters, and the quality of the estimates of those parameter values depends on the scope and quantity of the data from which they are estimated. In order to properly evaluate our models we need to collect a more comprehensive data set of human subjective randomness judgments for binary sequences and use statistical methods to evaluate the performance of different models.

Experiment 1: Evaluating more complex models

In the last section, we introduced four probabilistic machines that differed by their memory systems. This allowed us to define a set of models that generalize the Difficulty Predictor in several ways, although they retain the same basic commitment to regularities being based on repetition and alternation. We can add further complexity to the model by increasing the number of motifs it can detect. For example, if we include all binary strings up to length 4 that are not repeats of shorter motifs, we increase the total number of motifs to 22 and the total number of states to 72. The result is a comprehensive model that can capture a variety of regularities in binary sequences.[11]

In this section we present an experiment designed to evaluate the performance of this more complex model. Our probabilistic machines differ in the kind of memory they use, so we investigate the effects of memory constraints on randomness judgments by asking participants to decide whether a coin flip sequence comes from a random or non-random source. To test the effect of memory demands, each participant is placed under one of two conditions: *Sequential* or *Simultaneous*. In the Sequential condition each coin flip in the sequence appears one at a time, so that the participant is only able to view the current flip. This is akin to the natural task of observing the outcomes of a person flipping a single coin repeatedly. In the Simultaneous condition the full sequence appears all at once, as if the results had been tallied and displayed. When the participant is asked to choose whether a sequence is random or non-random, we do not specify the nature of the non-random source; it is left to participants to decide what non-random means. Our experimental data will allow us to test the efficacy of our four machines—finite state, pushdown, queue, and stack—and help us decide how high the Chomsky hierarchy must be climbed to capture

[11] In addition, as we no longer need to compare to DP directly we will use a transition matrix in which the rows sum to one. We alter the matrix by dropping C and normalizing each row, dividing each row by its sum to ensure that each row sums to 1. This change carries some minor implications: for example, sequences such as `HHHH` and `HTHT` now not only differ in their priors (α vs. α^2), but also by their motif continuation probabilities (each δ has a different normalization constant). Thus, the value of δ will be scaled higher for some motifs and scaled lower for others, enabling a more fine-grained look at the structure of complexity. The generative model still has only two parameters – α and δ – that generate all these transition probabilities.

people's intuitions.

Method

Participants. Participants were 40 undergraduates. Each participant was randomly assigned to one of the two conditions.

Stimuli. Stimuli were sequences of heads (`H`) and tails (`T`) presented in 130 point fixed width sans-serif font on a 19" monitor at 1280 × 1024 pixel resolution.

Procedure. Participants were seated at a computer and presented with a series of heads and tails sequences. In the Simultaneous condition, the sequence was presented in its entirety; in the Sequential condition, each element of the sequence was displayed after another, with each character displayed for 300ms with a 300ms inter-stimulus interval. Participants were instructed to classify each sequence as either coming from a random process (flipping a fair coin) or by some other process which was not random. After a practice session, each participant classified 128 unique sequences of length 8 in random order.[12] Participants took a short break every 32 trials.

Results

By measuring the proportion of participants classifying each sequence x as random, our results provide an estimate of the distribution of $P(x|\text{random})$. From this, we can fit the models using Equation 10. The models using 22 motifs outperformed those using 4 motifs; as such only the results of the 22 motif models are reported here. The results are displayed in Figure 6.

Because some of the models are nested, we can perform log-likelihood ratio tests to determine if the extra parameters contribute to a statistically significant model improvement. We find that for the Simultaneous condition, the best fit is provided by the stack automaton, while for the Sequential condition, the best fit is provided by the queue

[12] Presenting 128 sequences is sufficient if we assume that each sequence is equivalent to its complement, e.g., `HHTTHHTH` is equivalent to `TTHHTTHT`. The choice of which of the equivalent sequences was displayed was decided pseudo-randomly by the computer.

automaton. A Bayesian model comparison framework using Bayes factors yielded similar conclusions.[13] To test if our models were overfitting the data, we used 5-fold cross-validation[14] to calculate the average out-of-sample correlation. We found that the average out-of-sample correlations were 0.79 and 0.73 for the Simultaneous and Sequential conditions, respectively.

These results suggest that when the full sequence is available to the participant in the Simultaneous condition, the participant is sensitive to all three methods of production: symmetry, complement symmetry, and duplication. However, when viewing each coin flip one-by-one in the Sequential condition, participants are only sensitive to duplication. Adding in symmetry and complement symmetry as methods of production does not significantly improve the model's fit (see Figure 6(d)). This aligns with what we know about human cognition: working memory acts more like a queue than a pushdown stack. Repeating this analysis at the level of individual participants also supported this conclusion. In the Simultaneous condition, 3 participants were classified as best fit by the Finite State model, 7 by the Pushdown, 3 by the Queue, and 7 by the Stack. In the Sequential condition, 2 participants were classified as best fit by the Finite State model, 6 by the Pushdown, 10 by the Queue, and 2 by the Stack.

The stack automaton's best fit parameters for the group data in the Simultaneous condition are $\delta = 0.50$, $\alpha = 0.03$, $\lambda = 0.33$, $\psi = -1.05$, with production method probabilities $P(M = \text{repetition}) = 0.298, P(M = \text{symmetry}) = 0.494, P(M =$

[13] A Bayes factor is defined as the relative evidence the data gives for one model over another model, formally $\frac{P(D|M_1)}{P(D|M_0)}$, where D is the data, M_0 is the null model, and M_1 is the alternative model. To compute the Bayes factor, an integration over the parameters must be performed, with $P(D|M) = \int P(D|\theta, M)P(\theta|M)d\theta$ where $P(D|\theta, M)$ is the model's likelihood function and $P(\theta|M)$ is the prior distribution over the models parameters. For the HMM, this value cannot be computed analytically, so a Monte Carlo estimate of the probability of the data can found by sampling from the prior, as described in Kass and Raftery (1995). For the models described in this section, we used uniform priors on δ and α, a Gaussian (normal) distribution on λ with mean and standard deviation 1, a Gaussian distribution on ϕ with mean 0 and standard deviation 1, and a uniform distribution on the vector of probabilities for each method of production.

[14] 5-fold cross-validation is a standard technique to test for overfitting. The procedure works by randomly partitioning the data into 5 equal parts, training on each distinct set of 4 parts, and calculating the out-of-sample error for the remaining part. The 5 out-of-sample errors are then averaged to yield an estimate of test error.

complement symmetry) = 0.013, and $P(M = \text{duplication}) = 0.195$. Thus, symmetry and repetition are by far the most important patterns used by our participants to evaluate randomness. For the Sequential condition, the queue's best fit parameters are $\delta = 0.50$, $\alpha = 0.04$, $\lambda = 0.30$, $\psi = -0.50$, with production method probabilities $P(M = \text{repetition}) = 0.835$ and $P(M = \text{duplication}) = 0.165$. Because δ is much greater than α in both conditions, our models indicate that non-random sequences are biased towards continuing, rather than switching, motifs.[15]

It is insightful to see how the stack automaton model's predictions of randomness vary with the number of heads in a sequence. In Figure 7, we plot the number of heads in the sequence versus the model's predicted randomness(x) score. We find that flipping one bit of the sequence towards its majority symbol decreases the randomness of the sequence, on average. There is a small deviation from the pattern when there are exactly four heads and four tails–this is because perfect symmetry, `HHHHTTTT` or `TTTTHHHH` is perceived as quite non-random.

Comparing different strategies for defining simple machines

The models that we explored in the previous sections focused on one method for defining simple machines: specifying probabilistic automata on the Chomsky hierarchy. Gauvrit et al. (2014; 2016) presented a different method of measuring complexity using simple machines: explicitly calculating the probability of a sequence being produced by a

[15]Since these models have a large number of parameters, we conducted a parameter recovery analysis to confirm that our estimation procedure could identify the correct parameter values for simulated data. For each of the four models (Finite State, Pushdown, Queue, Stack), sets of randomly simulated data were generated. To generate a set of simulated data, parameters from the parameter space of the each model were uniformly sampled. Once we have the parameters, we can produce a $P(\text{random}|X)$ value for each of the unique 128 binary sequences $X_1, ..., X_{128}$. Then, we can simulate data in that same format as Experiment 1, sampling judgments for 128 binary sequences. Here, we choose to simulate the data 1,000 times–this is as if 1,000 participants had rated the 128 sequences. This procedure is repeated 5,000 times. To test for identifiability, we compute the correlation between the randomly sampled set of "true" parameters and the parameter values found when fit to the simulated data generated from the "true" parameters. There was a close correspondence between the true and estimated parameter values – of the 25 resulting correlations, 10 were greater than .95, 7 were between .90 and .95, 7 were between .85 and .90 and the lowest (the weight of repetition in the Pushdown model) was .81.

small Turing machine. The results of Experiment 1 provide a way to compare these two approaches.

Following Gauvrit et al. (2014), we took $P(x|\text{regular})$ to be the proportion of sequences produced by Turing machines with five states and two symbols that matched x. This model is referred to as $D(5)$ by Gauvrit et al. Since the sequences used in Experiment 1 were all of length 8, we can filter this distribution to contain only those sequences – this contributes just a normalizing constant that disappears in our subsequent analysis.

Taking this definition of $P(x|\text{regular})$, we can compute randomness(x) as specified in Equation 2. We can evaluate model performance by calculating the log-likelihood from the same logistic regression model we used for evaluating the other models presented above:

$$P(\text{random}|x) = \frac{1}{1 + \exp(-\lambda \text{randomness}(x) - \psi)} \tag{11}$$

where the parameters λ and ψ are optimized for best fit with the data. Fitting this model to the Sequential and Simultaneous conditions of Experiment 1 yielded best fitting parameters of $\lambda = 0.696, \psi = 0.162$, and $\lambda = 0.749, \psi = 0.229$ respectively.

The log-likelihood score for this model can be compared against the log-likelihood produced by the models based on the hidden Markov model, and the correlation between model predictions and data provides a sense of how well the predicted and observed probabilities of sequences being judged random correspond to one another. For the Sequential condition, the HMM (Queue) model has a log-likelihood of -1648.62 and $r = 0.75$, while the $D(5)$ model gives -1715.80 and $r = 0.45$. For the Simultaneous condition, the HMM (Stack) model has a log-likelihood of -1531.59 and $r = 0.83$, while the $D(5)$ model gives -1697.10 and $r = 0.38$. Admittedly, one of the advantages of the $D(5)$ model is that it is parameter-free; for this experiment we only use the logistic transformation to stretch and scale its output. To evaluate each model's relative strengths, we can use information criteria such as the Akaike Information Criterion (AIC; Akaike,

1973) and the Bayesian Information Criterion (BIC; Schwarz, 1978) which explicitly penalize the number of parameters. Smaller values of these criteria correspond to better fitting models taking into account any differences in the number of parameters. In the Sequential condition, AIC = 3307, BIC = 3339 for the HMM, AIC = 3436, BIC = 3449 for $D(5)$; in the Simultaneous condition, AIC = 3077, BIC = 3123 for the HMM, AIC = 3398, BIC = 3411 for $D(5)$. Choosing the model which minimizes AIC and BIC for both cases, we find that the HMM outperforms $D(5)$ on this experimental evidence.

To see why the $D(5)$ model is performing worse in each condition, we can analyze the total squared error for each of the different kinds of regularity expressed by the stimulus sequences. The stack automaton model has the ability to produce symmetry, complement symmetry, and duplication – 18.8% of sequences have at least one of these features. The percentages of squared error contributed by sequences that are symmetric, complement symmetric, or duplicated, in the HMM models were 17.5% and 12.3% in the Simultaneous and Sequential conditions respectively. This indicates that these sequences contributed about as much error as other sequences. However, for the $D(5)$ model, 43% of the error came from those special sequences in the Simultaneous condition, and 31% in the Sequential condition, indicating that an outsize proportion of error results from misclassifying sequences containing symmetry, complement symmetry, and/or duplication.

