Supplement to: *Evolving building blocks of rhythm: How human cognition creates music via cultural transmission*

1. Experimental Methods

1.1. Participants

12 participants (7 males) were recruited using the University of Edinburgh's graduate employment service (mean age 24.25). The sample size was established a priori and based on number of experimental chains (instead of participants) for comparability with the previous study. All participants were paid £30. Conditions for participation were the lack of any formal musical education, to any degree.

1.2. Materials

Stimuli for the experiment were sequences of midi drum beats, stored as lists of numbers. Each drum beat was defined by two values, velocity (on a scale of 0-127), representing hit strength, and interonset-interval (IOI), representing time in seconds between beats. To play a sequence to the participant, the experiment code transformed the velocity values into midi output (midi note 40, snare drum), separated by intervals determined by the IOI values. After all beats had been played, there would be 1.5 seconds of pause, and a single midi note of 32 (cymbal) would signal to the participant that the sequence had ended. Midi output was converted into audio by the software Garageband, set to its percussion channel (standard rock kit).

The training stimuli for each round were 32 snare drum sequences. Presentation of these 32 basic sequences was repeated twice, each time randomizing their order. After a sequence was played, the participant would copy it on a midi drum set. A custom Python script collected values of velocity and IOI from the participant's input and stored them. The 32 basic training sequences for the first generation of each chain were identical to the 32 basic training sequences used in our previous experiment (Ravignani et al., 2016). Briefly, first-generation sequences comprised 12 beats, whose values of velocity and IOI had been randomly generated (each IOI was randomly sampled from a uniform distribution between 100 and 1,000 milliseconds). For all other rounds of the experiment, training sequences were built using the last 32 output sequences of the participant's previous round, again produced twice in random order. Sequence length was left free to vary, with minimum length being 1 beat.

The experiment was run on a MacBook Air (OS 10.8.5), which received midi input from a midi drum-kit (Alesis SamplePad). Drum-kit and laptop were connected via an USB-midi external interface (Roland Duo-Capture Ex). Sound were played over headphones connected to the USB-midi interface, whose volume the participant was free to adjust at the beginning of the experiment. The participant received a single standard wooden drumstick.

The script used to collect data was a modified version of the Python script in (Ravignani et al., 2016). The only difference consists in the new script using the (better documented) mido and time modules instead of the rtmidi module. This alteration did not lead to a noticeable difference when replaying the sequences.

1.3. Design

This study aimed at replicating the six diffusion chains in (Ravignani et al., 2016). Each of the original chain was used here as generation 0 for two participants. This lead to a total of 12 chains (versus 6 in the previous study). All rounds of a single experiment were run on the same day, to control for potential beneficial effects of sleep on learning (Maquet, 2001). In an effort to reduce the overall duration of the experiment, and considering the little variation in the original chains after generation 5, each chain consisted of 6 generations: one computer-generated generation 0, and 5 experimental generations. Chains were named after the shared input with original chains, with chains 1_1 and 1_2 being replications of chain 1. Participants underwent five rounds of 20-25 minutes each, separated by four 15-minute breaks. Each experiment lasted approximately 3 hours.

1.4. Procedure

Participants were given consent forms, which included, among others, the instructions: "You are about to participate in a study which involves listening to sequences of drum beats and immediately repeating them on the electronic drum kit provided by the experimenter, to the best of your abilities. You will be shown how to use the kit before starting. The experiment is split into five rounds, each lasting approximately 30 minutes and followed by a break of approximately 15 minutes. Your session should last for up to 3.5 hours. You will be given full instructions shortly and will be able to ask any questions you may have."

After signing the consent forms, participants were given headphones and a drumstick. Apart from the written information in the form, participants were debriefed by the experimenter. They were told they would listen to sequences comprised of "certain number of drum beats", and they would know the sequence had ended when they heard a cymbal sound. They were given a few moments to test the pads and recognise that the top-right pad produced the cymbal noise, and were told that that pad should only be used to signal the end of their own sequence. Once they were satisfied with the range of sounds they could produce, participants were told a round would last approximately 25 minutes, and there would be five of these rounds, separated by four breaks of up to 15 minutes. Participants were unaware whether they would listen to stimuli produced by a computer, a previous learner, or themselves at a previous time.

Before starting, participants were given three recommendations. Firstly, they were not expected to replicate the sequences perfectly, but they were expected to "make an honest effort to replicate them to the best of" their abilities.

