Parameter-Free Audio Motif Discovery in Large Data Archives

Yuan Hao, Mohammad Shokoohi-Yekta, George Papageorgiou, Eamonn Keogh
University of California, Riverside
{yhao, moshok002, gpapag, eamonn}@cs.ucr.edu

***Abstract*—The discovery of repeated structure, i.e. motifs/near-duplicates, is often the first step in exploratory data mining. As such, the last decade has seen extensive research efforts in motif discovery algorithms for text, DNA, time series, protein sequences, graphs, images, and video. Surprisingly, there has been less attention devoted to finding repeated patterns in audio sequences, in spite of their ubiquity in science and entertainment. While there is significant work for the special case of motifs in music, virtually all this work makes many assumptions about data (often to the point of being genre specific) and thus these algorithms do not generalize to audio sequences containing animal vocalizations, industrial processes, or a host of other domains that we may wish to explore.**

In this work we introduce a novel technique for finding audio motifs. Our method does not require any domain-specific tuning and is essentially parameter-free. We demonstrate our algorithm on very diverse domains, finding audio motifs in laboratory mice vocalizations, wild animal sounds, music, and human speech. Our experiments demonstrate that our ideas are *effective* in discovering objectively correct or subjectively plausible motifs. Moreover, we show our novel probabilistic early abandoning approach is *efficient*, being two to three orders of magnitude faster than brute- force search, and thus faster than real-time for most problems.

Keywords-audio motif; spectrogram; anytime algorithm

I. INTRODUCTION

The first step in most exploratory data mining endeavors is the discovery and enumeration of *repeated structure*. This has been true even for data analysis that predates computers. For example, the decipherment of documents written in ancient unknown languages first requires the discovery of *repeated* elements in the scripts [7]. Given this, there has been significant research effort in the last decade focused on repeated pattern (motif/near-duplicate) discovery in text, DNA, graphs, time series, images, and video [20][25][27]. In contrast, the discovery of *audio* motifs, with the sole exception of music data, has not received much attention [21]. However, identifying structure in general audio sequences is an important and challenging task with applications in many diverse domains. Some representative examples include:

- Acoustic wildlife monitoring has been shown to allow effective and non-invasive measurement of the health of ecosystems [36].
- A powerful tool for investigating the role of genetics in human disorders modifies ("knocks out") various genes in mice and examines their vocalizations for changes that may be linked to those genes, and hence the analogue genes in humans [35][40]. Figure 1 hints at the utility of this idea, which we revisit in Section V.D. In recent years this framework has emerged as an extremely promising tool for understanding human cognitive and memory disorders.
- Audio content analysis has been shown to assist with *video* segmentation and summarization [20][25][33].

The above are in addition to the more obvious applications in the music domain, such as analysis, thumbnailing, retrieval, and summarization [3].

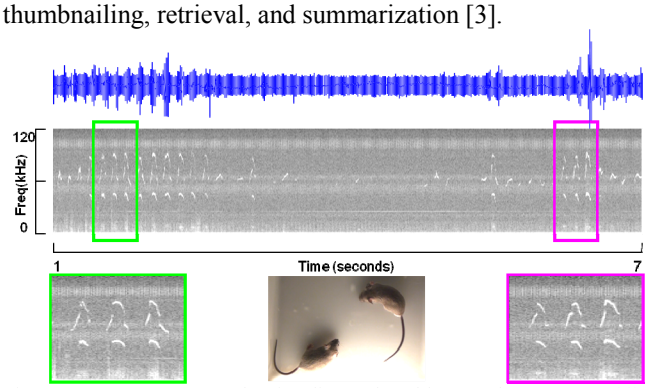

Figure 1. *top*) Seven seconds of audio produced by a male mouse. *middle*) We searched the spectrogram of this data for repeated patterns of length 0.5 seconds. *bottom left*, *right*) A zoom-in of the two repeated occurrences reveals their similarity. We will revisit this domain in Section 5.4.

