
ELSEVIER
Volume 122, Issue 3, March 2012
ISSN 0010-0277
pp. 267–460
COGNITION

Cognition 122 (2012) 346–362

Contents lists available at SciVerse ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/COGNIT

What's magic about magic numbers? Chunking and data compression in short-term memory

Fabien Mathy [a,*], Jacob Feldman [b]

[a] Université de Franche-Comté, 30-32 rue Mégevand, 25030 Besançon Cedex, France
[b] Rutgers University, New Brunswick, USA

ARTICLE INFO

Article history:
Received 1 February 2011
Revised 25 October 2011
Accepted 2 November 2011
Available online 15 December 2011

Keywords:
Chunk
Chunking
Compression
Short-term memory
Capacity
Span

ABSTRACT

Short term memory is famously limited in capacity to Miller's (1956) magic number 7 ± 2—or, in many more recent studies, about 4 ± 1 "chunks" of information. But the definition of "chunk" in this context has never been clear, referring only to a set of items that are treated collectively as a single unit. We propose a new more quantitatively precise conception of chunk derived from the notion of Kolmogorov complexity and compressibility: a chunk is a unit in a *maximally compressed* code. We present a series of experiments in which we manipulated the compressibility of stimulus sequences by introducing sequential patterns of variable length. Our subjects' measured digit span (raw short term memory capacity) consistently depended on the length of the pattern *after compression*, that is, the number of distinct sequences it contained. The true limit appears to be about 3 or 4 distinct chunks, consistent with many modern studies, but also equivalent to about 7 uncompressed items of typical compressibility, consistent with Miller's famous magical number.

1. Introduction

In a famous paper, Miller (1956) proposed that the capacity of short-term memory (STM) is limited to a "magical number" of about seven (plus or minus two) items.[1] This limit is usually expressed in terms of "chunks" (Anderson, Bothell, Lebiere, & Matessa, 1998; Gobet et al., 2001; Simon, 1974; Tulving & Patkau, 1962), meaning groups of items that have been collected together and treated as a single unit, in part to accommodate the observation that apparent span may be increased if items can be readily grouped together into larger units. For example, amid a sequence of letters the familiar string USA or the repeating pattern *BBB* might each serve as a single chunk, rather than as three separate items each. An extreme example of chunking is the subject S.F. discussed in Ericsson, Chase, and Faloon (1980), who despite average intelligence was able to increase his apparent digit span to almost 80 digits by devising a rapid recoding system based on running times, which allowed him to group long sequences of digits into single chunks.

The capacity limit is traditionally attributed to forgetting by rapid time-based decay (Baddeley, 1986; Barouillet, Bernardin, & Camos, 2004; Barouillet, Bernardin, Portrat, Vergauwe, & Camos, 2007; Burgess & Hitch, 1999; Henson, 1998; Jonides et al., 2008; Nairne, 2002; Page & Norris, 1998) or mutual interference between items (Lewandowsky, Duncan, & Brown, 2004; Nairne, 1990; Oberauer & Kliegl, 2006). The span is also substantially influenced by the spoken duration of the constituent items, a result which runs against a constant chunk hypothesis and which has been interpreted in terms of a phonemically-based store of limited temporal capacity (Baddeley, Thomson, & Buchanan, 1975; Burgess & Hitch, 1999; Estes, 1973; Zhang & Simon, 1985). Though verbal STM is well known to depend on phonological encoding (Baddeley, 1986; Chen & Cowan, 2005), the sometimes dramatic influence of chunking points to abstract unitization mechanisms that are still poorly understood.

* Corresponding author.
E-mail address: fabien.mathy@univ-fcomte.fr (F. Mathy).

[1] According to the Science Citation Index (Kintsch & Caciopo, 1994) this paper is the most frequently cited article in the history of *Psychological Review*.

0010-0277/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.cognition.2011.11.003

Notwithstanding the fame of Miller's number (Baddeley, 1994), many more recent studies have converged on a smaller estimate of STM capacity of about four items (Baddeley & Hitch, 1974; Brady, Konkle, & Alvarez, 2009; Broadbent, 1975; Chase & Simon, 1973; Estes, 1972; Gobet & Clarkson, 2004; Halford, Baker, McCredden, & Bain, 2005; Halford, Wilson, & Phillips, 1998; Luck & Vogel, 1997; Pylyshyn & Storm, 1988, 2008). The concept of working memory (Baddeley, 1986; Engle, 2002) has emerged to account for a smaller "magic number" that Cowan (2001) estimated to be 4 ± 1 on the basis of a wide variety of data.

Broadly speaking, the discrepancy between the two capacity estimates seems to turn on whether the task setting allows chunking (Cowan, 2001). Generally, four is the capacity that has been observed when neither rehearsal nor long-term memory can be used to combine stimulus items (i.e., to chunk), while seven is the limit when chunking is unrestricted. Hence the two limits might be fully reconciled if only chunking were more completely understood.

Yet half a century after Miller's article, the definition of a chunk is still surprisingly tentative. Chunks have been defined as groups of elements (Anderson & Matessa, 1997; Bower & Winzenz, 1969; Cowan, 2010; Cowan, Chen, & Rouder, 2004; Farrell, 2008; Hitch, Burgess, Towse, & Culpin, 1996; Ng & Maybery, 2002; Ryan, 1969; Wickelgren, 1964), but exactly which groups remains unclear unless they result from statistical learning (Perruchet & Pacton, 2006; Servan-Schreiber & Anderson, 1990). Cowan (2001) defines a chunk as "a collection of concepts that have strong associations to one another and much weaker associations to other chunks concurrently in use" and Shiffrin and Nosofsky (1994) as "a pronounceable label that may be cycled within short-term memory". Most attempts to define chunks are somewhat vague, ad hoc, or severely limited in scope, especially when they apply only to verbally encoded material (Shiffrin & Nosofsky, 1994; Stark & Calfee, 1970), making it difficult for them to explain the existence of chunking-like processes in animal learning (Fountain & Benson, 2006; Terrace, 1987, 2001). The current consensus is that (1) the number seven estimates a capacity limit in which chunking has not been eliminated (2) there is a practical difficulty in measuring chunks and how they can be packed and unpacked into their constituents.

In this paper we propose a new conception of chunk formation based on the idea of *data compression*. Any collection of data (such as items to be memorized) can be faithfully represented in a variety of ways, some more compact and parsimonious than others (Baum, 2004; Wolff, 2003). The size of the most compressed (lossless) representation that faithfully represents a particular sequence is a measure of its inherent randomness or complexity, sometimes called its *Kolmogorov complexity* (Kolmogorov, 1965; Li & Vitányi, 1997).

Simpler or more regular sets can be represented more compactly by an encoding system that takes advantage of their regularities, e.g. repetitions and symmetries. As an upper bound, a maximally complex sequence of N items will require about N slots to encode it, while at the other extreme an extremely repetitive string may be compressed into a form that is much smaller than the original string. Incompressibility as a definition of subjective randomness has some empirical support (Nickerson, 2002). Kolmogorov complexity has a number of cognitive correlates (Chater & Vitányi, 2003); for example simpler categories are systematically easier to learn (Feldman, 2000; Pothos & Chater, 2002).

In this paper, we ask whether complexity influences the ease with which material can be committed to short-term memory. Our hypothesis, that simpler material is more easily memorized, follows directly from the fact that—by definition—complexity determines the size of a maximally compressed representation.

If so, the true limits on capacity depend on the size of this compressed code, leading to our view that a "chunk" is really a unit in a maximally compressed code. The following experiments test this hypothesis by systematically

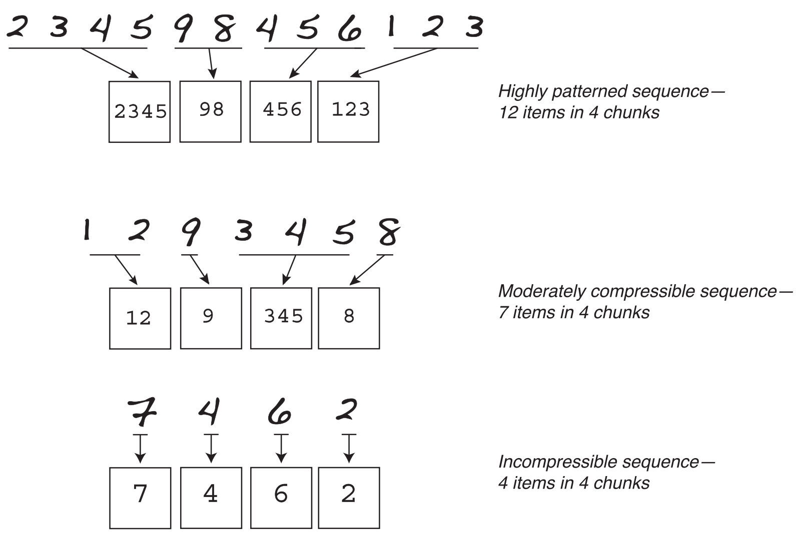

Fig. 1. The number of items that can be compressed into four "chunks" depends on the complexity of these material. Completely incompressible (maximum Kolmogorov complexity) sequences (bottom) require one chunk per item. Sequences of moderate complexity (middle) might allow 7 items to be compressed into 4 chunks, leading to an apparent digit span of 7. Highly patterned (regular) sequences might (top) allow even larger numbers of items to be compressed into the same four slots.

manipulating the complexity of material to be remembered. In contrast to most memory tasks where chunking is either unrestricted or deliberately suppressed, our goal is to *modulate* it by systematically introducing sequential patterns into the training materials.

If correct, this approach entails a new understanding of the difference between the magic numbers four and seven. To preview, our conclusion is that four is the true capacity of STM in maximally compressed units; while Miller's magic number seven refers to the length of an *uncompressed* sequence of "typical" complexity—which, for reasons discussed below, has on average compression ratio of about 7:4 (e.g., second sequence in Fig. 1).

Note that our conception is not intended to replace existing processing models of forgetting and remembering in working memory (Jonides et al., 2008). Rather, our goal is to develop a mathematically motivated model of chunks, paving the way for a better quantification of working memory capacity. Clearly, our model of chunking can and should be integrated with processing accounts, though in this paper we focus narrowly on the definition of chunk and its consequences for measured capacity.

2. Experiment 1: Sequences with variable complexity

In Exp. 1, subjects were given an immediate serial list recall task in which 100 lists of digits were created by using increasing or decreasing series of digits (runs) of variable lengths and increments (step sizes). In a given list, the increments were constant within chunks, but generally varied between runs. For example, three runs (say, 1234, 864, 56), using three different increments (here 1, −2, and 1) would be concatenated to produce a single list (123486456). Fig. 2 graphically illustrates the structure of two such lists, one more complex (more shorter runs) and one simpler (fewer longer runs). On each trial, the entire list was presented sequentially to the subject at a pace of 1 s per digit (without any indication of its division into runs). The subject was asked to immediately recall as many digits as possible in the order in which they were presented.

The length of the list was random (from 3 to 10), rather than progressively increasing, to avoid confounding fatigue or learning effects with task difficulty effects (and to avoid other peculiar effects, see Conway et al., 2005, p. 773). We used proportion correct as our dependent measure, focusing on performance as a function of the number of runs as well as the number of raw digits.