Analyzing the patterns of errors shows that the $D(5)$ model systematically overestimated $P(\text{random}|x)$ for sequences that display symmetry, complement symmetry, or duplication. In other words, people perceive sequences with the features to be less random than predicted by the model. This makes a certain amount of sense – these are high-level regularities that might be rare among the products of very simple Turing machines. However, this is not the only factor behind the success of the probabilistic automata – even the simplest automaton model, without these regularities, produced a better fit to the human data than the $D(5)$ model.

The use of simple Turing machines to define a constrained measure of algorithmic

complexity is extremely elegant, and the resulting parameter-free model does an impressive job of capturing the human data. However, getting a closer match to human judgments of randomness, including how factors such as memory constraints arise based on presentation format, may require a broader exploration of ways that simple computing machines can be linked to randomness. Here, we focused on different kinds of automata in the Chomsky hierarchy, but it may be possible to achieve similar results within the framework introduced by Gauvrit et al. by looking at a wider range of distributions defined using Turing machines of different sizes.

Exploring $P(x|\textbf{regular})$

Experiment 1 provided clear results indicating what kinds of regularities people are sensitive to under different memory conditions. The main limitation of this experiment is that the proportion of participants classifying a particular sequence as random is, at best, a noisy estimator of the perceived randomness of the sequence. Furthermore, it is difficult to analyze any individual differences between participants' perceptions of randomness.

To explore our account of subjective randomness further, we devised an experiment designed to directly estimate $P(x|\text{regular})$. The experiment is based on a commonly-used tool in statistics and machine learning known as Markov chain Monte Carlo, a stochastic approximation method used to sample from complex distributions that cover a high-dimensional space (Gilks, Richardson, & Spiegelhater, 1996). In a traditional Monte Carlo simulation, an algorithm is used to generate samples from a particular probability distribution directly. In contrast, Markov chain Monte Carlo algorithms produce a series of samples – a "chain" – from a series of probability distributions. If the algorithm is run for long enough, the probability distribution from which the samples are drawn converges to the probability distribution of interest, and the samples can be treated as samples from that distribution.

Sanborn, Griffiths, and Shiffrin (2010) proposed using Markov chain Monte Carlo as

the basis for experiments with people that are intended to estimate psychological quantities that can be expressed as probability distributions. Markov chain Monte Carlo with People (MCMCP) is an adaptive experimental method that can be used to elicit participants' beliefs without explicitly asking them. MCMCP has been used to infer people's expectations about categories composed of high-dimensional stimuli such as stick-figure animals (Experiment 3, Sanborn et al., 2010) and facial expressions (Martin, Griffiths, & Sanborn, 2012). In our experiment, we use the Markov chain Monte Carlo technique known as Gibbs sampling (Geman & Geman, 1984) to estimate people's beliefs about regular sequences.

Assume there is an n-dimensional probability distribution $P(x_1, x_2, \ldots, x_n)$ that we wish to sample from. In the case of a distribution on binary sequences, we would have n binary variables each corresponding to the value of each element in the sequence. The sequence `HHTHTHTT` would be $n = 8$ and $x_1 = \texttt{H}$, $x_2 = \texttt{H}$, $x_3 = \texttt{T}$ and so on. If we can efficiently sample from the conditional distributions $P(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, then we can use Gibbs sampling to generate samples from $P(x_1, x_2, \ldots, x_n)$. The algorithm is simple: First, we arbitrarily assign values to $x_1, x_2, \ldots, x_n$. Then we draw a new value for x_1 given the current values assigned to $x_2, \ldots, x_n$, sampling from $P(x_1|x_2, \ldots, x_n)$. We continue this process, drawing new values for x_2, x_3, and so on, in each case replacing the value of x_i with a sample from $P(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$. One pass through all n variables is a single iteration of the Markov chain. After this process is repeated enough times (called the "burn-in" period), the Markov chain converges to the target distribution and the values of $x_1, x_2, \ldots, x_n$ behave as samples from $P(x_1, \ldots, x_n)$.

We can adapt this procedure to a behavioral experiment that can be used to estimate $P(x|\text{regular})$. Instead of sampling from a pre-defined conditional probability distribution to perform Gibbs sampling, we "sample" from people's implicit expectations over coin flip sequences.[16] To perform this task, in each trial of our experiment the participant views a

[16]The procedure samples from people's subjective distributions as long as their choices can be described in terms of Luce's choice rule (for the underlying assumptions of the method see Sanborn et al., 2010).

sequence of coin flips with one flip covered up by a black square, e.g.,

HTTH■TTH

and indicates whether he or she believes the black square is covering an H or a T, given that the sequence was created by a non-random process. In other words, the participant is choosing a new value for x_i, which is covered by a black square, conditioned on the values of all other $x_k, k \neq i$. This is exactly the distribution that is required for Gibbs sampling. Once the new value for x_i is set, the participant views the same sequence on a later trial but with a different location covered by a black square. Performing this process repeatedly creates a chain of binary sequences which will ultimately behave as samples from $P(x|\text{regular})$.

Figure 8 illustrates our approach. The initial states of three chains are sampled at random (without replacement). We set the chain length to 16, so that each element in the coin flip sequence is sampled twice. The participant completes a set of trials corresponding to one iteration in each of the chains in a random order before moving onto the next iteration in each chain so as to prevent the participant from detecting that their choices are affecting future trials. The box locations are selected as random permutations of the eight possible locations for each chain, with one full random permutation selected for the first eight steps in the chain and another random permutation for the second eight steps. Our experiment used 40 chains, each of length 16, resulting in a total of 640 trials for each participant to complete.

Experiment 2: Sampling from $P(x|\textbf{regular})$

Method

Participants. Participants were 25 undergraduates participating for course credit. Two participants were excluded after examination of their data revealed they had repeatedly entered a single response, yielding usable data from 23 participants.

Stimuli. Stimuli were sequences of heads and tails of length 8, as in Experiment 1.

Procedure. In each of the 640 trials each participant completed, the computer prompted the participant to examine a coin flip sequence of length eight with one flip outcome hidden and to respond with whether they thought an `H` or `T` belonged where the ■ appeared. The instructions read:

> The sequence of heads (H) and tails (T) below was produced by a process that is NOT random. One outcome has been hidden with a box. Do you think what was underneath the box was heads or tails?

Each participant completed 640 trials, corresponding to 40 chains each of length 16. After every 40 trials, the participant was encouraged by a computer prompt to take a short break.

The binary sequences used to initialize the chains were sampled uniformly over all binary sequences of length eight without replacement so that no two chains began with the same sequence. Participants completed the trials in all 40 chains before moving onto the trials corresponding to the next iteration of those 40 chains. In each set of 40 trials, the chains from which the trials were drawn were presented in randomly permuted order. In addition, the location of the ■ in each chain was randomly permuted so that the location of the box appeared at each index in the sequence exactly twice.

Results

We use the chains-crossing criterion (Johnson, 1996) to determine convergence. Once two or more chains cross at a particular step, all future realizations of sequences in the converged chains are aggregated into the estimate of the target distribution $P(x|\text{regular})$. Because we are working with a discrete probability distribution, a cross occurs when the same binary sequence is present in two or more chains at a particular iteration. We call the estimate of $P(x|\text{regular})$ found from the MCMCP experiment $\hat{P}(x|\text{regular})$. Data from the experiment coincides with observations made in other studies of human subjective randomness, validating the MCMCP approach. For example, the most common sequence,

TTTTTTTT, made up 8.5% of the mass of $\hat{P}(x|\text{regular})$, and the second-most common sequence, HHHHHHHH, made up 6.8% of the mass. The least common sequence, HHTTTTHT, made up only 0.01% of the mass. A full ranking of all sequences from least to most random can be seen in Figure 9.

Table 1 shows the goodness-of-fit characteristics of all models considered. For each model, we computed a log-likelihood by calculating the probability of the samples generated by people under the distribution implied by the model. To provide a heuristic measure of fit, we also computed the Pearson correlation coefficient between $\hat{P}(x|\text{regular})$ and $P(x|\text{regular})$ under the model. We calculated the average out-of-sample correlation using 5-fold cross validation; we found that increasing the complexity of the models did not reduce the out-of-sample correlation, which suggests the models are not overfitting. The 4-motif and 22-motif stack automatons provide the best fits for our data set, yielding correlations of $r = 0.91$ and $r = 0.92$, respectively. Because the 4-motif and 22-motif stack automata are not nested models, we cannot formally apply a log-likelihood ratio test to see if the 22-motif machine lends a statistically significant improvement in fit. However, since the performance is similar despite the difference in complexity these results would seem to favor the 4-motif model.

A comparison of the 4-motif models using log-likelihood ratio tests (see Figure 10) reveals that the stack automaton yields the best fit for the data, corroborating the results found for the Simultaneous condition in Experiment 1. A Bayesian model comparison analysis using Bayes factors (also shown in Figure 10) produced the same conclusion.[17] The fact that the stack automaton outperforms the other models indicates once again that people use the pattern grammars of symmetry, complement symmetry, and duplication to evaluate randomness.

The optimal parameters found for the 4-motif stack automaton are $\delta = 0.5493$, $\alpha = 0.2073$, with production method probabilities $P(M = \text{repetition}) = 0.7074, P(M =$

[17]The same priors for δ, α and the methods of production in Experiment 1 were used for this analysis.

symmetry) = 0.1631, $P(M = \text{complement symmetry}) = 0.0652$, and $P(M = \text{duplication}) = 0.0984$. These yield transition matrix and initial state vector:

$$P(z_i|z_{i-1}) = \begin{pmatrix} 0.6497 & 0.2491 & 0.05063 & 0 & 0 & 0.05063 \\ 0.2491 & 0.6497 & 0.05063 & 0 & 0 & 0.05063 \\ 0.2078 & 0.2078 & 0 & 0.5421 & 0 & 0.0422 \\ 0.2078 & 0.2078 & 0.5421 & 0 & 0 & 0.0422 \\ 0.2078 & 0.2078 & 0.0422 & 0 & 0 & 0.5421 \\ 0.2078 & 0.2078 & 0.0422 & 0 & 0.5421 & 0 \end{pmatrix}$$

$$P(z_1) = \begin{pmatrix} 0.4155 & 0.4155 & 0.0845 & 0 & 0 & 0.0845 \end{pmatrix}$$

The difference between the motif continuation probabilities (0.6497 vs. 0.5421) illustrates that participants perceive alternating subsequences as less likely to continue than a repeating heads or repeating tails motif.

To check the robustness of the best-fitting parameters from Experiment 2, we use the data in the Simultaneous condition from Experiment 1 to re-fit Equation 10 while keeping the parameters found for the 4-motif stack automaton from this experiment fixed. In other words, the randomness(x) component in Equation 10 is fixed, but we freely vary λ and ψ. We find that $\lambda = 1.018$ and $\psi = -0.0057$, with $r = 0.75$. In Experiment 1, the best fit was provided by $\lambda = 0.33$ and $\psi = -1.05$, with $r = 0.83$. The new value for ψ is notable: because $\psi = \log \frac{P(\text{random})}{P(\text{regular})}$, a value near zero implies random and regular events are approximately equally likely. The new value for λ further validates the MCMCP experiment: the log-likelihood is weighted correctly when $\lambda = 1$, implying that the responses of the participants are conforming more closely to Bayesian inference with this choice of $P(x|\text{regular})$.

Another advantage of using MCMCP is that we can examine individual participants' differences in randomness judgments. Because each participant only worked on 40 chains,

requiring a chain to converge before its sequences are allowed into the estimation procedure would reduce the workable data from each participant to an unusably low size. Instead, we can estimate the models from the choices the participant makes in each trial. If we assume that the participant chooses heads or tails in proportion to their probability, we can construct a likelihood function and identify which model provides the best fit for each participant.