Thus far virtually all research efforts aimed at finding repeated patterns in audio sequences use feature extraction algorithms to produce low cardinality symbolic representations of the data and use suffix trees, hashing, or similar techniques to search these symbolic strings for approximately repeated elements [3]. The problem with this approach is that the feature extraction step must be highly tuned to the domain. For example, Zakaria et al. [40] demonstrate a technique to find motifs in vocalizations of a specific strain of lab mice called *Fmr1-KO*. However, it is not clear if this multi-stage algorithm (which requires significant human intervention) generalizes to other strains of mice, much less to other rodents.

In contrast to these efforts, we propose an algorithm which is completely general, makes essentially zero assumptions about the data, and is essentially parameter-free. We achieve this by leveraging off the growing realization that for at least some audio similarity problems, we can best measure similarity when the data is transformed into the *image space* (i.e. spectrograms) [18][27]. Image processing algorithms themselves are not generally devoid of the need for feature extraction. However, we propose to use the CK distance measure [8], a recently introduced compression-based measure that avoids explicitly extracting any features, thus remains parameter-free. We will show that the CK distance measure is so efficient that even a brute-force implementation can run in about real-time for a typical song.

For longer audio sequences we introduce two ideas to mitigate the time complexity. First, we show that we can cast the search for audio motifs into an anytime framework [28]. Second, we can derive confidence bounds that allow searches to return the optimal audio motifs with some bounded probability of error. As we shall show, even if we allow a very conservative probability of error, we can achieve a massive speedup.

The rest of the paper is organized as follows. In Section II we review related work. In Section III we introduce the necessary notation to formalize our algorithm in Section IV. Section V sees a detailed empirical evaluation of our ideas on diverse domains, and we offer conclusions and directions for future work in Section VI.

II. RELATED WORK

By far the most common approach to finding repeated patterns in audio is to "*use string-matching techniques on a symbolic representation learned from the data*" [3]. Given a high quality symbolic representation of the data, the problem becomes much simpler; we can just use an off-the-shelf symbolic repeated pattern discovery tool. This approach has been used in music [3] and in mice vocalizations [40]. However, it is obvious that the symbol extraction algorithms used for pop songs are unlikely to generalize to classical music, much less mice or insects [40]. Likewise, it is not clear that the symbol extraction method discussed in [40] will generalize to other strains of mice, much less other mammals. It is difficult to overstate how poorly existing audio motifs discovery algorithms can be expected to generalize. For example, [34] introduces an algorithm that is specialized for just Hindustani vocal music compositions.

There is, however, research work in the speech recognition community that is very close in spirit to our work. For example, in a recent but highly cited paper, the authors ask how well we can do in finding repeated speech elements with "*zero resources*" [23]. By zero resources they mean that they assume no "*models or training data for the target language*." Note, however, that even here researchers assume human speech. We would like to remove even that assumption, and have a completely unsupervised, parameter-free, and zero resource algorithm that can detect repeated sounds in sources as diverse as human speech, wildlife surveillance, music, and industrial applications.

It is important to recognize that the framing of our problem precludes many *apparent* solutions. For example, there is significant work in very fast audio search for commercial music applications. Such work is often called audio thumbnailing or audio fingerprinting [9]. One might imagine that such techniques and representations could be adapted to the task at hand. However, most such methods assume that there is a "platonic ideal" sound snippet, say a master recording of a song. The instances matching this idealized template might not be bit-for-bit identical due to different encodings, or in the case of Shazam/SoundHound[1], corruption by background noise as the user records the music with a mobile device. Nevertheless, the problem reduces to matching two objects that are essentially identical, except that one has minor noise/distortions. Most critically, the two snippets are assumed not to have *time warping*.

