

Eberhard Karls Universität Tübingen
Faculty of Economics and Social Sciences
Degree Program Quantitative Data Science Methods,
M.Sc.

Interpretable Markovian Models of Animal Communication

Master's Thesis

Ananya Albrecht-Buehler
Born Nov. 10, 2002 in Madison, Wisconsin

Matriculation Number 6978208

1. Supervisor: Prof. Dr. Dmitry Kobak
2. Supervisor: Dr. Alex Jordan

Submission date: September 1, 2026

Declaration of Academic Integrity

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such. This thesis has neither been previously submitted as a whole or in any significant part as part of any other examination process, and the thesis has not been published as a whole or in any significant part, and that the copy submitted in electronical form is identical in content to the bound copies submitted.

Ananya Albrecht-Buehler
Name

Tübingen, 24.08.2026
Place, Date


Signature

Abstract

Animal communication can be represented as sequences of discrete, categorical signals. These signals alone often do not convey information, but instead are combined to form complex, meaningful behavioral motifs. Therefore, a prerequisite in understanding the meaning behind communication is understanding how these signals are organized. Achieving this requires both exploratory methods for identifying structure, as well as experimental paradigms for testing hypotheses about that structure.

This thesis proposes a pipeline for approximating complex historical dependencies in behavioral sequences using fully interpretable analysis methods. Specifically, it proposes a family of probabilistic Markovian models based on generalizations of Markov chains. These models can integrate more complex historical information in sequences, referencing customizable strategies for extracting key historical information in sequences. This thesis presents two such strategies, both motivated by biological applications and theory, and evaluates their performance in synthetic datasets mimicking features of collected empirical corpora.

These models are applied to two animal communication datasets, featuring two different behavioral modalities. The first dataset consists of postural behavioral sequences collected from six species of cichlid during dyadic territory disputes. The second consists of acoustic recordings of song from six male Bengalese finches.

The analysis revealed that Bengalese finch song demonstrates a highly deterministic process at the beginnings which violates modeling assumptions. To address this, this thesis describes a method for detecting and extracting this process from sequences by adapting tries, a data structure typically used in computer science applications.

Finally, novel corpora were generated using fitted models, which can be used for future playback experiments to move from exploratory analysis to hypothesis testing. Possible experimental designs and hypotheses were explored, as well as strengths and limitations of the proposed methods.

Contents

1	General Introduction	1
1.1	Representing Communication	1
1.2	Detecting Structure in Communication	2
1.3	The Model Species	3
1.4	Playback Studies	4
1.5	Thesis Outline	4
2	Probabilistic Markovian Models	6
2.1	First-order and k -order Markov Chains	6
2.2	Estimating k using Model Comparisons	6
2.3	Simulation Study	10
2.3.1	Methods and Analysis	10
2.3.2	Results	12
2.4	Interim Discussion	15
3	Applications to Animal Communication	16
3.1	Cichlid Postural Communication	16
3.1.1	Materials and Preprocessing	16
3.1.2	Descriptive Statistics and String Behavior	17
3.1.3	Markovian Modeling Results	20
3.1.4	Transition Matrix Heatmaps	22
3.2	Bengalese Finch Song	24
3.2.1	Materials and Preprocessing	24
3.2.2	Descriptive Statistics and String Behavior	25
3.2.3	Prefix Tries for Extracting Regularities	30
3.2.4	Cropped Corpora	35
3.2.5	Sanity Checks for Prefix Extraction	38
3.2.6	Markovian Modeling Results	39
3.2.7	Transition Matrix Heatmaps	41

4	Naturalistic Sequence Generation	43
4.1	Generation App	43
4.1.1	Adding Prefixes	44
4.1.2	Generated Corpora	44
4.2	Suggestions for Future Applications	47
5	Main Discussion	50
5.1	Species Comparisons	50
5.2	Strengths and Limitations	51
5.3	Future Directions	52
5.4	Conclusion	53
	Bibliography	55
A	Sanity Checks for Prefix Extraction	60
B	Generated and Empirical Sequence Comparisons	66
C	Code and Software	69

Acknowledgments

I am incredibly grateful to Jun.-Prof. Dr. Lena Veit for acting as an unofficial third supervisor to this project and giving such helpful advice and feedback on my thesis. Thank you very much for the enthusiasm and encouragement you showed towards my research. I was thrilled to be able to include birdsong in this project, and I am very grateful that you gave me the opportunity to do so. I would additionally like to thank Priya Binwal for generously providing me with her finch datasets.

I would particularly like to thank Dmitry for being open to supervising this project, despite it being incredibly unrelated to his own research. I appreciate your unending patience with me and the exciting directions to which you guided this project.

A great thanks also goes out to Alex, who first allowed me to pursue these research questions, even though we had originally planned a completely different project. I really appreciate the immense encouragement you gave me to dive (haha) head-first into the topic when I first brought it up, with no questions asked. Thanks also goes to Dr. Noori Choi for providing me with the cichlid datasets and answering all my naive questions.

A massive thank you goes to everyone else from the Behavioral Evolution Lab, and to Carla, Teal, Finn, and Sabi, for all being so incredibly welcoming during my visit over the summer. You made my time in Konstanz so enjoyable and memorable, and it was very hard for me to say goodbye at the end of my stay. Likewise, thank you to everyone from the Data Visualization Group and the CogSciPrag Lab in Tübingen, especially for participating in so many discussions about birds and fish despite severely limited relevance to your own work.

Finally, I would like to thank Devid, Aarón, Papa, and Mama for supporting me in all my half-baked ideas, run-throughs, and rants throughout this project, because you cared.

Chapter 1

General Introduction

1.1 Representing Communication

Many natural communication systems consist not simply of isolated signals, but sequences of signals whose meaning depends on temporal organization. Understanding how these signals are organized is therefore a prerequisite for understanding the meaning behind communicative behaviors.

Communication is commonly represented as sequences of discrete events. This discretization is useful for modeling, but it is also a transformation that communicators must perform at some stage during processing. The onset and offset of signals are discrete events in time, but the behavior of signals themselves may be physically continuous (e.g., vocal pitch, limb movements). However, successful communication requires information to be transferred between individuals and subsequently interpreted by receivers. Therefore, the meaningful units of a natural communication system must ultimately be categorical.

This is because senders cannot perfectly repeat a signal due to motor noise, and even if they could, receivers could not perceive two instances of the same signal in exactly the same way due to neural and environmental noise. If every difference was treated as semantically significant, each signal would be unprecedented, preventing generalization beyond that single instance. Instead, signals must be grouped into categories of signals that are treated as sufficiently similar. Variation within these categories can then be treated as meaning-irrelevant, while differences between categories can be considered informative.

For researchers attempting to model and decipher communication systems, the task therefore consists of identifying the functional categories of signals, and the structural rules that link signals together. The former task requires accurately identifying the appropriate granularity for discretization. Advancements in machine learning methods have offered new data-driven op-

tions to discover the meaningful units of communicative signals. These types of methods have been implemented on a large variety of animal communication datasets [1, 2], including cichlid postural communication [3] and Bengalese finch song [4].

Once communicative signals are parsed into categorical units, research can focus on the structural rules linking signals together to form informative communicative sequences. This thesis focuses on this aspect of animal communication by estimating historical dependencies in pre-discretized communicative sequences.

1.2 Detecting Structure in Communication

Transformer models have become state-of-the-art approaches for detecting and predicting complex temporal structure in sequences, particularly for written and spoken word [5]. Transformer-based Large Language Models (LLMs) demonstrate impressive predictive accuracy across a wide variety of natural language tasks [6], all of which involve predicting sequences of discrete tokens.

However, animal communication datasets are often relatively small in size due to limitations in cost, access, and ethical considerations, making highly parameterized and data-hungry models like these difficult to train reliably [7, 8].

A more fundamental issue for scientific applications is that prediction is insufficient unless it accompanies an explanation. Science is built on hypothesis testing which requires evaluating potential mechanisms behind observations, not just accurate prediction of observations alone. Therefore, for any model of animal communication, we should be able to access not just a predicted behavioral sequence, but also the calculations justifying the output. Transformer models usually cannot provide this in fully explicit terms. Many advancements have been made in the explainability of model outputs, but these local insights still fall short of explaining the underlying global behavior of the model [9]. This interpretability is necessary for understanding the causal relationships used by models for predictions [10], and in the context of animal communication, for truly identifying the structural rules of the system.

For this reason, this thesis focuses on another class of probabilistic models based on generalizations of Markov chains. Rather than maximizing predictive performance, these models are designed to estimate historical dependencies and give a completely explicit read-out of the detected patterns through transition matrices. Markovian approaches have been especially popular in birdsong research, perhaps because the presence of complex structure is so obvious. Basic Markov chains fail to sufficiently explain the syntax of birdsong [11,

12], but partially observable Markov models with adaptation (POMMA) [11], branching chain networks [12, 13], probabilistic finite automata [14], and neural networks [15] have all been used to model context dependencies and sequential organization of birdsong.

1.3 The Model Species

This thesis will use probabilistic Markovian models to analyze the structure of communicative sequences from two sets of model species representing two different communicative modalities.

The first dataset consists of annotated postural behavior recorded from six species of Lake Tanganyika Lamprologine cichlids (family *Cichlidae*). This group of fish exhibits notably diverse social behaviors between species, despite occupying similar ecological niches in the lake. Nevertheless, most cichlid species use dynamic body movements to facilitate social interactions, including aggressive displays [16] and peacekeeping signals [17].

This dataset was collected by Ref. [3], who discretized the postural behavior of male cichlids during dyadic territory disputes using unsupervised machine learning methods. This analysis produced corpora of paired strings tracking the postural state by frame of each fish in a dyad throughout aggressive interactions.

The second dataset consists of annotated birdsong recorded from six adult male Bengalese finches (*Lonchura striata domestica*). Bengalese finches are a domesticated subspecies of estrildid finch, and are popular among birdsong researchers because of their diverse vocalizations. This is thought to have been influenced by domestication, which reduced predation pressure which would otherwise favor quieter birds who are less likely to be heard by predators [18].

Singing is widely considered to be a communicative behavior in birds [19, 20], even in undirected instances with no immediate recipient. The songs of Bengalese finches are composed of a fixed set of syllables arranged into variable sequences and following syntactic rules [21]. The specific rules and syllables differ between individuals, but all songs demonstrate strong structural organization. The dataset used here was formatted as strings representing the sequential order of syllables over the course of songs.

In both sets of model species, it is unlikely that individual states or syllables themselves are meaningful, but instead are the blocks of which meaningful behaviors are composed. Identifying these full behaviors is necessary for parsing meaning behind behavioral sequences, but this first requires understanding the sequential organization of the building blocks.

1.4 Playback Studies

A powerful approach to probing the structure and meaning behind communication is the playback experiment, in which behaviors by real animals are played back to other individuals. Using this technique, the meaning behind certain behaviors can be inferred by how animals react to them.

There has been a strong precedent of playback experiments for bird communication research, for both calls (e.g., [22–24]) and songs (e.g., [25, 26]). Many of these studies have focused on exploring innate and learned grammatical and syntactical structures. Playback studies are less common for underwater species, but successful examples come from female swordtail fish [27], and sticklebacks [28].

Playbacks have historically been limited by the stimuli they can present, which are often recordings of previously observed behavior. It is difficult to acquire a recording set large enough for experimental designs while still achieving a high quality. For example, visual stimuli must be fully visible and in focus, and auditory stimuli should have comparable background noises to discount effects of acoustic environmental information.

To address this issue, studies in recent years have made use of animated stimuli [29], which allow for much more standardized presentations of stimuli [30]. Even though animations offer more controlled presentations of behavior, the behavioral sequences themselves often still fall short. Animated sequences are usually either extracted from recordings of real behavior, or manually coded by researchers. In the former case, it is still difficult to manipulate these behavioral sequences without risking lower playback quality from splicing or cutting. In the latter case, it is unclear whether stimuli are ecologically valid and represent how real animals would actually behave. This creates a gap in playback experimental methodology for stimuli that are standardized, easily manipulable, but still show naturalistic sequences of behavior.

1.5 Thesis Outline

The remainder of this thesis presents an analysis pipeline for estimating and replicating historical dependencies in observed behavioral sequences. The resulting models are intended for developing hypotheses which can be experimentally tested, and generating stimuli for playback experiments that are both standardized and easily manipulable.

Chapter 2 introduces a family of probabilistic Markovian models derived from generalizations of Markov chains. These models predict the continuation of a sequence using selected information in the sequence’s history. The strategy used to identify this historical information is completely customizable,

and two options motivated by biological theory are proposed: Lagged History and Collapsed History. Their performance is compared in a simulation study using synthetic corpora with known generative structure, and characteristics designed to mimic those of the empirical datasets analyzed later in this thesis.

Chapter 3 applies these probabilistic Markovian models to corpora of natural communicative sequences from cichlid postural behavior and Bengalese finch song. It will demonstrate that Bengalese finch song can be approximated as a dual-process system consisting of a highly deterministic "prefix" section followed by a second, more stationary process. These prefixes violate time-homogeneity assumptions necessary for Markovian modeling. To resolve this, a method based on 'tries' from computer science is introduced to automatically identify and extract prefixes from songs.

Based on the results of the simulation study, the better-performing modeling strategy is used for each empirical corpus, resulting in a best-fitting model of the historical dependencies detected in each corpus. The calculations used by these models are explicitly accessible through transition matrices, and can be visualized as heatmaps. These plots will be generated for each corpus, and used to explore potential communicative structures.

Chapter 4 describes a pipeline for generating novel sequences from the fitted models. The chapter includes example generated corpora, and compares the structures of these corpora with the original empirical datasets. It will then discuss possible applications for hypothesis testing and experimental playback studies in the context of cichlid and finch communication.

Finally, Chapter 5 provides a general discussion on the results across the two sets of model species. It considers similarities and differences between models of these communication systems, evaluates the strengths and limitations of the proposed methods and their applications, and concludes with future directions in this line of research.

Chapter 2

Probabilistic Markovian Models

2.1 First-order and k -order Markov Chains

A Markov chain is a stochastic process in which the probability of the next state in a discrete sequence $S = \{s_1, s_2, \dots\}$ at time t depends only on the value of the immediately preceding state at time $(t - 1)$ [31]:

$$P(s_t | s_{t-1}, s_{t-2}, \dots) = P(s_t | s_{t-1}) \quad (2.1)$$

The changes of state in a sequence are referred to as transitions, and the probabilities of observing each unique $s_{t-1} \rightarrow s_t$ change are referred to as transition probabilities. These transition probabilities can be compiled in a square transition matrix, where each entry (i, j) corresponds to the probability of state i transitioning to state j .

Markov chains are often described as "memoryless", because the predicted value of the next state in a sequence is based solely on the value of the directly preceding state, and nothing else in the sequence's longer history. However, this is insufficient to fully represent many processes where the value of the next state depends on more than one preceding state.

To address this, this process can be generalized to a family of k -order Markov chains, where the probability of the next state instead depends on a historical tuple $\mathbf{h}$, consisting of k previous states extracted at certain key points in the sequence's history:

$$P(s_t | s_{t-1}, s_{t-2}, \dots) = P(s_t | \mathbf{h}), \mathbf{h} = \{h_1, \dots, h_k\} \in S \quad (2.2)$$

2.2 Estimating k using Model Comparisons

Formula 2.2 can be used not only to generate sequences with higher-order dependencies, but also to estimate the dependencies of observed sequences.

For k -order Markov chains, transition matrices are no longer necessarily square, but instead of size $(n^k \times n)$, with each row i corresponding to a unique historical tuple of length k , and the entry (i, j) corresponding to the probability of observing a transition from tuple i to state j . Consequently, the rows of a transition matrix with order k are expansions of the rows of a transition matrix with order $(k - 1)$. In this way, transition matrices of increasing k -order can be thought of as a set of nested models. To approximate higher-order historical dependencies in a sequence, one can therefore evaluate model fit for different orders of k .

Model fit is evaluated in the following analyses using an 80%/20% train-test split. For each value of k , a transition matrix is compiled for all observed k -order transitions in the training set. An additive smoother is added to each entry of this matrix, which allows the model to account for potentially unseen transitions in the training set which are nonetheless still possible. The entries of this matrix are then normalized to produce a Markovian model of smoothed k -order transition probabilities.

Next, a raw k -order count matrix of transitions is compiled from the testing set. The fit of the Markovian model is evaluated on this count matrix using the cross-entropy log-likelihood:

$$\mathcal{L} = \frac{\sum_h \sum_s C_{\text{test}}(s | \mathbf{h}) \log P_{\text{train}}(s | \mathbf{h})}{\sum_h \sum_s C_{\text{test}}(s | \mathbf{h})} \quad (2.3)$$

where $\mathbf{h}$ is the historical tuple of length k , s is the next character, $C_{\text{test}}(s | \mathbf{h})$ is the raw count of transition $\mathbf{h} \rightarrow s$ from the testing set, and $P_{\text{train}}(s | \mathbf{h})$ is the smoothed and normalized probability of the transition $\mathbf{h} \rightarrow s$ from the Markovian model.

This likelihood function is used to evaluate how well the Markovian model's probability distribution extends to the unseen testing set, which is assumed to be generated from the same latent process as the training set. A model which approximates this latent process well should generalize the best to the testing set, because it will feature the important distinctions in the historical tuples that are also present in the testing set, without overfitting to incidental patterns from the training set.

Markovian modeling relies on two key assumptions about the underlying generative structure of sequences: irreducibility and time-homogeneity.