Analyzing all 23 participants, 13.04% of participants made randomness judgments best characterized by the pushdown automaton, while the randomness judgments of the remaining 86.96% were best characterized by the stack automaton.[18] Hence, all participants utilized the notions of symmetry and complement symmetry to make randomness judgments, but a small minority did not associate duplication with non-randomness. For some participants, sequences such as `HHHTHHHT` look very random, even though they contain duplication. We are not generally arguing that some people's cognitive processes do not parse duplicated patterns, as `HHHT` repeated enough times would surely make people reject the sequence's randomness; but for our particular domain of binary sequences of length eight, existence of duplication may not provide enough evidence for some participants to reject the hypothesis of randomness.

The results of this experiment can also be used to further evaluate the performance of definitions of subjective randomness based on simple Turing machines (Gauvrit et al., 2014; 2016). Using the $D(5)$ model to define $P(x|\text{regular})$, we can compare the results directly to those for the probabilistic automata. The $D(5)$ model yields a log-likelihood of $-47,092$ and correlation of $r = 0.65$ (AIC = BIC = 94184), while the stack automaton gives a log-likelihood of $-44,820$ and correlation of $r = 0.92$ (AIC = 89650, BIC = 89685). As in Experiment 1, these results show that defining algorithmic complexity using simple Turing machines does well in predicting human judgment, but indicate that a closer fit to human data can be produced by using probabilistic automata to characterize the

[18] The same result was achieved whether the 22-motif or 4-motif model was used.

regularities people are sensitive to in binary sequences.

One weakness of the MCMCP approach is that drawing samples from $P(x|\text{regular})$ will not provide uniformly good estimates of the randomness of sequences. The issue is that sampling will provide better estimates of $P(x|\text{regular})$ for sequences where this probability is relatively high, simply because it will produce more samples of those sequences. By contrast, sequences with low probability under this distribution will, by definition, appear only rarely. This means that we get better estimates of highly regular sequences than highly random sequences.

While we do not perform such a study here, one strategy for addressing this weakness is to recursively apply MCMCP, drawing samples that push into increasingly low probability regions of the space. More formally, one can construct a sampling procedure that samples from the distribution that results from conditioning $P(x|\text{regular})$ such that $P(x|\text{regular}) < \epsilon$ for some probability ϵ. First, one uses MCMCP to get a rough estimate of $P(x|\text{regular})$. Then, one identifes all x such that the estimated probability under this distribution is greater than ϵ. Then one runs another round of MCMCP, but automatically rejects choices that fall into this set. The result will be a sample from $P(x|\text{regular})$ renormalized over low-probability values of x. This procedure can be repeated until a sufficiently high-resolution estimate of the distribution is obtained.

Mixtures of restricted distributions

Defining $P(x|\text{regular})$ using simpler computing machines is an effective strategy for defining a model of the subjective randomness of binary sequences: using this approach, we have been able to account for existing data on randomness judgments (Falk & Konold, 1997; Lopes & Oden, 1982), predict how people will classify binary sequences, and capture the kinds of regularities people seem to view as violations of randomness. However, it is not obvious how this approach can be applied with other kinds of stimuli. People have no difficulty assessing the randomness of individual numbers or arrays of points in space, but

these stimuli are harder than binary sequences to link to hypothetical computing machines.

To address subjective randomness judgments with a wider range of stimuli, we turn to the second approach to specifying $P(x|\text{regular})$: mixture distributions. Equivalence of statistical inference and algorithmic complexity is obtained when we take $P(x|\text{regular})$ to be a mixture of all computable distributions, as shown in Equation 4, but by restricting this set in specific domains we can obtain models of subjective randomness that are adapted to human judgments.

More precisely, we assume that in a given domain people maintain a set of hypotheses h about the kinds of regularity that can be exhibited in that domain. We then define $P(x|\text{regular})$ to be a mixture of the distributions associated with those hypotheses, namely

$$P(x|\text{regular}) = \sum_h P(x|h)P(h|\text{regular}) \tag{12}$$

where $P(x|h)$ is the distribution over stimuli x associated with hypothesized regularity h, and $P(h|\text{regular})$ corresponds to the weights with which these distributions are mixed, capturing the expected prevalence of those regularities.

In the remainder of the section we apply this approach to two other kinds of stimuli that have played a prominent role in the literature on subjective randomness: binary matrices and spatial clustering.

Binary matrices

Zhao, Hahn, and Osherson (2014) presented a series of experiments exploring human judgments of randomness in binary matrices. In particular, their Experiment 3 was designed to provide evidence against the idea that randomness is related to complexity as measured by encoding difficulty – the "encoding hypothesis" advocated by Falk and Konold (1997). In this section we show that these results are not problematic for our framework, and can in fact be accommodated with a relatively simple choice of regular generating process.

The experiment conducted by Zhao et al. had two conditions. In one condition, the researchers investigated the relative perceived randomness for 16×16 binary matrices. On every trial, the participant viewed a *mirror matrix* and a *switch matrix*. Each mirror matrix was generated by randomly sampling equiprobable values along one half of the matrix, and then reflecting the values over one of the matrix's diagonals to create symmetry. Each switch matrix was generated by populating its squares in either row-major or column-major order with a process which stochastically alternated with probability p between 0.1 and 0.9 in increments of 0.05. Participants were asked to indicate which matrix they thought was more random. The results are shown in Figure 11 — we see a familiar U-shape, where high and low rates of alternation increase the chance the participant views the mirror matrix as more random.

In the other condition of the experiment, participants were shown two matrices in succession and were asked whether a change had occured. On each trial, the computer programs randomly chose to either flip or not flip 10 bits from the first matrix. The results showed that participants were able to detect changes in switch matrices more easily than mirror matrices for every alternation rate – a different pattern of results than that seen for randomness judgments. Zhao et al. suggested that these results provide evidence against Falk and Konold's (1997) encoding hypothesis, which predicts that people would be able to encode less random-looking matrices into memory more quickly, and thus be able to detect changes more easily. Because their experiment showed that (1) mirror matrices are less random, and (2) changes in mirror matrices are more difficult to detect, they argued that the encoding hypothesis cannot be true.

Zhao et al.'s argument was specifically directed at Falk and Konold's (1997) interpretation of the relationship between complexity and randomness in algorithmic information theory. However, these results do not rule out the more general approach of explaining randomness perception in terms of algorithic complexity, particularly when subjective randomness is interpreted in terms of statistical inference rather than encoding.

Under the view we have presented in this paper, the reason why complexity matters to randomness is the evidence that it provides about the underlying generating process, which is independent of any kind of cognitive processing difficulty.

In support of this account, it is straightforward to show that the randomness judgments produced by the participants in Zhao et al.'s experiment can be accommodated within our framework. Assume that the participant evaluates randomness(x) for both the mirror matrix X_m and the switch matrix $X_s(p)$, and chooses the matrix for which this quantity is largest. Then we obtain the model

$$P(\text{choose } X_m) = \frac{1}{1 + \exp(-\psi - \lambda[\text{randomness}(X_m(p)) - \text{randomness}(X_s)])}, \tag{13}$$

where λ and ψ have a similar interpretation to our previous models: ψ is the log prior odds in favor of the mirror matrix and λ is the weight given to the statistical evidence of randomness.

We then need to define $P(X|\text{regular})$. We assume the same generative model as Zhao et al.: that regular matrices are generated by a switching process with unknown switching probability θ, giving

$$P(X|\theta) = \theta^s(1-\theta)^{n-s} \tag{14}$$

where s is the number of switches and n the total number of possible switches. We further assume that θ follows a Beta(α, β) distribution,

$$P(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \tag{15}$$

where Γ denotes the generalized factorial function, with $\Gamma(n) = (n-1)!$ when n is an integer. The mean of this distribution is $\alpha/\alpha+\beta$, so the relative values of α and β encode the bias towards or against switches in the prior. The shape of the distribution depends on the values of α and β: when both are greater than 1, it concentrates probability around the

mean; when both are less than 1, it concentrates probability around values of θ that are close to 0 and 1. Having defined this model in this way allows us to integrate over values of θ, to define $P(X|\text{regular})$ to be a Beta-Binomial(α, β) distribution,

$$\begin{aligned} P(X|\text{regular}) &= \int P(X|\theta)P(\theta)\, d\theta && (16) \\ &= \int \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{s+\alpha-1}(1-\theta)^{n-s+\beta-1} && (17) \\ &= \frac{\Gamma(\alpha+\beta)\Gamma(s+\alpha)\Gamma(n-s+\beta)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(n+\alpha+\beta)}. && (18) \end{aligned}$$

Under this distribution, all that matters is the number of switches in the matrix s and the size of the matrix n.

Since the mirror matrices were generated by flipping a fair coin and then reflecting, the number of switches in those matrices is roughly constant. As a result, our model simplifies to

$$P(\text{choose } X_m) = \frac{1}{1+\exp(-\psi + \lambda \text{randomness}(X_s(p)))} \quad (19)$$

where ψ absorbs the constant for the randomness of X_m. The data collected by Zhao, Hahn, and Osherson (2014) did not include the specific matrices their participants saw because they were generated on the fly. As a result, we do not know what X_m or $X_s(p)$ look like on each trial. Fortunately, we can sample from a population of matrices to generate statistics with which we can build a model. We then estimate the mean number of switches that appears in a switch matrix $X_s(p)$ for each alternation rate p with a simple Monte Carlo estimate.

The model parameters were estimated by fitting to the data from Zhao et al., yielding $\alpha = 0.617, \beta = 0.566, \lambda = 13.2, \psi = 2.98$. This corresponds to a distribution $P(X|\text{regular})$ in which switching and not-switching are considered roughly equally likely, but where matrices that contain either a lot of switches or very few are both given high probability. The results are shown in Figure 11. While the absolute probabilities depend on ψ and λ,

the U-shape depends only on α and β (and is also produced with any other value of α and β less than 1). The resulting model yields a rank-order correlation coefficient of $\rho = 0.97$. These results demonstrate that there is no theoretical obstacle to explaining the results of this experiment within a framework based on algorithmic complexity.

Spatial clustering

During the second World War London was repeatedly hit by German rockets. When newspapers published the locations of such bombings, people saw a pattern: the Germans seemed to be targeting certain parts of London more than others, especially the poorer parts (Johnson, 1981). After the war, an analysis by Clarke (1946) revealed that the pattern of bombings did not deviate from a uniform distribution over the city – the apparent clusters were the result of chance.

What makes people think an array of points in space is not random? This question can be answered using the framework we have developed so far. Under $P(x|\text{random})$, each bomb $i = 1, \ldots, n$ falls in a location x_i sampled uniformly at random. $P(x|\text{regular})$ is defined by taking a mixture of hypotheses. In each hypothesis, each bomb i is aimed at a common target location μ with probability α and is otherwise strikes a random location with probability $1 - \alpha$. To account for the possibility that the rocket misses the common target, we assume that the actual location where the bomb explodes, x_i, is selected from a two-dimensional Gaussian distribution with mean μ and covariance matrix Σ.

The distributions used in Bayesian inference are then as follows. For $P(x|\text{random})$ we have

$$P(x_1, \ldots, x_n|\text{random}) \approx \left(\frac{1}{|\mathcal{R}|}\right)^n \tag{20}$$

where $\mathcal{R}$ is the region over which the bombs are falling and $|\mathcal{R}|$ is its area. For

$P(x|\text{regular})$, we have the mixture

$$P(x_1, ..., x_n|\text{regular}, L_c, \Sigma) = \int\int \prod_{i=1}^{n} \left(\alpha\phi(x|\mu, \Sigma) + (1-\alpha)\frac{1}{|\mathcal{R}|} \right) d\mu \, d\Sigma \tag{21}$$

where $\phi(x|\mu, \Sigma)$ is the probability density function of the two-dimensional Gaussian with mean μ and covariance matrix Σ.