In contrast, we consider the more general case where the two similar sounds are different *physical* (not digital) instantiations of a "process". Here the process could be two bird calls, two belt slip screeches from an overloaded industrial machine [41], or two repetitions in a mouse's song (cf. Figure 1). Thus, we are interested in finding repetitions in the face of a much broader set of noise and distortions, including time scaling (*global* shrinking/stretching), *local* time warping, pitch shifts, and echos, etc.

The most important difference between our work and previous audio motif discovery approaches is that our algorithm finds repeated objects in audio sequences without making *any* assumptions about the intrinsic properties of the objects. For example, we did not need domain knowledge of rodent physiology to find the motifs discovered in the mice vocalizations shown in Figure 1 [40]. For the task of *music* motif discovery, researchers have considered a huge number of possible features, including static music information (key, beat, and tempo), acoustic information (loudness, duration, pitch, bandwidth, and brightness), thematic features (melodies, rhythms, and chords), and higher-level composite features (i.e. hierarchical rules, Markov models) [21]. These features may be helpful for motif discovery, but they require a huge amount of feature engineering and there is evidence that they do not generalize across music genres [34], much less generalize to the diverse domains we consider.

Non human-produced sounds offer no fewer difficulties. For example, in [14], researchers attempt to find repeated patterns in bird songs. Their algorithm requires extracting features from syllables, and the authors bemoan the effort of human intervention: "*Syllable templates were formed by aligning and averaging four to five manually chosen clips corresponding to each syllable…*," "*…manually chosen based on visual inspection*," etc. It is exactly this kind of manual tweaking that we wish to avoid.

Our algorithm leverages off the idea of analyzing sounds directly in the image space (i.e. spectrograms). This idea has been increasing in popularity recently [27][40]. For example, [27] analyzes music data by computer vision techniques; however, current work is limited to query-by-content, not motif discovery, and is explicitly specialized to music data.

III. NOTATION

Before describing our audio motif discovery algorithm, we provide the necessary definitions.

We are interested in mining audio sequences:

***Definition* 1**: An *audio sequence* A of length $m > 0$ is a sequence $\mathbf{A} = (A_1, A_2, \ldots, A_m)$ of m real-valued numbers corresponding to the amplitude at that time stamp.

Inspired by recent work [4][27], we plan to leverage off several advantages of analyzing *audio* sequences in a *visual* representation, called a *spectrogram*.

***Definition* 2**: A *sound spectrogram* **S** is an image of time-varying spectral representation, produced by applying the Short Fast Fourier Transform to successive overlapping frames of an *audio sequence*. The horizontal dimension

[1] Shazam/SoundHound are commercial mobile phone based music identification services. A cell phone's built-in microphone is used to gather a brief sample of music being played. An acoustic fingerprint is created based on the sample, and is compared against a central database for a match.

corresponds to time and the vertical dimension corresponds to frequency. The relative spectral intensity of a *sound* at any specific time and frequency is indicated by the color/grayscale intensity of the *image*.

We have already encountered an example of a spectrogram in Figure 1. A more detailed discussion of spectrograms is beyond the scope of this paper, so we refer the reader to [5] and the references therein.

We are not interested in the *global* properties of a *sound spectrogram*, because any repeated patterns are typically only manifest in small *local subsequences*:

***Definition* 3**: An *audio subsequence* of length n of an *audio* sequence $\mathbf{A} = (A_1, A_2, \ldots, A_m)$ is a time series $\mathbf{A}_{i,n} = (A_i, A_{i+1}, \ldots, A_{i+n-1})$ for all integers i, where $0<i<m-n+1$.

Informally, audio motifs are the most *similar* subsequences within a longer audio sequence. Thus, we must compute similarity with some measure of *distance*:

Definition* 4**: The *distance* between a subset of $\mathbf{S}$, comprised of $\mathbf{S}_{i,n}$ and another subsequence $\mathbf{S}_{j,n}$ is the CK distance [8], denoted ***dist$(\mathbf{S}_{i,n}, \mathbf{S}_{j,n})$.