A sequence is irreducible if it is possible to reach any state from any other state in a finite number of steps. This is important for sequence generation because the model's transition matrices only specify the immediate next state in a sequence. Therefore, by iteratively sampling from a transition matrix

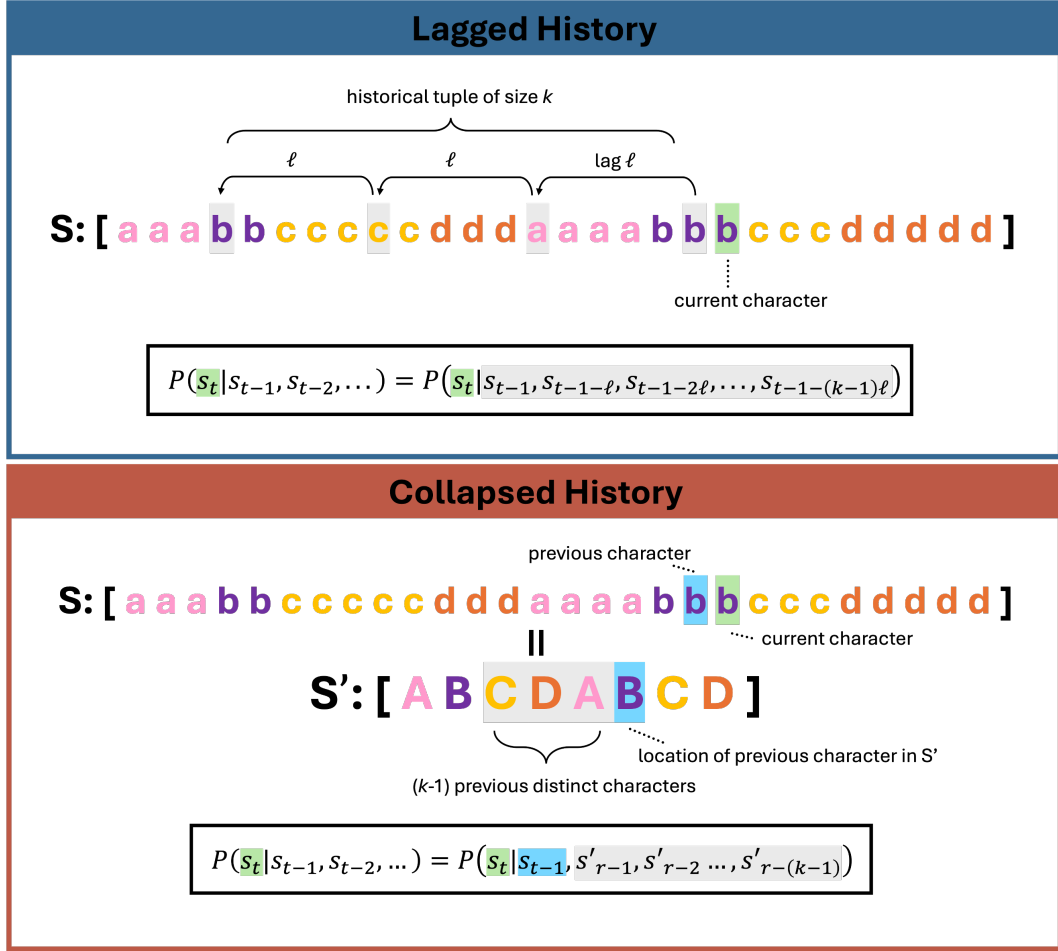


Figure 2.1: A diagram of Lagged and Collapsed History strategies for Markovian modeling.

compiled from a system which violates irreducibility, it is possible to follow paths to states that are impossible for the true system to reach: each transition step may be possible, but the destination state from a starting state is not.

A sequence is time-homogeneous if transition probabilities remain stationary across all positions in a sequence. This assumption is important because Markovian models compile a single transition matrix across all positions in a sequence, meaning they cannot account for position effects.

A full specification of Formula 2.2 requires not only a value for k , but also a strategy to choose the previous k states in $\mathbf{h}$. This is completely customizable, but this thesis proposes two options motivated by empirical applications to animal communication, using sequences represented as strings of characters. A diagram of these two strategies is presented in Figure 2.1.

The first strategy will be referred to as Lagged History. For each current character s_t in a string $S = \{s_1, s_2, \dots\}$, the historical tuple of k previous characters is composed of the directly previous character and the $(k - 1)$ previous characters spaced apart by a fixed lag (ℓ). ℓ acts as an additional parameter that can be fine-tuned during model fitting. This method allows transition matrices to capture long-range dependencies that may be present in the string, without conditioning on every intermediate character. Long range sequential dependencies characterize human language, which utilizes a temporal structure where meaning can be altered by words spaced very far apart [32]. Similar dependencies have been found in birdsong [32, 33].

The Lagged History strategy compiles a transition matrix for a string as:

$$P(s_t | s_{t-1}, s_{t-2}, \dots) = P(s_t | s_{t-1}, s_{t-1-\ell}, s_{t-1-2\ell}, \dots, s_{t-1-(k-1)\ell}) \quad (2.4)$$

The second strategy will be referred to as Collapsed History. For each current character in a string, the historical tuple of k previous characters is composed of the directly previous character and the $(k - 1)$ most recent *distinct* characters.

For a string $S = \{s_1, s_2, \dots\}$, let $S' = (s'_1, s'_2, \dots, s'_r)$ be the sequence of characters from that string without repeats. For example, if $S = \{a, a, a, b, b, c, c, b, b, b, a, c\}$, then $S' = \{a, b, c, b, a, c\}$. The length of S' (r) is consequently shorter than that of S (t), so for each current character s_t in a string, its position is mapped to the place it would lie in S' as s'_r .

This strategy is more consistent with the idea that behavior composed of smaller functional units that are combined to form more complex and meaningful behavioral motifs [34–36]. These motifs are thought to depend more on the sequence of preceding behaviors rather than the durations spent performing each one.

The Collapsed History strategy compiles a transition matrix for a string as:

$$P(s_t | s_{t-1}, s_{t-2}, \dots) = P(s_t | s_{t-1}, s'_{r-1}, s'_{r-2}, \dots, s'_{r-(k-1)}) \quad (2.5)$$

Probabilistic Markovian models provide several benefits for a preliminary exploration of sequences with unknown structure. First, they can integrate longer historical dependencies than basic Markov chains can account for. Second, their transition matrices provide a completely explicit read-out of the detected historical dependencies. Finally, model fit can be evaluated not only between k -orders, but also strategies for identifying the historical tuple. Better performing strategies can indicate the true nature of historical dependencies in sequences.

2.3 Simulation Study

Nevertheless, there are inherent limitations in the types of structure these models can detect. Therefore, it is important to assess under what conditions they can reliably recover the true generative structure of a string. In particular, the likelihood function used to evaluate model fit relies on adequate discriminability between rows of collected transition matrices in order to differentiate between transition matrices of different k -orders. In other words, changing the identity of characters at each position in the historical tuple should lead to different next-character probability distributions. Otherwise, child historical tuples of order k might collapse into a parent tuple of order $(k - 1)$, meaning the two matrices provide identical information.

This may be an issue when the generative system behind a string has high transition entropy, meaning that transition probability given a certain historical tuple is distributed across many next characters rather than a single character. At high entropy levels, the transition probability distribution may become so uniform that differentiating between certain characters in a string’s history is completely redundant for prediction, despite them actually having an effect.

Besides high entropy, other dataset qualities may contribute to low discriminability between k -orders. A dataset consisting of very few strings, or very short strings, may not represent enough transitions from a high-order generative structure, even if that true system has low entropy. This effect may be heightened if the set of unique string characters is very large, because there is a lower probability of observing each combination of them.

The following simulation study evaluated the performance of Markovian models using Lagged History and Collapsed History strategies across these dataset conditions, using synthetic corpora of strings with known generative structures.

2.3.1 Methods and Analysis

First, a cache of transition matrices was created to represent different ground-truth generative rules for corpora of strings. Transition matrix structure varied systematically along three parameters (bolded for reference): true generative order ($\mathbf{K} \in \{0, \dots, 5\}$), character set size ($\mathbf{nChar} \in \{2, 4, 6, \dots, 10\}$), and concentration parameter ($\alpha \in \{0.1, 0.5, 1, 10\}$). The size of each transition matrix was $(\mathbf{nChar}^{\mathbf{K}} \times \mathbf{nChar})$. Each entry corresponding to a self-transition (i.e., the most recent previous character is the same as the current character: $\{a\} \rightarrow a$; $\{a, b\} \rightarrow b$; $\{a, b, a\} \rightarrow a$) was populated with a probability of 0.95¹. This

¹Except for the $K = 0$ transition matrices, which lacked historical tuples.

was done to mimic the long repeated-character runs observed in the upcoming empirical datasets.

The remaining probability mass in each row was distributed across the $(\text{nChar} - 1)$ non-self characters by sampling from a Dirichlet distribution with concentration parameter α , to simulate different levels of transition entropy. At high values of α , the probability mass is distributed more uniformly across all characters. At low values, transition probability becomes more peaked towards fewer characters, approaching a deterministic distribution with only one possible next character as $\alpha \rightarrow 0$.

Since Dirichlet sampling is stochastic, α does not map to a consistent transition entropy. Therefore, entropy was measured explicitly for each generated transition matrix. Entropy was calculated only over the probabilities of non-self character transitions and re-normalized accordingly. This was done to ensure that entropy was not trivially pulled to near-zero due to the high self-transition probabilities added to the transition matrices.

Given a row of a transition matrix corresponding to the historical tuple $\mathbf{h}$, where the most recent character is h_{-1} , the non-self transition probability ($\tilde{P}$) to a next character c is given as:

$$\tilde{P}(c|\mathbf{h}) = \frac{P(c|\mathbf{h})}{1 - P(h_{-1}|\mathbf{h})}, c \neq h_{-1} \quad (2.6)$$

The Shannon entropy ($\mathcal{H}$) of this new distribution is calculated as the negative non-self transition probability multiplied by its logarithm, summed over all possible next c 's:

$$\mathcal{H}_{\mathbf{h}} = - \sum_{c \neq h_{-1}} \tilde{P}(c|\mathbf{h}) \log \tilde{P}(c|\mathbf{h}) \quad (2.7)$$

To ensure that entropy can be compared across systems with different character set sizes, entropy is normalized over the maximum possible entropy from $(\text{nChar} - 1)$ characters (since self-transitions have been removed), i.e., $\log(\text{nChar} - 1)$:

$$\mathcal{H}_{\mathbf{h}}^{\text{norm}} = \frac{\mathcal{H}_{\mathbf{h}}}{\log(\text{nChar} - 1)} \quad (2.8)$$

These calculations produce a normalized entropy value for one row of a transition matrix. For each matrix, this calculation was computed and averaged over all rows to create a single summary statistic $\tilde{\mathcal{H}}$ describing the overall entropy behavior of the matrix.

Next, corpora of strings were generated from each cached transition matrix while varying two additional parameters (bolded for reference): number of strings generated (**corpusSize** $\in \{10, 100, 500, 1000\}$), and length of each string (**length** $\in \{10, 100, 500, 1000\}$).

Finally, for each generated corpus, models of orders $\mathbf{k} \in \{0, \dots, 5\}$ were run using Lagged History and Collapsed History strategies. For Lagged History models, the lag parameter was additionally varied: ($\ell \in \{1, 10, 50\}$). An additive smoother of 0.1 was added to every cell of training transition matrices as a reasonable value to account for unseen testing transitions across dataset conditions, particularly small corpus sizes.

This pipeline resulted in a total of 46,080 unique combinations of structural parameters $\{\mathbf{k}, \mathbf{nChar}, \alpha\}$, generative parameters $\{\mathbf{corpusSize}, \mathbf{length}\}$, modeling strategy (Lagged or Collapsed), and modeling parameters $\{\mathbf{k}, \ell\}$. This process was repeated over 25 simulations for a total of 1,152,000 iterations.

For each generated corpus, the predicted order $\hat{\mathbf{k}}$ was identified for each modeling strategy as the order achieving the highest log-likelihood. This $\hat{\mathbf{k}}$ was cross-checked with K and coded as either an overestimate, underestimate, or correct detection of K .

2.3.2 Results

The overall correct detection rate across all parameters was 48.67% for models using the Lagged History strategy, and 61.57% for those using the Collapsed History strategy. Figure 2.2 presents a comprehensive comparison between modeling strategies over all values of the structural and generative parameters.

Lagged History detection accuracy was better than Collapsed History when $K \leq 1$, but performance dropped dramatically as K increased, sinking below chance when $K = 5$. In contrast, Collapsed History detection remained more stable across values of K .

Collapsed History accuracy also started lower than Lagged History for binary strings with only two possible characters, but steadily increased as the number of unique characters in the corpus increased, while Lagged History performance remained around 50% regardless of character set size.

Both modeling strategies resulted in stable detection for most values of corpus size and string length, with Collapsed History consistently outperforming Lagged History. However, both strategies demonstrated drops in performance when corpus size was very low, and the Lagged History strategy showed worse performance when string lengths were very short.

Finally, the two modeling strategies showed different detection accuracy patterns when system entropy was varied. Lagged History performance peaked at intermediate entropy levels between 0.4 – 0.5, but decreased as entropy became more extreme on either end. In contrast, Collapsed History performance was poorer than Lagged History at very low entropy values, but showed a better and more stable performance than Lagged History at intermediate to high levels of entropy.

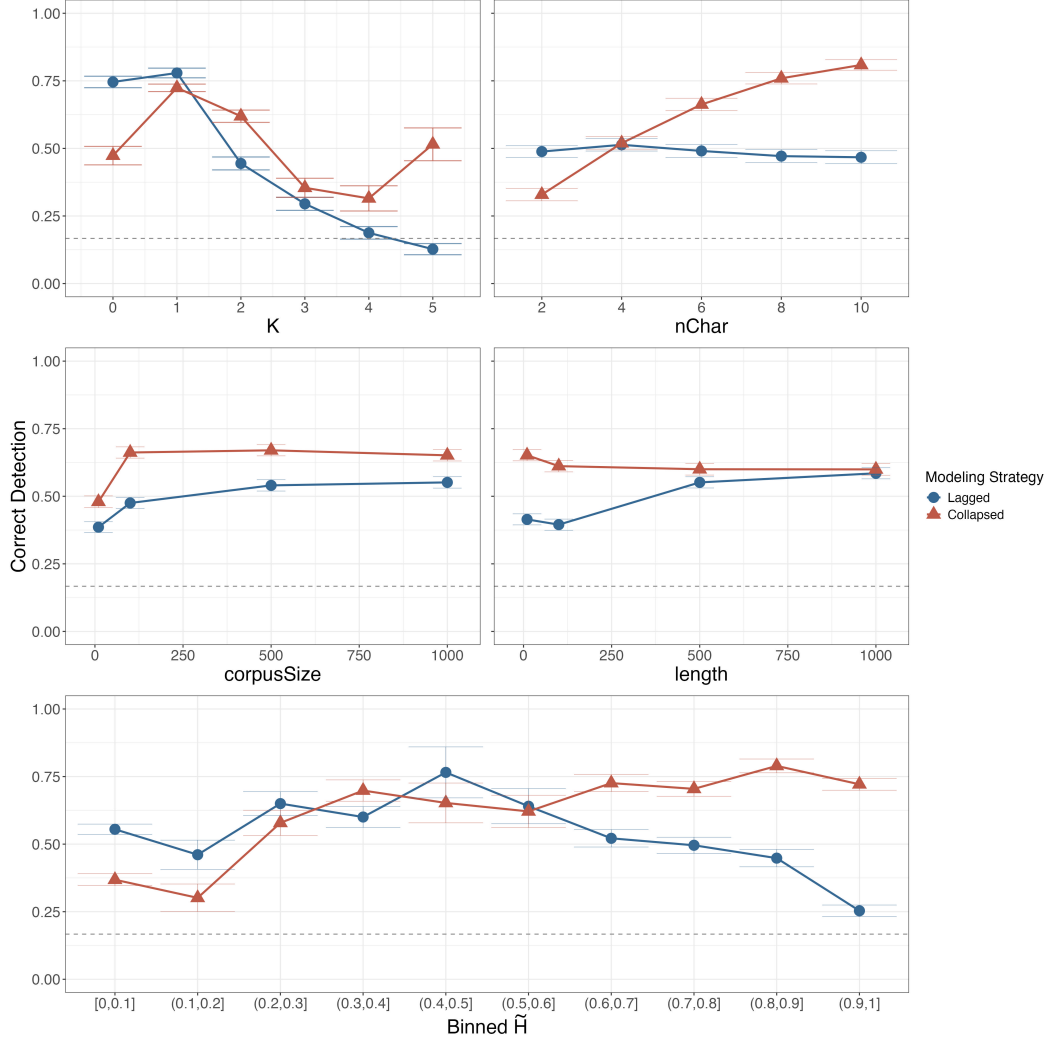


Figure 2.2: Line plots comparing modeling strategy performance. Each facet depicts the correct detection rate (y-axis) of each modeling strategy (line color) at each value (x-axis) of a structural or generative parameter. The plots show from left-to-right and top-to-bottom: true generative order (K), number of unique characters ($nChar$), number of strings generated ($corpusSize$), length of each string ($length$), and binned transition entropy ($\tilde{H}$). At-chance detection performance ($\frac{1}{6}$) is indicated by the gray horizontal dashed line.

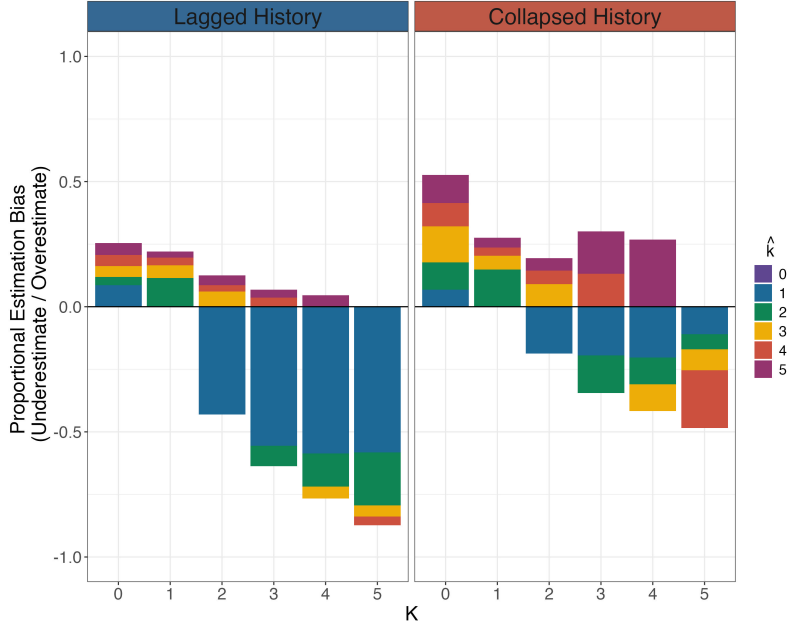


Figure 2.3: Bar plots comparing estimation bias of each modeling strategy. Each facet depicts the proportion of over- and underestimated $\hat{k}$ (y-axis) at each true generative order K (x-axis) for one modeling strategy (Lagged History, left; Collapsed History, right) across all simulations. Overestimates are plotted on the positive y-axis, and underestimates are plotted on the negative y-axis. Each bar is divided by color to indicate the proportion of each incorrectly detected $\hat{k}$.