With these two distributions defined we can once again use our measure of randomness, Equation 2, to see how well the model predicts human judgments about random clustering in two-dimensional space. To evaluate the predictions of the model we conducted an experiment in which we asked people to judge the randomness of different spatial arrays.

Experiment 3: Spatial clustering

Method

Participants. Participants were 118 undergraduates, participating for course credit.

Stimuli. The stimuli were taken from Griffiths and Tenenbaum (2007), and consisted of 12 images which contained points at different locations within a 10 by 10 square, with each axis spanning from -5 to 5. No markings or axes were displayed to the participants; they are only described here to make our calculations sensible to the reader. Nine of the stimuli were generated from the mixture of the uniform and Gaussian distributions described above, each using a different set of parameters. The varied parameters were (1) number of bombs, n, (2) proportion of bombs generated from the clustering point, α, (3) the location of the cluster, μ, and (4) the spread of the cluster, Σ. Each parameter had a basic value, each of which was varied twice to yield nine different images. The range of parameter values is shown in Table 2. The remaining three stimuli were generated by sampling from the uniform distribution over the square.

Procedure. Participants completed the questionnaire for this experiment as part of a booklet of other short psychology experiments. Each participant saw all 12 images, in one of six random orders. The instructions read as follows:

> Each of the images below shows where bombs landed in a particular part of London for a given month, with a single point for each bomb. On the lines at the bottom of the page corresponding to each image, please rate HOW RANDOM the distribution of bombs seems to you. Use a scale from 1 to 10, where 1 means 'Not at all random', and 10 means 'Very random'.

Participants then provided ratings for all 12 images, presented in one of six random orders.

Results and Discussion

Model predictions were evaluated using the Monte Carlo method outlined in Griffiths and Tenenbaum (2007). The mean responses given by participants and the model's predictions are shown in Figure 12. The rank-order correlation between the randomness model's predictions and the responses (which makes no assumption about the form of the relationship other than that it is monotonic) is $\rho = 0.769$. The values given by the model shown in the figure were transformed by the equation $y = \text{sign}(z)\text{abs}(z)^{\gamma}$ where $z = \text{randomness}(x)$ and $\gamma = 0.05$, and yield linear correlation of $r = 0.951$ with the data.

The model does a fairly good job of capturing people's intuitions about randomness, or at least the distinction between arrays that are considered random and arrays that are not. However, there are also some discrepancies between the model predictions and the data. For example, the Bayesian model predicts an effect of the number of points in the array that is not observed in the human data. This may be a consequence of a floor effect, since the judged randomness of these stimuli is consistently low, but it may also indicate that people are less sensitve to this aspect of the data than an account based on statistical inference would predict. For example, human perception may group together points and therefore prevent forming an accurate estimate of the number that appear within a cluster.

The fact that our simple statistical model does so well in the absence of any perceptual modeling is thus all the more impressive.

General Discussion

One of the challenges of understanding subjective randomness is developing an appropriate definition of randomness that we can use to make sense of human cognition. In this paper we have argued that such a definition can be found if we recognize that subjective randomness is fundamentally a statistical inference – an inference about the process that generated observed data. Under this definition, stimuli that seem random are those that provide the strongest evidence for a random generating process over some other more regular process. The challenge is then to define what constitutes a regular generating process.

Algorithmic information theory provides a starting point for characterizing regularity – defining regular generating processes using distributions specified using universal Turing machines results in a standard definition of randomness in terms of algorithmic complexity. However, this characterization of regularity is far too general – it includes regularities people are unlikely to detect and is incomputable. In order to develop an account of human subjective randomness based on this approach, we need to restrict the kinds of regularity included in the model. Inspired by the two ways of defining probability distributions based on computing machines, we have shown how restrictions can be introduced either on the computing machines used to define the distributions (as in previous work by Griffiths and Tenenbaum (2003; 2004) and Gauvrit et al. (2014; 2016)) or on components of a mixture of regular distributions.

Our results illustrate how these two approaches can be used to capture human subjective randomness for a range of different stimuli. For the binary sequences that have classically been studied in the psychological literature on subjective randomness, a model in which regularity is defined using a finite state automaton augmented with the capacity

to identify symmetry and duplication provides a good characterization of people's randomness judgments and of the kind of regularities they expect to see in binary sequences (with symmetry not being necessary if those sequences are presented sequentially, one element at a time). This approach also outperforms an approach based on using simple Turing machines (Gauvrit et al., 2014; 2016), although this approach shows a lot of promise as an alternative way of constraining algorithmic complexity. For binary matrices, a simple model of regularity can explain results that have previously been taken as providing evidence against accounts of subjective randomness based on complexity. Finally, taking a mixture of possible clusters allows us to predict whether people will consider a spatial array of dots to be random or not.

In the remainder of the paper we consider four questions raised by these results: Where do regularities come from? Are people bad at assessing randomness? How does randomness generation relate to perception? And how is randomness linked to coincidences?

Where do regularities come from?

A key insight behind our approach is that defining randomness really requires defining regularity: to provide evidence for a random generating process, a stimulus needs to be inconsistent with expectations about regular generating processes. In each of the domains we studied, the set of regularities we considered was different – one of the disadvantages of abandoning the universality of algorithmic complexity is that we need to consider a distinct set of detectable regularities in each domain. This is not a weakness of the approach – it is necessary in order to actually develop an account of subjective randomness in each of these domains, and the experimental method that we used in Experiment 2 shows how regular generating processes can be identified and then used to model randomness judgments. However, an important question remains: in each domain, why are these the regularities that people are sensitive to?

In the spirit of Anderson's (1990) approach of rational analysis, we anticipate that the answer to this question lies in the environment in which human cognition and perception are carried out. Detecting randomness is not, in itself, an ability that is likely to have adaptive consequences, but detecting violations of randomness – recognizing that a pattern may be present – is an important human ability. The regularities that we expect people to be sensitive to are thus those that are likely to indicate an underlying causal relationship, capturing the kinds of patterns that appear (and have consequences) in human environments. For example, the simple clustering model used in Experiment 3 can naturally be described in terms of a causal process that influences the location of points (see Griffiths & Tenenbaum, 2007, for examples of such descriptions).

Placing the source of regularity in the environment raises the tantalizing idea that we might be able to develop appropriate models of $P(x|\text{regular})$ by measuring the structure of human environments. Support for this idea comes from the results of Hsu, Griffiths, and Schreiber (2010), who showed that people's judgments of the randomness of small binary arrays (4×4 grids in which each cell was colored black or white) could be predicted well by the frequency with which a similar pattern appeared in images of natural scenes. Patterns that were less frequent were considered more random, just as would be expected if $P(x|\text{regular})$ assigned higher probability to those patterns that appeared in the natural world. Conducting similar studies with other stimuli would provide a way to evaluate whether this account holds more broadly.

Are people bad at assessing randomness?

As we discussed at the start of the paper, much previous research on subjective randomness has argued that people are irrational and have a flawed notion of the workings of chance. While it is certainly true that people make mistakes when reasoning about chance – such as considering `HHHHHHHH` to be less likely than `HTHTTHTT` as the result of flipping a fair coin – we have argued that at least some of people's inferences can be

reconciled with probability and statistics when the problem that they seem to be solving is appropriately characterized. Specifically, if we consider people to be inferring the process that generated a stimulus – computing $P(\text{random}|x)$ rather than $P(x|\text{random})$ – then their judgments can be seen to be consistent with the Bayesian models we have presented in this paper.

This view of subjective randomness as a relatively well-calibrated statistical inference is consistent with other recent research on randomness judgment. Williams and Griffiths (2013) showed that judging the randomness of binary sequences is fundamentally a difficult problem: analzing the task as one of Bayesian inference, they showed that it is only ever possible to obtain weak evidence in support of a random generating process (while regular processes can receive strong evidence). This asymmetry in the strength of evidence is characteristic of randomness judgments but not of other kinds of decision tasks – it is an intrinsic consequence of the structure of the statistical problem being solved. Once this is taken into account, people turn out to be no worse at judging randomness than making other kinds of statistical decisions, and in fact perform very similarly to optimal Bayesian models.

In line with these results, the models we used to capture people's assessment of binary sequences in this paper indicate that a surprisingly high proportion of coin flip sequences should be classified as non-random. Equation 10 measures the probability that a sequence x being random. Using the equation, we can employ a classification procedure: if $P(\text{random}|x) < 0.5$, then the sequence is classified as non-random, as there is more evidence indicating it is non-random. To simplify our analysis, we fix $\lambda = 1$ (corresponding to a well-calibrated Bayesian inference), and use the best-fit parameters for the stack automaton in Experiment 3 as the definition of randomness(x) to yield an equation whose only unknown variable is ψ, the log prior odds

$$P(\text{random}|x) = \frac{1}{1 + \exp(\text{randomness}(x) - \psi)} \tag{22}$$

where positive values of ψ indicate a bias in favor of the random process. We can now plot Equation 22 as a function of ψ (Figure 13). We find that even when $\psi = 0$, which corresponds to no a priori bias towards either random nor regular events, the percentage of sequences classified as non-random is 28.15%. When $\psi = -0.5$, indicating that the observer has a slight bias towards perceiving regular events, 46.09% of sequences are classified as non-random. Depending on the choice of ψ, our ideal observer analysis suggests that about one quarter to one half of the 256 possible coin flip sequences are seen as non-random occurrences. This intrinsic asymmetry is an important factor in understanding why judging randomness is hard, and why we should expect people to have difficulty with such tasks even when they are approaching optimal performance.

How does randomness generation relate to perception?

Our focus in this article has been on explaining the perception of randomness. However, an equally fascinating question is how people generate randomness. The emphasis that we are placed on perception is a consequence of the significant role that detecting deviations from randomness may have played in the evolution of human cognition. By contrast, the primary function of randomness generation arises in adversarial settings, where one needs to act in a way that cannot be predicted by an opponent. In such settings, acting in a way that is genuinely random is less important than acting in a way that an opponent perceives to be random. As a consequence, we should expect that people's perception of randomness should also play a role in the generation of randomness. That is, we should expect that people are trying to generate sequences that they themselves perceive to be random.

Griffiths and Tenenbaum (2001) explored this possibility, showing that the kind of statistical model of subjective randomness we have defined here could be leveraged in order to predict people's generation of binary sequences. During the 1930s the Zenith Radio Corporation conducted a series of broadcasts in which they tried to test people's psychic

abilities. During the broadcast a psychic would transmit a binary sequence of length five, and listeners will write down the sequence that they "received" and mail it to the radio station. The results provide a nice picture of human randomness generation for binary sequences. Griffiths and Tenenbaum suggested that these data could be modeled by assuming that each time people generated the next element in their sequence, they were considering the contribution that element would make to its overall randomness. For example, having already generated `HHT` they would evaluate the relative randomness of `HHTH` and `HHTT` when deciding whether the next element should be `H` or `T`. Using a model similar to that we used for binary matrices, where only the relative number of heads and tails mattered, Griffiths and Tenenbaum showed that this approach could provide a remarkably good account of the Zenith radio data.

The analysis presented by Griffiths and Tenenbaum revealed two interesting features of randomness generation. First, in order to account for people's behavior they had to assume that the evaluation of the randomness of the binary sequences being produced also included its most recent subsequences. So in evaluating `HHTT`, people also considered the randomness of `HTT` and `TT`. This kind of sensitivity to local structure is compatible with the idea of "local representativeness" that Kahneman and Tversky (1972) appealed to in their discussion of subjective randomness. Second, the most significant failure of the model was predicting the most probable random sequence to be `HTHTH`, which people actually generated with relatively low probability. This is perhaps a consequence of the global symmetry of the sequence, something that the more sophisticated models we have considered in this paper may be able to redress. Having developed more accurate models of randomness perception opens the door to exploring randomness generation across a variety of domains.