The CK distance measure is a relatively new, compression-based similarity measure, which exploits MPEG video encoding to measure the similarity between *real-valued* images [8]. The distance between two equal-sized images (denoted as **x** and **y**) is calculated as:

dist(**x**,**y**)=((mpegSize(**x**,**y**)+mpegSize(**y**,**x**))/(mpegSize(**x**,**x**)+mpegSize(**y**,**y**)))-1

In Section IV, we will explain and justify the choice of this particular distance function [8].

We are finally in a position to define audio motifs more formally. To find the audio motif pair of (a *user* given) length w in a long audio sequence, we consider the pair-wise distances between each subsequence and all others. The pair with the smallest distance is the *audio motif:*

***Definition* 5**: The *audio motif* of an *audio sequence* is the unordered pair of subsequences $\{\mathbf{A}_{i,n}, \mathbf{A}_{j,n}\}$ of a long audio sequence $\mathbf{A}$ of length n that is the most similar. More formally, $\exists i,j\ \forall a,b$, the pair $\{\mathbf{A}_{i,n}, \mathbf{A}_{j,n}\}$ is the *audio motif* iff $dist(A_{i,n}, A_{j,n}) \leq dist(A_{a,n}, A_{b,n})$, $|i-j| \geq w$ and $|a-b| \geq w$ $(i \neq j, a \neq b)$for $w>0$, where w is the audio motif length.

$$\{(A_{i^*,n}, A_{j^*,n})\} = \underset{\substack{(A_{i,n}, A_{j,n}) \in A \\ |i-j| \geq w}}{\arg\min}\ dist(A_{i,n}, A_{j,n}) \qquad (1)$$

Note that we refer to *audio motifs* even though we are searching the image space (the spectrograms). In Figure 2, we illustrate an example of an audio motif pair (leftmost and rightmost boxes) together with the other concepts introduced in this section.

Note that our audio motif definition excludes trivial matches of *audio subsequences* that match a part of a sound with itself, such that $i=j$ or $|i-j|< w$. Thus the motif pair must be strictly non-overlapping. Our experience analyzing real world audio has shown us that sections of pure silence (which we denote as $\mathbf{S}_{ps}$, shown as pure black section in Figure 2) are quite common in scientific data [32]. These silent elements (not to be confused with simply *quiet* but non-zero time periods) may be caused by disconnected wires or data deliberately written with zero energy to denote the beginning/ending of an event. These sections can be problematic since all silences "sound" the same, and thus allow for perfect yet meaningless audio motifs.

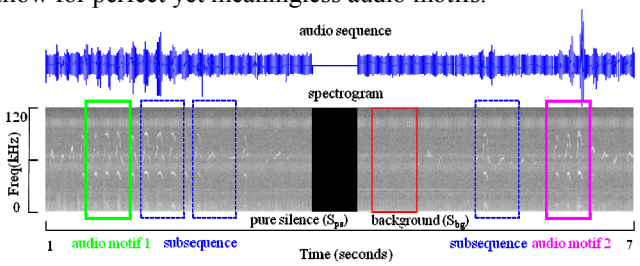

Figure 2. An illustration of our definitions.

Another special case we have to consider is that of a constant background sound (denoted as $\mathbf{S}_{bg}$, and illustrated in the area surrounded by the red box in Figure 2), which occurs *everywhere* in audio sequence. For example, if we are interested in finding audio motifs near a highway on a rainy day, then the *entire* background will have the sound of rain, which we would like to ignore. We exclude both pure silence and regions containing *only* the background sound if they last more than one motif length w.

It is important to note that our definition of *closest pair* does not preclude other possible definitions. For example, for some applications it might be convenient to consider the K closest pairs, or *all* objects within a user-given radius R. As noted in [31] in the context of time series motifs, if one can solve the closest pair problem efficiently, then the K closest pairs and user-given radius variants can also be solved using the *closest pair* subroutine with some linear time post processing. In particular, we have explored finding the top K motifs (for K equals up to 10) in the birds dataset discussed in Section V.B; this required just a few minutes modifying the code, and took less than twice as long as finding the *closest pair* (or $K = 2$). Nevertheless for clarity of presentation and consistency we limit discussion and experiments to the *closest pair* case in this work.