2.1. Method

2.1.1. Participants

Nine Rutgers University students and 39 Université de Franche-Comté students received course credit in exchange for their participation.

2.1.2. Stimuli

The stimuli were displayed visually on a computer screen. Each digit stimulus was about 3 cm wide and 4 cm tall, presented in the middle of the screen at a pace of 1 s per item, printed in a white Arial font against a black background. In a given list of digits, each digit replaced the previous one in the same spatial location. Each stimulus (i.e., a list of digits) was composed of a maximum of 10 digits. The stimuli were composed of monotonic series of constant increments (runs). As explained above, the increments were hold constant within runs but could vary between them. Runs varied in length from 1 digit (meaning in effect no run) to 5 digits. To construct each list, the number of runs was drawn randomly from the range 1–10. For each

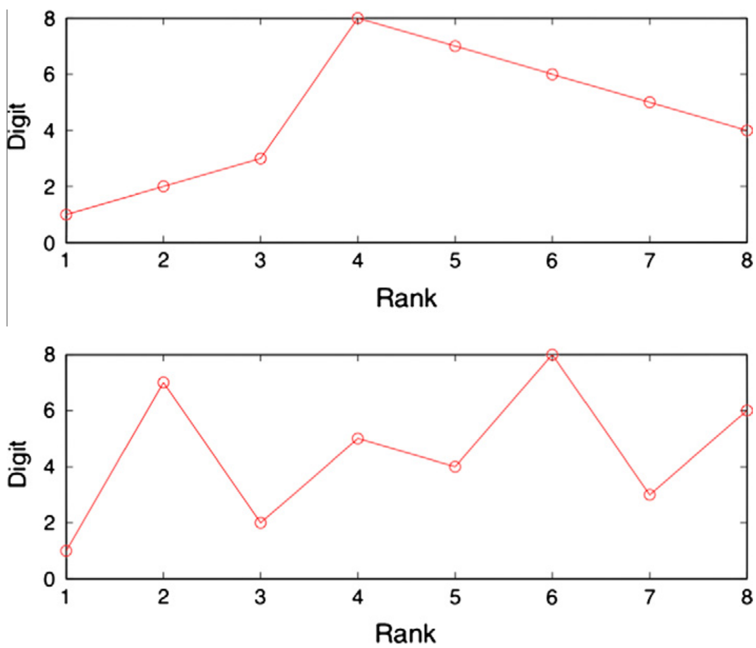

Fig. 2. Graphical depiction of sequence structure in two digit sequences, 12387654 (top) and 17254836 (bottom). The upper example contains two runs (123–87654), and is thus relatively compressible, corresponding mentally to two "chunks." The lower example is relatively erratic, containing no apparent runs at all, and as a result is approximately incompressible, requiring approximately 8 chunks to encode the 8 digits.

run, the first digit, the increment (1, 2, or 3), and the sign of the increment (+ or −) were chosen randomly. If the first digit did not allowed the series to go all the way, the run was terminated where it could end. For instance, if "3" was chosen as the first digit of a chunk, and if "−1" was chosen as the increment, the length of the chunk was limited to 3. Had the increment been "−2", the length of the chunk would have been limited to 2, and so on. Therefore, in this experiment, the number of digits per run (mean 2.8) was generally less than the expected value of 3.

At the end of the first run, the next run was drawn, and so forth, as long as the series did not go beyond 10 digits.

Using this technique, the expected value of the number of runs was 3.6, a value of particular interest given the discussion above.

2.2. Procedure

Each experimental session lasted approximately half an hour and included a maximum of 100 separate stimulus lists. Subjects were not given any special indications concerning the presence of monotonic sequences. After the presentation of the last digit of a given list, subjects could enter their response on a keyboard. The subjects were instructed to recall the digits in order. The successive digits entered by subjects were displayed in pink ink (1 cm wide and 1.5 cm tall Arial letters) and placed side by side forming a single row from the subject's left to right. The subjects could read their response to make sure the list they entered was what they intended. Once their response confirmed by a press of the space bar, they were presented with the next list.

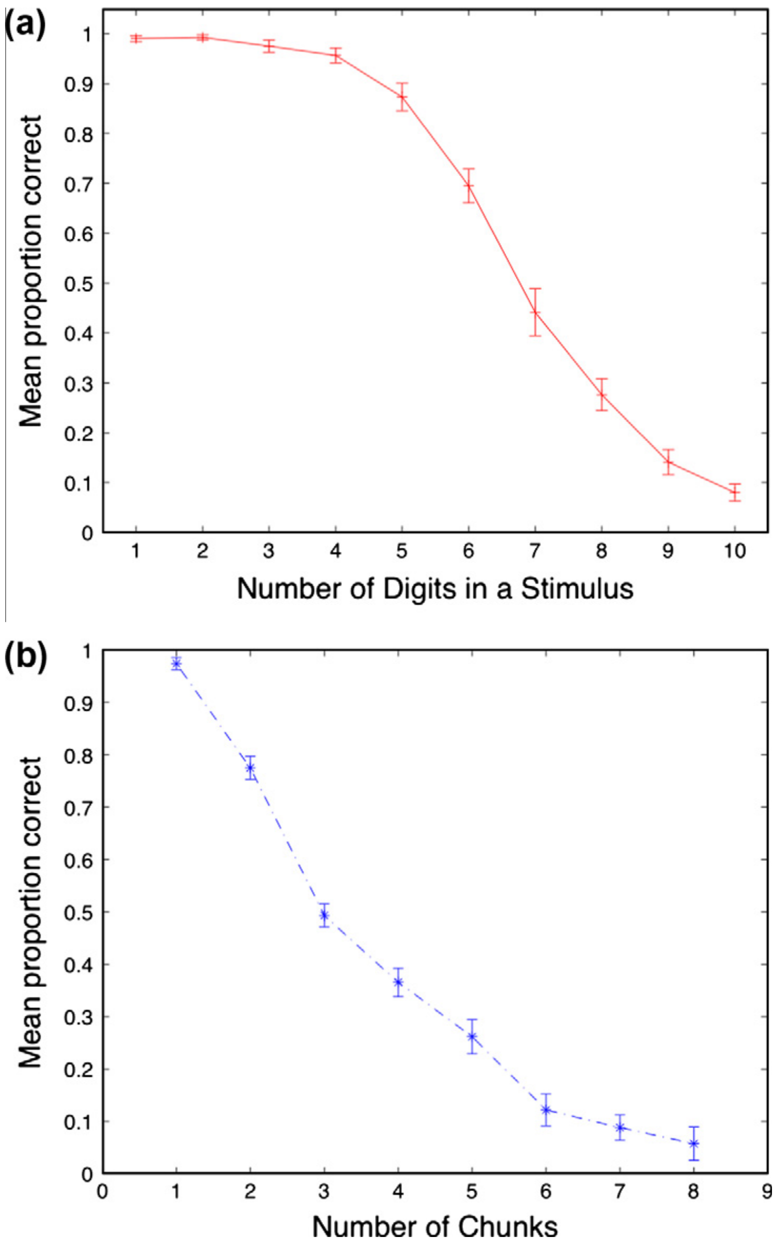

Fig. 3. Mean proportion of sequences correctly recalled (Exp. 1) as a function of (a) the number of digits and (b) the number or runs in the stimulus sequence. Error bars indicate ±s.e.

No feedback was given to subjects, but the subjects were debriefed and offered a chance to look at their datafile by the end of the experiment.

2.3. Results

All 48 subjects were included in the analysis. Seventy-seven percent of the subjects completed at least 95% of the 100 trials.[2]

Figs. 3a and b show performance as a function of the number of digits and runs respectively. The number of digits and the number of runs are, obviously, not independent of each other, and the plots show that performance steadily deteriorates with both. The decline has a more exponential form (the functional form one would expect in this situation; see Crannell & Parrish, 1957) with runs than with digits (for digits $R^2 = .51$, SSE = 35.64, RMSE = .277; for runs: $R^2 = .69$ SSE = 14.22, RMSE = .198, when both are fitted by an exponential decay function). The difference between the two correlations is significant, with $z = 4.2$, $p < .001$.

Mean memory span (integral under the performance curve) was about 6.4 digits or 2.8 chunks, consistent with both classical limits.[3] Likewise, analysis of the rate of decline in performance shows that subjects' performance falls below 50% at about 3 runs or 7 digits, corroborating the usual limits. Odds ratios (ratio of likelihood of recollection at n against $n - 1$) at 7 digits and 3 runs are respectively 2.05 and 1.57.

Fig. 4 shows more clearly how digits and runs contribute independently to performance. The plot shows the influence of the number of runs, broken down by the number of digits, indicating (for sequences longer than six digits) a steady decline in performance as sequences get more complex (more runs in the same number of digits). For each fixed number of digits, performance tends to decline with increasing number of runs (i.e. more numerous shorter runs, making a more complex sequence). The decline is separately significant by linear regression[4] for N digits = 7, 8, 9, and 10 (details in Table 1). This confirms that even when the length of the sequence is held constant, more complex or erratic sequences (more runs within the same number of digits) are harder to recall. The fact that increasing the number of runs in 5- and 6-digit sequences does not lead to worse performance might confirm MacGregor's (1987) suggestion that chunking is beneficial only when the number of items is above capacity. In that respect, the fact that many of our 4-, 5-, or 6-digit sequences reduced to a similar low number of chunks (for instance, respectively, four 1-digit chunks, three 1-digit chunks and one 2-digit chunk, and, for instance, three 2-digit chunks) can account for why most of these short sequences do not lead to a sufficiently high load to degrade performance.

In Fig. 4, a pure effect of digits would appear as vertical separation between the individual —horizontal— curves; a pure effect of chunks would appear as a decreasing trend within each curve, with curves overlapping. Accordingly, in addition to the decreased performance observed with runs, Fig. 4 shows a very large digits effect. However, the dependent measure used in this figure is a coarse measure of performance, scoring 0 for trials that were not recalled correctly, regardless of how closely the recalled string actually matched the stimulus.

We also attempted a more accurate evaluation of performance given the number of chunks, based on the proportion of digits recalled, by giving "partial credit" for sequences recalled. To evaluate performance in a more graded and more informative manner, we used a sequence alignment method (Mathy & Varré, submitted for publication) in order to compute the actual number of digits that were recalled in correct order for each sequence,[5] irrespective of response accuracy (a dependent measure previously used by Chen & Cowan, 2009). For example, given the sequence 12345321, a response of 1234321 would be scored as seven digits correct out of eight rather than 0 as in a conventional accuracy score.[6] With this less coarse measure, we expected to obtain more overlapping curves showing that subjects had greatly benefited from the reduced memory load conferred by more compressible sequences. The actual number of digits recalled in correct order plotted as a function of the number of runs, broken down by the number of digits is shown in Fig. 5. To better estimate the effect of regularity on performance, we tried to maximize the chance of having a larger coefficient for runs than for digits in the

[2] Each experimental session was limited to half an hour and included at most 100 separate stimulus lists. Certain participants did not have sufficient time to finish the experiment. In the absence of cues from the screen on the number of lists already completed, the experimenter was unable to know that a participant was, for instance, one or a couple of trials short of finishing the experiment (this is the reason why sometimes, in our experiments, the total number of trials is very close to the maximum; there was no cut-off in the data).