Estimation biases of each modeling strategy were calculated to further investigate model behavior during incorrect detections. Figure 2.3 presents the proportion of over- and underestimates of K , by $\hat{k}$ from each modeling strategy.

The two modeling strategies showed very different behaviors with incorrect detections. Lagged History was more likely to underestimate K , especially for larger orders. However, the vast majority of its incorrect detections were $\hat{k} = 1$, regardless of the true K order. Overall, if models using this strategy did not correctly detect K , they reverted to a $\hat{k} = 1$ model. For larger values of K , models chose a $\hat{k} = 1$ order more often than the correct K -order model.

Models using the Collapsed History strategy tended to overestimate K slightly more often than underestimate it. In contrast to Lagged History, the distribution of $\hat{k}$ values was more varied for different K -orders. Notably, the majority of incorrect detections when $K \geq 0$ were $\hat{k}$ values that were only one order higher or lower than the K . This contrasts strongly with Lagged History, where the $\hat{k}$ tended to equal 1 regardless of K .

2.4 Interim Discussion

Markovian models may appear outdated or unsophisticated given the current state of machine learning with neural networks and transformer models, but they were not designed to compete with these new methods. Rather, these models serve a very specific purpose of identifying and replicating historical dependencies in sequences in a completely explicit manner. The transition matrices compiled by these models provide an interpretable read-out of the observed frequencies of historical patterns in sequences, with customizable parameters to change the strategy for identifying key historical information. Model fit can then be evaluated between different strategies and parameters to estimate the nature of the underlying generative process in sequences. A more sophisticated transformer model may produce better sequence predictions, but if it cannot give a clear read-out of which historical information in a string mattered to the prediction, it is not useful for generating biological hypotheses.

The simulation study gives important insights into the conditions where Markovian models are reliable and when they might fail. Overall, models using the Lagged History strategy were worse at detecting K than those using the Collapsed History strategy, particularly at high K orders. The performance of Collapsed History was also more stable across dataset conditions. This is particularly notable because the Lagged History models made use of an additional parameter, ℓ , to finetune their search strategies even further, and still failed to perform at the same level as Collapsed History models.

The estimation bias comparisons in Figure 2.3 depict the behavior of each modeling strategy during inaccurate k detection. Reassuringly, neither strategy resulted in models indicating complete lack of temporal structure ($\hat{k}=0$) when there actually was a higher-order generative structure. However, these results also illustrate that Lagged History models had a strong bias towards a first-order $\hat{k}$ estimate, regardless of the complexity of the underlying generative structure. In contrast, incorrectly-detected $\hat{k}$ values in Collapsed History models remained close to K as generative order increased. Exact recovery of K is arguably not very important in this application, since the underlying sequence structures in the upcoming empirical datasets are likely far more complex than either modeling strategy allows for. Rather, it is more important to have a method that flexibly adapts to the dataset, which the Collapsed History strategy seems to do better than Lagged History.

For these reasons, I chose to only use the Collapsed History modeling strategy for the empirical applications in the next chapter. These results do not necessarily indicate that the Lagged History strategy performs poorly in general, but rather that it performs poorly for datasets with sequence structures similar to those in the upcoming empirical corpora.

Chapter 3

Applications to Animal Communication

3.1 Cichlid Postural Communication

3.1.1 Materials and Preprocessing

The cichlid postural dataset was collected by Ref. [3] and represents six species: *Neolamprologus pulcher*, *Neolamprologus multifasciatus*, *Neolamprologus brevis*, *Lamprologus ocellatus*, *Lamprologus ornatipinnis*, and *Neolamprologus meeli*.

Individuals were collected from their habitats in Lake Tanganyika and subjected to contest experiments in experimental arenas. In each trial, two size-matched males of the same species were placed on opposite sides of a 30×20 cm experimental arena, separated by an opaque barrier within a larger concrete tank. The fish were allowed to acclimate there for at least 20 hours to establish their territories on their side of the arena. After acclimation, five consecutive contest trials, each separated by 30 minutes, were conducted by removing the opaque barrier and inciting the fish into territory disputes for control of the entire tank. During these contest trials, the experimental arena was left open so that fish always had an escape option to end the interaction at any time.

Video data from each contest trial was collected at 1080p resolution and 120 frames per second. Social interactions were defined as periods of time in each contest trial during which the head of one fish was within a specified distance away from the other fish. Of course, fish may have been communicating on opposite sides of the tank, but this is difficult to differentiate from times where they were resting. When two males are in proximity, especially in this aggressive context, they are much more likely to be truly interacting. The

open escape option further ensured that the fish were meaningfully interacting when they stayed in the experimental arena.

Relevant postural features were extracted using a 60-frame window length (0.5 seconds) and a 10-frame hop length, resulting in one multivariate time series per fish per interaction. These time series were discretized using a Gaussian Mixture Model Hidden Markov Model (GMMHMM) applied across all cichlid species, creating a shared set of categorical postural states. The optimal number of states was determined by comparing model fit using the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and log-likelihood, and validated with a random forest model. This resulted in a set of ten postural states $\{0, 1, 2, 3, 4, 5, 6, 7, 9, 10\}$ ¹. Each multivariate time series was finally converted into a sequence of [state, duration] vectors.

For this thesis, postural states were recoded as corresponding characters $\{a, b, c, d, e, f, g, h, j, k\}$. Each vector sequence was converted into a string, with each character representing one time point, and characters repeating for the duration of time spent in each postural state. These strings were separated by species into six corpora and analyzed independently.

3.1.2 Descriptive Statistics and String Behavior

Basic descriptive statistics are presented in Table 3.1 for full strings, and the collapsed versions of strings where character self-transitions (i.e., repeated states) are ignored. Although only the full strings were used for Markovian modeling, collapsed string length statistics have been included to describe cichlid behavior in terms of unique postural states displayed over time.

Character distributions over string position were visualized as ribbon plots to establish whether these corpora meet the modeling assumptions of irreducibility and time-homogeneity (See Figure 3.1). As demonstrated in Table 3.1, string length was quite variable for all species, with a large difference between the maximum length and upper quantiles (Q3). This means only a few strings had very long lengths per corpus, which could bias state distribution estimates at very late positions. To avoid this, ribbon plots were cropped at the string length Q3.

¹Two additional states (8 and 11) were originally identified by the GMMHMM but merged into adjacent states based on the random forest model results.

Species	N	Full Strings			Collapsed Strings		
		Q1	Q3	Max	Q1	Q3	Max
<i>N. brevis</i>	180	700	1670	8270	4	18	199
<i>N. meeli</i>	78	475	1930	11710	3	15	561
<i>N. multifasciatus</i>	778	640	1440	6420	9	30	226
<i>L. ocellatus</i>	164	400	1470	12860	9	34	253
<i>L. ornatipinnis</i>	478	450	1885	21540	5	26	426
<i>N. pulcher</i>	894	723	1828	51980	5	23	321

Table 3.1: *Basic descriptive statistics for each species' corpus.* From left to right: species, corpus size (N), lower quantile of string length (Q1), upper quantile of string length (Q3), and maximum string length (Max), separated by full strings (left) and repeat-collapsed strings (right). Quantiles have been rounded to the nearest whole number.

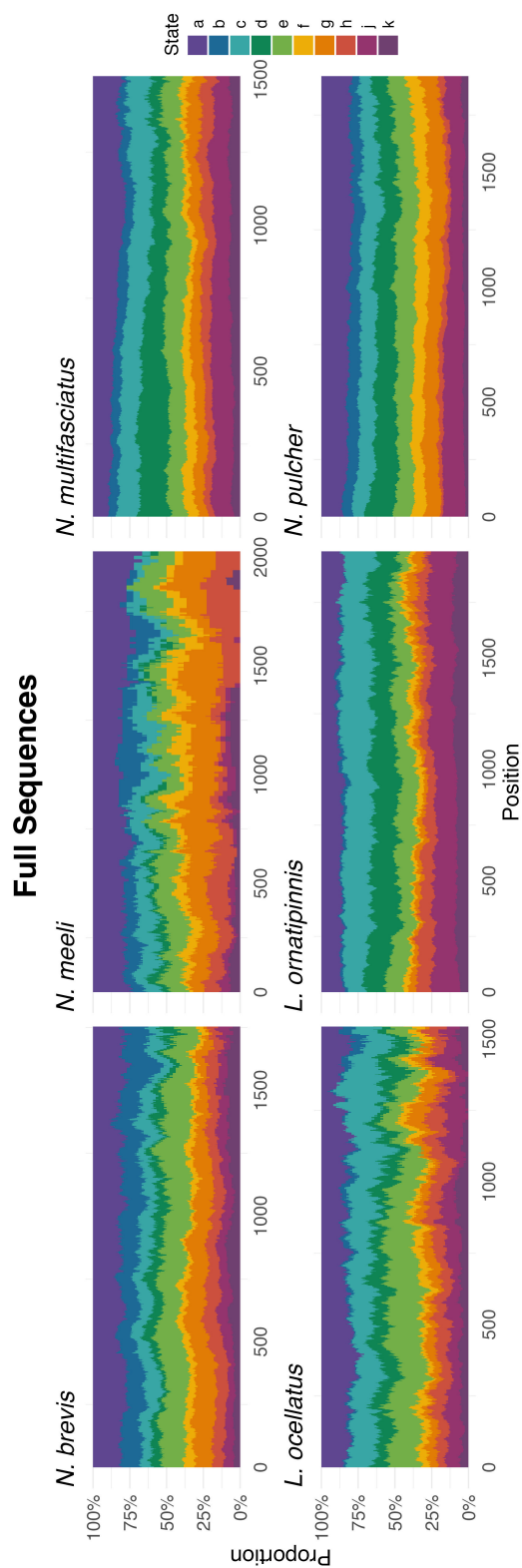


Figure 3.1: *Cichlid Postural State Distributions over Time.* Each facet depicts the proportion (y-axis) of each state (color) at each position (x-axis) of full strings in each species's corpus. Plots are cropped to the string length Q3.

Figure 3.1 demonstrates that both modeling assumptions were reasonably met for all corpora. All corpora showed relatively stable state distributions over time, meaning the strings in each dataset did not show systematic preferences for certain states at specific positions, changes in the overall state probability distribution by position, or limitations on the states reachable from other states. The strongest deviations in stability appeared in the corpus of *N. meeli*, whose state distribution shifted towards the end of the ribbon plot. However, this was by far the smallest corpus of all species (see Table 3.1), meaning that this distribution could have been more affected by noise from individual strings. Overall, it appeared valid to use Markovian modeling for all species without any further processing.

3.1.3 Markovian Modeling Results

Following the conclusions presented in Chapter 2, 100 iterations of Markovian modeling using the Collapsed History strategy were run for each species' corpus, for orders $k \in \{0, 1, 2, 3, 4, 5\}$ and additive smoother values $s \in \{0.001, 0.01, 0.1\}$ (parameters bolded for reference). Each iteration used a unique random train/test split of the corpus. For each iteration, the k value corresponding to the highest log-likelihood was extracted along with the corresponding s . Within this subset, the best-fitting k -order (Best- k) was defined as the mode k , and the best-fitting smoother value (Best- s) was identified as the mode s associated with the Best- k . Figure 3.2 presents the average performance of each k -value for each species.

Table 3.2 presents the Best- k , the proportion of iterations indicating the Best- k , the corresponding average log-likelihood over all iterations for the Best- k , and the corresponding Best- s .

Species	Best- k	Prop.	AvgLL	Best- s
<i>N. brevis</i>	1	1.00	-0.07336135	0.001
<i>N. meeli</i>	1	0.91	-0.06655062	0.1
<i>N. multifasciatus</i>	2	0.75	-0.11711392	0.1
<i>L. ocellatus</i>	1	0.70	-0.12318247	0.001
<i>L. ornatipinnis</i>	1	0.97	-0.07993485	0.001
<i>N. pulcher</i>	1	1.00	-0.06353527	0.1

Table 3.2: *Best- k by species.* From left to right: species, best-fitting k -order compiled from 100 iterations of Collapsed History modeling (Best- k), proportion of iterations of best performance by that k value (Prop.), corresponding average log-likelihood of the best-fitting k over all iterations (AvgLL), and mode smoother associated with the Best- k (Best- s).

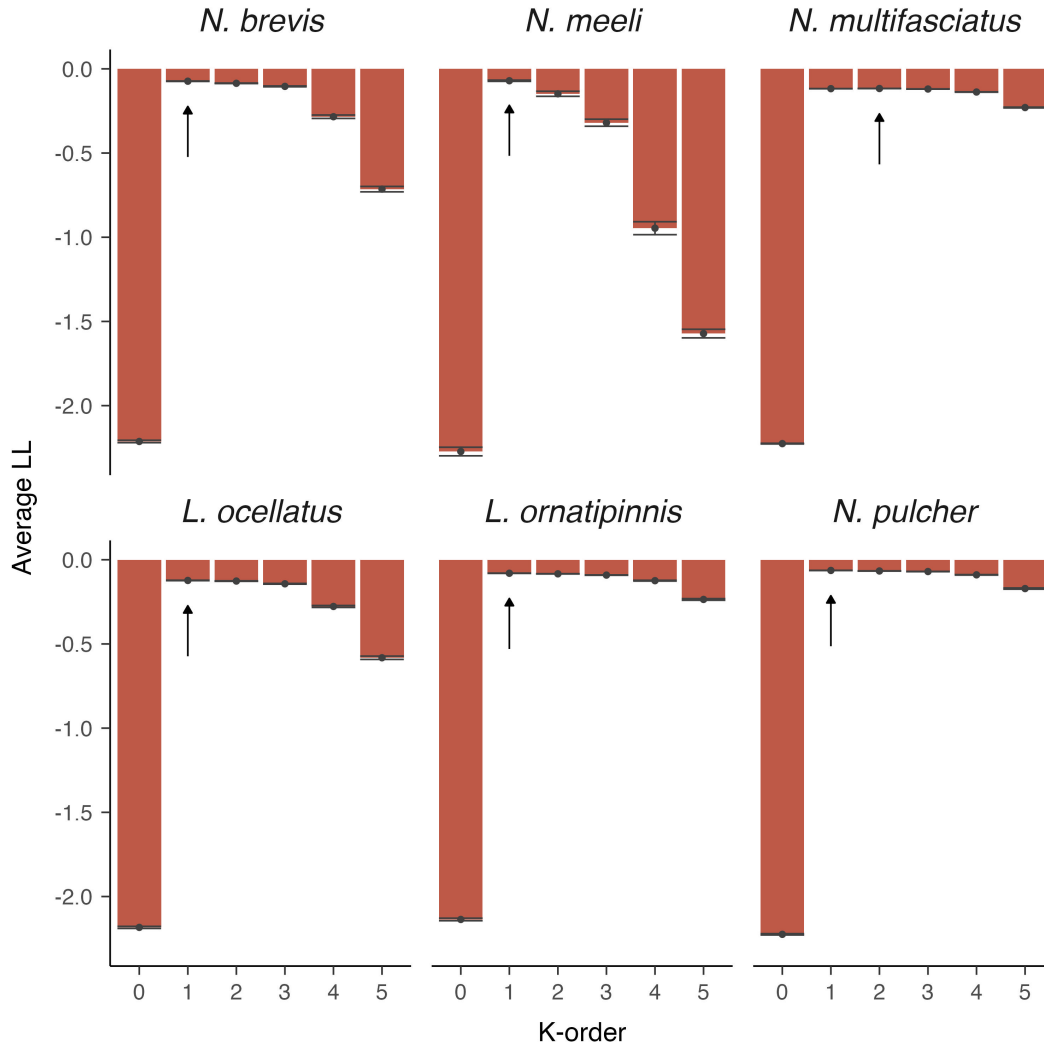


Figure 3.2: *Cichlid Markovian modeling results.* Each facet shows the average log-likelihood (y-axis) for each k -order (x-axis), compiled from 100 iterations of Collapsed History modeling, for each cichlid species. The black arrow in each facet points to the best-fitting k -order, calculated as the most-occurring k value corresponding to the highest AvgLL over each iteration. Error bars depict the 95% confidence interval for the AvgLL of each k -order.

All cichlid corpora except that of *N. multifasciatus* were best described by first-order Collapsed History Markovian models, meaning that next states in postural sequences were best predicted using only a single previous unique postural state. The corpus of *N. multifasciatus* was best described by a second-order Collapsed History Markovian model, meaning that the previous two unique states predicted the next state in postural sequences.

The Best- s varied by species. The models for *N. brevis*, *L. ocellatus*, and *L. ornatipinnis* most frequently selected an additive smoother value of 0.001, but 0.1 for *N. meeli*, *N. multifasciatus* and *N. pulcher*. This smoother reflects how much uncertainty the model allowed for transitions not seen in the training set. Therefore, higher Best- s values may indicate that these corpora required more smoothing because the training sets did not fully represent the transitions seen in the testing set.