IV. Finding Audio Motifs

We outline a detailed formal explanation and statement of our audio motif discovery algorithm in Section IV.B. However, for simplicity and clarity, we give some simple intuitions behind our ideas in the next section.

A. Intuition behind Audio Motif Discovery

Our entire approach is predicated on the following assumption. Similar sounds will produce similar images when transformed into spectrograms, and we can efficiently and effectively compute the similarity in the *visual space*. The idea that audio patterns can be revealed and measured in the image space has been exploited in some specialized domains [4][27][40]. However, these works require domain-specific feature extraction techniques to allow the similarity computation, a step we are anxious to avoid in order to create a universal and highly usable tool.

To be clear, using the spectrogram representation is, by itself, *not* the solution to our problem. To see this, we performed a simple clustering experiment using Scale-

Invariant Transform Features (SIFT) [29][37] on a small dataset. SIFT is arguably the state-of-art for image matching, and the most obvious strawman to compare against [22].

In Figure 3, we show seven pairs of two-second audio snippets of diverse sounds produced by *coyotes*, *crickets*, *squirrels*, *katydids*, *ravens*, *owls*, and *explosions*. For all images we extract SIFT features to form a feature description using Lowe's algorithm [29]. We use the number of matched keypoints/features points as a similarity measure between two images[2]. Figure 3 shows the clustering of the seven pairs using SIFT; the results are only slightly better than random.

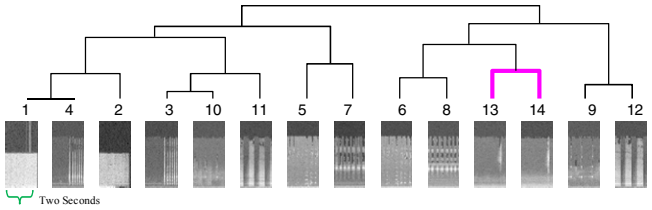

Figure 3. A clustering of seven pairs of two-second audio recordings of various sounds using SIFT. Only one pair {13,14} is correctly clustered (*katydids*).

These results are not promising. In contrast, we tested the same dataset shown in Figure 3 using CK distance measure, and the result is shown in Figure 4.

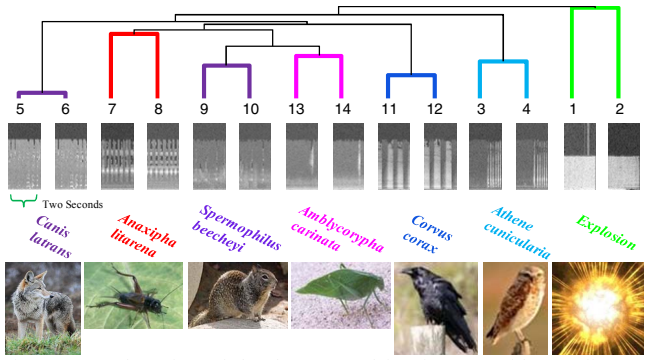

Figure 4. A clustering of the dataset used in Figure 3 with CK distance measure. All pairs are correctly clustered, and the explosion sound is an outlier to animal sounds. No parameters were adjusted here.

This result highly suggests that the CK distance measure on spectrogram images is measuring similarity in a way that maps to human notions of sound similarity.

Surprisingly, the CK distance measure can be very effective even on *human speech*, which is obviously the most studied audio source [9][17][23]. To see this, in Figure 5, we show a reading of *A Dream within a Dream* by Edgar Allan Poe. We naturally expect repeated structure in most poetry [13], and although this short poem only has 24 lines in two stanzas, we do find two obvious repetitions as the audio motif (the last line of both verses).