[3] Scoring is fundamental, but the choice of scoring procedures can change the estimates for a given task (St Clair-Thompson & Sykes, 2010). See Conway et al. (2005, pp. 774–775), who compare four basing scoring procedures; some examples are given by Cowan (2001, p. 100); see also Martin (1978), in the context of immediate free recall. Note that integrating under the performance curve (Brown, Neath, & Chater, 2007; Murdock, 1962) corresponds to an all-or-nothing unit scoring (Conway et al., 2005).

[4] We use linear regression as a simple test for declining performance. Tests for exponential declines revealed the same pattern of significance.

[5] The nwalign function of the MATLAB Bioinformatics Toolbox used for the analysis is based on the Levenshtein distance (i.e., the minimum number of operations needed to transform one string into another, with the allowable operations being deletion, substitution, or insertion), except that nwalign allows to test different costs for the different operations (for instance, allowing to set an insertion operation as less probable than a simple deletion). This sequence alignment method is particularly useful when repetition is allowed in the stimuli. For instance, given a list 321.24.2345 (the chunks are separated by a dot symbol) and a response 3212345, the computation of the alignment '|||**||||' between the two sequences clearly indicates that the 4^{th} and 5^{th} items are omitted in the subject's response (the other digits are aligned). Without such a technique, it is difficult to know which of the three '2' digits have been recalled in correct order. Consider another hypothetical response 32123452: in that case, the last '2' digit would not be considered as a digit correctly recalled *in order* since it cannot be aligned with a digit in the stimulus list. Given that permutation rates are low (Henson, Norris, Page, & Baddeley, 1996; Mathy & Varré, submitted for publication), we focused on a basic algorithm run with default parameters (i.e., to compute a Levenshtein distance), in view of searching for omissions, confusions, and insertions.

[6] This method meshes nicely with the idea that a Kolmogorov distance can be computed between two objects (Hahn, Chater, & Richardson, 2003). The simpler the transformation distorting the stimulus to the response, the more similar they are.

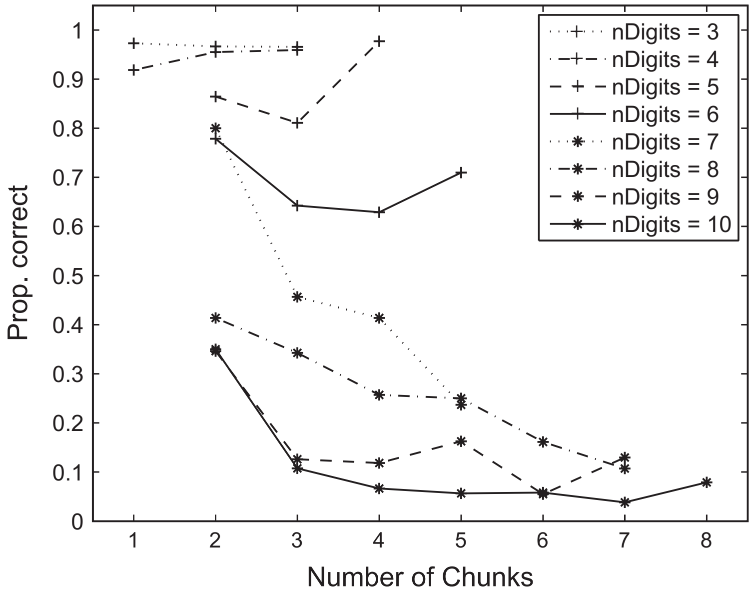

Fig. 4. Mean proportion of sequences correctly recalled as a function of the number of runs per stimulus sequence, broken down by the number of digits per stimulus sequence. For sufficiently long runs (>6), the plot shows a substantial decreasing trend. This shows how the number of runs contributes to memory load in a way that is distinct from the contribution of the number of digits. A sequence of a given length is more difficult to remember if it has more distinct runs, which increases its complexity and decreases it compressibility.

Table 1
Statistics for declines in performance as a function of runs, broken down by the number of digits (Exp. 1).

N digits	*r*	*p*	*N* sequences
3	.01	.921	98
4	.10	.291	115
5	−.03	.791	99
6	−.05	.547	165
7	−.37	<.001	137
8	−.23	.001	219
9	−.14	.028	247
10	−.19	.002	276

multiple regression analysis. With that goal in mind, we removed the data at sub-span level that naturally showed nearly ceiling performance (where the number of digits was less than 7), and we averaged across conditions where the number of chunks was greater than 5 (to group less numerous data). In that condition, we again obtained four significant linear regressions for the curves ranging from *n*Digits = 7 to *n*Digits = 10 (respectively, $r = -.15$, $r = -.15$, $r = -.21$, and $r = -.10$). However, the multiple linear regression analysis showed that performance did not solely depend on the number of runs, with the number of digits

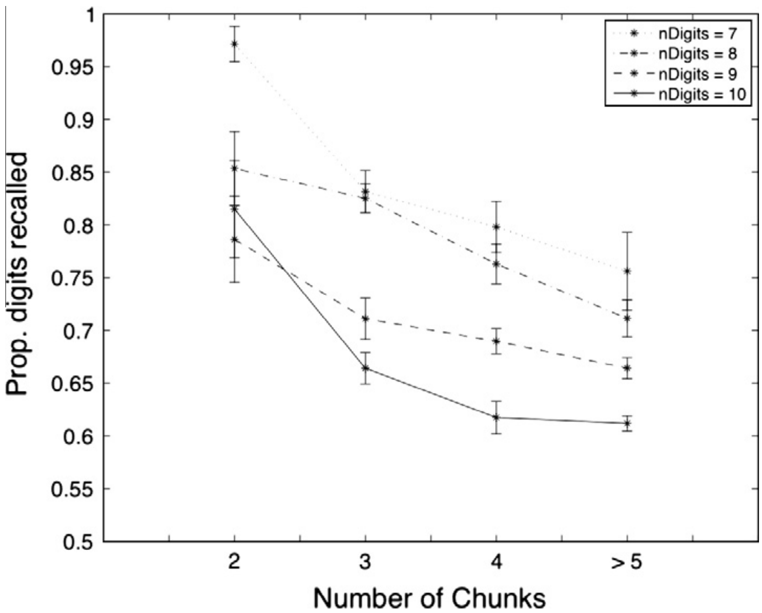

Fig. 5. Mean proportion of digits (Exp. 1) recalled per sequence as a function of the number of runs per sequence, broken down by the number of digits per sequence. The plot only shows the curves having a significant negative slope (the other curves at sub-span level showed nearly ceiling performance). Error bars indicate ±one standard error.

having again a greater influence on performance than the number of chunks ($\beta_{nDigits} = -.255$, $p < .001$; $\beta_{nRuns} = -.122$, $p < .001$; $F(2,3111) = 175$, $R^2 = .10$, $p < .001$).

2.3.1. Quantifying complexity

We can quantify the complexity of each digit string by considering how much memory space its component runs would actually require to encode, given their length. Each string of length N consists of K runs (numbered $i = 1 \ldots K$), with the i-th run having length L_i. For example the string 12387654 illustrated in Fig. 2 (top) has $K = 2$ (two runs) with $L_1 = 3$ and $L_2 = 5$. The memory space required to encode a number N is generally proportional to log (N), with the units determined by the base of the log (see Cover & Thomas, 1991). For example, the number of decimal digits required to express a number N is about $\log_{10} N$ digits (the number 10 requires 2 decimal digits, 100 requires 3, 1000 requires 4, and so on), while in base 2 the number of bits required is proportional to $\log_2 N$. We usually use base 2 and thus bits. So the representation of each digit string will have length about

$$\text{complexity} = \sum_{i=1}^{K} \log_2(1 + L_i), \tag{1}$$

which sums up the space required over all the component runs, i.e. their aggregate complexity. (We add one to each length so that singletons ["runs" of length 1] have nonzero minimum complexity of $\log_2 2 = 1$ bit.) Our goal was to choose the simplest formulation that could account for both the variations in the number of chunks and the number of digits to which the runs were subject to. This formula admittedly oversimplifies the situation a bit, because it ignores the "overhead" involved in encoding the decompression routine itself (e.g. the "for… end" structure implicit in the expression of a run), which contributes a constant amount of complexity. But it approximates the space required to express the *variable* parameters in the run structure, which make a contribution which increases as the runs get longer. In our sequences these include the number of runs (K) and the length of each run (L_i). So the complexity measure should be understood as an approximation of the amount of information required to encode the runs within the sequences we created. Also, a choice was deliberately made not to fully develop a Minimum Description Length (MDL) which work best with situations where the to-be-recoded patterns can be assigned a probability value (Perlman, Pothos, Edwards, & Tzelgov, 2010; Robinet & Lemaire, 2009; Servan-Schreiber & Anderson, 1990). The MDL approach that more thoroughly takes into account the probabilities of various patterns could probably lead to a superior generalization of our approach, sharing the same idea that storage capacity relates to maximal effective compression.

We admit that this simple formulation is not completely general, but rather is tailored to the specific class of digit sequences that were used in our experiments. Our formula for example neglects the possibility of increments of 0 (leading to repeated digits) or systematically changing increments (leading to nonlinear runs), which were not included in our experimental materials. Such patterns would require a more elaborate complexity metric because they involve additional parameters, which must be encoded.[7] Nor is our measure intended to encompass broader classes of patterns as might apply to items other than digits. A *completely* general complexity measure, i.e. one that encompasses any kind of pattern at all, would be Kolmogorov complexity itself, which is not actually computable (see Schöning & Pruim, 1998). In practice, any computable complexity measure only reflects the compression possible given the patterns within its scope, and the more pattern classes are entertained, the more parameters must be included. Our formulation aims to encompass only the limited set of regularities possible in the runs we included in our experiments, and the parameters (number and length of runs) that are required to encode them.[8]

Fig. 6 shows memory performance as a function of complexity. The plot shows a dramatic fall-off in performance for digits strings above about 4 compressed units ($R^2 = .98$ using a sigmoid function). Digit strings that are short enough and/or regular enough to be compressed into this size are easily accommodated in available memory space, while longer and/or more complex ones are not. This finding encapsulates our claim that "chunks" are best understood as units in a compressed code. It is the length of such a code, quantified by Eq. (1), that most directly determines whether a particular sequence can be squeezed into available memory space.

We also tried a more complex formula for complexity in which the length, the increment, and the starting point of the runs are all taken into account, $\sum_{i=1}^{K}[\log_2(1 + L_i) + \log_2(1 + \text{Incr}_i) + \log_2(1 + \text{Start}_i)]$. This leads to a lower portion of explained variance ($R^2 = .96$), but usefully reveals that recall is about perfect for 10 bits, about 50% correct for 20 bits, and about null for 40 bits. This 10 bit limitation is similar to the one found by Brady et al. (2009).