Figure 3.2 visualizes the trends in AvgLL as k -order varied. A model using the order $k = 0$ effectively represents sequences as a distribution of state frequencies across the entire corpus, without incorporating any historical information from the sequence at all. Therefore, the poor fit of $k = 0$ models in all cichlid corpora suggests that cichlids' postural sequences do not consist of random combinations of behaviors, but rather demonstrate sequential rules.

For each species, model fit improved when moving from $k = 0$ to the Best- k (indicated by the black arrow), after which performance declined again for higher orders. This decline suggests that models with orders greater than the Best- k began to overfit to the training data rather than capturing additional generalizable structure.

3.1.4 Transition Matrix Heatmaps

One of the main advantages of Markovian modeling is that the fitted models produce explicit transition matrices which give full read-outs of the calculations they use for sequence prediction. These matrices can be plotted as heatmaps, which provide interpretable representations of the sequential structures observed in each corpus.

After identifying the best-fitting model for each cichlid corpus, a complete Collapsed History transition matrix was compiled using the corresponding Best- k and - s parameters. Heatmaps of these transition matrices are presented in Figure 3.3. Due to the sheer number of rows for transition matrices at higher orders, y-axis labels of historical tuples were omitted for print. Rows are instead ordered alphabetically by the most recent state in the historical tuple.

The heatmaps show that the identity of the historical tuple strongly influenced the distribution of probabilities for non-self transitions, shown by the

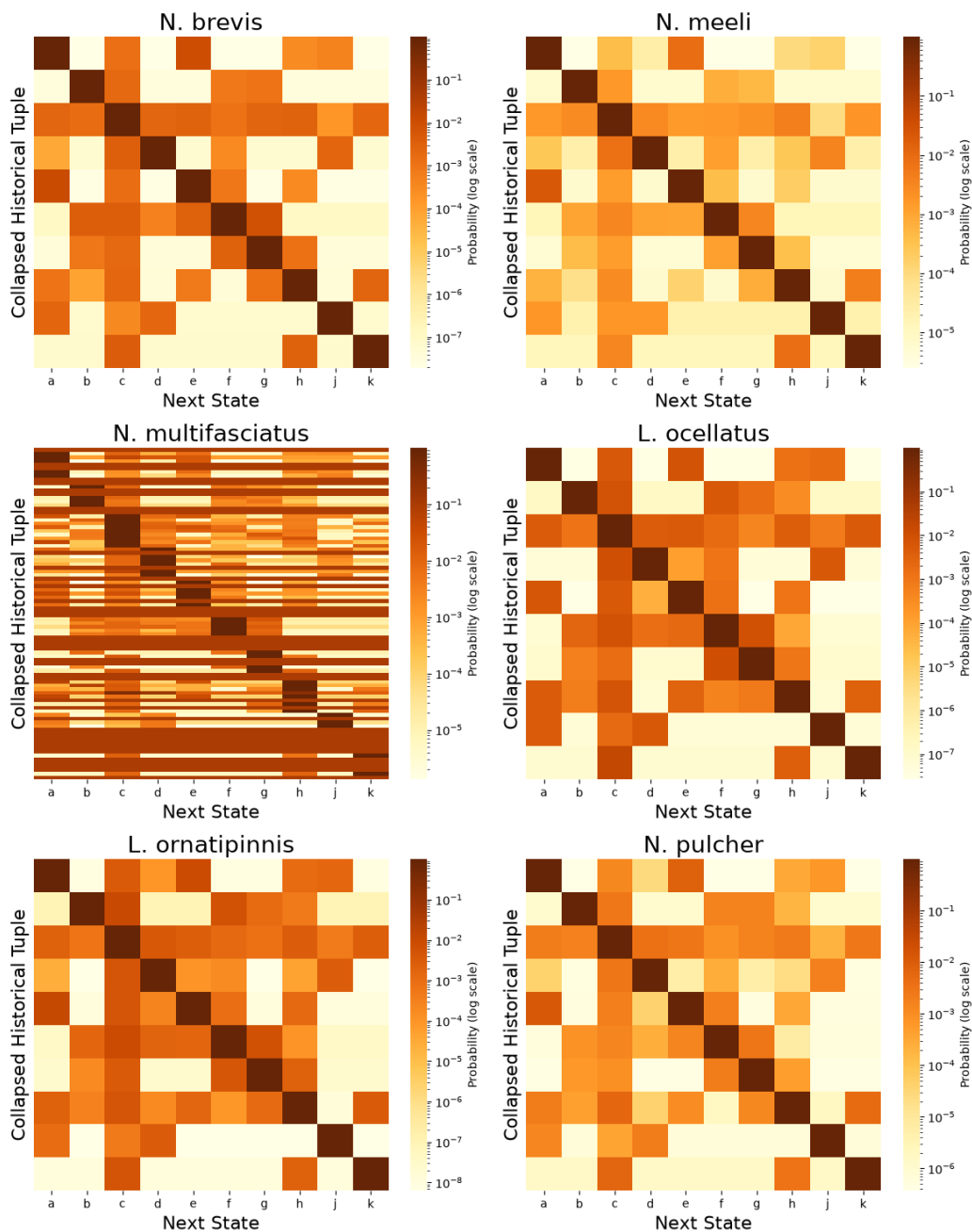


Figure 3.3: Transition matrix heatmaps of best-fitting Markovian models for each cichlid species. Cells of each heatmap portray the probability (log scale) of a next state (cell column) given a historical tuple (cell row). Y-axis labels are omitted, but rows are ordered alphabetically by the most recent character in the historical tuple.

distinct color patterns by row. This indicates that the fitted models successfully captured statistical differences between historical contexts. If historical states in a tuple were redundant, multiple rows would instead show identical color patterns because the added historical information would not have influenced the next-state distribution.

All six species also showed strong rates of self-transitions, visible as a diagonal band of high-probability cells in each heatmap. This reflects the long runs of repeated characters in the original corpora, because posture was tracked frame-by-frame. The horizontal lines visible in the *N. multifasciatus* heatmap are artifacts of additive smoothing. These lines correspond to historical tuples that were not observed anywhere in the corpus but were nonetheless assigned non-zero transition probabilities by the smoothed model.

The heatmaps further demonstrate that historical tuples differed in how specifically they predicted next states. In particular, state *c* appears to function as some form of intermediary posture. State *c* was relatively likely to be reached from any historical tuple. When *c* was the most recent state in the historical tuple, it was also relatively likely to transition to any other next state. This pattern suggests that state *c* is a posture that fish enter to reach other states when a direct transition is physically impossible.

The species also differed in their overall postural transition "repertoires", represented by the global heatmap patterns. Some state transitions that were common for one species were uncommon for another (e.g., the cell (7, 8) for *N. brevis* and *L. ocellatus* had a near-zero probability, but for *N. meeli*, *L. ornatipinnis*, and *N. pulcher* had intermediate values). Other cells were not populated in any species, indicating that some postures may be unnatural to reach directly from particular preceding states. Instead, fish may need to accomplish these behavioral trajectories by first transferring through intermediary postures like state *c*.

3.2 Bengalese Finch Song

3.2.1 Materials and Preprocessing

This dataset originated from Ref. [37] and represents songs from six adult male Bengalese finches (*Lonchura striata domestica*), named Birds 1-6. Birds were housed individually in sound-attenuating boxes for several days, and sound was continuously recorded. These recordings were segmented and annotated into discrete syllables using a trained deep neural network (TweetyNet [4]), and hand-checked to ensure quality and resolve any syllable ambiguities.

Song bouts were defined as continuous sequences of syllables, each separated by more than 2 seconds of silence. Syllable sequences were further

simplified by collapsing certain recurring combinations or repeated syllables into single condensed representations, called introductory states or inter-notes depending on whether they occurred only at the start of song bouts or also in later positions. Other highly stereotypical repeated-syllable runs were also condensed into single states.

This process produced a corpus of strings for each bird, where each string represented one song bout consisting of characters, each corresponding to a unique syllable (including introductory/inter-notes). Table 3.3 lists the syllable set for each bird and any corresponding introductory or inter-notes. Each bird’s corpus was analyzed separately for all further analyses.

Bird	Syllables	Introductory/Inter-notes
1	a, b, e, g, h, l, q, u	q
2	a, b, c, d, e, f, g	N/A
3	b, c, d, e, i	i
4	b, e, i, k	i
5	b, e, f, g, h, i, m	i
6	a, b, e, f, h, i, m	i

Table 3.3: *Syllable sets for each bird.* Introductory or inter-notes are listed if applicable.

3.2.2 Descriptive Statistics and String Behavior

Basic descriptive statistics are presented in Table 3.4 for full and collapsed strings.

Syllable distributions over full string position were plotted to establish whether these corpora meet the analysis assumptions of irreducibility and time-homogeneity (see Figure 3.4). Ribbon plots were cropped at Q3 to avoid biases from a small set of very long strings.

Bird	N	Full Strings			Collapsed Strings		
		Q1	Q3	Max	Q1	Q3	Max
1	657	109	164	261	29	47	75
2	500	37	57	106	21	33	67
3	985	50	80	152	14	24	52
4	788	58	108	229	10	20	42
5	501	48	105	205	5	16	34
6	300	62	139	367	24	57	153

Table 3.4: *Basic descriptive statistics for each bird's song corpus.* From left to right: Bird, corpus size (N), lower quantile of string length (Q1), upper quantile of string length (Q3), and maximum string length (Max), separated by full strings (left) and repeat-collapsed strings (right). Quantiles have been rounded to the nearest whole number.

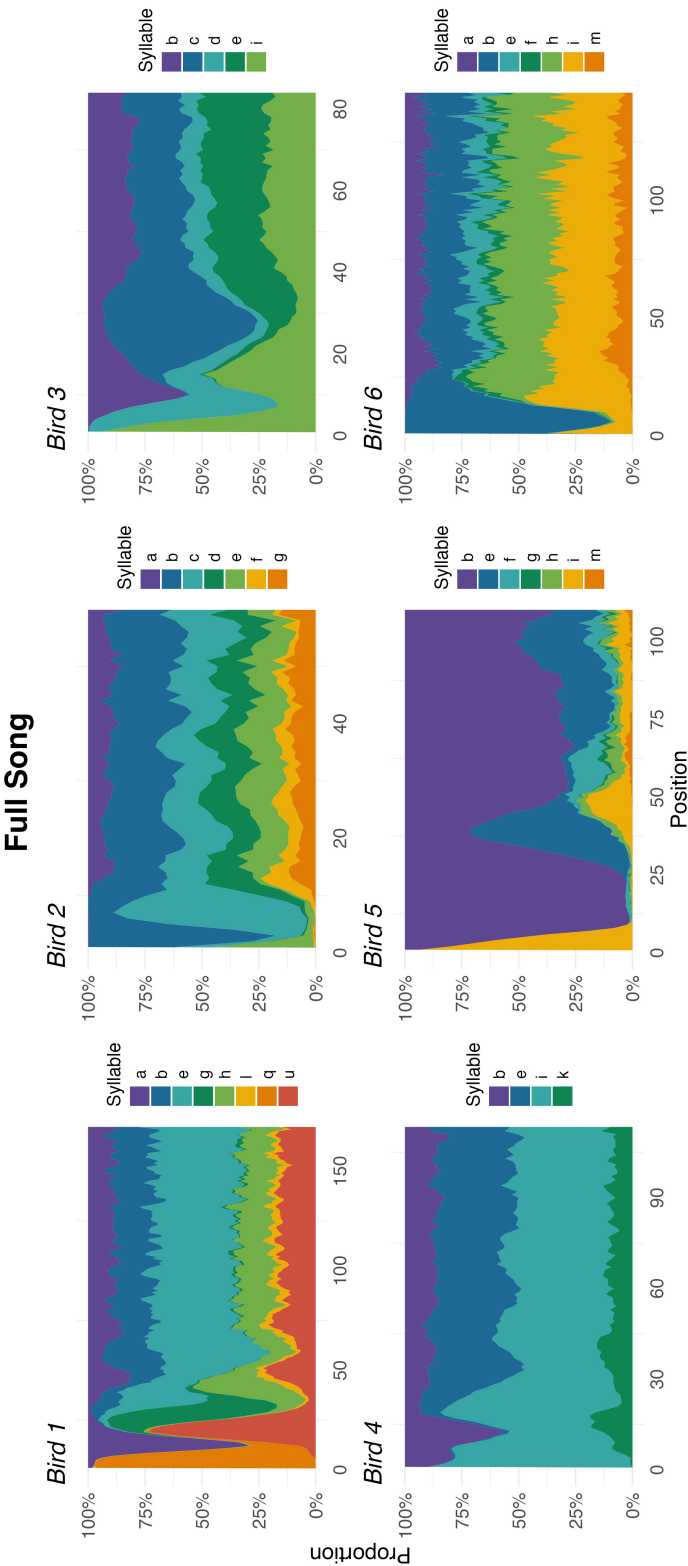


Figure 3.4: *Finch syllable ribbon plots for full-song corpora.* Each facet depicts the proportion (y-axis) of each syllable (color) at each position (x-axis) of full songs in each bird's corpus. Plots are cropped to the string length upper quantile (Q3).

These plots contrast strongly with those of the cichlids, as there seem to be strong systematic shifts in syllable distributions over position. Specifically, each bird's ribbon plot appears to consist of two processes: an initial Process 1 where each position is dominated by one syllable, followed by a second, relatively stationary Process 2 which is more reminiscent of the cichlid ribbon plots in Figure 3.1.

This pattern persists in collapsed strings, shown in Figure 3.5. Because this partitioning persists at the unique-syllable level, this indicates that the structure of Process 1 can be modeled at the unique-syllable level, relatively independently of the length of each syllable run.

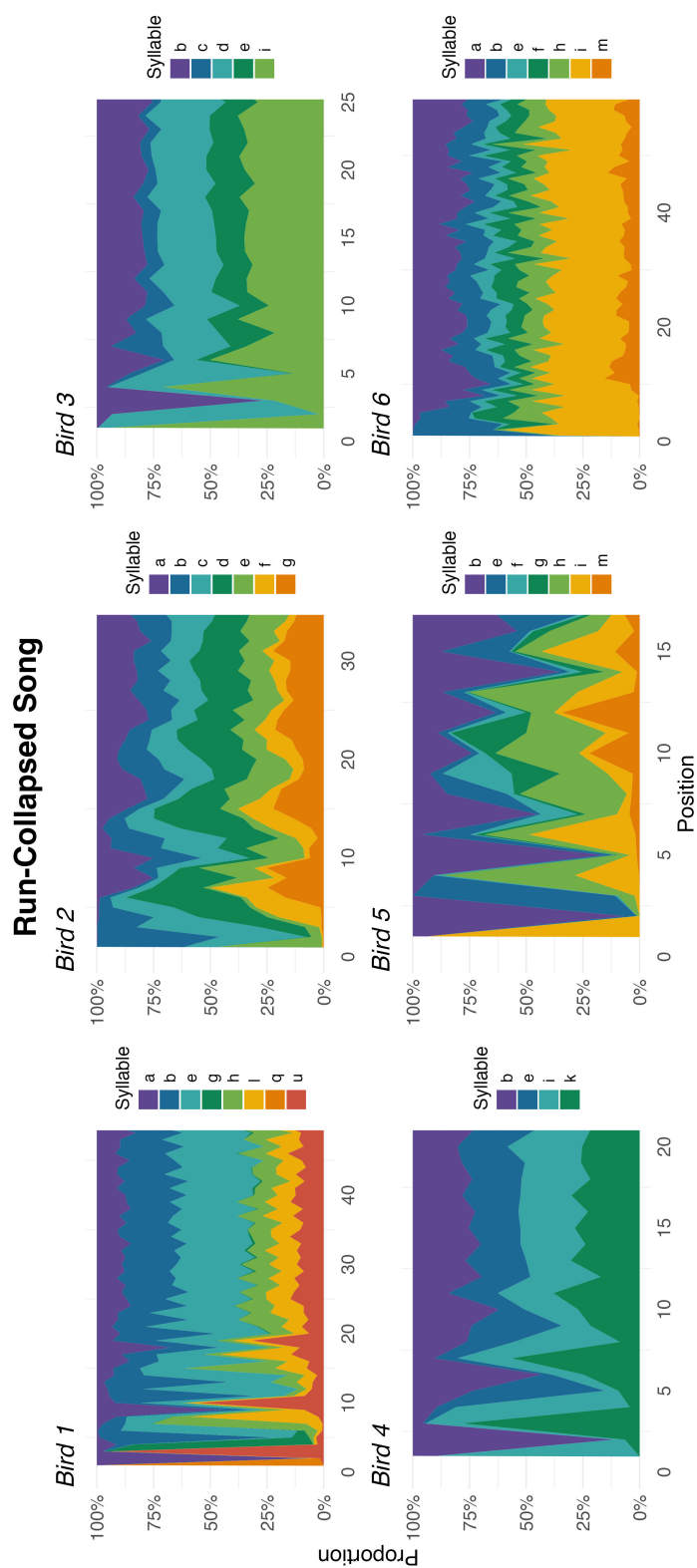


Figure 3.5: *Finch syllable ribbon plots for collapsed-song corpora.* Each facet depicts the proportion (y-axis) of each syllable (color) at each position (x-axis) of repeat-collapsed strings combined over each bird's corpus.

As is, these songs would violate the time-homogeneity assumption necessary for Markovian modeling and generation. However, one can still model these songs as a dual-process system, where each process is generated separately and then stitched together. Process 1 is very deterministic on both the full and collapsed level, so generation may be able to rely on a simple look-up table of example Process 1 "prefixes" extracted directly from the corpora. Process 2 is relatively stationary and can therefore still be analyzed and generated using Markovian models, as with the cichlids. During generation, Process 2 can be generated and stitched behind a Process 1 prefix to recreate a full dual-process string.

3.2.3 Prefix Tries for Extracting Regularities

However, this method hinges on identifying the point in each string where Process 1 transitions to Process 2. Figures 3.4 and 3.5 also indicate that the prefixes within Process 1 are not completely identical, since each position is still not 100% dominated by one syllable. One way to identify the transition from prefixes to Process 2 is by using 'tries'. Tries are a type of tree data structure used to encode and retrieve strings from dictionaries, and are often used for keyword searches and predictive word completion. They create a condensed mapping of all possible string continuations in the dictionary by tracing characters from each word at each position through a tree graph.