[2] This (carefully annotated) code, along with *all* code and data used in this work is archived at [19].

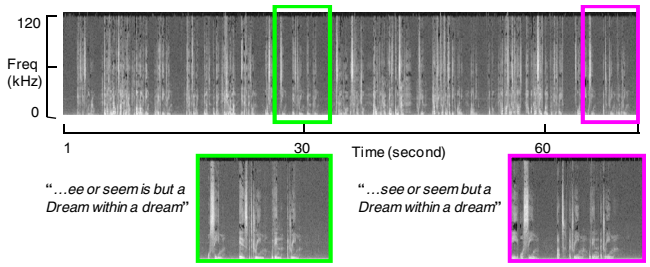

Figure 5. *top*) A performance of *A Dream Within A Dream* has a motif of length seven seconds. *bottom left*, *right*) A zoom-in of the two occurrences and the corresponding sentences. The reader can go to [19] to hear the original sound file and the discovered motifs.

As the audio motif shown in Figure 5 demonstrates, the CK distance measure can accurately find the *only* repeated pattern ("…*see or seem* (*is*) *but a dream within a dream*") of this audio recording. Note that the distance measure was *exactly* the same as used in Figure 4, no tuning or adjustments were necessary to go from (mostly) animals to poetry.

In this example, generating and testing *all* possible motifs to find the best one (cf. Definition 5) requires about 15 minutes, although the audio clip is barely a minute long. Such languor may be tenable for music and other relatively short audio files; however, in scientific domains we need to be able to find motifs in datasets that are orders of magnitude longer. In the next section, we will outline our strategy for making motif discovery tenable even in such massive datasets.

B. *A Formal Description of our Algorithm*

Given a spectrogram S transformed from a long *audio sequence* A, and a user specified length *w*, our goal is to find audio motifs as described in Definition 5.

For ease of exposition we will begin by explaining the generic search algorithm, and then we will then introduce our novel modifications that make it more tractable.

In TABLE I. Line 1, our algorithm begins by initializing the *best-so-far* CK distance corresponding to the audio motif pair to infinity. In Line 2 we generate all N subsections D. This consists of *all* subsections except those excluded because they are $\mathbf{S_{ps}}$ or $\mathbf{S_{bg}}$ (cf. Section III). One pair (with indices differing by at least w) from this set will eventually become the motif pair.

In Line 4 we are finally in a position to test the $N(N\text{-}1)/2$ possible pairings of subsections for the pair that minimizes the CK distance, our audio motif. However, in what order should we search? Clearly, if we search exhaustively then the order makes no difference. However, there are two reasons why we may want to avoid exhaustive search and terminate early. The first is to respond to a user request to stop, so the user can treat the algorithm as an *Anytime* search algorithm [1]. The other reason is that we may wish to frame the search probabilistically, supporting a user request of the type "*stop searching when there is only a one in a thousand chance that the current best-so-far is not the true motif.*"

As we will shortly show, we can support these useful variants by using different heuristic functions as defined in Sections 4.3, 4.4, 4.5 and 4.6.

TABLE I. GENERIC AUDIO MOTIF DISCOVERY

Procedure *AudioMotif_Discovery*(***S***, *w*, *p*)	
Input: spectrogram ***S*** transformed from original audio archive A; Audio motif length *w*; Early abandoning probability threshold *p*;	
Output: Audio motif pair *D*;	
1	*best-so-far* ← Inf
2	Discard S_{ps} and S_{bg} area from S and generate only meaningful subsections into *D*. (i.e. no *silence* and no *constant* background sounds)
3	$N \leftarrow |D|$
4	**for** *loopCt* ← 1 to $N(N\text{-}1)/2$ **do**
5	[*i*, *j*, *stopFlag*] ← heuristicFunction(*loopCt*, *type*, *p*, *best-so-far*)
6	*distance* ← dist(D_i, D_j)
7	**if** *distance* < *best-so-far* **and** $|i\text{-}j| \geq w$ **then**
8	*best-so-far* ← *distance*
9	$Pos_1 \leftarrow i$
10	$Pos_2 \leftarrow j$
11	**end if**
12	**if** *stopFlag* == **True then**
13	Break out of the loop
14	**end if**
15	**end for**
16	**Return** D_{Pos1}, D_{Pos2}