2.4. Discussion

Our results show that the compressibility of a string, reflected in its regularity and redundancy, exerts a measurable influence on subjects' ability to remember it. While longer sequences are, on average, more difficult to remember, simplicity of pattern within the string wields an independent influence. When the number of digits is held constant, simpler strings (fewer longer runs) are easier to remember, while more complex strings (more numerous short runs) are harder. The complexity of the string, which expresses how much memory space is required to encode it when

[7] For example, the pseudo-code "$x = 1$; Print x; For $i = 1{:}3$, $x = x + 2$ Print x; End" generates 1357. However, a more complicated algorithm is necessary to make the increment vary as a function of the loop index in order to generate 1247 (i.e., "$x = 1$; For $i = 1{:}4$, $x = x + \boldsymbol{i} - 1$ Print x; End"). The second procedure uses the index i twice instead of once, requiring more space to encode.

[8] To support the idea that the evaluation of compression is material-specific, one of the present authors (F. M.) has run experiments in which the participants were required to memorize and recall sequences of categorizable multi-dimensional stimuli (that is, objects with possible associations between them). In this latter case, compression was better accounted for by the length of the shortest algorithm made of If-Then structures instead of simple loops (Chekaf & Mathy, submitted for publication).

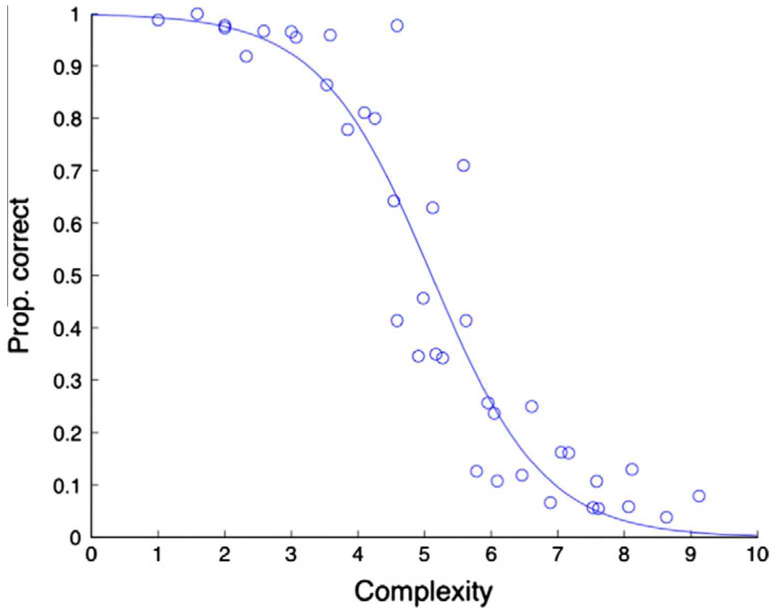

Fig. 6. Plot of performance (Exp. 1) as a function of the complexity (size after compression) of the digit string, computed from Eq. (1). The plot shows an abrupt decline in performance when complexity exceeds about 3 or 4 units of complexity (chunks). Each data point results from the mean complexity and the mean proportion correct that were computed for each cell of a *n*Digits × *n*Chunks matrix.

maximally compressed, determines the success of memory processes.

In summary, Exp. 1 demonstrated a systematic relationship between capacity and compression in patterns with moderate but variable complexity.

Consequently, our nonstandard chunking memory task enables us to recover two different well-known estimates of the span: about 3 chunks and 7 unpacked items. Unfortunately, our observation is weakened by the fact that the number of digits in a sequence prevailed over the number of chunks. A first explanation is that chunks are not immune to disturbance; a second is that our subjects were simply not able to notice the regularities in the sequences and therefore failed to encode the runs. Exp. 2 was carried out to make subjects better encode the chunks in order to let the number of chunks override the number of digits.

3. Experiment 2: Sequences with variable complexity

Because there is a possibility that the participants in Exp. 1 could not benefit from stimulus regularity (either because they did not notice the runs, or because they had difficulties encoding them), we devised a new chunking memory span task designed to offer every possible way for subjects to both notice and encode the same regularities that were available in Exp. 1.

3.1. Method

3.1.1. Participants

Twenty-three Université de Franche-Comté students received course credit in exchange for their participation.

3.1.2. Procedure

Exp. 2 was identical to Exp. 1 except that the presentation of the stimuli was not sequential. A similar procedure was used by O'Shea and Clegg (2006), in which the digits to be chunked were not presented individually in order to induce optimal recall performance (for instance, a participant would view three digits for 3 s, followed by three other digits, etc.). In our experiment, all the digits of a given trial were simultaneously shown on the screen during a time proportional to the number of digits (one second per digit). Whenever a new chunk was built by the program, the digits were located on a new line. The runs could therefore easily be identified by their location.The number of lines simply reflected the number of runs in a list. The participants were instructed that each line would correspond to a regularity found by the computer, to facilitate their task. The experimenter then proceeded to a demonstration on how the digits should be read, and emphasized that the digits had to be read from left to right, and from top to bottom to benefit from the facilitation introduced in the task. The experimental setting was optimized to simplify the memorization process in that the participants could use both verbal and visual information to chunk the digits that were displayed at once. The participants were given a list of 100 trials, but the experiment was again limited to half an hour.

3.2. Results

All 23 subjects were included in the analysis. As shown in Fig. 7, both the mean number of chunks and the mean number of digits recalled were superior to those of Exp. 1 (chunk and digit span were respectively 4.0 and 7.9 by integrating under the performance curve). Analysis of the rate of decline in performance showed that subjects' performance fell below 50% at about 4 runs and 8 digits. Fig. 8 shows how digits and runs contributed to performance, when the actual number of digits recalled in correct order for each list was computed using the alignment method already applied to the data in Exp. 1. To follow the analysis

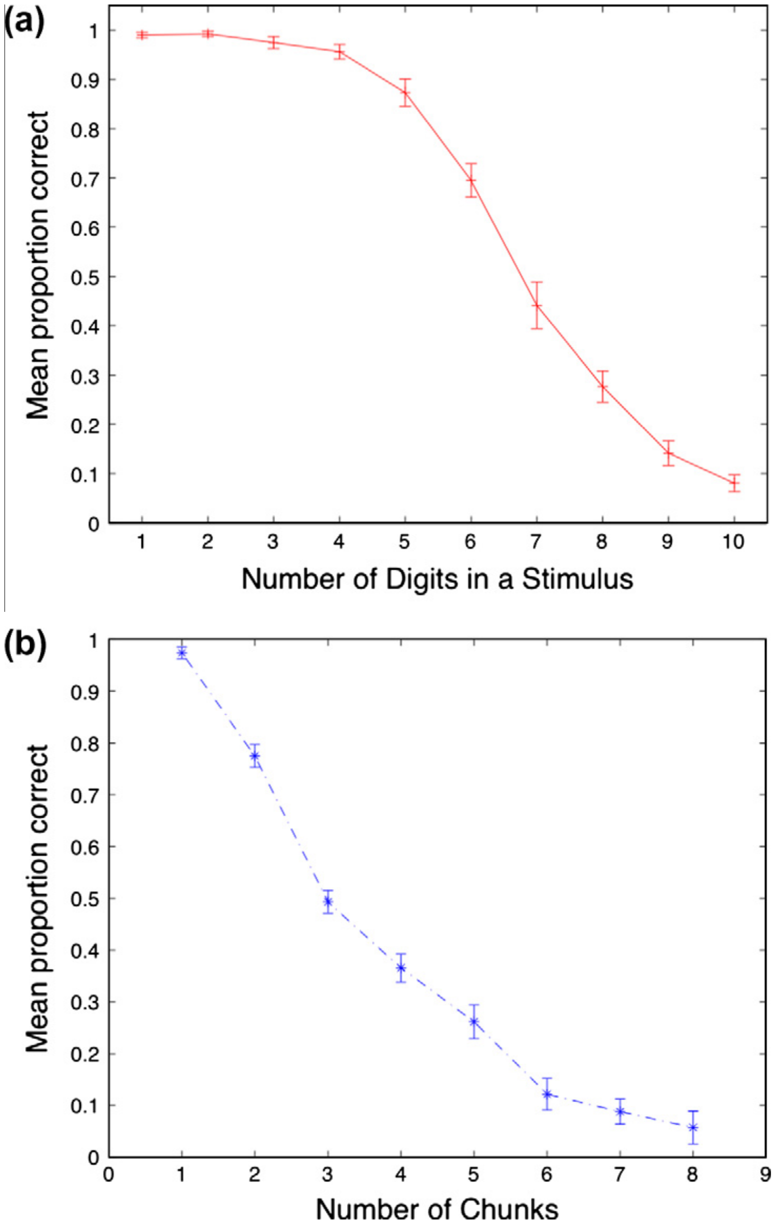

Fig. 7. Mean proportion of sequences correctly recalled (Exp. 2) as a function of (a) the number of digits per sequence and (b) the number or runs per sequence. Error bars indicate ±s.e.

conducted in Exp.1, we removed the conditions at sub-digit-span level to avoid ceiling performance (where the number of digits was less than 8), and the conditions where the number of chunks was greater than 5 were again averaged (to gather less numerous data). The main result is that, contrary to Exp. 1, the number of chunks now had a greater influence on performance than the number of digits in the multiple regression analysis, a result that is clearly visible in Fig. 8 which shows more closely spaced curves than in Fig. 5 ($\beta_{nDigits} = -.190$, $p < .001$; $\beta_{nRuns} = -.219$, $p < .001$; $F(2, 1285) = 78$, $R^2 = .11$, $p < .001$; the three regression lines as a function of the number of runs were all significant: respectively $F(1, 272) = 7.5$, $p = .01$, $\beta_{nRuns} = -.164$, for nRuns = 8; $F(1, 411) = 9.5$, $p = .01$, $\beta_{nRuns} = -.150$, for nRuns = 9; $F(1, 599) = 53$, $p = .001$, $\beta_{nRuns} = -.285$, for nRuns = 10).

Fig. 9 shows memory performance as a function of complexity. The plot indicates that performance decreases for digits strings above about 4 compressed units ($R^2 = .97$ using a sigmoid function). As in Exp. 1, using the more complex formula $\sum_{i=1}^{K}[\log_2(1 + L_i) + \log_2(1 + \mathrm{Incr}_i) + \log_2(1 + \mathrm{Start}_i)]$ leads to a lower portion of explained variance ($R^2 = .87$), but again reveals that recall is about perfect for 10 bits, about 50% correct for 20 bits, and about null for 40 bits. This 10 bit limitation is similar to the one found by Brady et al. (2009).

In a final analysis, we again took advantage of the alignment found between the stimulus sequence and the response in order to compute the actual number of chunks that were correctly recalled for each sequence, irrespective of response accuracy. Let us give an example of how useful a correct alignment between the list and the response is, particularly when digits are repeated (when there is no repetition in the material, a simple search function can be used to recover which of the items are correctly recalled). The

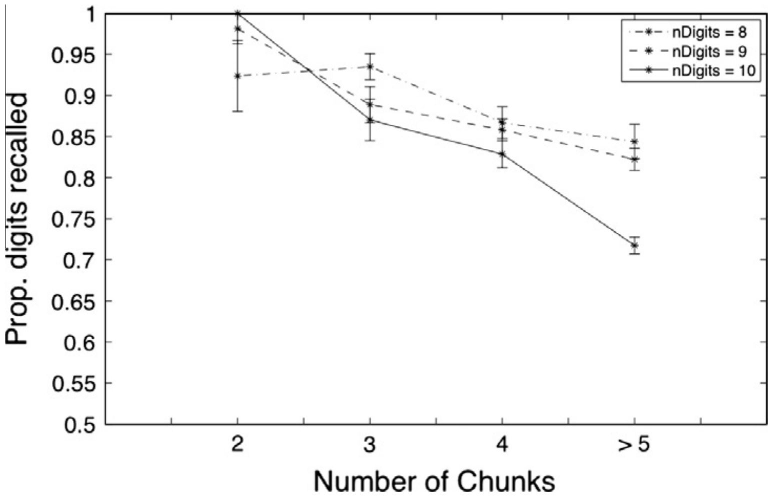

Fig. 8. Mean proportion of digits (Exp. 2) recalled per sequence as a function of the number of runs per sequence, broken down by the number of digits per sequence. The plot only shows the curves which have a significant negative slope (the other curves were at sub-span level). Error bars indicate ±one standard error.