Figure 3.6 depicts a trie for the keys "apple", "apricot", "banana", "peach", and "pear". Each child node shares a common start with its parent. Using this structure, one can track string continuations and extract commonly occurring string starts within the trie, since these prefixes will be ancestors to many differentiating child nodes. By treating each bird corpus as a dictionary, tries can be used to extract the common prefixes contributing to the strong regularities in Process 1.

Figure 3.5 demonstrates that the structure of the bird prefixes seems to be founded at the unique-syllable level, relatively independent of syllable run length. Therefore, tries were constructed using the collapsed string corpora. This option is more robust to variability in syllable repetition in prefixes. Otherwise, each variation would be treated as its own branch, leading to an extremely dense and likely extremely redundant prefix trie.

A trie was constructed for each bird corpus with a maximum length of 40 characters. This boundary was chosen because it was considerably longer than the maximum Process 1 length across all birds in run-collapsed string ribbon plots. In fact, this maximum length is arbitrary, provided it exceeds the longest prefix, because beyond the prefix length there should not be enough sequential regularities for consistent extraction anyway.

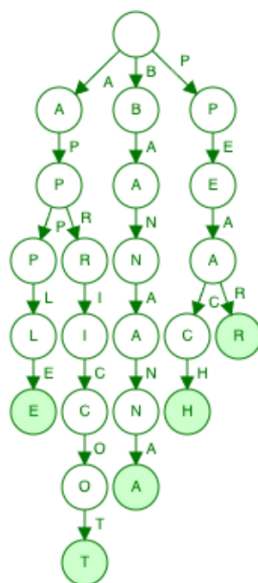


Figure 3.6: An example of a trie with the keys "apple", "apricot", "banana", "peach", and "pear". "ap" is a common prefix to "apple" and "apricot", and "pea" is a common prefix to "peach" and "pear". Adapted from [38].

For each unique root-node syllable, all trie entries starting with this root were identified, isolating all possible continuations of this root. Entries occurring at least 5% of the time were identified as prefixes. This criterion was chosen because it allows for detecting multiple prefixes rather than only the most common one per bird. Later during generation, this allows sampling from several prefix options so generated strings are less stereotyped.

Table 3.5 presents all identified prefixes for each bird. Because of the trie's inherent structure and this search strategy, these prefixes can be grouped into nested "families" originating from the same root node and diverging into sub-families at later nodes.

Table 3.5: *Bird prefix families.* Prefix and the proportion of strings containing that prefix (Prop.) are listed by bird and coded by color. The lightest shade indicates the start of one prefix family, determined by a shared root node. Shades of the same color indicate divergences into sub-families.

Bird	Prefix	Prop.
Bird 1	qau	0.92
	qaugbe	0.76

Continued on next page

Table 3.5 continued from previous page

Bird	Prefix	Prop.
	qaugbehe	0.09
	qaugbehlau	0.52
	qaugbehlaube	0.51
	qaugbehlaubebe	0.44
	qaugbehlaubebehe	0.30
	qaugbehlaubebehela	0.23
	qaugbehlaubebehelau	0.23
	qaugbehlaubebehelaube	0.21
	qaugbehlaubebehelaubebe	0.11
	qaugbehlaubebehelaubebeh	0.06
	qaugbehlaubebehelaubeh	0.10
	qaugbehlaubebelau	0.07
	qaugbehlaubebelaube	0.06
Bird 2	bcbcdfgaebcdfdga	0.06
	bcbcdfgaebcdfdga	0.06
	bcbcdfgaebcdfdga	0.06
	ebcbcd	0.15
	ebcbcdfd	0.14
	ebcbcdfdgae	0.13
	ebcbcdfdgaeb	0.10
	ebcbcdfdgaebcd	0.09
	ebcdfdgaeb	0.09
	ebcdfdgaebc	0.08
	ebcdfdgaebcdfdgaeb	0.06
	ebcdfdgaebcdfdga	0.05
Bird 3	i	0.94
	idb	0.73
	idbi	0.69
	idbice	0.05
	idbid	0.63
	idbidb	0.28
	idbidbi	0.27
	idbidbic	0.09
	idbidbice	0.09
	idbidbid	0.16
	idbidbidb	0.06

Continued on next page

Table 3.5 continued from previous page

Bird	Prefix	Prop.
	idbidbidi	0.10
	idbidbidice	0.05
	idbidi	0.35
	idbidice	0.18
	idbidicedb	0.10
	idbidicedbi	0.10
	idbidicedbid	0.06
	idbidicei	0.06
	idbidiceid	0.06
	idbidiceidb	0.05
	idbidiceidbi	0.05
	idbidid	0.13
	idbididb	0.06
	idbididbi	0.06
	idbididi	0.06
	idi	0.18
	ididbi	0.10
	ididi	0.06
Bird 4	i	0.89
	ibki	0.75
	ibkibk	0.16
	ibkibki	0.15
	ibkibkie	0.13
	ibkibkiebk	0.10
	ibkibkiebki	0.07
	ibkibkiebkie	0.06
	ibkie	0.57
	ibkieb	0.52
	ibkiebk	0.51
	ibkiebkb	0.11
	ibkiebkbbie	0.07
	ibkiebki	0.39
	ibkiebkib	0.05
	ibkiebkie	0.31
	ibkiebkieb	0.22
	ibkiebkiebk	0.22
	ibkiebkiekb	0.07

Continued on next page

Table 3.5 continued from previous page

Bird	Prefix	Prop.
	ibkiebkiebkbk	0.06
	ibkiebkiebkbki	0.05
	ibkiebkiebki	0.13
	ibkiebkiebkie	0.11
	ibkiebkiebkieb	0.07
	ibkiebkiebkiebk	0.06
Bird 5	be	0.05
	ibe	0.89
	ibeh	0.44
	ibehb	0.43
	ibehbi	0.30
	ibehbib	0.29
	ibehbibe	0.14
	ibehbibh	0.14
	ibehbibhfhgm	0.13
	ibehbibhfhgmhb	0.12
	ibehbibhfhgmhbi	0.10
	ibehbibhfhgmhbib	0.10
	ibehbibhfhgmhbibe	0.09
	ibei	0.22
	ibeib	0.22
	ibeibe	0.11
	ibeibhfhgmhb	0.08
	ibeibhfhgmhbib	0.07
	ibeibhfhgmhbibe	0.06
Bird 6	bib	0.35
	bifh	0.19
	bifhiaiaiam	0.07
	bifhiaiaiamb	0.06
	bifhiaiaiambi	0.06

Table 3.5 shows extensive prefixes present in all corpora, with several birds exhibiting remarkably long and stereotyped song beginnings. For example, 92% of Bird 1's songs began with the syllables 'qau', and 6% of this bird's 657 songs continue to the 24-syllable collapsed sequence 'qaugbehlaubebehelaubebeh'. Bird 5 also showed highly stereotyped beginnings of songs, with 89% of songs beginning with the prefix 'ibe' and 9% continuing to the 17-character

prefix 'ibehbibhfhgmhbibe'.

Bird 2 had the least established prefixes, with the most common prefix 'ebcbcd' occurring in only 15% of strings. This corresponds to the less deterministic Process 1 visible in the bird's full and collapsed ribbon plots. Bird 6 also showed comparatively weak prefix regularity, with the most common prefix 'bib' occurring in 35% of strings.

The single-syllable prefix 'i', extracted in Birds 3 and 4, is a trivial prefix because the original processing of the datasets already identified these as collapsed representations of introductory/inter-notes. However, the continuations following this initial syllable are not trivial, because they continue to show highly deterministic structure despite no longer representing the originally compressed syllable sequences.

Importantly, though prefix tries were searched to a maximum length of 40 characters, the longest extracted prefix was 24 characters long ('qaugbehlaubebehlaubebeh' from Bird 1). This is consistent with the approximate maximum length of Process 1 in the collapsed ribbon plots from Figure 3.5. The lack of longer extracted prefixes provides further evidence that the sequence structure of songs becomes substantially less deterministic beyond this point.

3.2.4 Cropped Corpora

The extracted prefixes were then used to separate Process 1 from Process 2. Each bird's full corpus was searched for strings starting with an uncollapsed version of one of its prefixes. These prefixes were collected in a list for generation later on, and were cropped out of each bird's full string corpus, resulting in cropped corpora intended to isolate Process 2.

Figure 3.7 presents ribbon plots for the resulting cropped corpora. Compared to the original full-song ribbon plots, most of the strong structure associated with Process 1 has been removed, resulting in much more stationary syllable distributions. This is particularly notable because this processing pipeline only referred to the run-collapsed versions of songs, yet transferred successfully to the full songs. This provides further evidence that the structure of Process 1 is primarily determined at the unique-syllable level, and is relatively insensitive to syllable run length.

However, this cropping procedure did not perfectly isolate Process 2. The first several positions of the cropped sequences still show some irregularities in their syllable distributions. One possible explanation for this is the assumption that Processes 1 and 2 are completely independent, with no transitional period between processes. The beginnings of the cropped-song ribbon plots may instead indicate a transition section in which the syllable structure is not sufficiently regular to be detected by the prefix trie method, but still contains

a different structure than Process 2.

Another possibility is that the 5% prefix criterion was too strict to identify all prefix continuations. This method could therefore be improved by using an optimization routine to find the best value rather than relying on a fixed heuristic. Such a procedure could also allow the threshold to vary between corpora rather than applying the same criterion to every bird.

Bird 5 was a notable outlier in the effectiveness of the cropping procedure. Its cropped ribbon plot remained substantially less stationary than those of the other birds. This reflects the weaker distinction between Processes 1 and 2 already visible in the original song corpus. Nevertheless, the prefix trie still identified a large set of regularities in Bird 5's songs.

Furthermore, the maximum prefix length for Bird 5 was only 17 characters ('ibehbibhfhgmhbibe') even though the trie was constructed at a maximum length of 40 characters. This indicates that the post-prefix sequence does not exhibit the same extent of deterministic structure as Process 1.

Despite these residual irregularities, the Markovian modeling approach is sufficiently robust to model the cropped corpora. Because transition matrices are compiled across sequence positions, the remaining irregularities near the beginnings of the cropped songs may introduce some bias, but the resulting matrices should still primarily reflect the relatively stationary Process 2.

Most of Process 1 was successfully removed in all corpora, which is important because otherwise the deterministic prefixes would have strongly biased the transition matrices toward the prefixes. Markovian modeling was therefore continued using the cropped corpora.

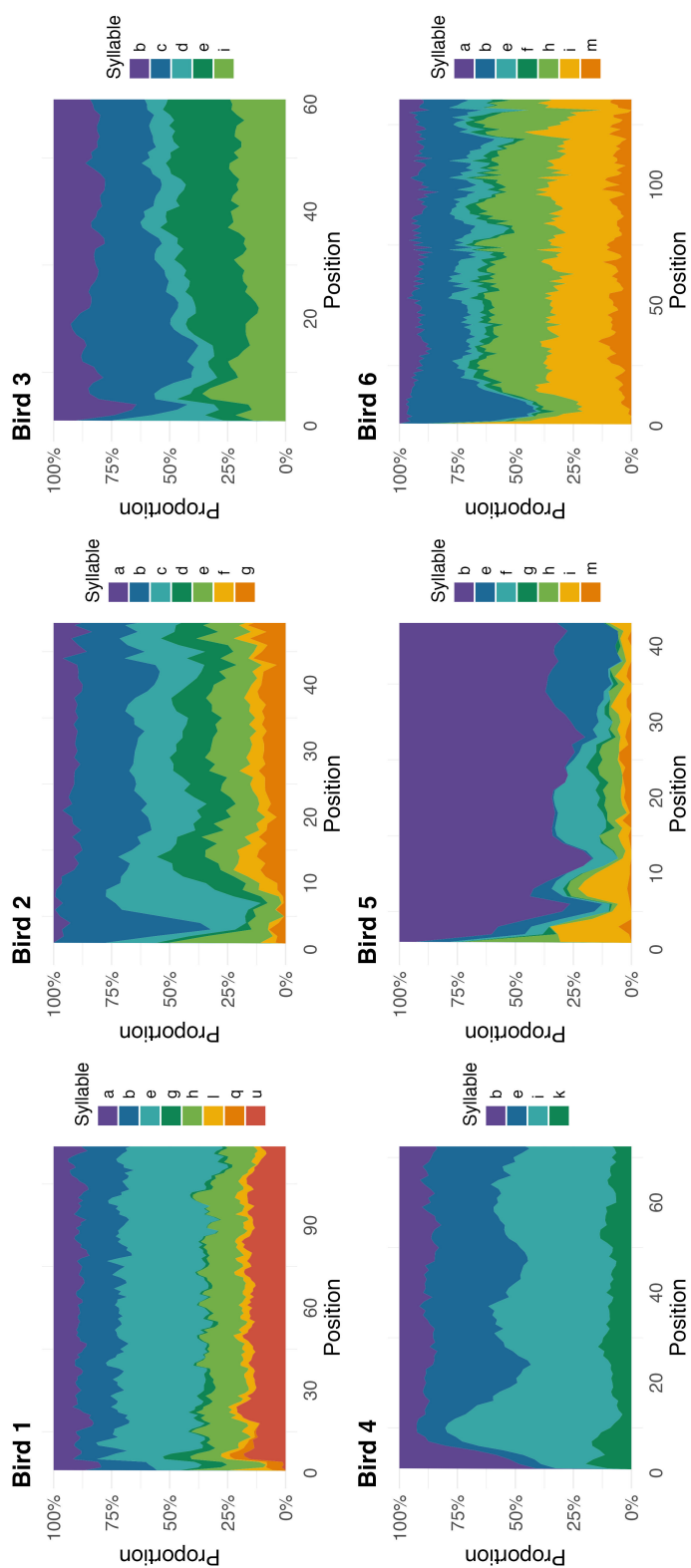


Figure 3.7: *Cropped full corpora syllable distributions.* Each facet depicts the proportion (y-axis) of each syllable (color) at each position (x-axis) of cropped strings combined over each bird's corpus.

3.2.5 Sanity Checks for Prefix Extraction

Given the strong indications of prefixes from the ribbon plots and prefix tries, it was necessary to verify that the structures identified by this analysis pipeline were not simply an artifact of aligning all strings at their beginnings. Because songs varied in length, patterns further down the sequences could have simply been washed out by these visualization techniques, which in turn informed the maximum lengths of prefix tries and the string locations for searching for regularities (i.e., only at the beginnings of strings).

To check whether string alignment influenced the detection of sequence regularities, the run-collapsed song corpora were transformed in several ways and run through this analysis pipeline again. Specifically, this pipeline was run on versions of each corpus which were: a) fully reversed by string; b) character-randomized by string; c) rotated by cropping the first ten characters of each string and appending them to the end; and d) rotated by cropping up to the third occurrence of a commonly occurring syllable in each string and appending that segment to the end. Ribbon plots and tries of these transformed corpora can be found in Appendix A.

Overall, no regularities to the extent of the prefixes were identified in the transformed corpora. The strongest regularities identified were from the reversed songs, which extracted common sequences from all birds except Bird 1. The extracted sequences can be thought of as "suffixes", the same way the previous analysis extracted prefixes. However, for most birds these suffixes were less regular and extensive than the corresponding prefixes.

Tries for other transformations detected very few regularities. Two sequences for each of Birds 3 and 4, and six sequences were extracted for Bird 5 in the randomized corpora. This likely reflects the high frequencies of particular syllables in those corpora combined with small syllable set sizes, rather than genuine sequential structure.

The strongest evidence that the original prefixes are genuine structures comes from the lack of detected regularities in the two rotated corpora. Only two regularities were detected in Bird 5's corpus of character-rotated songs, and one, six, and four regularities were detected for Bird 3, 4, and 5, respectively, in the syllable-rotated corpora.

In contrast to the randomized corpora, the rotated sequences preserve the original sequential structure but shift the alignment of sequences to later positions. If prefix detection were an artifact of alignment at the beginnings of songs and similar deterministic structures continued throughout the songs, the rotated corpora should have produced comparable numbers of regularities, which was not the case.

Overall, these findings support the conclusion that the detected prefixes

represent genuine structure in Bengalese finch song rather than being artifacts of string alignment. They also provide evidence that suffixes may exist for some birds, although these regularities were generally weaker.

3.2.6 Markovian Modeling Results

100 iterations of Markovian modeling using the Collapsed History strategy were run on the cropped corpus of each bird for orders $\mathbf{k} \in \{0, 1, 2, 3, 4, 5\}$ and additive smoother values $\mathbf{s} \in \{0.001, 0.01, 0.1\}$ (parameters bolded for reference). For each iteration, the k value corresponding to the highest log-likelihood was extracted along with the corresponding s . Within this subset, the best-fitting k -order (Best- k) was defined as the mode k , and the best-fitting smoother value (Best- s) was identified as the mode s associated with the Best- k . Figure 3.8 presents the average performance of each k -value for each bird.

Table 3.6 presents the Best- k , the proportion of iterations indicating the Best- k , the corresponding average log-likelihood over all iterations for the Best- k , and the corresponding Best- s .

Bird	Best- k	Prop.	AvgLL	Best- s
1	3	0.71	-0.6928811	0.1
2	4	0.56	-0.6463401	0.1
3	2	0.53	-0.6260475	0.1
4	3	0.81	-0.4783114	0.1
5	2	0.69	-0.3054980	0.1
6	3	0.56	-0.7119472	0.1

Table 3.6: *Best- k by bird.* From left to right: bird, best-fitting k -order compiled from 100 iterations of Collapsed History modeling (Best- k), proportion of iterations of best performance by that k value (Prop.), corresponding average log-likelihood of the best-fitting k over all iterations (AvgLL), and mode smoother associated with the Best- k (Best- s).