Returning for a moment to the generic version of our algorithm, in Line 6 we measure the CK distance between the two candidate subsections and in Line 7 we check to see if this pair of *audio subsequences* has a smaller distance than the current *best-so-far* distance. If that is the case, we update the *best-so-far* distance, and record the relevant locations (Lines 8 to 10).

Having seen the generic algorithm we now consider four variants produced by using different heuristic functions and stopping criteria.

C. Brute-force Algorithm

The brute-force heuristic is outlined in TABLE II. This heuristic simply lists every possible combination of pairs of audio subsequences {Pos_1, Pos_2} in lexical order and allows search to exhaustion. Note the last two arguments are just place-keeping dummy variables.

TABLE II. BRUTE-FORCE SEARCH

Procedure *heuristicFunction*(*index*, *bruteForce*, *dummy*, *dummy*)	
1	Generate testing candidates in a lexical order, which is from left to right with the sliding window
2	[idx_i, idx_j, **False**] ← Return an array containing the candidates' positions

Run to completion this heuristic clearly lists the pair of audio subsequences {(D_{Pos1}, D_{Pos2})} that are optimal.

Note that for many real world problems there may be *many* motifs that are of high quality, and finding *any* pair may be sufficient. For example, if a full day's recording in a forest in Kenya has dozens of the stereotypical calls of the *Common Scimitarbill* (cf. Section V.B) then reporting any pair as a motif will suffice for many applications. However, if the recording started at midnight, and the bird is most vocal just before dusk, then the linear-ordered brute-force search will not produce a good motif (i.e. have a low *best-so-far* value) until very late in the search process. As we show in the next section, we can mitigate this with a random ordered search.

D. Random Search Algorithm

In contrast to lexical-ordered search discussed in the previous section, we can consider random ordered search, which has long been used to guard against pathological situations where an iterative improvement algorithm (i.e. *best-so-far* linear search) does not improve much until the last few iterations [26].

The rate at which the *best-so-far* decreases does not matter if we run to completion. But if we allow users to interrupt the search and peek at the best current motif, we expect that random search as shown in TABLE III. to do better, as its *best-so-far* value will converge faster.

TABLE III. RANDOMIZED SEARCH

Procedure *heuristicFunction*(*index*, *Random*, *dummy*, *dummy*)	
1	Generate testing candidate in a *random* or*der, which produced by random permutation*
2	[idx_i, idx_j, **False**] ← Return an array containing the candidates' positions

The idea of supporting interruptions (possibly followed by continuations) of an algorithm is known as creating an *anytime algorithm*, and anytime algorithms have seen a recent surge of interest in the data mining community [1][28]. As we shall empirically show in Section V, random ordering does greatly improve the "*early returns*" property of our search. However, in the next section we show that we can do even better.

E. Euclidean Distance Ordering Algorithm

Anytime algorithms for searching tend to work best if they can test promising solutions early. This seems to open a *chicken-and-egg* type paradox, since we do not know if a pair of subsequences will make a good motif until *after* we test them. However, if we had an approximate test of quality that was much faster than the CK distance itself, then we could sort by that measure and increase our chances of seeing good solutions early on. In the most general case, CK distance measure has resisted attempts at fast approximation [22]. However, it has been shown that in the special case of spectrograms, the Euclidean distance between the images is a reasonable approximation of CK distance measure [18], but can be computed at least three orders of magnitude faster. Moreover, the Euclidean distance computations are amenable to many tried-and-tested speedup techniques including early abandoning, triangular inequality, and indexing.