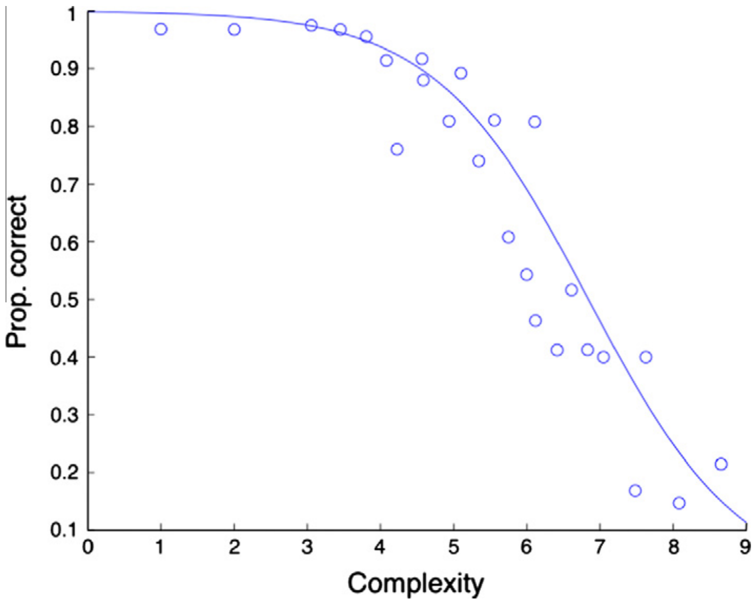

Fig. 9. Plot of performance (Exp. 2) as a function of the complexity (size after compression) of the digit string, computed from Eq. (1). The plot shows an abrupt decline in performance when complexity exceeds about 4 units of complexity (chunks). Each data point results from the mean complexity and the mean proportion correct that were computed for each cell of a *n*Digits × *n*Chunks matrix.

computation of the alignment is particularly helpful to avoid assigning correct recall to one chunk incorrectly (false alarm) and to fail to credit a chunk not strictly recalled at the expected position but nevertheless entirely recalled. For instance, given a 123.2.432 stimulus, and a 123.432 response, the alignment '|||∗|||' signals the omission of the '2' digit/chunk in the middle of the sequence, the correct recall of '123' at correct positions, and the correct recall of '432' at lower positions than expected. The recall of the chunks was computed in order, beginning with '123', and so on. Once '123' credited, it could be removed from the subsequent search function. Then, when searching for the '2' one-digit chunk, the algorithm encountered an omission symbol which was associated with incorrect recall. Because the search function followed the alignment, the '2' digit could not be wrongly associated with the one present in the '432' chunk. The resulting estimation of the number of correctly recalled chunks for this example is 2. This method is not without difficulties (Mathy & Varré, submitted for publication), but to the best of our knowledge, it is the best method available to estimate the number of chunks correctly recalled in order with such complex material, regardless of the strict position of the digits recalled in the subject's response.

Fig. 10 indicates the number of chunks that were actually encoded as a function of the number of chunks in the list. The figure shows a logarithmic performance with asymptote of about 4 chunks, $R^2 = .99$ ($N = 7$), $p < .001$, $y = .838 + 1.7 \times ln(x)$. By comparison, the number of encoded stimuli in the first experiment was systematically lower, $t(6) = 3.88$,

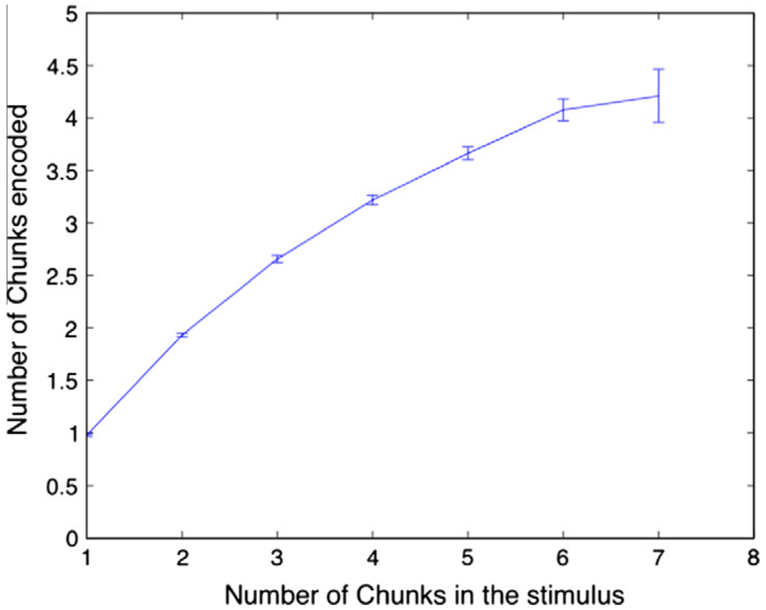

Fig. 10. Number of chunks correctly recalled (Exp. 2) as a function of the number of chunks per sequence. Error bars indicate ±one standard error.

$p < .01$, which tends to confirm that subjects had more difficulties encoding the runs in the first experiment.

A final point relating to encoding processes concerns the sensitivity of our participants to the type of increment within runs. A mean proportion correct was first computed across subjects for the six sorts of increments $(-3, -2, -1, +1, +2, +3)$, from which we obtained the respective means: .69, .74, .78, .77, .75, .76. Three different ANOVAs with the within-subject factor being Increment Type were carried out on proportion correct, after we removed the data that included one-digit long chunks. We observed neither a significant effect of the increments $(-3, -2, -1, +1, +2, +3)$, $F(5,110) = 1.99$, $p = .08$, nor a significant effect of the sign of the increment ($+i$ vs. $-i$), $F(1,22) = .06$, $p = .81$. However, we observed a significant effect when the absolute value of the increments were taken as the independent variable (respectively, .78, .75, and .73 for 1, 2, and 3), $F(2,44) = 4.07$, $p = .02$.

3.3. Discussion

Our second experiment allowed us to recover the usual estimate of the span: about 4 chunks, although this time the number of unpacked items surpassed the magical number 7. Because this experiment was carried out to make participants better encode the chunks than in Experiment 1, the number of chunks this time prevailed over the number of digits at supra-span level. Exp. 3 tests the limits of this phenomenon by using much higher levels of regularity (longer runs) in the sequences, which in principle should yield more compressible sequences and thus higher apparent capacity.

4. Experiment 3: Highly regular sequences

Our argument suggest that it ought to be possible to greatly inflate apparent memory capacity by using more regular (less complex) material, and indeed that with sufficiently regular sequences the 7 ± 2 limit should effectively vanish. To test this prediction, in Exp. 3 we use highly regular sequences with relatively long runs. If our account is correct this should yield more chunking and thus larger apparent capacity, though still corresponding to the same compressed size and the same actual memory capacity.

4.1. Method

Exp. 3 was identical to Exp. 1 except that runs were fixed to 4 digits in length (with constant increments equal to 1 or -1), and we simply varied the number of runs per sequence. Following the more classic procedure of Exp. 1, the presentation was sequential.

4.1.1. Participants

Thirty-two (new) Université de Franche-Comté students received course credit in exchange for their participation.

4.1.2. Procedure

The procedure was similar to Experiment 1, except that the number of digits per run was set to 4 and the increments could only take two values (+1 or -1) to facilitate the encoding process. The number of runs was randomly drawn from the range 2–5 (inclusive). For example, a trial with 3 runs and respective increments of +1, -1, -1 might yield the sequence 123487657654.

4.2. Results and discussion

Fig. 11 shows the proportion correct as a function of the number of runs and digits in the sequence. (In this experiment the number of digits is always four times the number of runs, so the plot shows the abscissas for digits at the bottom and runs at the top.) When analyzed in terms of

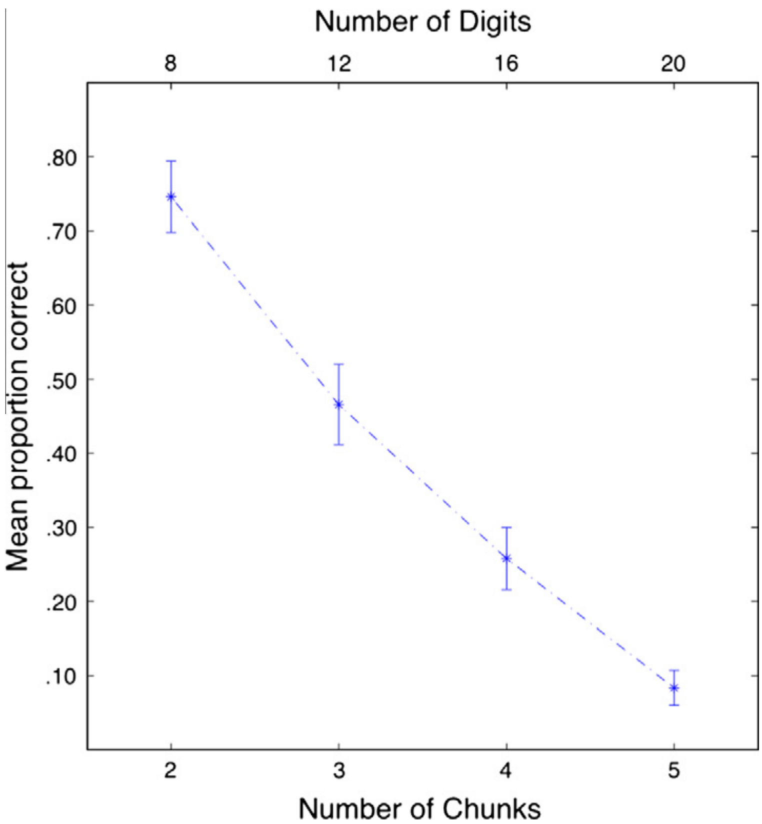

Fig. 11. Mean proportion correct (Exp. 3) as a function of the number of runs (upper abscissa) or digits (lower abscissa). Note that in this experiment all runs were 4 digits long, so the graph shows performance with respect to both. Error bars indicate ±one standard error.

raw digits, performance greatly exceeds that in Exp. 1. In Exp. 1, performance had crossed the 50% line at about 7 digits (Fig. 3a) or 3 runs (Fig. 3b), corresponding respectively to Miller's and Cowan's magic numbers. Here in Exp. 3, performance again crosses that same threshold at about three runs—but with the higher compression ratio this now corresponds to 12 digits rather than 7. Raw digit span was now 10.2 digits (compared to 6.4 digits in Exp. 1) while "chunk span" was 2.5 runs (similar to though actually slightly lower than the 2.8 found in Exp. 1). Also, we found that 1.7 (sd = .02), 2.3 (sd = .04), 2.7 (sd = .05), and 2.9 (sd = .07) chunks were correctly recalled on average, for a number of chunks respectively equal to 2, 3, 4, and 5 in the stimulus sequence.