Compared to the cichlid models, the bird models showed greater variety and higher Best- k values. Birds 3 and 5 were best described by second-order Collapsed History Markovian models; Birds 1, 4, and 6 by third-order models, and Bird 2 by a fourth-order model. However, these fits were less consistent than for the cichlid corpora. This suggests that although the Collapsed History strategy may provide a reasonable approximation of Process 2, it may not fully capture the rules governing song structure. The true generative process may

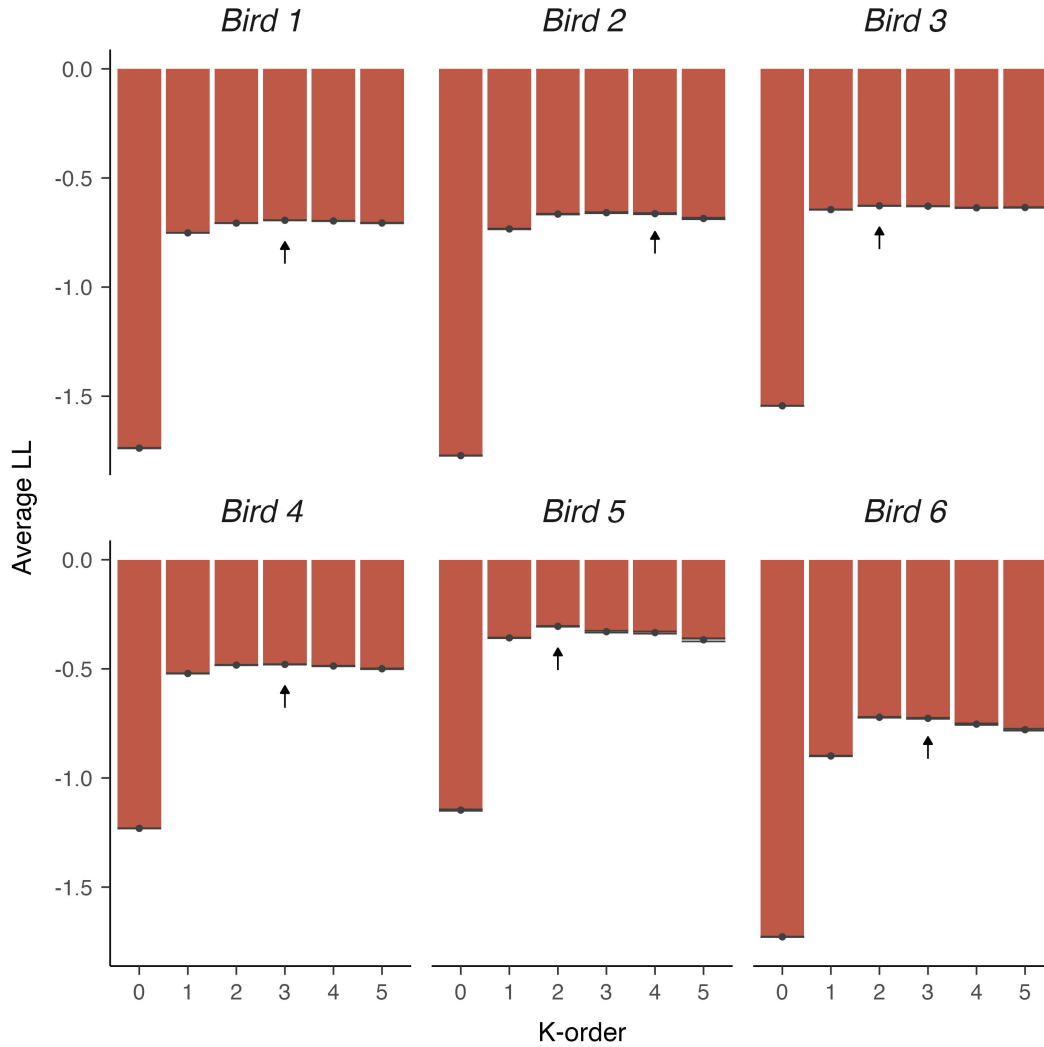


Figure 3.8: *Bird Markovian modeling results.* Each facet shows the average log-likelihood (y-axis) for each k -order (x-axis), compiled from 100 iterations of Collapsed History modeling, for each bird. The black arrow in each facet points to the best-fitting k -order, calculated as the most-occurring k value corresponding to the highest AvgLL over each iteration. Error bars depict the 95% confidence interval for the AvgLL of each k -order.

therefore involve more complex dependencies than can be represented by the strategies used here.

Unlike the cichlid models, all bird models converged on the same Best- s value of 0.1. This was the largest smoother value tested and therefore allowed for the greatest uncertainty in transitions not seen in the training set. This result may be related to the higher Best- k values of the bird models. As order increases, the number of possible historical tuples increases exponentially, while the corpus size remains limited. In this way, the training split may fail to represent all historical tuples that are possible and observed in the testing set.

Figure 3.8 visualizes the trends in AvgLL as k -order varies. As with the cichlids, corpora show the worst fit for $k = 0$ models, indicating that a simple distribution of syllable frequency across each entire corpus does not adequately explain the structure of songs.

3.2.7 Transition Matrix Heatmaps

After identifying the best-fitting model for each bird, a complete Collapsed History transition matrix was compiled using the corresponding Best- k and - s parameters. Heatmaps of these transition matrices are presented in Figure 3.9. Again, y-axis labels of each historical tuple were omitted for print, but rows are ordered alphabetically by the most recent syllable in the historical tuple. It is important to note that these heatmaps only reference the cropped-song corpora and therefore only describe Process 2.

Compared to the cichlids, these heatmaps are much sparser despite using higher k -orders, and self-transitions are less dominant. This is particularly notable because the bird models used higher k -orders than the cichlid models, meaning there are exponentially more possibilities for historical-tuple-to-next-syllable transitions, though most are not observed in the corpora. This indicates that these birds are following highly specific structural rules even in Process 2.

Therefore, the difference between Processes 1 and 2 is *not* that Process 1 is structured and Process 2 is not, but rather that Process 1 uses prefixes with a globally deterministic syllable order, while Process 2 uses rules that determine the local "grammar" of a sequence, but not necessarily the global syllable order.

This distinction is important for sequence generation described in Chapter 4. The deterministic structure of Process 1 can be reproduced by sampling from extracted prefixes, but the flexible, but still highly structured Process 2 can be reproduced by sampling from the fitted models' transition matrices.

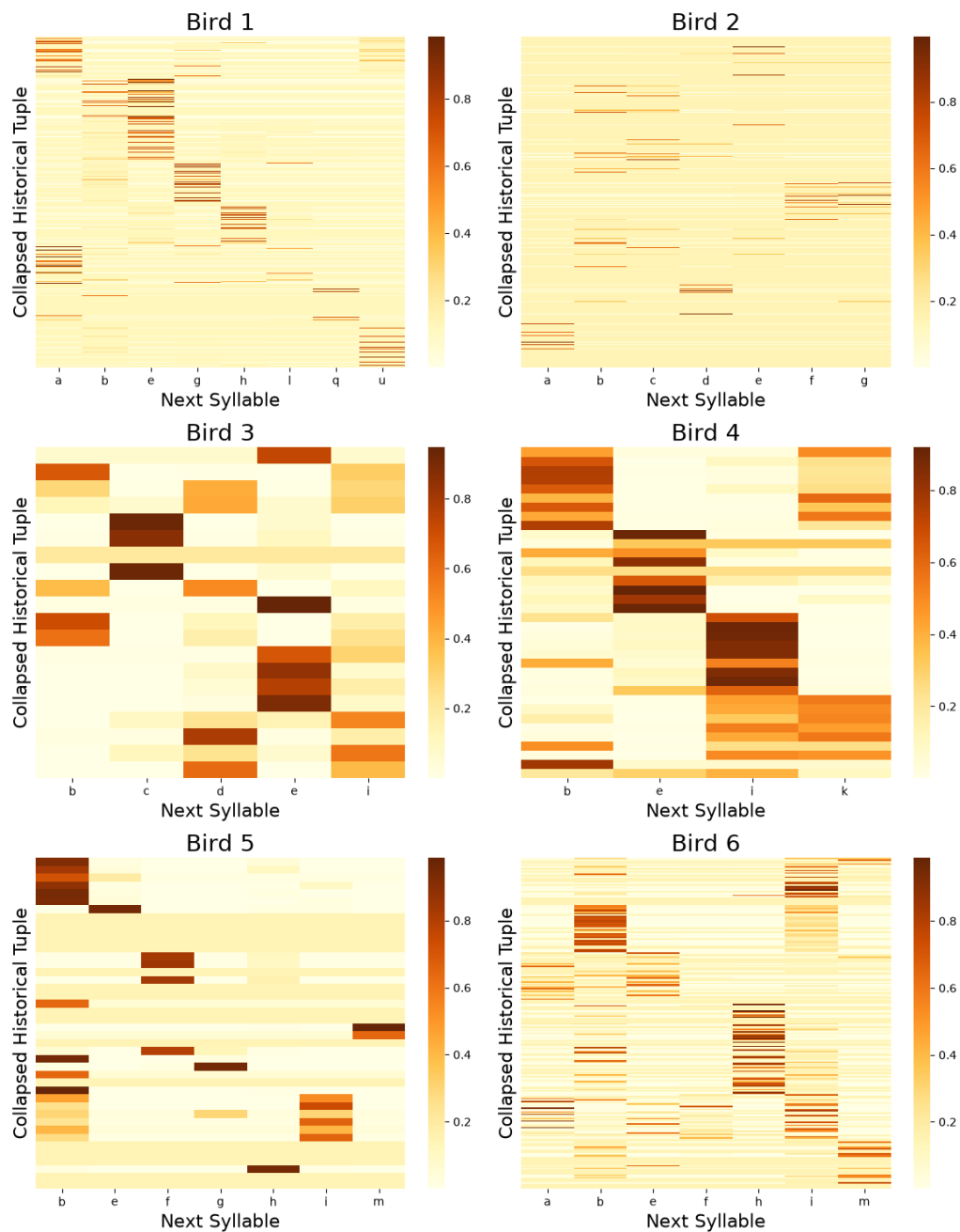


Figure 3.9: *Transition matrix heatmaps of best-fitting Markovian models for each bird.* Cells of each heatmap portray the probability of a next state (cell column) given a historical tuple (cell row). Y-axis labels are omitted, but rows are ordered alphabetically by the most recent character in the historical tuple.

Chapter 4

Naturalistic Sequence Generation

4.1 Generation App

The Markovian modeling pipeline detailed previously indicates which smoothed training model best generalized to an unseen testing set, with a Best- k and Best- s as output. These parameters can be used to generate novel strings.

Each iteration of generation (one full string from one corpus) can be organized in several steps. First, a smoothed transition matrix was compiled over each *entire* corpus with the Collapsed History strategy, using the corresponding Best- k and Best- s (i.e., corresponding to the heatmaps in Figures 3.3 and 3.9). A count version of this matrix (un-smoothed and non-normalized) was also compiled. Then, a starting tuple was randomly sampled from the set of historical tuples from the count matrix, weighted by frequency. This tuple acted as the initiation for generation, since reading from the transition matrix requires a history which at the beginning of a string does not yet exist.

Next, iterative generation of the next character was performed by sampling over the next characters in the smoothed transition matrix, given the most recent Best- k historical tuple of characters in the generated string. The sampled character was added to the end of the string, and the historical tuple moved one character further for the next round of sampling. This process was done iteratively to create a string of characters. The length of this string was determined by sampling from a normal distribution centered around the mean string length from the corresponding empirical corpus, and truncated at the first and third quartiles (Q1 and Q3).

The result of this procedure is a string following the historical dependencies approximated by the corresponding best-fitting Collapsed History Markovian model. This can be repeated many times to produce full corpora of generated strings.

4.1.1 Adding Prefixes

Generation of finch song requires an extra step. As demonstrated in Chapter 3, these songs present a dual-process structure: an initial prefix with strong regularities and structure, followed by a second stationary component.

These processes likewise needed to be generated separately. A list of uncollapsed instances of the extracted prefixes (see Table 3.5) was compiled for each bird’s corpus, and during every iteration of generation, one prefix was randomly sampled from this list. Then, strings were generated following the process above, but referencing the modeling results and corpora of the *cropped* songs. Generated string lengths were consequently sampled from a truncated normal distribution using the mean, Q1, and Q3 of the cropped songs. Finally, the sampled prefix was stitched in front of the generated sequence to produce a single sequence per generation iteration.

4.1.2 Generated Corpora

A 200-line generated corpus for each species can be found in the linked repository in Appendix C.

Sequence length and state/syllable distributions were compared between the empirical and generated corpora for each cichlid species (Figures 4.1 and 4.2) and for each bird (Figures 4.3 and 4.4). Appendix B additionally presents examples of strings from each empirical and generated bird corpus ¹.

¹Cichlid examples were not included for print because the sequences are thousands of characters long. Please refer to the linked repository for full sequences.

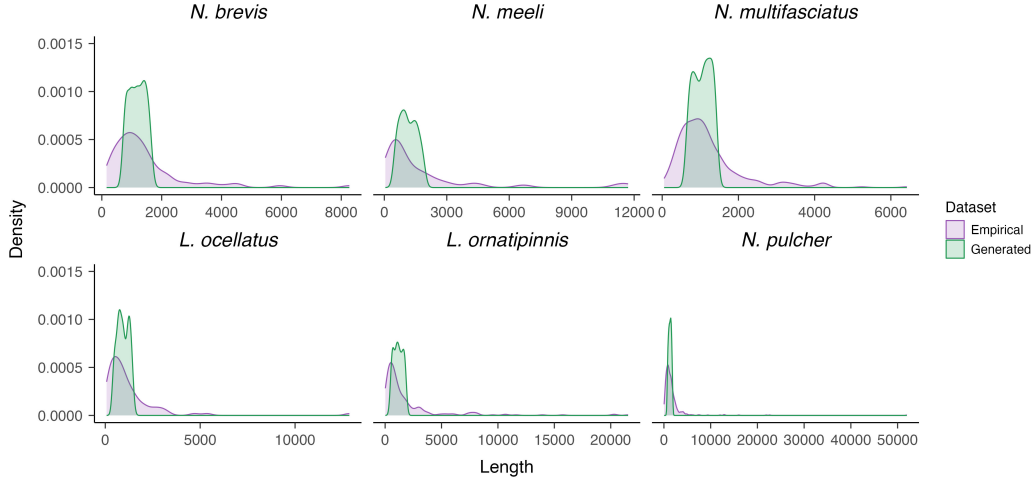


Figure 4.1: Sequence length distributions for empirical and generated cichlid corpora. Each facet compares the density (y-axis) of sequence length (x-axis) from the empirical (shaded purple) vs. generated (shaded green) corpora of one species of cichlid.

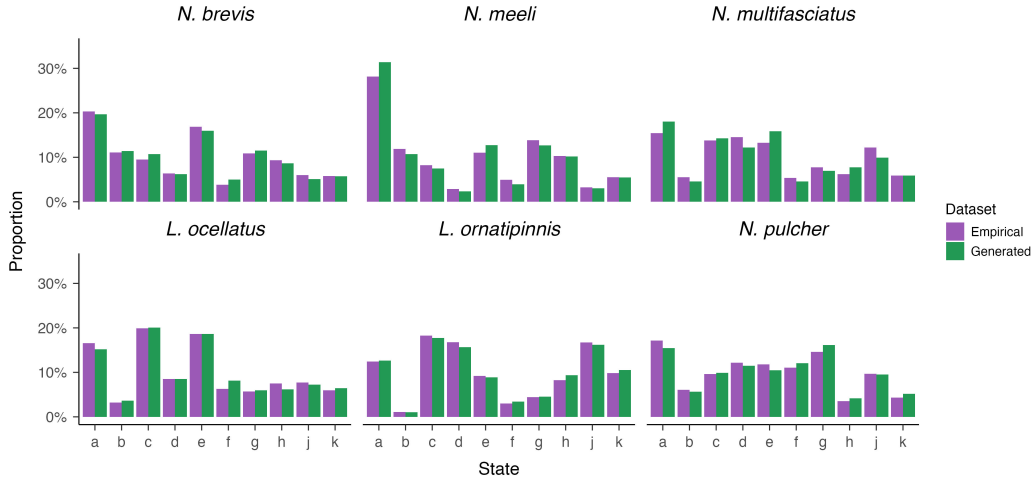


Figure 4.2: State distributions for empirical and generated cichlid corpora. Each facet compares the proportion (y-axis) of each unique state (x-axis) within the empirical (colored purple) vs. generated (colored green) corpora of one species of cichlid.

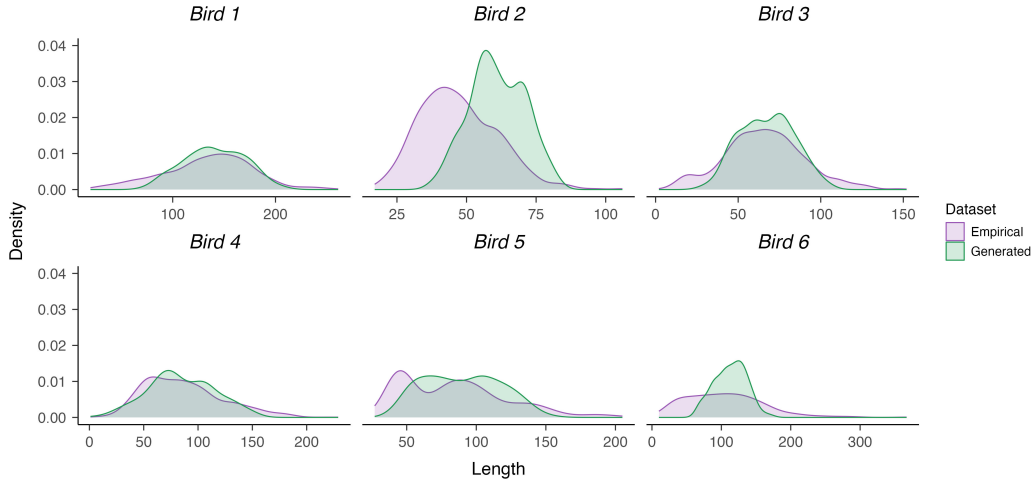


Figure 4.3: *Sequence length distributions for empirical and generated bird corpora.* Each facet compares the density (y-axis) of sequence length (x-axis) from the empirical corpus (shaded purple) vs. generated corpus (shaded green) of one bird.