How does randomness link to coincidences?

Subjective randomness is just one of many phenomena related to the perception of chance. Another equally striking phenomenon is that of coincidence: when we observe an event and it strikes us as being unlikely to be the result of chance. Intuitively, randomness and coincidences seem related to one another. Our formal framework makes this relationship explicit.

Griffiths and Tenenbaum (2007) presented a mathematical definition of coincidences: events that provide strong evidence for a hypothesis that was previously considered unlikely to be true. When the current hypothesis is that no causal relationship or causal force exists, and the alternative is that it does, this becomes the comparison of the hypothesis of randomness against some other, more regular process. Consequently, Griffiths and Tenenbaum (2007) defined the strength of a coincidence to be the likelihood ratio in favor of the hypothesis of a regular generating process over a random generating process

$$\text{coincidence}(x) = \log \frac{P(x|\text{regular})}{P(x|\text{random})}. \tag{23}$$

Comparison of this equation with Equation 2 reveals a clear prediction of our account: that randomness and coincidence should be inversely related.

The stimuli we used in Experiment 3 were drawn from an experiment conducted by Griffiths and Tenenbaum (2007) in which participants were asked to judge the strength of coincidence of different patterns of bombing. As a consequence, we can directly compare the coincidence judgments and the randomness judgments that were given for these stimuli. The correlation between the two sets of judgments was $r = -0.96$, consistent with the prediction that the should be inversely related. Randomness and coincidences may thus be considered two sides of the same (tossed) coin, both being an inference about generating processes but supporting opposite conclusions.

Conclusion

We have shown that people's judgments of subjective randomness for a range of different stimuli can be captured in a single formal framework, which has deep links to ideas from statistics and algorithmic information theory. At the heart of this framework is the idea that subjective randomness is statistical inference: an inference about the process that generated the observed data. This idea helps to clarify how randomness can be a property of a stimulus, not just a process, and provides a foundation for exploring what makes something random – and, conversely, what regularities people are sensitive to – in any domain of interest. In addition to evaluating this formal framework, our results illustrate how it can be extended to other domains, providing an empirical method for identifying regular generating processes. Using these tools, it should be possible to pin down what makes something seem random – and what does not – across the full scope of human experience.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csáki (Eds.), *2nd International Symposium on Information Theory* (p. 267-281).

Baddeley, A. D. (1966). The capacity of generating information by randomization. *Quarterly Journal of Experimental Psychology*, *18*, 119-129.

Bar-Hillel, M., & Wagenaar, W. A. (1991). The perception of randomness. *Advances in Applied Mathematics*, *12*, 428-454.

Budescu, D. V. (1987). A Markov model for generation of random binary sequences. *Journal of Experimental Psychology: Human Perception and Performance*, *12*, 25-39.

Chaitin, G. J. (1969). On the length of programs for computing finite binary sequences: statistical considerations. *Journal of the ACM*, *16*, 145-159.

Chater, N. (1999). The search for simplicity: A fundamental cognitive principle? *Quarterly Journal of Experimental Psychology*, *52A*, 273-302.

Chomsky, N. (1959). A review of B.F Skinner's Verbal Behavior. *Language*, *31*, 26-58.

Clarke, R. D. (1946). An application of the Poisson distribution. *Journal of the Institute of Actuaries (London)*, *72*.

Delahaye, J.-P., & Zenil, H. (2012). Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of randomness. *Applied Mathematics and Computation*, *219*, 63–77.

Falk, R. (1981). The perception of randomness. In *Proceedings of the fifth international conference for the psychology of mathematics education* (Vol. 1, p. 222-229). Grenoble, France: Laboratoire IMAG.

Falk, R., & Konold, C. (1997). Making sense of randomness: Implicit encoding as a bias for judgment. *Psychological Review*, *104*, 301-318.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature*, *407*, 630-633.

Feldman, J. (2003). A catalog of Boolean concepts. *Journal of Mathematical Psychology*, *47*, 98-112.

Feldman, J. (2004). How surprising is a simple pattern? Quantifying "Eureka!". *Cognition*, *93*, 199-224.

Gauvrit, N., Singmann, H., Soler-Toscano, F., & Zenil, H. (2016). Algorithmic complexity for psychology: A user-friendly implementation of the coding theorem method. *Behavior research methods*, *48*, 314–329.

Gauvrit, N., Zenil, H., Delahaye, J.-P., & Soler-Toscano, F. (2014). Algorithmic complexity for short binary strings applied to psychology: a primer. *Behavior research methods*, *46*, 732–744.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (1995). *Bayesian data analysis*. New York: Chapman & Hall.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721-741.

Gilks, W., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1996). *Markov chain Monte Carlo in practice*. Suffolk, UK: Chapman and Hall.

Gilovich, T., Vallone, R., & Tversky, A. (1985). The hot hand in basketball: On the misperception of random sequences. *Cognitive Psychology*, *17*, 295–314.

Griffiths, T. L., & Tenenbaum, J. B. (2003a). From algorithmic to subjective randomness. In *Advances in Neural Information Processing Systems 16.*

Griffiths, T. L., & Tenenbaum, J. B. (2003b). Probability, algorithmic complexity, and subjective randomness. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society.*

Griffiths, T. L., & Tenenbaum, J. B. (2007). From mere coincidences to meaningful discoveries. *Cognition*, *103*, 180-226.

Hahn, U., & Warren, P. A. (2009). Perceptions of randomness: why three heads are better

than four. *Psychological Review*, *116*, 454-461.

Hsu, A., Griffiths, T. L., & Schreiber, E. (2010). Subjective randomness and natural scene statistics. *Psychonomic Bulletin & Review*, *17*, 624-629.

Johnson, D. (1981). *V-1, V-2: Hitler's vengeance on London.* New York: Stein & Day.

Johnson, V. E. (1996). Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *Journal of the American Statistical Association*, *91*, 154–166.

Kac, M. (1983). What is random? *American Scientist*, *71*, 405-406.

Kahneman, D., & Tversky, A. (1972). Subjective probability: A judgment of representativeness. *Cognitive Psychology*, *3*, 430-454.

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

Kareev, Y. (1995a). Positive bias in the perception of covariation. *Psychological Review*, *102*, 490-502.

Kareev, Y. (1995b). Through a narrow window: Working memory capacity and the detection of covariation. *Cognition*, *56*, 263-269.

Kass, R. E., & Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, *90*, 773-795.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, *1*, 1-7.

Kubovy, M., & Psotka, J. (1976). The predominance of seven and the apparent spontaneity of numerical choices. *Journal of Experimental Psychology: Human Perception and Performance*, *2*, 291-294.

Li, M., & Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications* (3rd ed.). New York: Springer Verlag.

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

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

Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing.* Cambridge, MA: MIT Press.

Martin, J. B., Griffiths, T. L., & Sanborn, A. N. (2012). Testing the efficiency of Markov chain Monte Carlo with people using facial affect categories. *Cognitive Science*, *36*, 150–162.

Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods* (Tech. Rep. No. CRG-TR-93-1). University of Toronto.

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

Nickerson, R. S. (2002). The production and perception of randomness. *Psychological Review*, *109*, 330-357.

Rabinowitz, F. M., Dunlap, W. P., Grant, M. J., & Campione, J. C. (1989). The rules used by children and adults to generate random numbers. *Journal of Mathematical Psychology*, *33*, 227-287.

Reichenbach, H. (1934/1949). *The theory of probability.* Berkeley: University of California Press.

Sanborn, A. N., Griffiths, T. L., & Shiffrin, R. (2010). Uncovering mental representations with Markov chain Monte Carlo. *Cognitive Psychology*, *60*, 63-106.

Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461-464.

Shannon, C. E. (1948). The mathematical theory of communication. *Bell System Technical Journal*, *27*, 379-423, 623-656.

Sipser, M. (2012). *Introduction to the theory of computation.* Boston, MA: Cengage Learning.

Skinner, B. F. (1942). The processes involved in the repeated guessing of alternatives. *Journal of Experimental Psychology*, *39*, 322-326.

Soler-Toscano, F., Zenil, H., Delahaye, J.-P., & Gauvrit, N. (2014). Calculating kolmogorov complexity from the output frequency distributions of small turing machines. *PloS One*, *9*, e96223.

Solomonoff, R. J. (1964). A formal theory of inductive inference. part i. *Information and Control*, *7*, 1-22.

Spencer-Brown, G. (1957). *Probability and scientific inference*. London: Longmans Green.

Tune, G. S. (1964). Response preferences: A review of some relevant literature. *Psychological Bulletin*, *61*, 286-302.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *2*, 230-265.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: heuristics and biases. *Science*, *185*, 1124-1131.

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

Wiegersma, S. (1982). Can repetition avoidance in randomization be explained by randomness concepts? *Psychological Research*, *44*, 189-198.

Williams, J. J., & Griffiths, T. L. (2013). Why are people bad at detecting randomness? A statistical argument. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *39*, 1473-1490.

Zhao, J., Hahn, U., & Osherson, D. (2014). Perception and identification of random events. *Journal of Experimental Psychology: Human Perception and Performance*, *40*(4), 1358.

Table 1
Performance of probabilistic models approximating $P(x|regular)$.

No. of Motifs	Machine Type	Log-Likelihood	Corr.	5-Fold CV Corr.
4	Finite State	-46,240	0.74	0.59
	Pushdown	-45,084	0.86	0.76
	Queue	-45,478	0.84	0.66
	Stack	-44,820	0.91	0.77
22	Finite State	-45,866	0.80	0.28
	Pushdown	-44,864	0.90	0.43
	Queue	-45,286	0.87	0.40
	Stack	-44,647	0.92	0.47

Table 2
Parameters used for stimuli in Experiment 3.

Property	Parameters		
Number	$n = 20$	$n = 50$	$n = 200$
Proportion	$\alpha = 0.5$	$\alpha = 0.3$	$\alpha = 0.1$
Location	$\mu = \begin{bmatrix} -3 \\ -3 \end{bmatrix}$	$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\mu = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$
Spread	$\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$	$\Sigma = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$	$\Sigma = \begin{bmatrix} \frac{1}{5} & 0 \\ 0 & \frac{1}{5} \end{bmatrix}$

Note: The default values of the parameters for the experimental stimuli were $n = 50$, $\alpha = 0.3$, $\mu = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$, and $\Sigma = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$. The stimuli shown in Figure 12 thus correspond to modifying one parameter setting away from this default, with the order matching that given here.

A longer program is required to produce sequences of high complexity.	*A shorter program can produce sequences of low complexity.*

`function complexSequence():`
`  print(H)`
`  print(T)`
`  print(T)`
`  print(H)`
`  print(H)`
`  print(T)`
`  print(H)`
`  print(T)`
	`function notSoComplexSequence():`
	`  for i = 1..8:`
	`  print(H)`

Figure 1. Algorithmic complexity. Random sequences are more complex, being harder to generate using simple programs. The function `complexSequence` outputs the sequence `HTTHHTHT`, and the function `notSoComplexSequence` outputs the sequence `HHHHHHHH`.

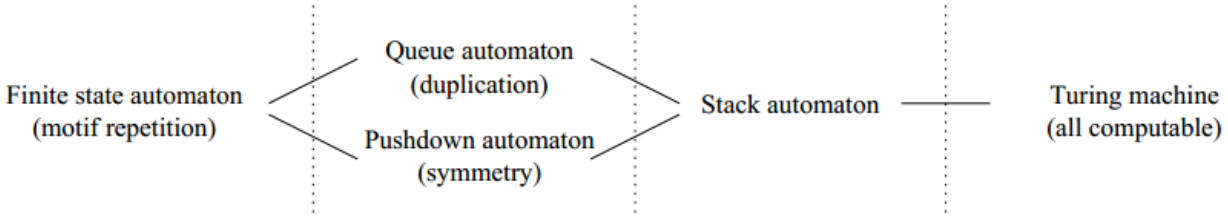

Figure 2. Hierarchy of computing machines. The regularities each machine is capable of recognizing is written below in parentheses. Each machine to the left is a special case of the right.