These results shows very clearly how the two magic numbers can be deconfounded when compressibility is ratcheted up. As discussed above, the historical conflation between these two benchmarks depends on an approximately 7:4 compression ratio typical of random strings but violated in the current experiment.[9] With more compressible material, while capacity in chunks remains the same, digit spans can grow substantially. Our Exp. 3 subjects were still exceeding 25% correct with sequences of 16 digit in length, performance that would certainly be impossible with completely random sequences.

[9] The 7:4 ratio only applies to material that would show the same ratio of compressibility. Other random sequences such as those resulting from bernoulli trials (e.g., sequences of 0s and 1s) would probably show other ratios. However, we argue that 4 is a constant reflecting capacity.

5. General discussion

This study questions whether working memory and short term memory are different constructs relating to separate capacities of 4 and 7 (Unsworth & Engle, 2007). Our results suggest that memory involves a kind of reflexive data compression, whereby material to be encoded is automatically analyzed for structure and redundancy, and compressed as much as each pattern allows. Chunks in this account correspond to elements of a maximally compressed encoding. An immediate consequence, confirmed in our experiments, is that more compressible material is easier to fit into available encoding space, and less compressible (more complex) material is harder. This observation leads to our main conclusion: that the memorability of a sequence depends at least in part on its complexity or incompressibility. This conception compares very closely with common computer data compression techniques such as Lempel–Ziv (e.g. see Ziv & Lempel, 1977), which underlies the widespread compression utility gzip. The Lempel–Ziv algorithm searches for repeated blocks of data, builds a table of such blocks, and then recodes the original data sequences using symbolic references to the table. We believe that our results can add to other related findings, such as the effect of chunks on task inhibition (Koch, Philipp, & Gade, 2006), the growth of chunk capacity with age (Burtis, 1982; Cowan et al., 2010; Gilchrist, Cowan, & Naveh-Benjamin, 2009) the relationships between long-term memory and working-memory spans (Ericsson &

Kintsch, 1995; Logan, 2004), the role of chunking in perception (Suge & Okanoya, 2010) or in skill acquisition (Cohen & Sekuler, 2010), the spontaneous organization of sequences by participants (Li, Blair, & Chow, 2010), chunking in dyslexia (De Kleine & Verwey, 2009), and chunking in spatial memory (Gmeindl, Walsh, & Courtney, 2011; Sargent, Dopkins, Philbeck, & Chichka, 2010). One advantage of our chunking memory span tasks is the study of various chunk sizes, in contrast to other studies in which chunking was restricted to pairs or triplets (Burtis, 1982; Chen & Cowan, 2009; Cowan et al., 2004; De Kleine & Verwey, 2009; Li et al., 2010; O'Shea & Clegg, 2006).

This conclusion allows the capacity of short-term memory to be restated as a very simple "law:" the material that can be stored in the short-term buffer *always has Kolmogorov complexity of about 3 or 4 units.* The apparent capacity, meaning the number of raw (uncompressed) items, depends on the compressibility of the material. But when compressed, it must occupy at most about 3 or 4 units of space in order to be retained. Our estimation relies on different scoring criteria: (Conway et al., 2005)'s all-or-nothing scoring (which amounts to integrating under the performance curve), recall achieved 50 % of the time (which does not correspond to Broadbent's (1975) view that a 3 or 4 capacity limit relates to perfect recall), and on the computation of the actual number of chunks correctly encoded irrespective of response accuracy (in Exp. 2; see Fig. 10).

Naturally, subjects are not perfect encoders and our theory does not intend to account for the heterogeneous error patterns in memory that are generally observed (omissions, confusions, transpositions, insertions; Brown, Preece, & Hulme, 2000; Farrell & Lewandowsky, 2002; Henson, 1998; Henson et al., 1996; Maylor, Vousden, & Brown, 1999; McCormack, Brown, Vousden, & Henson, 2000), not to mention some complex mechanisms such as response suppression during recall (Lewandowsky, 1999), redintegration processes (Roodenrys & Miller, 2008), and transposition gradients (Brown et al., 2000). Our conception is perhaps entirely compatible with certain serial order models which have been refined for years (e.g., Brown et al., 2000; Estes, 1972; Lewandowsky & Murdock, 1989; Page & Norris, 1998), which consider for instance that recall functions by position-item associations or by means of chains of item-item associations. These models might be able to account for the dramatic decrease of performance with list length in Fig. 4, which tends to show that there are other strong determinants of memory performance beyond chunking processes.

One can note that our complexity measure gives a higher count for patterns that are longer (for example, the string 1234512345 has a greater complexity than the string 123123), which seems to go counter the idea that both strings are made of two chunks. This idea is nevertheless entirely compatible with the formalization of compression in computer science: two optimized programs of equivalent size (with equivalent Kolmogorov complexity) can nevertheless have different logical depths. Logical depth represents the time taken to calculate the results of a program of minimal length and is thus a combination of computational complexity and Kolmogorov complexity (Bennett, 1995). Two programs that may therefore have equivalent lengths will not necessarily have equivalent complexity. For instance, a program that generates the value of π is very short, although it can run for ages. In this view, the complexity of one chunk is not totally independent of the decompression time. An example is one participant who gave a 987.1.4567 response for a 987.1.45678 stimulus sequence. This might also account in part for the decrease of the proportion correct with list length in Fig. 4. We believe that future modeling of the compressibility in the to-be-learned material will need such elaborated conceptions of complexity in order to better account for the mechanisms that might take place in chunk formation and chunk recall.

Still, the level of performance depends heavily on compression, reaching a breaking point at about 4. This result parallels other arguments that relate capacity limits to optimization processes (Cabrera, 1995; Dirlam, 1972; MacGregor, 1987). Our phrase "maximally compressed code" is intended to serve as an ideal definition with a clear motivation, but in practice chunk structure will depend on the compression actually achieved. In this paper, we attempt to give an account of human chunk formation based on the assumption that the structure of the environment can be compressed optimally, but this goal can be impeded by many obstacles (limits in rationality such as not being able to reverse-engineer the Fibonacci sequence, interferences and decay in phonological recoding, etc.), some of the obstacles being intricately related to the incomputability of Kolmogorov complexity. Any real compression system, after all, including Lempel–Ziv, will miss regularities it does not understand.

The main contribution of this paper is to shed light on how compression relates to capacity, and this contribution is clearly in line with other recent research (Brady et al., 2009). The present proposal can be applicable to verbal chunks, by considering that compression is achieved by the use of pointers to LTM content or by the use of chunk's first words as indices of over-learned word associations (Chen & Cowan, 2009; Cowan et al., 2004), and to visual material for which compression can be achieved by clustering items by proximity in tasks such as matrix recall or corsi blocks (since subjects can take profit of accidental regularities; see Gmeindl et al., 2011). Although we targeted a precise (i.e., quantitative) evaluation of how chunks are built using information that can easily been recoded without necessarily relying on specific long-term-memory knowledge, our conception is not incompatible with other processes more directly implying long-term-memory. For more specific material, the compressibility can also depend on long-term-memory knowledge that can help organizing or grouping the input into familiar units, a process that recalls the original definition of a chunk (e.g., relabeling the letters of the alphabet by the word "alphabet"), although this way of characterizing the chunks as unique "pointers" is more straightforward.

5.1. The distribution of complexity in random sequences

Our approach allows the capacity of working memory to be understood in a more mathematically rich way, a benefit we illustrate by briefly elaborating on several quantitative relationships.

Our argument is that Miller's magical number 7 is essentially an artifact of statistically typical compression ratios. That is, random sequences (as were typical of the studies on which Miller based his estimate) randomly contain some number of accidental patterns which result in chunking and thus compressed representations (Feldman, 2004). More specifically, each successive item in the sequence (after the first) will accidentally continue a pattern established by the previous item or items with some fixed probability ϵ. With our monotonic runs the probability of any given run continuing by accident is about $\epsilon = 1/10$, because each class of runs (e.g. +1, +2, +3, −1, −2, or −3 increments) can be continued only by one specific digit. For example the run 23 can only be continued by a 4. (In fact ϵ is slightly less than 1/10, actually about *0.085*, because runs cannot be continued past limits of 0 and 9, which reduces the possible continuations.) There are R disjoint classes of patterns (in our case $R = 6$). Each successive digit initiates a new chunk *unless* it accidentally continues a pattern, so the expected number of runs is N minus the expected number of accidental pattern continuations.

The number of accidental pattern continuations is in effect the number of successes on $(N-1)$ independent tosses of an ϵ-weighted coin (a Bernoulli sequence), which means that the total number of pattern continuations will follow a binomial distribution with mean $\epsilon R(N-1)$ and variance $\epsilon(1-\epsilon)R(N-1)$. So the expected complexity (number of resulting chunks) will be binomially distributed with mean (expectation)

$$E(\text{complexity}) = N - \epsilon R(N-1), \qquad (2)$$

and variance $\epsilon(1-\epsilon)R(N-1)$. The expected compression ratio will be

$$E(\text{compression ratio}) = \frac{N}{E(\text{complexity})}, \qquad (3)$$

which with values $\epsilon = .085$, $R = 6$ and $N = 7$ equals 1.78—not far from the 7/4(=1.75) ratio between Miller's number and Cowan's. This analysis is admittedly approximate, but it illustrates that for moderately probable accidental patterns and moderately long sequences, the expected number of chunks in a random pattern is roughly consistent with our account.

Finally, we can invert this reasoning to consider the distribution of raw pattern length that compresses to a *fixed* complexity, say 4 chunks. That is, how long on average are the random strings that happen to be just long enough (and complex enough) to compress to 4 units? Technically, this is the marginal distribution of raw pattern length conditioned on complexity = 4, which (because the binomial is a conjugate family) will also be binomial but with standard deviation expanded by the compression ratio. With values given above the variance of complexity is about 2.8, meaning that the standard deviation is about $\sqrt{2.8}$ or 1.67 chunks, which decompresses to about 2.97 raw digits in the uncompressed string. That is, the distribution of strings that compress to four chunks have lengths that follow a binomial distribution with mean about 7 and standard deviation about 3—echoing (though not quite matching) Miller's 7 ± 2. So our account gives an explanation not only of Miller's mean value but also, very roughly, of its spread. In other words, digit span has a random distribution (7 ± 2) because *compressibility* has a random distribution, stemming from the accidental patterns that occur in randomly chosen strings.