Secondly, if they heard a different drum sequence in their headphones while still replicating the previous one, it meant they had hit the cymbal pad by accident, and the code had given them the next sequence: they should therefore stop, listen to the new sequence and replicate that one once cued by the cymbal sound. Thirdly, the experimenter would be just outside the experimental room if they were feeling ill or faint or something important had come up, but barring those events, they should only come out once they failed to hear a new sequence, signalling the end of the round. Participants were then given another minute to familiarise themselves with the range of movements they could make and ensure the volume and headphone fit were comfortable. Once they were satisfied, the experimenter would start the script. The experimenter remained in the room with the participants for the first 3 sequences (of the 32 warm-up trials), to ensure they had understood the procedure and the code was working. At the end of the last round, participants filled in an information sheet asking for their age, gender and degree of familiarity with music, before leaving.

2. Model: Latent Variables

2.1. Data

Let $\vec{y} = (y_1, y_2, \ldots, y_n)$ be a time series of onsets for one drum pattern. Let $\vec{r} = (r_1, r_2, \ldots, r_{n-1})$ be the series of interval ratios, such that $r_i = \frac{y_i - y_{i-1}}{y_{i-1} - y_{i-2}}$.

2.2. Clustering datapoints into interval ratio categories

We assume that participants do not just perceive and represent the absolute datapoints (r_i) veridically, but potentially reduce the variation among datapoints by assigning each to a rhythmic category. These categories are assumed to be learned in an unsupervised fashion by the learner. This category structure is modeled as a weighted mixture of k Gaussian distributions. Each category is defined by an unknown mean (μ_k) and variance (σ_k), and its weight in the mixture (ω_k), where $0 \leq \omega_k \leq 1$ and $\sum_k \omega_k = 1$. Let $\vec{z} = (z_1, z_2, \ldots, z_{n-1})$ be a vector of category assignments indices such that we would write $z_i = k$ if datapoint r_i has been assigned to rhythmic category k. When modelling a learner estimating these quantities, we assume a standard statistical model for Bayesian inference of an unknown mean and variance (the Normal-Inverse-Chi-Square model with uninformative prior parameters, see Gelman et al, 2006), and a standard statistical model for inference of the weights (weights $\vec{\Omega} = (\omega_1, \ldots, \omega_k)$ are modelled as a multinomial probability vector estimated under a uniform Dirichlet prior, see Gelman et al. 2006).

2.3. Chunking subseries into motifs

We wish to describe the structure in drum patterns as transitions between predictably structured, prototypical categories of subseries, or *motifs*. A motif is a kind (type), and we observe instances (tokens) of that motif in the data. Every subseries is an instance of a motif. Let $\vec{m} = (m_1, m_2, \ldots, m_{n-1})$ be notation that tells us which motif each datapoint has been assigned to. As before, $m_i = j$ tells us that datapoint r_i has been judged to be part of an instance of motif j.

2.4. *Boundary Variables*

Allow the notation $\vec{b} = (b_1, b_2, \ldots, b_n)$ to denote the boundaries between subseries. Arbitrarily, we will keep track of the onset in a subseries (rather than the coda). If datapoint r_i is the first interval ratio of a motif, then $b_i = 1$, otherwise $b_i = 0$. By definition $b_0 = 1$.

2.5. *Motif internal Structure*

We pursue one of the simplest possible formulations of motif categories: each category of motif is defined by a single prototype series of interval ratio categories. The probability that a newly observed subseries belongs to a particular category of motif depends on the subseries being an exact instance of the prototype sequence, or a close match.

3. Model: A sequential learning algorithm for the latent variables

Our focus is on approximating the representations of patterns that participants induce upon observing these patterns. As a result we choose to model individual participants using a lightweight, sequentially dependent posterior approximation algorithm for the latent variables defined by our model. We take McCauley and Christiansen's (2012) CAPPUCINO model as inspiration for our algorithm. This model provides an online, sequential updating procedure that segments sequences of observed events (in their case words, in our case IOI ratios) and chunks these sequences into predictable subsequences by keeping track of transition probabilities between events. Here we describe our adaptation of this algorithm, and the additional variables it requires.

3.1. *Transition Probabilities*

The core assumption underpinning this algorithm is that individual use statistical information about the co-occurrence of interval ratio categories (z_i) to make decisions about boundaries between motifs in patterns. Improbable sequences of interval ratio categories imply a boundary between motifs, because by definition a motif is a predictable sequence of interval ratio categories: transition probabilities between categories within a motif are likely to be high, whereas transition probabilities between the final category in a preceding motif and the onset category in a succeeding motifs are likely to be lower. Dips in transition probabilities provide cues to boundaries.