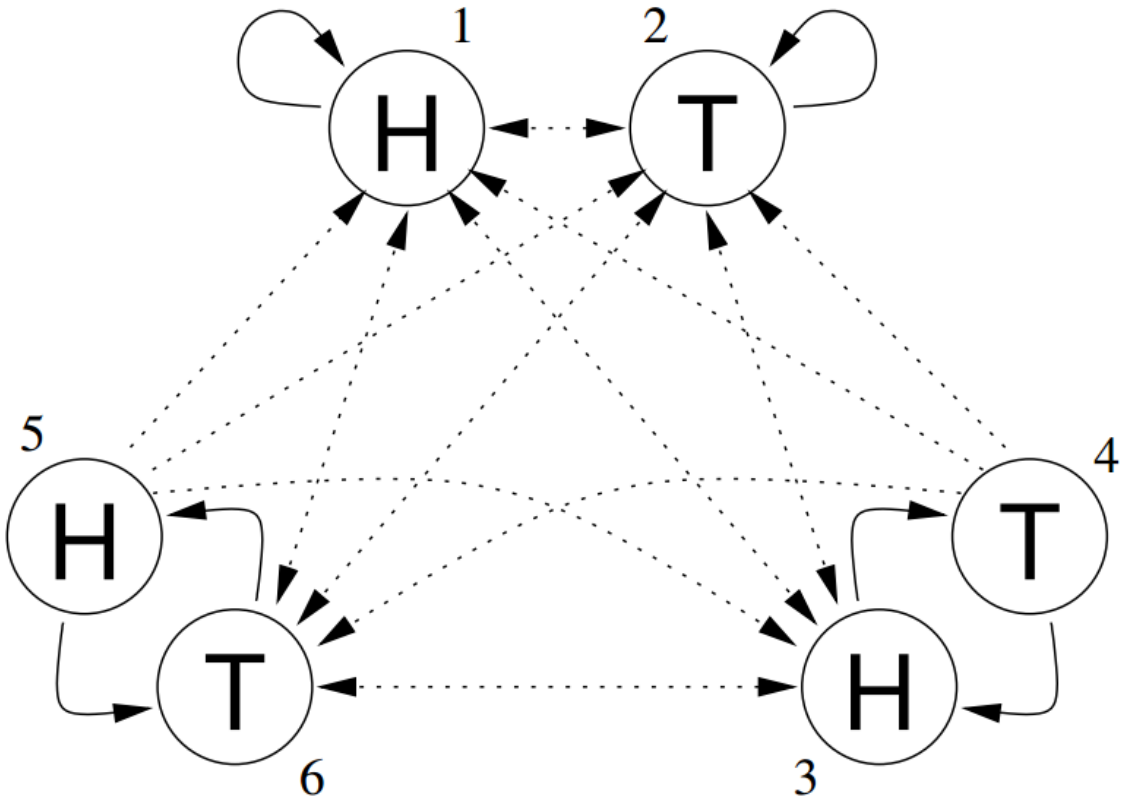

Figure 3. State diagram for the hidden states in the hidden Markov model. Solid arrows indicate continuation of a motif, whereas dashed arrows indicate permissible changes between motifs. The numbers used for each state correspond to the rows in the transition matrix (Equation 9).

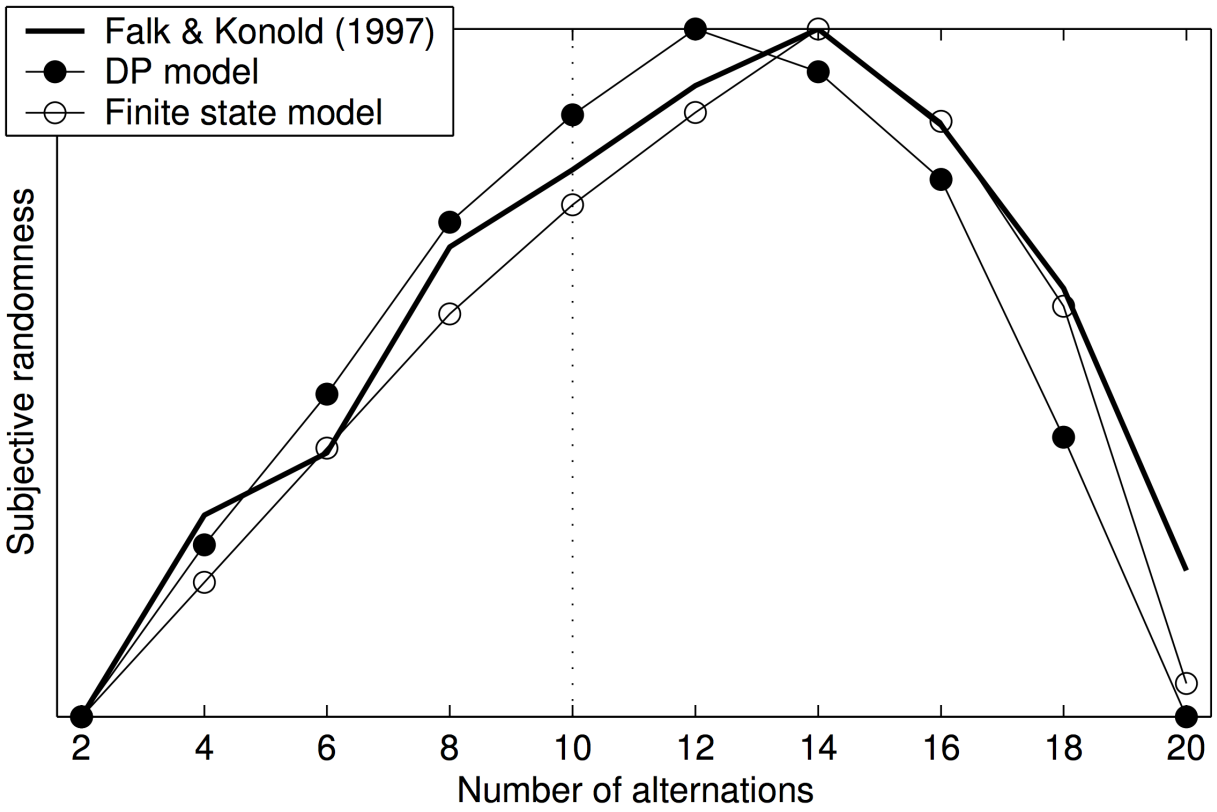

Figure 4. Subjective randomness and probability of alternation. The solid line shows the participants' mean apparent randomness ratings of binary sequences plotted against the number of alternations from Experiment 1 in Falk and Konold (1997). The dashed line at 10 provides a baseline for the number of alternations expected from a binary sequence of length 20. The two other lines show the predictions of the DP and our finite state automaton.

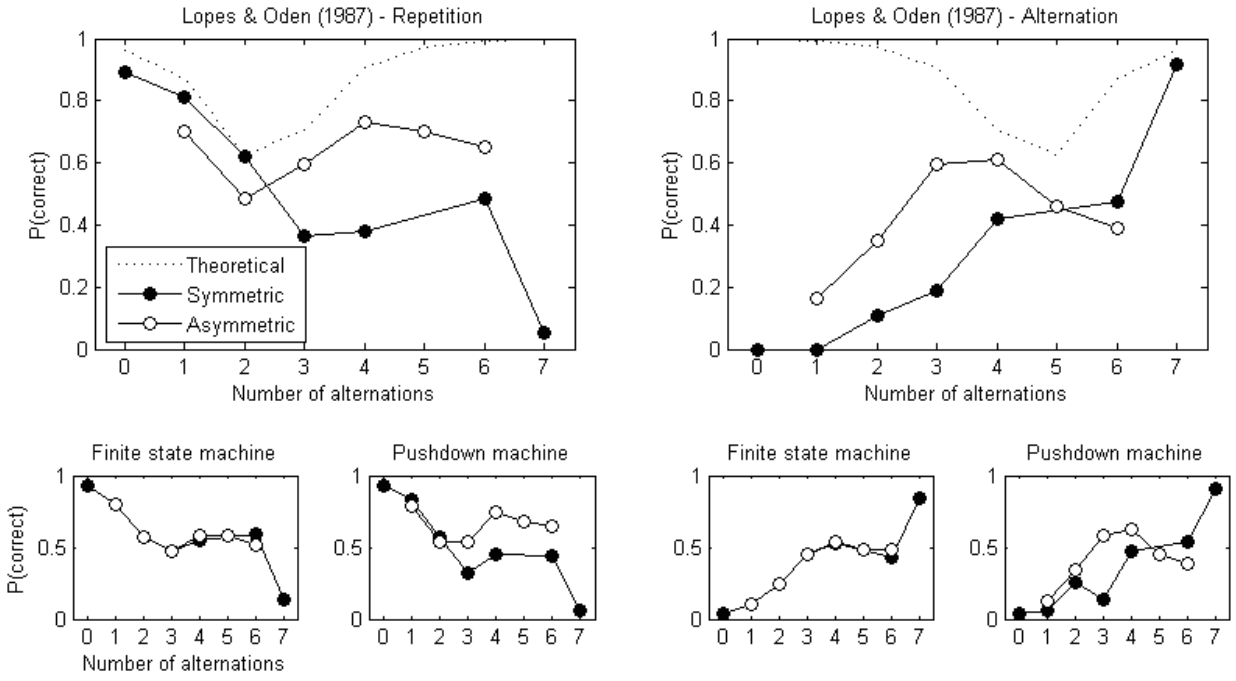

Figure 5. Results from Experiment 1 of Lopes and Oden (1987). The top-left panel shows the percentage of correct responses ("random" or "non-random") given by participants when the non-random source was composed of repetition-biased sequences. The top-right panel shows the same information but for the group that saw alternation-biased sequences. The results are decomposed into two plots, depending on whether the sequence shown was symmetric or asymmetric. The dashed line shows the theoretical performance of an observer who had perfect knowledge of the nature of the non-random source. The bottom four panels show the results that are obtained if we use our model of randomness, where the underlying machine used to judge non-random sequences is either a finite state automaton or a pushdown automaton. The pushdown automaton is able to capture the effects of symmetry, and thus more closely approximates people's intuitions about randomness.