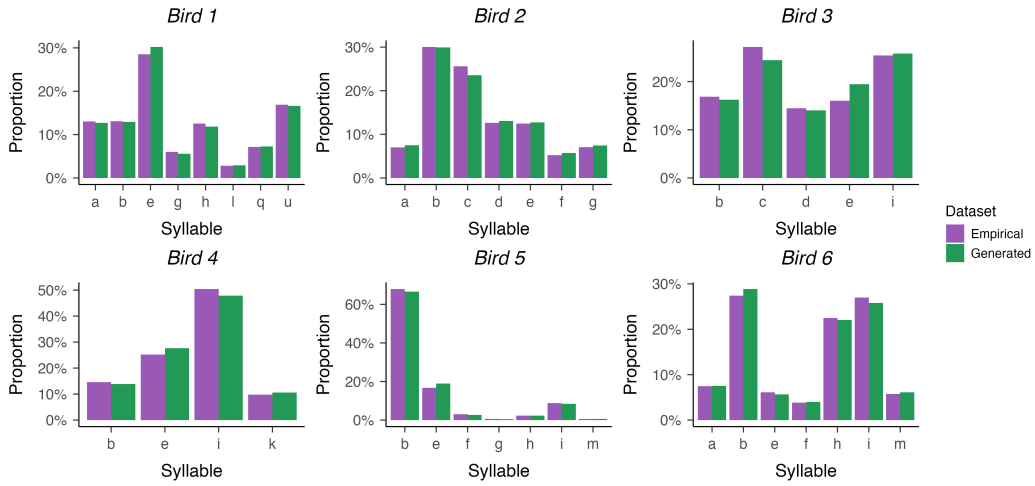


Figure 4.4: *Syllable distributions for empirical and generated bird corpora.* Each facet compares the proportion (y-axis) of each unique syllable (x-axis) within the empirical (colored purple) vs. generated (colored green) corpora of one bird.

The state distributions between the empirical and generated cichlid corpora are very similar, differing by only a couple percentage points across all species and states. The discrepancies in the length distributions from the two corpora were an intended result of the generation pipeline. The lengths of generated sequences were sampled from a truncated normal distribution centered around the mean of the corresponding empirical corpus, which effectively squeezes the empirical length density into a smaller interval. This is why the generated length distribution has a higher peak at the mode and drops steeply around it. This distribution guards the generation pipeline from being influenced by outlier sequence lengths from the empirical datasets.

The finch song corpora were created using a slightly altered process, because generation referenced the cropped songs and then added a sampled prefix to the beginning. With this in mind, it is remarkable that for both sequence length and syllable distribution, the generated corpora match the empirical corpora very well.

The exception to this is the length distribution of Bird 2’s generated corpus. Table 3.5 shows that this bird had the lowest proportion of prefixes in its corpus, with the most common prefix occurring in only 15% of songs. However, this generation method always adds a prefix, thus inflating the average sequence length in Bird 2’s generated corpus and resulting in a right-shifted distribution. This demonstrates the limitations of this stitching method for corpora where sequences with prefixes are common enough to be detected, but still are not the majority.

Nevertheless, the empirical and generated song sets (including Bird 2’s) also qualitatively match very well to each other (see Appendix B). Characters repeat for similar durations, transition to similar next characters, and follow similar higher-order patterns.

4.2 Suggestions for Future Applications

Though this analysis pipeline appears to produce promising results, it is inherently exploratory. True generation accuracy can only be measured through experimental paradigms. Without this, it is impossible to tell whether the patterns identified from these methods are meaningful or completely spurious, particularly because human researchers lack insider knowledge of the non-human communication systems analyzed here.

The characters from generated strings correspond to categorized behaviors that can be read back into the associated physical modalities. With playback experiments, researchers can probe whether these generated strings are naturalistic and meaningful by measuring the reaction of individuals to these

sequences.

Generated playback studies allow for much more control over the presented stimuli than true recording playbacks can. However, this assumes that generated sequences are treated the same way as true recordings by users of the communication system. Therefore, the first step with these generated strings should be to verify that they are valid models of communication. This means that individuals should not be able to discriminate between recordings and generated playbacks, but still discriminate between generated playbacks and less informed or inappropriate versions of behavioral sequences (e.g., random sequences, manually-coded sequences, sequences from another corpus).

Establishing non-discriminability of generated and prerecorded playbacks would justify further experiments using generated playbacks alone. The exploratory analyses in this thesis cannot probe into the meaning behind behaviors, but this can be done with well-designed experiments.

Several experimental approaches have already been developed to measure discrimination of social and communicative stimuli in both cichlids and finches. For example, Ref. [39] used the pS6 molecular marker as a proxy for neural activity in the social decision-making network (SDMN) of *L. ocellatus* during presentation of animated stimuli on a screen. Fish demonstrated differences in neural activation when presented with animated spheres compared to avatars of con- and heterospecific cichlids. This approach provides a way to distinguish responses to social and non-social events and also provides preliminary evidence for the use of avatars in cichlid playbacks, as neural activity distinguished them from non-social stimuli.

In finches, song discrimination has been successfully measured using forced-choice tasks [40], go/no-go tasks [41], and the presence of signature calls [42]. These metrics all provide ways to evaluate whether individuals can discriminate between empirical and generated playback sequences.

Establishing the validity of generated sequences enables a large range of future research. Researchers can explore whether different species or individuals react differently to the same generated sequence. If so, this may indicate that the same sequence could be used by different users to convey different meanings. Reactions towards the same sequence may also vary by context, particularly the presence of environmental variables like predators, food, and mates. This would link to many arguments in animal communication that lexicons are under-specified in pure meaning-mappings, and are only resolved through contextual information [43].

In finches, females select mates based on their courtship displays, in which songs play a heavy part [44]. For these birds, not only can researchers probe whether individuals react differently to their own songs compared to those of conspecifics, but also whether female finches show preferences towards certain

syllable patterns or motifs.

This analysis pipeline also offers the possibility of 'corrupting' structural patterns due to the completely explicit transition matrices used to generate sequences. By amending or knocking out the probabilities of certain transitions, one can explore at what point individuals start to discriminate between generated and prerecorded sequences again.

In this way, these generated sequences open the door for many experimental paradigms that can probe the meaning behind communicative behaviors, as well as behavioral differences between species or individuals.

Chapter 5

Main Discussion

5.1 Species Comparisons

Finch song differentiated itself from cichlid postural sequences by the presence of a deterministic prefix process at the beginnings of songs. This is particularly interesting given the pre-processing of these songs from the original paper [37], which already compressed some syllable sequences into introductory or inter-notes. The ribbon plots in Figures 3.4 and 3.5 instead indicated that there are further regularities in the beginning of these birds' songs extending past pure introductory notes.

Another difference between the model species was the range of their Best- k values, where cichlid postural sequences were best approximated with first- and second-order models, while finch song was best approximated with second-, third-, and fourth-order models. These results do **not** indicate a fish or bird's actual memory capabilities. While k -order can have implications for the structure and nature of different communication systems, it cannot be extended to the cognition of the users of that communication system. The k -order only gives an approximation of the amount of historical information that improves predictions about the next behavior.

Because the Collapsed History modeling strategy uses historical tuples of previous unique states, k -order could be interpreted as indicating the size of behavioral combinations that make up meaningful motifs. However, it may be better to consider k as an "inertia" of continuing a specific behavioral motif once it is begun, in light of a major difference between the two sets of corpora. Cichlid posture was tracked during dyadic interactions, while finch song was recorded from isolated birds. This means that the cichlid postural sequences unfolded not only based on historical dependencies within their own behavior, but also on another time series: the behavior of the other fish. Cichlids may indeed have longer postural motifs, but in this situation they likely were con-

stantly interrupted because they need to react to the other fish in the arena at the same time.

Alternatively, this finding may only be an artifact of the population sizes represented by each corpus. Cichlid corpora featured behavior from many individual fish, whereas each bird corpus featured behavior from only one bird. This means that the cichlid corpora inherently contain a noisier dataset due to individual differences. This may explain the smaller Best- k orders for cichlid models, because high variability in the types and structures of behavior would mask longer patterns.

The number of possible explanations for the same result highlights the limitations of purely exploratory analyses, and why it is crucial to move towards experimental designs. Each explanation listed here is a hypothesis that can be tested through playback experiments.

5.2 Strengths and Limitations

The methods in this thesis were designed with an emphasis on interpretability. Transition matrices from Markovian models give a completely explicit read-out of probabilities of different behavioral motifs. Prefix tries offer a systematic way to decide on cut-off points for deterministic processes, and these processes can be read back from the trie. Accessibility to the mechanisms behind estimation of sequence structures is vital for eventually understanding the meaning of these behaviors.

These methods are also compatible with biological datasets, which often suffer from very small sizes. Plugging these small datasets into a transformer model would have a high risk of overfitting, which for playback experiments would introduce the possibility of generating strings with unnatural behavioral sequences. In contrast, this generation method is almost entirely restricted to observed transitions, with leeway only from an additive smoother which can be amended to be more or less lenient in allowing unseen transitions in the training set. In general, these methods are highly customizable, which means they can be easily adapted to fit with the precedents and theory in the literature about each communication system.

However, this pipeline still has areas to improve, particularly in prefix identification. There were several thresholds that were selected qualitatively rather than quantitatively. The number of characters to search for prefixes was decided through visual inspection of ribbon plots and approximating the maximum length of Process 1. Mistakenly deciding on an overly long search length would not affect prefix detection, but making this value too small would cut off potential continuations of prefixes. Nevertheless, this heuristic was

validated, since the extracted prefixes were far shorter than the maximum search length, and by the removal of Process 1 in the cropped ribbon plots.

Another qualitative threshold was the 5% criterion for identifying true prefixes. This was decided for the finch dataset because it was a reasonable value so as not to discount the possibility of multiple prefixes, while remaining resistant to idiosyncratic sequences that were repeated by chance. However, in different systems, this threshold may not be as appropriate. For example, when there is truly only one prefix structure, a 5% threshold would be too lenient, but for a system with many prefixes (for example, human language), this may instead be too strict. Future work could address this by running an optimization routine to find the best cut-off criterion that isolates Process 2 by corpus.

In terms of finch song generation, the method of randomly sampling and stitching prefixes to a Markov chain assumes that Processes 1 and 2 are completely independent from each other, and there is no transition between processes. Additionally, because prefixes are only sampled from the empirical dataset, this means that generated songs are more stereotyped in Process 1 than for Process 2, especially because the generation method never sampled from songs without prefix starts. Furthermore, for birds like Bird 2 where prefixes are not as regular, this generation method inflates the presence of prefixes, since a prefix is stitched to every generated string.

Finally, this pipeline has computational constraints. The values of parameters evaluated in the simulation study and for modeling the empirical corpora were relatively restrained. This is because the compiled transition matrices grow exponentially in size as k -order increases, making these analyses computationally heavy and extremely time-intensive. This makes it difficult to detect higher-order structures without powerful computers.

However, these methods were ultimately only designed to provide explicit estimates of structures and patterns in the dataset which can be probed in further experiments. The quality of the resulting generated corpora demonstrates that this pipeline can reproduce complex statistical properties found in the empirical corpora.

5.3 Future Directions

Apart from playback studies, the other direction for development is improving modeling strategies. As discussed in Chapter 2, the strategy to identify key historical information in sequences is highly customizable. Two strategies, Lagged and Collapsed History, were proposed and utilized here with biological motivations, but these are far from the only two possible strategies. It

would be of particular interest to combine these strategies using a multivariate, vector-based Markovian model. In the current methods, the length of states is implicitly discovered from the modeling strategy, since the first entry in the historical tuple is the directly preceding character. However, it may be more accurate to treat these two variables as covariates which predict each other as well as their own future values. This is biologically plausible, since the duration of time spent in a behavior may also help determine what behavior comes next, and the type of behavior may determine how long future behaviors last. A model that treats these variables as potentially dependent may fit better than Lagged or Collapsed History strategies alone, which each only focus on one of these variables.

5.4 Conclusion

This thesis presented an analysis pipeline for approximating and replicating historical dependencies in behavioral sequences. Compared to other machine learning methods, the methods developed here provide explicit and interpretable statistical representations of sequence structure which can be used to develop hypotheses and evaluate them through experiments, particularly with playbacks.

This pipeline also provided methods for diagnosing whether Markovian assumptions are reasonably satisfied by a dataset. When these assumptions are violated, as was the case for Bengalese finch song, this thesis also presented a way to identify and extract deterministic processes from more stationary sequence structures.

When applied to cichlid postural behavior and Bengalese finch song, these Markovian models demonstrated that both communicative systems contain historical structures, but these structures differ between the two systems. Cichlid postural corpora were generally best described by lower-order models, while bird corpora were best described by higher-order models. Finch song also showed a dual-process structure, with Process 1 consisting of highly deterministic prefixes, followed by a relatively stationary Process 2.

The best-fitting models were then used to generate novel corpora containing the historical dependencies estimated from the empirical corpora. These generated corpora can be used to move from exploratory data analyses to experimental hypothesis testing. Ultimately, the intended contribution of this thesis is not a definitive model of either cichlid or Bengalese finch communication, but rather development of completely explicit models that provide access to the mechanisms underlying the estimation of historical dependencies.

As George E. P. Box stated, all models are wrong, but some are useful

[45]. This thesis presented a pipeline for developing useful models for future investigation into animal communication.

Bibliography

- [1] Tim Sainburg, Marvin Thielk, and Timothy Q Gentner. Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS computational biology*, 16(10):e1008228, 2020.
- [2] Quentin Bacquelé, Jean-Yves Barnagaud, Cyrille Violle, Frédéric Theunissen, and Nicolas Mathevon. The global biogeography of passerine songs. *Science*, 393(6809):381–386, 2026.
- [3] Noori Choi and Alex Jordan. Reorganization of shared behavioural elements drives behavioural evolution in an adaptive radiation. Unpublished manuscript.
- [4] Yarden Cohen, David Aaron Nicholson, Alexa Sanchioni, Emily K Mallaber, Viktoriya Skidanova, and Timothy J Gardner. Automated annotation of birdsong with a neural network that segments spectrograms. *Elife*, 11:e63853, 2022.
- [5] Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242, 2023.
- [6] Santiago Canchila, Carlos Meneses-Eraso, Javier Casanoves-Boix, Pascual Cortés-Pellicer, and Fernando Castelló-Sirvent. Natural language processing: An overview of models, transformers and applied practices. *Comput. Sci. Inf. Syst.*, 21(3):1097–1145, 2024.
- [7] Arjen van der Schaaf, Cheng-Jian Xu, Peter van Luijk, Aart A van’t Veld, Johannes A Langendijk, and Cornelis Schilstra. Multivariate modeling of complications with data driven variable selection: guarding against overfitting and effects of data set size. *Radiotherapy and Oncology*, 105(1):115–121, 2012.
- [8] Constantin Aliferis and Gyorgy Simon. Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best prac-

- tices in machine learning and ai. *Artificial intelligence and machine learning in health care and medical sciences: Best practices and pitfalls*, pages 477–524, 2024.
- [9] Mitali Raj. Clarifying model transparency: Interpretability versus explainability in deep learning with mnist and imdb examples. *arXiv preprint arXiv:2509.10929*, 2025.
- [10] Zachary C Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43, 2018.
- [11] Dezhe Z Jin and Alexay A Kozhevnikov. A compact statistical model of the song syntax in bengalese finch. *PLoS computational biology*, 7(3):e1001108, 2011.
- [12] Jason D Wittenbach, Kristofer E Bouchard, Michael S Brainard, and Dezhe Z Jin. An adapting auditory-motor feedback loop can contribute to generating vocal repetition. *PLoS computational biology*, 11(10):e1004471, 2015.
- [13] Dezhe Z Jin. Generating variable birdsong syllable sequences with branching chain networks in avian premotor nucleus hvc. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 80(5):051902, 2009.
- [14] Jeffrey E Markowitz, Elizabeth Ivie, Laura Kligler, and Timothy J Gardner. Long-range order in canary song. *PLoS computational biology*, 9(5):e1003052, 2013.
- [15] Takashi Morita, Hiroki Koda, Kazuo Okanoya, and Ryosuke O Tachibana. Measuring context dependency in birdsong using artificial neural networks. *PLoS computational biology*, 17(12):e1009707, 2021.
- [16] Jakob Gübel, Aneesh PH Bose, and Alex Jordan. Social and spatial conflict drive resident aggression toward outsiders in a group-living fish. *Behavioral Ecology*, 32(5):826–834, 2021.
- [17] Carsten Schradin and Jürg Lamprecht. Female-biased immigration and male peace-keeping in groups of the shell-dwelling cichlid fish *neolamprologus multifasciatus*. *Behavioral Ecology and Sociobiology*, 48(3):236–242, 2000.
- [18] Kazuo Okanoya. The bengalese finch: a window on the behavioral neurobiology of birdsong syntax. *Annals of the New York Academy of Sciences*, 1016(1):724–735, 2004.