Following McCauley and Christiansen, our algorithm keeps track of *backwards* transition probabilities. For all k categories of interval ratio, we keep track of a vector of probabilities $\vec{t_l} = (t_{l_1}, \ldots, t_{l,k})$ whose entries capture the probability that interval ratio category l is preceded by each other interval ratio category. Each vector t_l is a multinomial probability vector estimated under a uniform Dirichlet prior (unbiased Bayesian inference).

3.2. Creating and storing motifs

The algorithm described below assigns subseries to motifs based on the similarity between the subseries and a hypothesised motif. When a candidate subseries is an exact match for an existing motif which has been attested at least twice in previous patterns (this minimum count follows the CAPPUCINO model procedure), the model probabilistically assigns the new subseries to this motif category (we allow a small probability – 0.25 - that the model chooses instead to continue through the sequence before assigning a motif category). However, when a subseries of interval ratios is not an exact match for an existing motif, or if the model decides not to exploit the match, the model has the ability to create a new category of motifs entirely to account for the new datapoint. This decision process has a natural analogue statistical model in the Dirichlet Process, which is commonly formulated sequentially like our procedure. For any existing category of motifs, the model checks whether the motif matches the prototype motif, and assigns a likelihood of 1 to any matching categories. The prior for any motif category is the number of subseries already assigned to this category divided by a normalising constant $c = s - 1 + \alpha$, where s is the total number of subseries assigned to all motifs, and α is a parameter of the model. During every decision, the model also entertains the possibility that a new motif category must be created to account for this subseries. The marginal likelihood of a new motif is linearly inversely proportional to its length (1 over the number of interval ratios in the subseries), and the prior probability of a new motif is α divided by the same normalising constant c.

This formulation corresponds closely to a sequential understanding of the Dirichlet process, in which short motifs are privileged, and a parameter α controls how happy the learner is to invent new motif categories. In our analyses, we set $\alpha = 10^{-1}$, which corresponds to the assumption that participants weakly prefer to re-use existing motif categories where possible. We run the model ten times on each participants' data, and plot the average results in the main manuscript. This procedure is a close analogue of a particle filter, which is a commonly used technique in machine learning for sampling the posterior of a Dirichlet Process model.

3.3. Algorithm description

Here is procedure for running the algorithm to model an individual participant. We run this procedure only on the patterns observed (not produced), and only observed in the retest portion of the experiment (i.e. the final 32 patterns observed at each generation).

```
initialization:
1: construct an initial representation of interval ratio categories by categorizing, in order, every individual interval ratio in the initial sequence of 32 test patterns. For each individual datapoint, the categorisation procedure works by computing the posterior predictive distribution of each Gaussian interval ratio category, using this as the likelihood for the datapoint under consideration, and using the category mixture weight ($\omega_k$) as the category prior, then drawing a single sample from the posterior over categories.
2: Randomly sample initial transition probabilities from a uniform Dirichlet prior. During this sequence of categorisation decisions in step 1. of the initialization, update these transition probabilities sequentially by sampling the posterior, which is also Dirichlet thanks to the conjugacy of multinomial probability vector likelihoods and the Dirichlet prior.
for each drum pattern $p = 33, \ldots, 64$ do
    Set $b_1 = 1$
    for each interval ratio $r_i = 1, \ldots, n_p$ do
        i) categorise the interval ratio by sampling the posterior for $z_i$, as outlined in the initialisation;
        ii) if $b_i = 1$ or $b_{i-1} = 1$, skip the following steps (motifs cannot be a single interval ratio);
        iii) update the estimate of the transition probability $t_{z_{i-1},z_i}$ between $z_{i-1}$ and $z_i$ as decribed in section 3.1;
        iv) if the sequence $(z_*, \ldots, z_{i-1})$ is an exact match for any stored motif $x$, assign $(m_*, \ldots, m_{i-1}) = x$ probabilistically by drawing a Bernoulli sample with $p = .75$ (where the index $*$ represents the most recent onset boundary) and set $b_i = 1$, then skip the remaining steps;
        v) if the transition probability $t_{z_{i-1},z_i}$, between $z_{i-1}$ and $z_i$, is smaller than the overall mean of all transition probabilities ($\bar{t}$), set a boundary $b_i = 1$;
        vi) if step v) resulted in a boundary ($b_i = 1$), sample a motif assignment for the sequence $(z_*, \ldots, z_{i-1})$ as described in 3.2 ;
    end
end
```
Algorithm 1: Psuedo-code for the learning algorithm implementing our model. Our algorithm is based closely on the CAPPUCINO model, and is a close analogue of a particle filter posterior sampling procedure.