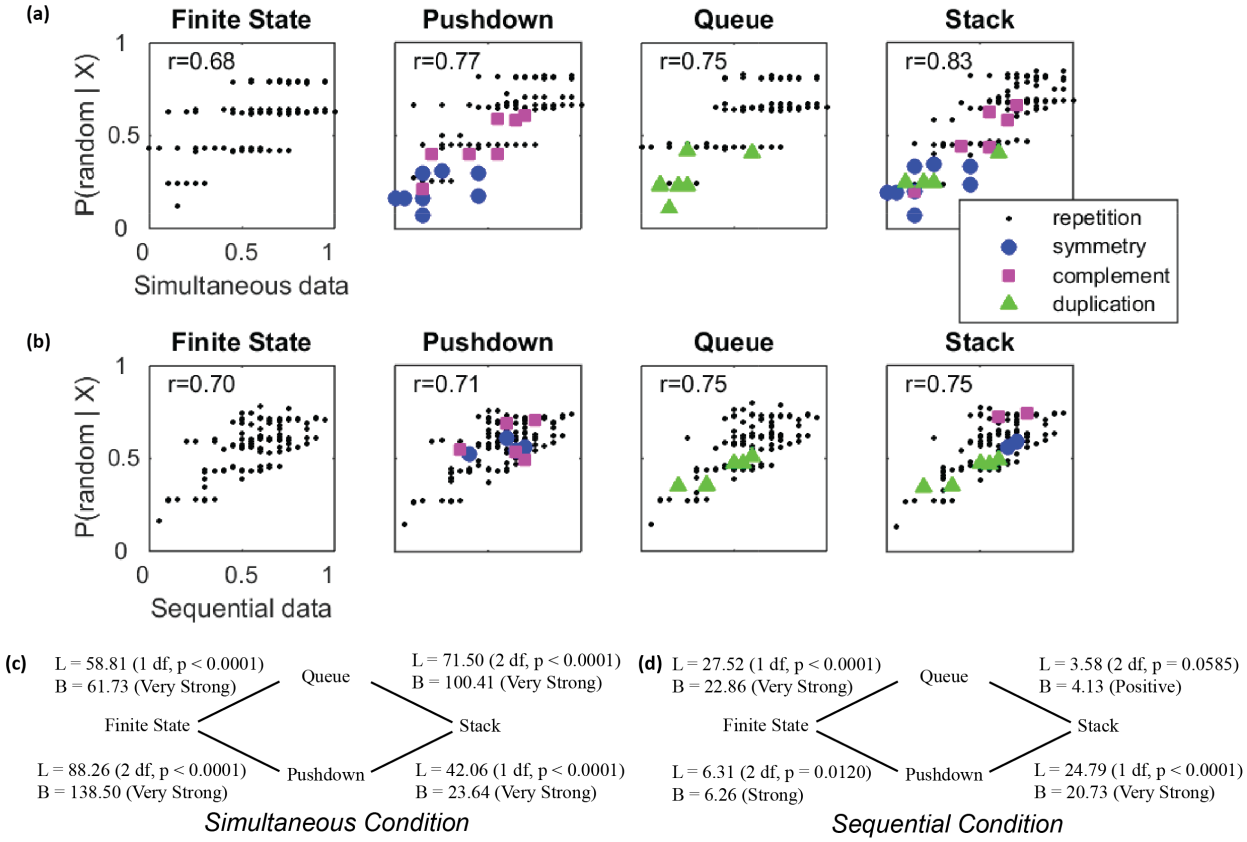

Figure 6. Results of Experiment 1 for (a) the Simultaneous condition and (b) the Sequential condition. The horizontal axis measures the fraction of participants labeling a sequence as random and the vertical axis shows the model's prediction. Each point is categorized according to its highest probability parse under each model—the legend can be found on the right side beneath the stack model in (a). For example, for the stack model in the Simultaneous condition, the sequence `HHHHHHHH` reaches its highest-probability representation when the machine produces it via symmetry; thus, the sequence is represented on the graph as a blue circle. Diagrams (c) and (d) show the χ^2 values, denoted L (df, p), and twice the natural logarithm of the Bayes factor, denoted B, of each model comparison. Listed alongside the B value value is the recommended interpretation of the relative evidence in favor of the alternative model according to the criteria used in Kass and Raftery (1995).

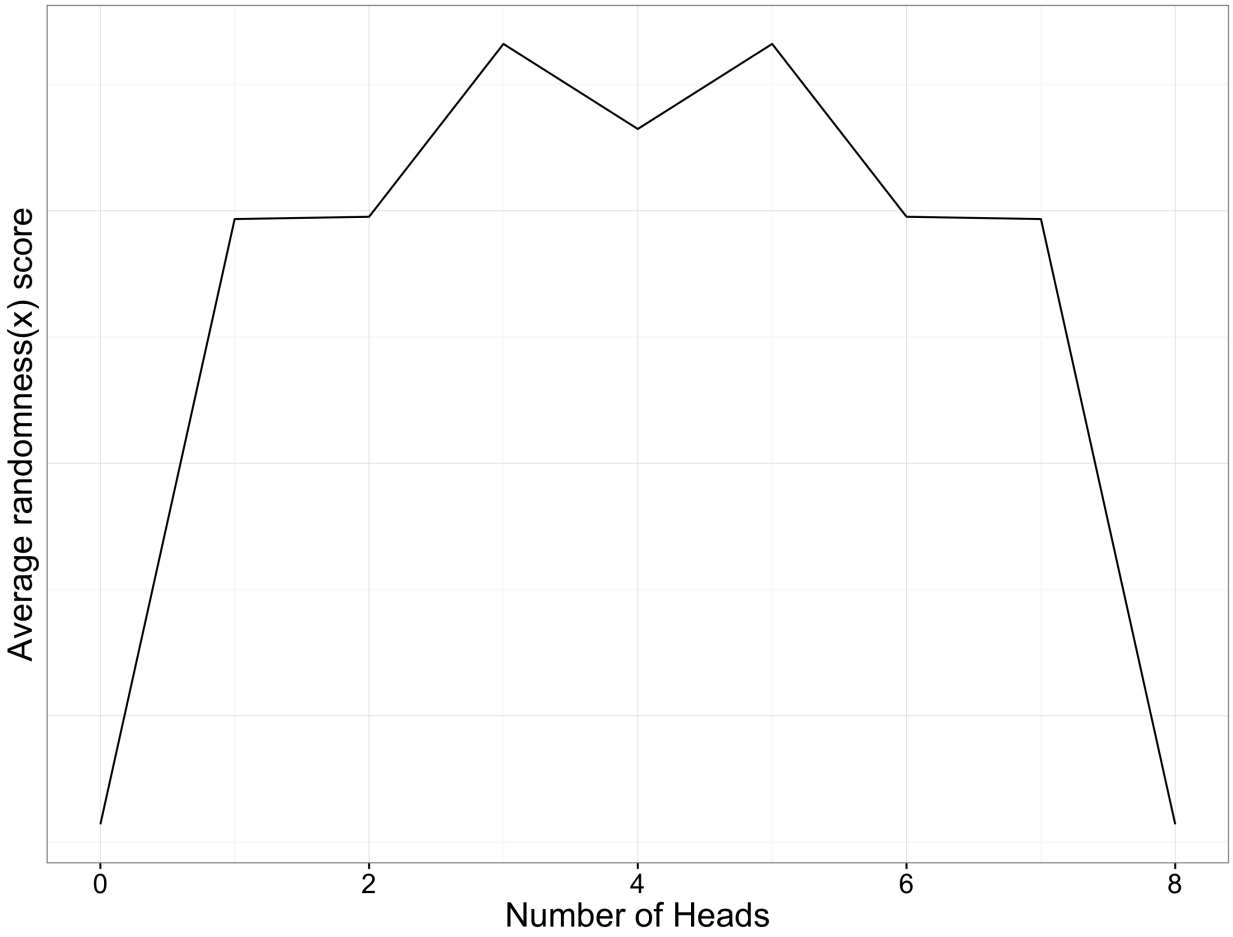

Figure 7. Effect of number of heads on the predictions of the stack automaton model used in Experiment 1. Deviations from an even division between heads and tails are predicted to be more random.

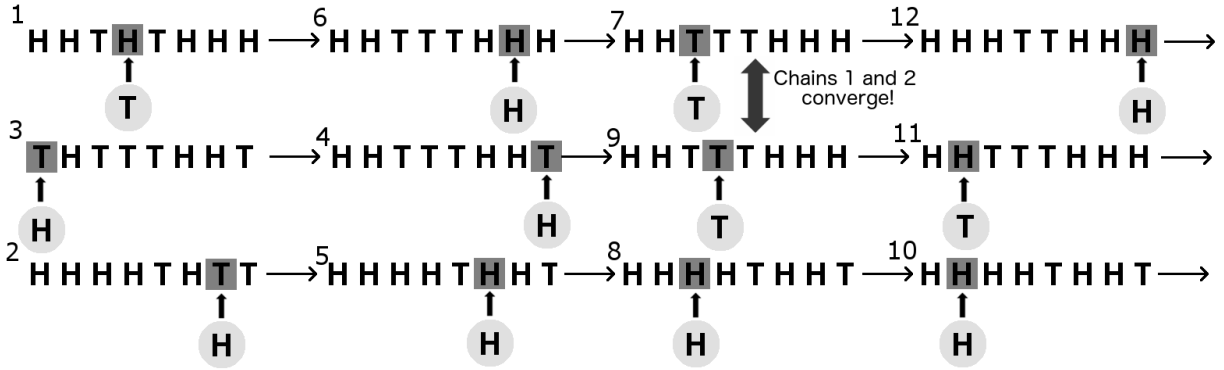

Figure 8. Experiment 2 uses Gibbs sampling with people to obtain an estimate of $P(x|\text{regular})$. 40 initial sequences are generated for each participant at random (only three are shown here to demonstrate). In each trial of the experiment, the participant views a sequence of coin flips with one symbol completely covered by a black box. In this figure the black box is made transparent in order to help explain the method – it was completely opaque for participants. The participant is asked to evaluate whether heads or tails would be more likely to appear in the missing spot if the sequence were generated by a non-random process. Participant responses are displayed in green. Notice how the participant's selection modifies the sequence for the next step in each chain. The participant completes trials corresponding to each iteration fo the three different chains in random order—one potential order is indicated by the numbers in the upper left corner of each sequence. In this case the first trial corresponds to the first chain, the second trial to the third chain, the third trial to the second chain, and then the fourth trial goes back to the second chain and the fifth to the third chain.

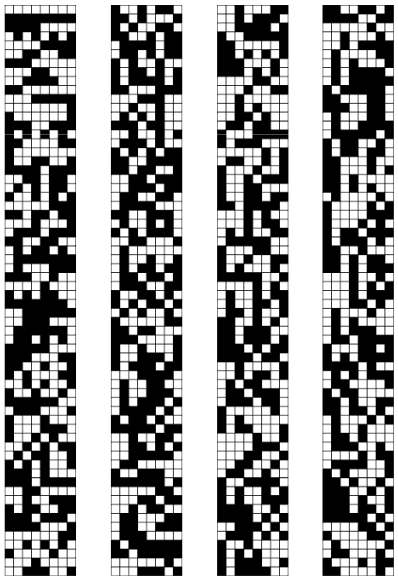

Figure 9. Rankings of binary sequences from the subjective stationary distribution $\hat{P}(x|\text{regular})$, where a head is encoded as a black square, and a tail is encoded as a white square. The least random sequence appears at the top of the leftmost column. Rankings proceed down and then across columns.

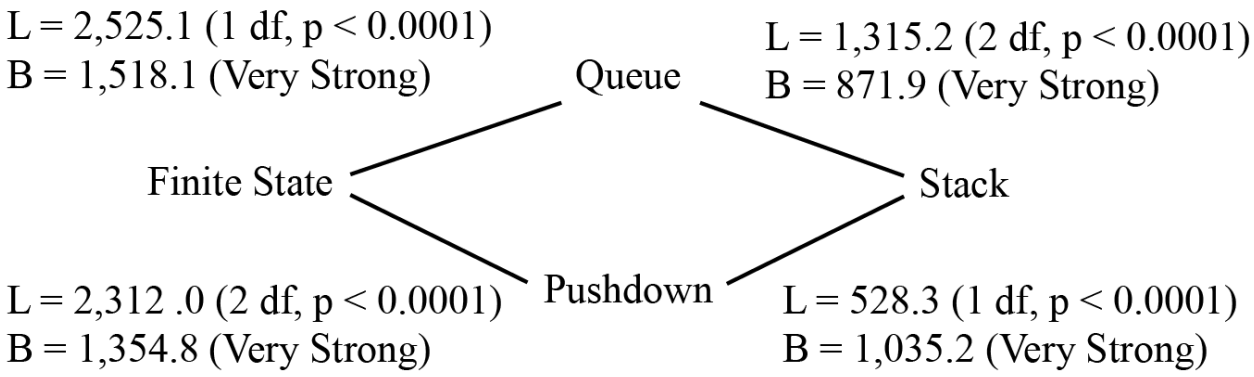

Figure 10. Statistical tests of models of $P(x|\text{regular})$ from Experiment 2. Along each edge are the results of the log-likelihood ratio tests χ^2 scores, denoted L (degrees of freedom, p-value), and twice the logarithm of the Bayes factor for the comparison, denoted B, with its intepretation in parentheses as designated by Kass and Raftery (1995).