- [19] Evangeline M Rose, Nora H Prior, and Gregory F Ball. The singing question: re-conceptualizing birdsong. *Biological Reviews*, 97(1):326–342, 2022.
- [20] Allison J Doupe and Patricia K Kuhl. Birdsong and human speech: common themes and mechanisms. *Annual review of neuroscience*, 22(1):567–631, 1999.
- [21] Avani Koparkar, Timothy L Warren, Jonathan D Charlesworth, Sooyoon Shin, Michael S Brainard, and Lena Veit. Lesions in a songbird vocal circuit increase variability in song syntax. *Elife*, 13:RP93272, 2024.
- [22] Toshitaka N Suzuki, David Wheatcroft, and Michael Griesser. Experimental evidence for compositional syntax in bird calls. *Nature communications*, 7(1):10986, 2016.
- [23] Ambre Salis, Jean-Paul Léna, and Thierry Lengagne. Both learning and syntax recognition are used by great tits when answering to mobbing calls. *Behavioral Ecology*, 34(6):941–949, 2023.
- [24] Toshitaka N Suzuki, David Wheatcroft, and Michael Griesser. Wild birds use an ordering rule to decode novel call sequences. *Current Biology*, 27(15):2331–2336, 2017.
- [25] Giacomo Costalunga, Carolina Sánchez Carpena, Susanne Seltmann, Jonathan I Benichov, and Daniela Vallentin. Wild nightingales flexibly match whistle pitch in real time. *Current Biology*, 33(15):3169–3178, 2023.
- [26] Kentaro Abe and Dai Watanabe. Songbirds possess the spontaneous ability to discriminate syntactic rules. *Nature neuroscience*, 14(8):1067–1074, 2011.
- [27] Trisha Butkowski, Wei Yan, Aaron M Gray, Rongfeng Cui, Machteld N Verzijden, and Gil G Rosenthal. Automated interactive video playback for studies of animal communication. *Journal of visualized experiments: JoVE*, (48):2374, 2011.
- [28] Laura Chouinard-Thuly. *Public information use in ninespine sticklebacks (Pungitius pungitius): Isolating the mechanisms using computer-animated stimuli*. PhD thesis, The University of St Andrews, 2012.
- [29] Laura Chouinard-Thuly, Stefanie Gierszewski, Gil G Rosenthal, Simon M Reader, Guillaume Rieucau, Kevin L Woo, Robert Gerlai, Cynthia Tedore, Spencer J Ingle, John R Stowers, et al. Technical and conceptual

- considerations for using animated stimuli in studies of animal behavior. *Current zoology*, 63(1):5–19, 2017.
- [30] Klaudia Witte, Stefanie Gierszewski, Laura Chouinard-Thuly, and Guest Editors. Virtual is the new reality, 2017.
 - [31] Søren Asmussen. *Applied probability and queues*. Springer, 2003.
 - [32] Tim Sainburg, Brad Theilman, Marvin Thielk, and Timothy Q Gentner. Parallels in the sequential organization of birdsong and human speech. *Nature communications*, 10(1):3636, 2019.
 - [33] William A Searcy, Jill Soha, Susan Peters, and Stephen Nowicki. Long-distance dependencies in birdsong syntax. *Proceedings of the Royal Society B: Biological Sciences*, 289(1967):20212473, 2022.
 - [34] Gordon J Berman, William Bialek, and Joshua W Shaevitz. Predictability and hierarchy in drosophila behavior. *Proceedings of the National Academy of Sciences*, 113(42):11943–11948, 2016.
 - [35] Richard Dawkins. Hierarchical organisation: a candidate principle for ethology. 1976.
 - [36] SIMON HA. The architecture of complexity. *Proc Am Philos Soc*, 106:467–482, 1962.
 - [37] Priya Binwal and Lena Veit. Syllable repetition reveals links between distant phrases in birdsong. *bioRxiv*, pages 2026–05, 2026.
 - [38] David Galles. Trie (prefix tree). Data Structure Visualizations, University of San Francisco, 2011. Accessed: 2026-07-02.
 - [39] Hannah Kniebeler. Activation of the social decision-making network in the shell-dwelling cichlid *lamprologus ocellatus* in response to digital avatars. Master’s thesis, University of Konstanz, 2025.
 - [40] Adam R Fishbein, Nora H Prior, Jane A Brown, Gregory F Ball, and Robert J Dooling. Discrimination of natural acoustic variation in vocal signals. *Scientific Reports*, 11(1):916, 2021.
 - [41] Michelle J Spierings and Carel Ten Cate. Budgerigars and zebra finches differ in how they generalize in an artificial grammar learning experiment. *Proceedings of the National Academy of Sciences*, 113(27):E3977–E3984, 2016.

- [42] Julie E Elie and Frédéric E Theunissen. Zebra finches identify individuals using vocal signatures unique to each call type. *Nature communications*, 9(1):4026, 2018.
- [43] Marlen Fröhlich, Gerhard Jäger, and Asya Achimova. Rethinking ambiguity across species. *Neuroscience & Biobehavioral Reviews*, page 106401, 2025.
- [44] Jeffery L Dunning, Santosh Pant, Aaron Bass, Zachary Coburn, and Jonathan F Prather. Mate choice in adult female bengalese finches: females express consistent preferences for individual males and prefer female-directed song performances. *PLoS One*, 9(2):e89438, 2014.
- [45] George EP Box and Norman R Draper. *Empirical model-building and response surfaces*. John Wiley & Sons, 1987.
- [46] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2026.

Appendix A

Sanity Checks for Prefix Extraction

To check whether string alignment influenced regularity detection, the run-collapsed Bengalese finch song corpora were transformed in several ways and run through the analysis pipeline again. The following ribbon plots depict versions of corpora that were: a) fully reversed (Figure A.1, Table A.1); b) character randomized by string (Figure A.2, Table A.2); c) rotated by cropping the first ten characters of each string and appending them to the end (Figure A.3, Table A.3); or d) rotated by cropping up to the third occurrence of a commonly occurring syllable in each string and appending that segment to the end (Figure A.4, Table A.4).

All ribbon plots were cropped at the Q3 of string length. Tries were collected with a maximum of 40 characters, and regularities were identified as entries occurring at least 5% of the time in each corpus.

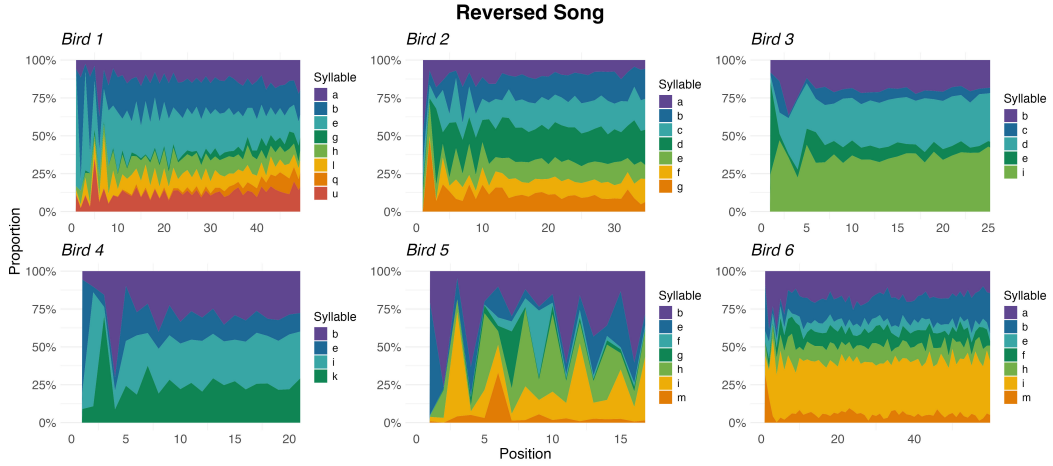


Figure A.1: Ribbon plot of reversed finch song sequences.

Table A.1: *Extracted families of reversed sequences.* Sequence and the proportion of strings containing that sequence (Prop.) are listed by bird and coded by color. Darker shades indicate divergences into sub-families.

Bird	Sequence	Prop.
Bird 2	agdfdcbeagdfdc	0.07
	beagdfdcbe	0.09
	beagdfdcbeagdfdc	0.07
Bird 3	bdi	0.07
	ecibdi	0.12
	ecibdibd	0.06
	eibdibdi	0.06
	i	0.24
	ibdi	0.10
	idi	0.06

Continued on next page

Table A.1 continued from previous page

Bird	Sequence	Prop.
Bird 4	eikb	0.66
	eikbeik	0.28
	eikbeikb	0.28
	eikbeikbi	0.13
	eikbi	0.20
	eikbikb	0.08
	eikbkb	0.14
	eikbkbeikb	0.11
	i	0.11
Bird 5	bheb	0.13
	bhebi	0.12
	eb	0.74
	ebi	0.68
	ebibheb	0.11
	ebibhebi	0.10
	ebibhmghfhib	0.19
	ebibhmghfhibi	0.18
	ebibhmghfhibibheb	0.10
	ebibhmghfhibibhebi	0.09
	ebibhmghfhibieb	0.06
	ebibhmghfhibiebi	0.06
	ebieb	0.12
	ebiebi	0.10
Bird 6	beih	0.07
	maiaiaihfib	0.10

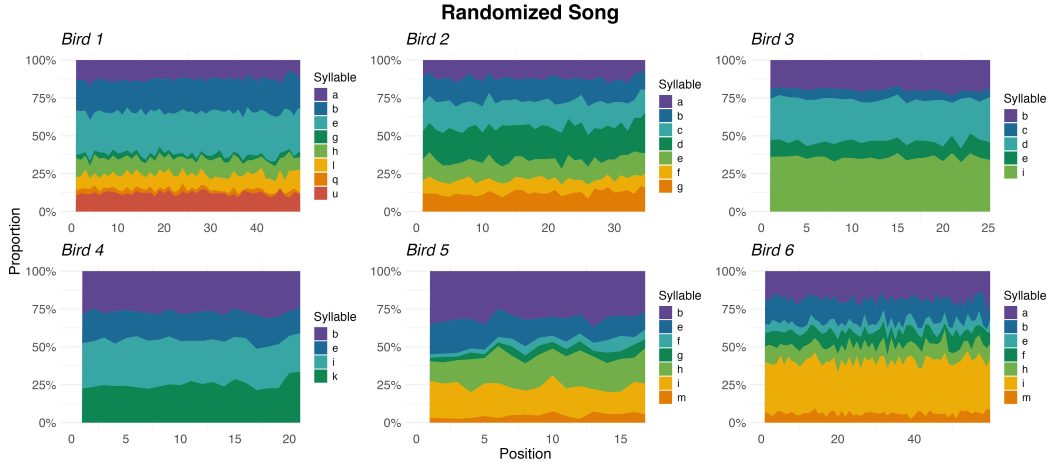


Figure A.2: Ribbon plot of randomized finch song sequences.

Table A.2: *Extracted families of randomized sequences.* Sequence and the proportion of strings containing that sequence (Prop.) are listed by bird and coded by color. Darker shades indicate divergences into sub-families.

Bird	Sequence	Prop.
Bird 3	i	0.36
	iid	0.05
Bird 4	i	0.30
	ib	0.08
Bird 5	bei	0.06
	bie	0.07
	eb	0.10
	ebi	0.05
	ibe	0.07
	ieb	0.06

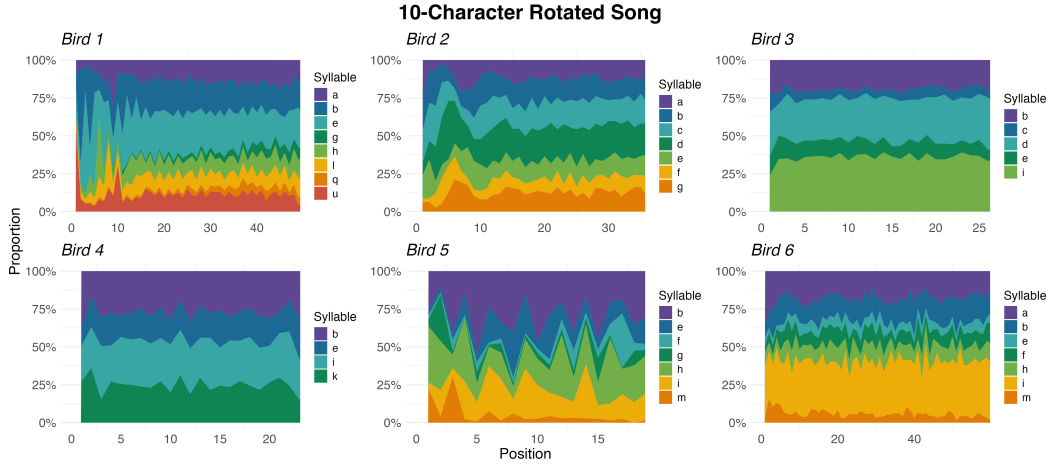


Figure A.3: Ribbon plot of 10-character rotated finch song sequences.

Table A.3: *Extracted families of 10-character rotated sequences.* Sequence and the proportion of strings containing that sequence (Prop.) are listed by bird and coded by color. Darker shades indicate divergences into sub-families.

Bird	Sequence	Prop.
Bird 5	hgmhbibeibehbibhf	0.16
	mhbibeibeibhfhg	0.08

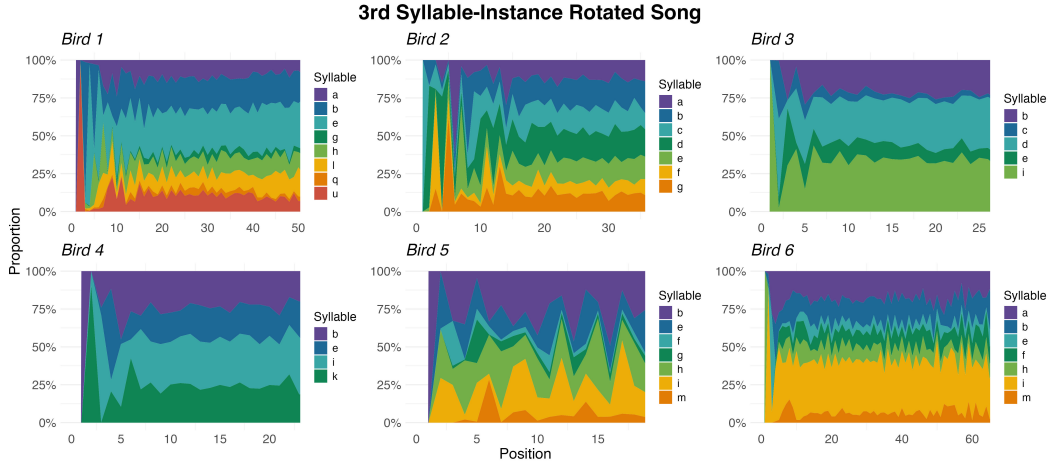


Figure A.4: Ribbon plot of finch song sequences rotated by the third instance of a common syllable.

Table A.4: *Extracted families of sequences rotated by the 3rd instance of a common syllable.* Sequence and the proportion of strings containing that sequence (Prop.) are listed by bird and coded by color. Darker shades indicate divergences into sub-families.

Bird	Sequence	Prop.
Bird 3	iceidbid	0.06
Bird 4	bkiebkiebk	0.09
	bkiebkiebki	0.05
	bkiebkieibki	0.06
	bkieibkiebk	0.10
	bkieibkiebki	0.07
	bkieibkiebkie	0.05
Bird 5	behbibehbi	0.05
	beibehbi	0.15
	bhfhgmhbibeibehbi	0.13
	bibeibeibhfhgmh	0.07

Appendix B

Generated and Empirical Sequence Comparisons

Figure B.1 presents examples of strings from each empirical and generated bird corpus ¹.

¹Cichlid examples were not included for print because the full sequences are thousands of characters long. Please refer to the linked repository for full sequences.

Figure B.1: *Examples of empirical and generated songs for each bird.*

[illegible]

Bird 4

Empirical strings

iiiiiiiibbbkkkiiiiiiiiiieeeeeeeebkkiiiiieeeeeebbkbkkieeee
 iiiiiiibbbbkkkiiiiiiiiiieeeeeeeebkkkkiiiiiiiieeeeeebbbkbbkiiiiiiiiee
 iiiiiiibbbbkkkiiiiiiiiiieeeeeeeebkkkkiiiiiiiieeeeeebbbkbbkiiiiiiiiee
 iiiiiiibbbbkkkiiiiiiiiiieeeeeeeebkkkkiiiiiiiieeeeeebbbkbbkiiiiiiiiee
 iiiiiiibbbbkkkiiiiiiiiiieeeeeeeebkkkkiiiiiiiieeeeeebbbkbbkiiiiiiiiee

Generated strings

iiiiiiiiibkiiiiiiiiiiiieeeeeeeebkkiiiiiiiiiiiieeeeeeebbkibebkkbiiiiiiikbiiiiiiiieeeeeeeeeeeeeeeee

iiiiiiiiibbbkbbkiiiiiiiiiiiieeeeeeeebkiiiiiiiiiiiieekieeeeeeeebkiiiiiiiieeeee

iiiiiiiiibbbkbbkiiiiiiiiiiiieeeeeeeebbbkbbkiiiiiiiiiiiieeeeeeeebbbkbbkibeeeeeeeeeeebkbbkiiiiiiiie

Bird 5

Empirical strings

[illegible]

Generated strings

iiiiiiiiiibbbbbbmbbbbbbbbeeeeeeeiiiiiiibbbbbbhbffrfffghmbbbbbeeeeeeee

Bird 6

Empirical strings

[illegible]

Generated strings

iibbbbbbbambiiiiieeeeeabiiiiifhhhhhiiiaiammbbbbffhhhhhnieabbobbbiffhhhhhhihhhhieeiffhhhhh

Appendix C

Code and Software

A GitHub repository for all analyses and datasets included in this thesis can be found at: <https://github.com/aalbrechtbuehler/fish-and-finches.git>

Data-wrangling, prefix tries, plots, and post-processing of modeling results were conducted using R version 4.6.0 [46] with the packages: *tidyverse*, *readr*, *tidyr*, *dplyr*, *paletteer*, *strs*, *stringi*, *stringr*, *purrr*, *xtable*, *patchwork*, *ggplot2*, *papaja*, *tidybins*, *phileentropy*, and *latex2exp*. Prefix tries were compiled using the *reasytries* package.

Markovian modeling and heatmap generation were conducted in Python 3.14.4 with the packages: *pandas*, *numpy*, *scipy*, *seaborn*, *matplotlib*, and *scikit-learn*. A full list of dependencies is available in the repository.

All random seeds were set at the value 123.