5.2. The role of phonological rehearsal

Considering that the to-be-remembered material is partially compressed using verbal information (verbal information can naturally support the recoding of the runs into shorter descriptions), the compressed verbal information probably shortens the pronunciation of the original list (for instance, 1–9, instead of 123456789), a process that should facilitate recall in a way that is parallel to the pure algorithmic compressibility that is manipulated here. Conversely, new kinds of verbal interference/acoustic confusion can probably occur during this verbal recoding process, as well as during the rehearsal and the recall phases, because of proper articulation constraints and auditory representations. Therefore, the more a sequence can be potentially redescribed, the more phonological confusion effects can arise. Hence this relates to a major debate in the STM literature concerning the distinctive role of the phonological loop component in the Baddeley and Hitch (1974) model. Because our task did not target articulatory suppression, any effects due to verbal strategies would fall outside our predictions. Our participants were therefore liable to confuse increments, starting points and the lengths of the newly formed chunks. It is also possible that our subjects needed to vocalize the numbers while unpacking them from the chunks, hence producing an additional subvocal output effect that depended on the length of the original sequence. Overall, this phenomenon might unfortunately result in a lower estimate of the chunking abilities based on algorithmic compression.

Because it is difficult to disentangle algorithmic complexity (information stored in conceptual form appropriate for underlying chunk formation) and verbal complexity (information stored in phonological form) in our tasks, such confounding seems an inevitable problem, although our results still suggest the overall presence of a compression factor. In conclusion, we agree that the limits in the subjects' ability to recall our chunkable material is probably modulated by a linguistic factor, although we believe that this factor mainly operates by degrading the available regularity (given that natural language cannot have a more powerful expressiveness than algorithmic complexity).

We are considering future experimentation that could potentially test the contribution of verbal memory in our chunking span tasks. For instance, the testing of localist and globalist assumptions (Cowan, Baddeley, Elliott, & Norris, 2003) could take profit of the analogy between the short words vs long words factor used in previous studies and the small chunks *vs.* large chunks opposition that calls into play in our study. For instance, it seems that both localist (Neath & Nairne, 1995) and globalist (Bryant, 1986; Cowan et al., 1992) approaches predict that the overall proportion correct decreases as the proportion of long words increase for lists of fixed length. This is not reflected in our data, which rather suggests that algorithmic complexity surpasses verbal complexity.

5.3. A comparison to the minimum description length approach

There has been a long tradition in psycholinguistics of using information theory to account for word recognition and word segmentation (Altmann & Carter, 1989), two processes that have often been studied through the learning of redundant strings generated by artificial grammars (Miller, 1958; Perruchet & Pacteau, 1990; Pothos, 2010; Pothos & Wolff, 2006; Robinet, Lemaire, & Gordon, 2011; Servan-Schreiber & Anderson, 1990). The present paper strengthens the hypothesis that recoding of information underpins chunking, with the limits of capacity set by optimal compression. Information directly leads to code lengths, and the computation of code lengths (such as those computed by Shannon-Fano, Sayeki, 1969 or Huffman coding) is straightforward to estimate compressibility. For instance, given a set of four objects associated to different counts (x,x,x,x,y,y,z,w), a basic encoding of objects ($x = 00, y = 01, z = 10, w = 11$) leads to a total of 16 bits after the different objects are concatenated using their code: 00, 00, 00, 00, 01, 01, 10, 11. Using a more optimal Huffman code such as ($x = 0, y = 10, z = 110, w = 111$) however allows more compression: 0, 0, 0, 0, 10, 10, 110, 111, comprising 14 bits. Usually, an MDL solution (Rissanen, 1978) is combined with this Huffman method because the length of the redescriptions (e.g., $x = 0$, $y = 10$, $z = 110$, $w = 111$, or $x = 00$, $y = 01$, $z = 10$, $w = 11$) still needs to be taken into account in order to quantify the total memory space taken by the compression process. The MDL principle states that the best explanation is the one that permits the greatest compression of the data, including the recoding process. Let's imagine that the following 15 character sequence 123123454512345 needs to be compressed. The redescription $a = 123$ and $b = 45$ allows optimal compression since it leads to an *aabbab* representation, for a total of 13 characters when both the recoding process and the writing of the new sequence are combined ($a + 123 + b + 45 + aabbab$). However, using $a = 123$, $b = 45$, and $c = 12345$, although leading to a shorter *aabbc* sequence, the overall compression process is more costly ($a + 123 + b + 45 + c + 12345 + aabbc = 18$ characters) than the original sequence.

Nevertheless, this very useful MDL approach is not totally advantageous when a few regular patterns show equivalent frequencies, which is the case of the short lists that our participants were given. Effectively, the recoding of 12349876 in $a = 1234$ and $b = 9876$ in order to retain *ab* does not lead to any sort of compression. This does not mean however that some other kind of recoding process cannot operate. For instance, the remembering of 123456789 per se seems very lengthy in comparison with a more structured algorithmic representation such as "1–9". The same remark applies to the 0112358132134... sequence (the Fibonacci number) that can be unpacked from the brief $i_n = i_{n-1} + i_{n-2}$ formula, as long as one figures it out. Algorithmic compression can underlie many coding schemes, and in that respect, Kolmogorov complexity is generic enough to express how optimal compression can be achieved when allowing every possible way for subjects to simplify a sequence. For instance, the sequence 011235813213443112318532110 is very compressible because the algorithmic distance between 0112358132134 and 43112318532110 is very low (the latter is the inverse of the former). By combining the fibonacci formula and the inverse function, an optimal compression is achieved using two chunks. The underlying principle in our study is that the ability to compress the data reflects how much the subject has learned about the regularities in the data, and that a pretty good estimate of regularity is the (approximate) Kolmogorov complexity.

6. Conclusion

Our results highlight the centrality of compression in working memory, and suggest that the size of a sequence after compression—in other words, its Kolmogorov complexity—determines how much space it occupies in memory. This means that both Miller's magic number (7 ± 2) and Cowan's (4 ± 1) are correct—but they refer to different quantities. Cowan's number refers to the number of chunks that can be stored, which in our account is proportional to the size of the string after compression. Miller's number refers to the number of digits in an *uncompressed* string of approximately typical compressibility, which when compressed reduces to about four chunks. The more meaningful limit is Cowan's, because it is the size after compression that determines how much can really be stored.

Acknowledgements

This research was supported in parts by grants from the Centre National de la Recherche Scientifique/Institut National des Sciences Humaines et Sociales (PEPS grant) to F.M. in 2009, from the Agence Nationale de la Recherche (Grant # ANR-09-JCJC-0131-01) to F.M., and from NSF EITM 0339062 to J.F. The material in this paper was presented in part at the 51st Annual Meeting of the Psychonomic Society, November 18–21, St. Louis, MO. We would like to thank Nelson Cowan and two anonymous reviewers for their comments on this paper.

References

Altmann, G., & Carter, D. (1989). Lexical stress and lexical discriminability: Stressed syllables are more informative, but why? *Computer Speech & Language, 3*, 265–275.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language, 38*, 341–380.

Anderson, J. R., & Matessa, M. (1997). A production system theory of serial memory. *Psychological Review, 104*, 728–748.

Baddeley, A. (1986). *Working memory*. New York: Oxford University Press.

Baddeley, A. (1994). The magical number seven: Still magic after all these years? *Psychological Review, 101*, 353–356.

Baddeley, A., & Hitch, G. (1974). Working memory. In G. H. Bower (Ed.), *Recent advances in learning and motivation* (pp. 647–667). New York, NY: Academic Press.

Baddeley, A., Thomson, N., & Buchanan, M. (1975). Word length and the structure of short-term memory. *Journal of Verbal Learning & Verbal Behavior, 14*, 575–589.

Barouillet, P., Bernardin, S., & Camos, V. (2004). Time constraints and resource sharing in adults' working memory spans. *Journal of Experimental Psychology: General, 133*, 83–100.

Barouillet, P., Bernardin, S., Portrat, S., Vergauwe, E., & Camos, V. (2007). Time and cognitive load in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 33*, 570–585.

Baum, E. B. (2004). *What is thought?* Massachusetts, MA: MIT Press.

Bennett, C. H. (1995). Logical depth and physical complexity. In R. Herken (Ed.), *The universal turing machine: A half-century survey* (pp. 227–257). New York, NY: Springer-Verlag.

Bower, G. H., & Winzenz, D. (1969). Group structure, coding, and memory for digit series. *Journal of Experimental Psychology, 80*, 1–17.

Brady, T. F., Konkle, T., & Alvarez, G. A. (2009). Compression in visual working memory: Using statistical regularities to form more efficient memory representations. *Journal of Experimental Psychology: General, 138*, 487–502.

Broadbent, D. (1975). The magic number seven after fifteen years. In A. Kennedy & A. Wilkes (Eds.), *Studies in long-term memory*. New York: Wiley.

Brown, G. D. A., Neath, I., & Chater, N. (2007). A temporal ratio model of memory. *Psychological Review, 114*, 539–576.

Brown, G. D. A., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review, 107*, 127–181.

Bryant, R. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Compilers, C-35*, 8.

Burgess, N., & Hitch, G. (1999). Memory for serial order: A network model of the phonological loop and its timing. *Psychological Review, 106*, 551–581.

Burtis, P. J. (1982). Capacity increase and chunking in the development of short-term memory. *Journal of Experimental Child Psychology, 34*, 387–413.

Cabrera, A. (1995). The"rational" number e: A functional analysis of categorization. In J. D. Moore & J. F. Lehman (Eds.), *Proceedings of the 17th annual conference of the cognitive science society*. Hillsdale, NJ: Lawrence Erlbaum.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology, 4*, 55–81.

Chater, N., & Vitányi, P. (2003). Simplicity: A unifying principle in cognitive science. *Trends in Cognitive Sciences, 7*(1), 19–22.

Chekaf, M., & Mathy, F. (submitted for publication). Chunking memory span of objects varying in number, order and compressibility.

Chen, Z., & Cowan, N. (2005). Chunk limits and length limits in immediate recall: A reconciliation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 31*, 1235–1249.

Chen, Z., & Cowan, N. (2009). Core verbal working-memory capacity: The limit in words retained without covert articulation. *Quarterly Journal of Experimental Psychology, 62*, 1420–1429.

Cohen, N., & Sekuler, R. (2010). Chunking and compound cueing of movement sequences: Learning, retention and transfer. *Perceptual & Motor Skills, 110*, 736–750.

Conway, A. R., Kane, M. J., Bunting, M. F., Hambrick, D. Z., Wilhelm, O., & Engle, R. W. (2005). Working memory span tasks: A methodological review and user's guide. *Psychonomic Bulletin & Review, 12*(5), 769–786.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: John Wiley.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences, 24*, 87–185.

Cowan, N. (2010). The magical mystery four: How is working memory capacity limited, and why? *Current Directions in Psychological Science, 19*, 51–57.

Cowan, N., Baddeley, A. D., Elliott, E. M., & Norris, J. (2003). List composition and the word length effect in immediate recall: A comparison of localist and globalist assumptions. *Psychonomic Bulletin & Review, 10*, 74–79.

Cowan, N., Chen, Z., & Rouder, J. N. (2004). Constant capacity in an immediate serial-recall task: A logical sequel to miller (1956). *Psychological Science, 15*, 634–640.

Cowan, N., Day, L., Saults, J. S., Keller, T. A., Johnson, T., & Flores, L. (1992). The role of verbal output time in the effects of word length on immediate memory. *Journal of Memory and Language, 31*, 1–17.

Cowan, N., Hismjatullina, A., AuBuchon, A. M., Saults, J. S., Horton, N., & Leadbitter, K. (2010). With development, list recall includes more chunks, not just larger ones. *Developmental Psychology, 46*, 1119–1131.