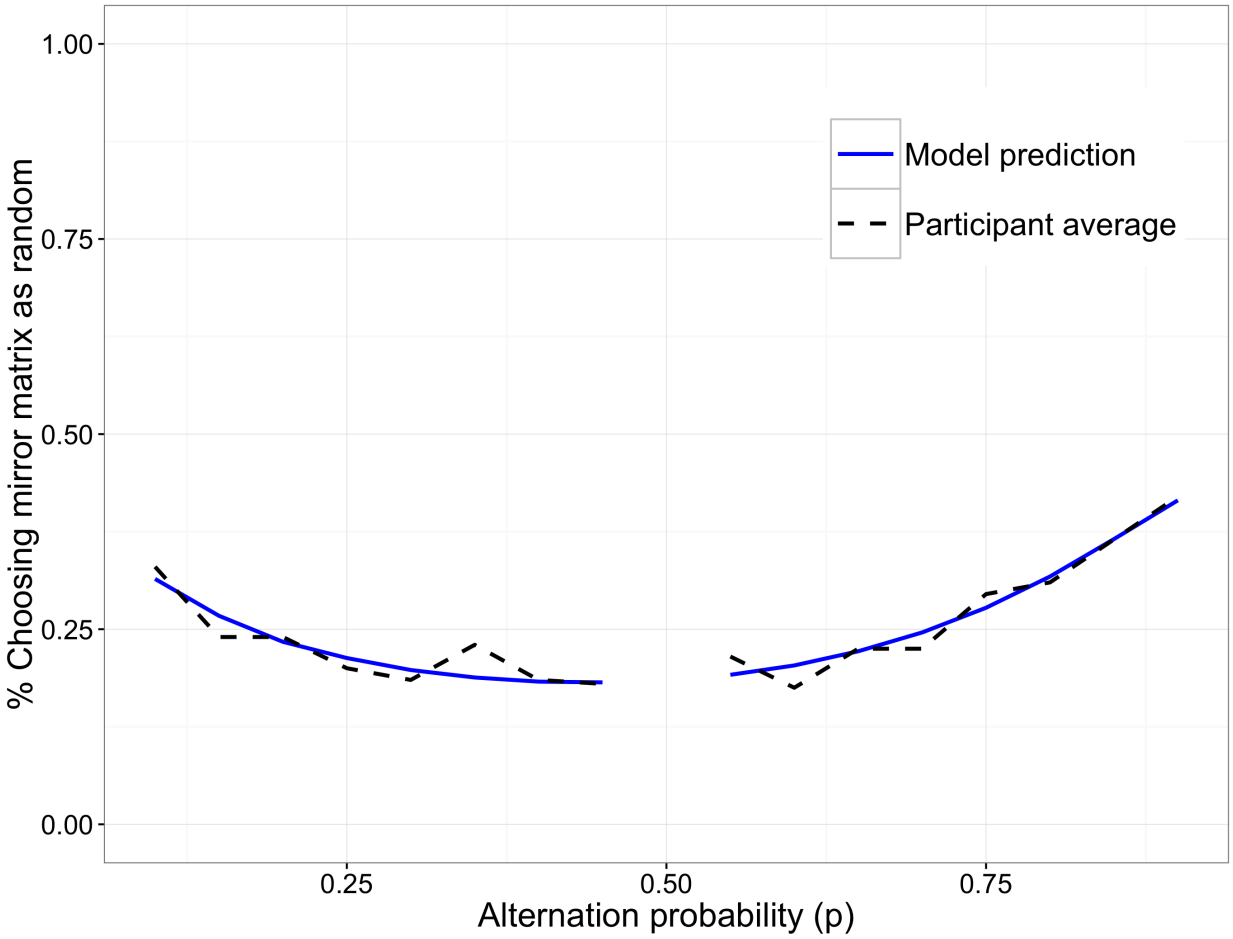

Figure 11. The dashed line shows the proportion of times participants classified the mirror matrix as more random than the switch matrix in Experiment 3 of Zhao, Hahn, and Osherson (2014). The solid line shows our model predictions.

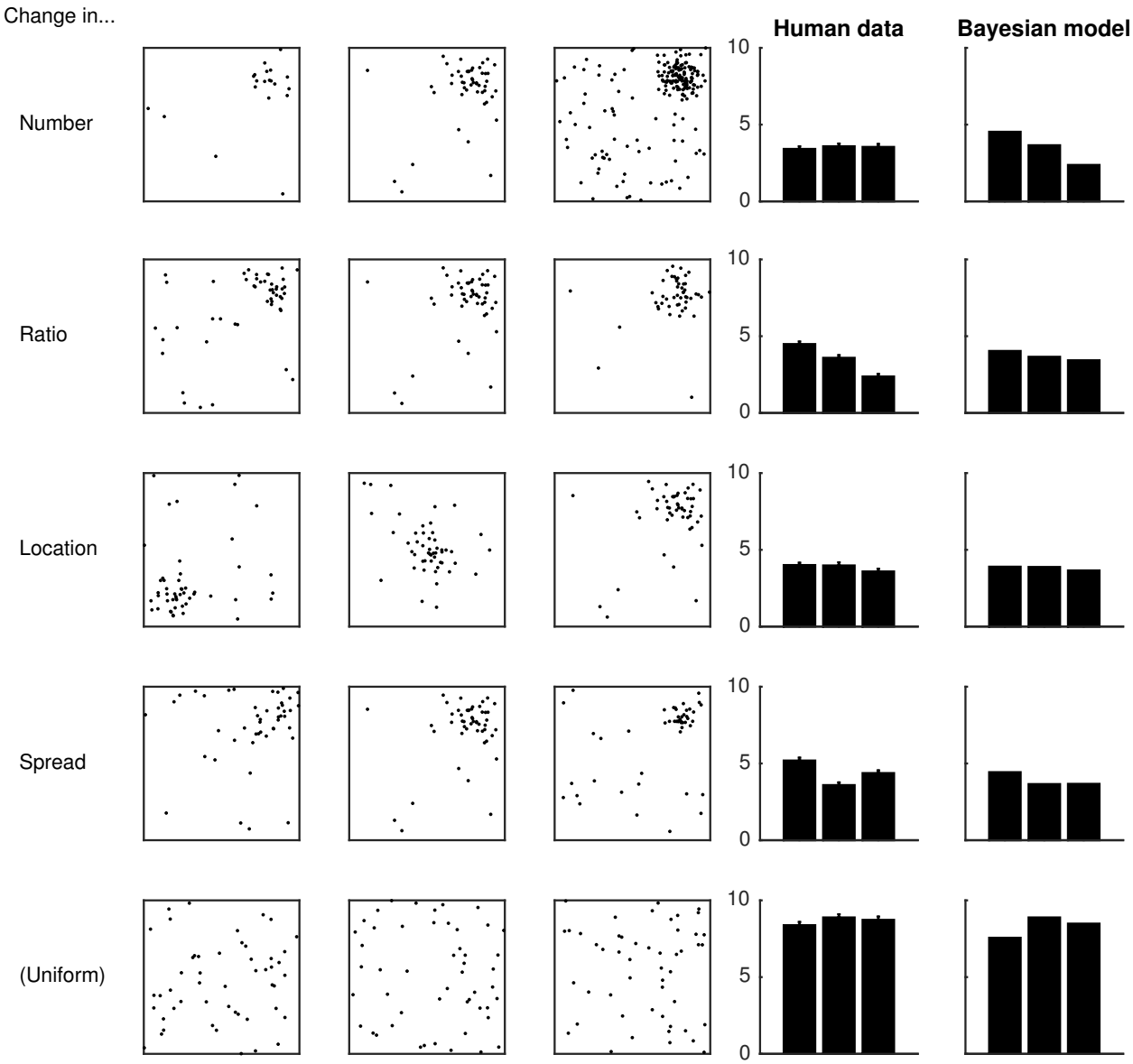

Figure 12. Randomness judgments for spatial arrays. On the left are the images shown to the participants, organized along each varied dimension. On the right, the mean randomness responses (from 1 to 10) from the human participants and the predictions from the Bayesian model. The error bars signify one standard error.

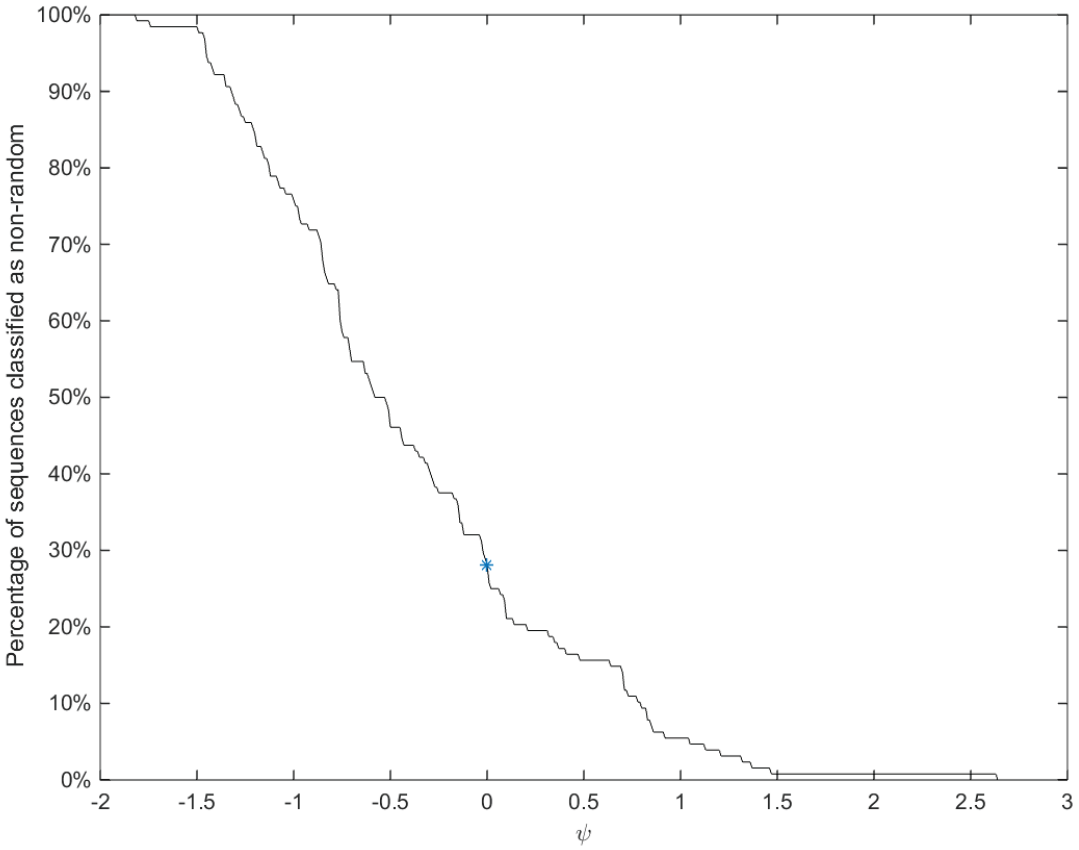

Figure 13. Percentage of sequences classified as non-random according to Equation 22 as a function of ψ. A point is starred where there is no a priori bias towards regular or random events ($\psi = 0$). Shifting the bias towards non-random events ($\psi < 0$) rapidly increases the number of events classified as non-random.