Crannell, C. W., & Parrish, J. M. (1957). A comparison of immediate memory span for digits, letters, and words. *Journal of Psychology: Interdisciplinary and Applied, 44*, 319–327.

De Kleine, E., & Verwey, W. B. (2009). Motor learning and chunking in dyslexia. *Journal of Motor Behavior, 41*, 331–337.

Dirlam, D. K. (1972). Most efficient chunk sizes. *Cognitive Psychology, 3*, 255–359.

Engle, R. W. (2002). Working memory capacity as executive attention. *Current Directions in Psychological Science, 11*, 19–23.

Ericsson, K. A., Chase, W., & Faloon, S. (1980). Acquisition of a memory skill. *Science, 208*, 1181–1182.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review, 102*, 211–245.

Estes, W. K. (1972). An associative basis for coding and organization in memory. In A. W. Melton & E. Martin (Eds.), *Coding processes in human memory* (pp. 161–190). Washington, DC: Winston.

Estes, W. K. (1973). Phonemic coding and rehearsal in short-term memory for letter strings. *Journal of Verbal Learning & Verbal Behavior, 12*(4), 360–372.

Farrell, S. (2008). Multiple roles for time in short-term memory: Evidence from serial recall of order and timing. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 34*, 128–145.

Farrell, S., & Lewandowsky, S. (2002). An endogenous distributed model of ordering in serial recall. *Psychonomic Bulletin & Review, 9*, 59–79.

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

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

Fountain, S. B., & Benson, D. M. (2006). Chunking, rule learning, and multiple item memory in rat interleaved serial pattern learning. *Learning and Motivation, 37*, 95–112.

Gilchrist, A. L., Cowan, N., & Naveh-Benjamin, M. (2009). Investigating the childhood development of working memory using sentences: New evidence for the growth of chunk capacity. *Journal of Experimental Child Psychology, 104*, 252–265.

Gmeindl, L., Walsh, M., & Courtney, S. (2011). Binding serial order to representations in working memory: A spatial/verbal dissociation. *Memory & Cognition, 39*, 37–46.

Gobet, F., & Clarkson, G. (2004). Chunks in expert memory: Evidence for the magical number four… or is it two? *Memory, 12*, 732–747.

Gobet, F., Lane, P. C., Croker, S., Cheng, P. C. H., Jones, G., & Oliver, I. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences, 5*, 236–243.

Hahn, U., Chater, N., & Richardson, L. B. (2003). Similarity as transformation. *Cognition, 87*, 1–32.

Halford, G. S., Baker, R., McCredden, J. E., & Bain, J. D. (2005). How many variables can humans process? *Psychological Science, 16*, 70–76.

Halford, G. S., Wilson, W. H., & Phillips, W. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental and cognitive psychology. *Behavioral and Brain Sciences, 21*, 803–831.

Henson, R. N. A. (1998). Short-term memory for serial order: The start-end model. *Cognitive Psychology, 36*, 73–137.

Henson, R. N. A., Norris, D. G., Page, M. P. A., & Baddeley, A. D. (1996). Unchained memory: Error patterns rule out chaining models of immediate serial recall. *The Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology, 49*, 80–115.

Hitch, G., Burgess, N., Towse, J., & Culpin, V. (1996). Temporal grouping effects and working memory: The role of the phonological loop. *Quarterly Journal of Experimental Psychology, 49A*, 116–139.

Jonides, J., Lewis, R. L., Nee, D. E., Lustig, C. A., Berman, M. G., & Moore, K. S. (2008). The mind and brain of short-term memory. *Annual Review of Psychology, 59*, 193–224.

Kintsch, W., & Caciopo, J. (1994). Introduction to the 100th anniversary issue of the psychological review. *Psychological Review, 101*, 195–199.

Koch, I., Philipp, A. M., & Gade, M. (2006). Chunking in task sequences modulates task inhibition. *Psychological Science, 17*, 346–350.

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

Lewandowsky, S. (1999). Redintegration and response suppression in serial recall: A dynamic network model. *International Journal of Psychology, 34*, 434–446.

Lewandowsky, S., Duncan, M., & Brown, G. D. A. (2004). Time does not cause forgetting in short-term serial recall. *Psychonomic Bulletin & Review, 11*, 771–790.

Lewandowsky, S., & Murdock, B. B. (1989). Memory for serial order. *Psychological Review, 96*, 25–57.

Li, K. Z., Blair, M., & Chow, V. S. (2010). Sequential performance in young and older adults: Evidence of chunking and inhibition. *Aging Neuropsychology, and Cognition, 17*, 270–295.

Li, M., & Vitányi, P. (1997). *An introduction to Kolmogorov complexity and its applications*. New York, NY: Spring Verlag.

Logan, G. D. (2004). Working memory, task switching, and executive control in the task span procedure. *Journal of Experimental Psychology: General, 133*, 218–236.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature, 390*, 279–281.

MacGregor, J. N. (1987). Short-term memory capacity: Limitation or optimization? *Psychological Review, 94*, 107–108.

Martin, M. (1978). Memory span as a measure of individual differences in memory capacity. *Memory & Cognition, 6*, 194–198.

Mathy, F., & Varré, J. S. (submitted for publication). Retention-error patterns in serial-recall tasks revisited by bioinformatics.

Maylor, E. A., Vousden, J. I., & Brown, G. D. A. (1999). Adult age differences in short-term memory for serial order: Data and a model. *Psychology and Aging, 14*, 572–594.

McCormack, T., Brown, G. D. A., Vousden, J. I., & Henson, R. N. A. (2000). Children's serial recall errors: Implications for theories of short-term memory development. *Journal of Experimental Child Psychology, 76*, 222–252.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, 81–97.

Miller, G. A. (1958). Free recall of redundant strings of letters. *Journal of Experimental Psychology, 56*, 485–491.

Murdock, B. B. (1962). The serial position effect of free recall. *Journal of Experimental Psychology, 64*, 482–488.

Nairne, J. S. (1990). A feature model of immediate memory. *Memory & Cognition, 18*, 251–269.

Nairne, J. S. (2002). Remembering over the short-term: The case against the standard model. *Annual Review of Psychology, 53*, 53–81.

Neath, I., & Nairne, J. S. (1995). Word-length effects in immediate memory: Overwriting trace decay theory. *Psychonomic Bulletin & Review, 2*, 429–441.

Ng, H. L., & Maybery, M. T. (2002). Grouping in short-term verbal memory: Is position coded temporally? *Quarterly Journal of Experimental Psychology, 55*, 391–424.

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

Oberauer, K., & Kliegl, R. (2006). A formal model of capacity limits in working memory. *Journal of Memory and Language, 55*, 601–626.

O'Shea, G., & Clegg, B. A. (2006). Stimulus and response chunking in the hebb digits task. *Psychological Research, 70*, 180–192.

Page, M. P., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review, 105*, 761–781.

Perlman, A., Pothos, E. M., Edwards, D. J., & Tzelgov, J. (2010). Task-relevant chunking in sequence learning. *Journal of Experimental Psychology: Human Perception and Performance, 36*, 649–661.

Perruchet, P., & Pacteau, C. (1990). Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *Journal of Experimental Psychology: General, 119*, 264–275.

Perruchet, P., & Pacton, S. (2006). Implicit learning and statistical learning: One phenomenon, two approaches. *Trends in Cognitive Sciences, 10*, 233–238.

Pothos, E. M. (2010). An entropy model for artificial grammar learning. *Frontiers in Psychology, 1*, 16.

Pothos, E. M., & Chater, N. (2002). A simplicity principle in unsupervised human categorization. *Cognitive Science, 26*, 303–343.

Pothos, E. M., & Wolff, J. G. (2006). The simplicity and power model for inductive inference. *Artificial Intelligence Review, 26*, 211–225.

Pylyshyn, Z., & Storm, R. (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision, 3*, 1–19.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica, 14*, 465–471.

Robinet, V., & Lemaire, B. (2009). MDLChunker: A MDL-based model of word segmentation. In N. A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31th annual conference of the cognitive science society* (pp. 2866–2871). Austin, TX: Cognitive Science Society.

Robinet, V., Lemaire, B., & Gordon, M. B. (2011). Mdlchunker: A mdl-based cognitive model of inductive learning. *Cognitive Science, 35*, 1207–1405.

Roodenrys, S., & Miller, L. M. (2008). A constrained rasch model of trace redintegration in serial recall. *Memory & Cognition, 36*(3), 578–587.

Ryan, J. (1969). Grouping and short-term memory: Different means and patterns of grouping. *Quarterly Journal of Experimental Psychology, 21*, 137–147.

Sargent, J., Dopkins, S., Philbeck, J., & Chichka, D. (2010). Chunking in spatial memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 36*, 576–589.

Sayeki, Y. (1969). Information seeking for object identification. *Organizational Behavior and Human Performance, 4*, 267–283.

Schöning, U., & Pruim, R. (1998). *Gems of theoretical computer science*. Berlin: Springer.

Servan-Schreiber, E., & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*, 592–608.

Shiffrin, R. M., & Nosofsky, R. M. (1994). Seven plus or minus two: A commentary on capacity limitations. *Psychological Review, 101*, 357–361.

Simon, H. A. (1974). How big is a chunk? *Science, 183*, 482–488.

Stark, K., & Calfee, R. C. (1970). Recoding strategies in short-term memory. *Journal of Experimental Psychology, 85*, 36–39.

St Clair-Thompson, H., & Sykes, S. (2010). Scoring methods and the predictive ability of working memory tasks. *Behavior Research Methods, 42*, 969–975.

Suge, R., & Okanoya, K. (2010). Perceptual chunking in the self-produced songs of bengalese finches (lonchura striata var. domestica). *Animal Cognition, 13*, 515–523.

Terrace, H. S. (1987). Chunking by a pigeon in a serial learning task. *Nature, 325*, 149–151.

Terrace, H. S. (2001). Chunking and serially organized behavior in pigeons, monkeys and humans. In R. G. Cook (Ed.), *Avian visual cognition*. Medford, MA: Comparative Cognition Press. <http://www.pigeon.psy.tufts.edu/avc/terrace/>.

Tulving, E., & Patkau, J. E. (1962). Concurrent effects of contextual constraint and word frequency on immediate recall and learning of verbal material. *Canadian Journal of Experimental Psychology, 16*, 83–95.

Unsworth, N., & Engle, R. W. (2007). On the division of short-term and working memory: An examination of simple and complex span and their relation to higher order abilities. *Psychological Bulletin, 133*, 1038–1066.

Wickelgren, W. A. (1964). Size of rehearsal group and short-term memory. *Journal of Experimental Psychology, 68*, 413–419.

Wolff, J. G. (2003). Information compression by multiple alignment, unification and search as a unifying principle in computing and cognition. *Artificial Intelligence Review, 19*, 193–230.

Zhang, G., & Simon, H. A. (1985). Stm capacity for chinese words and idioms: Chunking and acoustical loop hypotheses. *Memory & Cognition, 13*, 193–201.

Zhang, W., & Luck, S. (2008). Discrete fixed-resolution representations in visual working memory. *Nature, 453*, 233–236.

Ziv, J., & Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory, 23*, 337–343.