Given a spectrogram image **S** with size M×N, **S** can be written as $\mathbf{S} = \{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^{MN}\}$ according to the gray levels of each pixel. The Euclidean distance $dist_E(\mathbf{S}_1, \mathbf{S}_2)$ between two images $\mathbf{S}_1$ and $\mathbf{S}_2$ is defined as:

$$dist_E(\mathbf{S}_1, \mathbf{S}_2) = \sum_{k=1}^{MN} (\mathbf{S}_1^k - \mathbf{S}_2^k)^2 = (\mathbf{S}_1 - \mathbf{S}_2)^T (\mathbf{S}_1 - \mathbf{S}_2) \qquad (2)$$

Thus, as shown in TABLE IV. we propose to sort the pairs returned by the heuristic in ascending order (denoted as *Z*) of their Euclidean distance.

TABLE IV. EUCLIDEAN DISTANCE MEASURE ORDER SEARCH

	Procedure *heuristicFunction*(*index*, *ED*, *dummy*, *dummy*)
1	*sortOrder* $\tau \leftarrow$ Sort all the subsection pairs by the *Euclidean distance* between them in an ascending order
2	*stopFlag* $\leftarrow$ DidUserRequestAnInterruption();
3	[idx_i, idx_j, *stopFlag*] $\leftarrow$ Return an array containing the candidates' positions based on τ and *stopFlag*

The high degree of correlation between the Euclidean distance and CK distance is hinted at in Figure 6.*left*.

Before moving on, it is critical to note that while the Euclidean distance and the CK distance are correlated, we cannot simply use the Euclidean distance to directly find motifs. For example, it does not produce the correct answers for the examples shown in Figure 4 and Figure 5 (we shown this in [19]). Nevertheless, as we show in the next section, we can use the Euclidean distance as a heuristic to both guide the search order, and to tell us when we can abandon the search with a small, user-defined probability of missing the optimal answer.

F. *Probabilistic Motif Discovery Algorithm*

As noted above, the anytime algorithm framework is gaining increasing acceptance by both the data mining community and domain practitioners. However, at least some of the latter may be reluctant to use anytime algorithms as intended. Nevertheless, most biologists are much more comfortable with the idea of statistical significance, the idea of considering if a result could be explained by a chance at a given probability cutoff (i.e. the *significance level*). We can support this type of worldview by allowing the user to specify the probability of returning a non optimal motif pair. In essence, we propose to allow queries of the form "*stop searching when there is only a one in a million chance that the current best-so-far is not the true motif*."

By exploiting the Euclidean distance ordering heuristic, introduced in the previous section, we can support such queries. As we shall see later in our experimental section, we can trade a *small* probability of a *slightly* suboptimal result for several orders of magnitude speedup.

The intuition behind the Probabilistic Early Abandoning Audio Motif Discovery (PEAMD) algorithm is to internally estimate the likelihood that the current *best-so-far* motif is optimal, and signal to abandon the search once this likelihood exceeds the user's tolerance for a sub-optimal result. This signal is passed into the generic search algorithm in Line 5 of TABLE I. How can we estimate this probability? Figure 6 gives a visual intuition. In Figure 6.*left* we show the relationship between the Euclidean distance and CK (estimated from the dataset shown in Figure 5).

Let P_d(*best-so-far*) be the probability that the remaining pairs of subsequences in the Euclidean searching order Z (the y-axis ordering of Figure 6.*left*) contains a better match than the match represented by the current *best-so-far*. Given that the two measures are highly correlated, we can estimate P_d(*best-so-far*), which is monotonically non-decreasing as we search because the *best-so-far* can only decrease by definition (i.e. the red *bsf_dist* bar shown in Figure 6.*right* can only move *leftwards*), and the positive correlation means that the mean of the distribution of estimated values of untested pairs can only move *rightwards.*

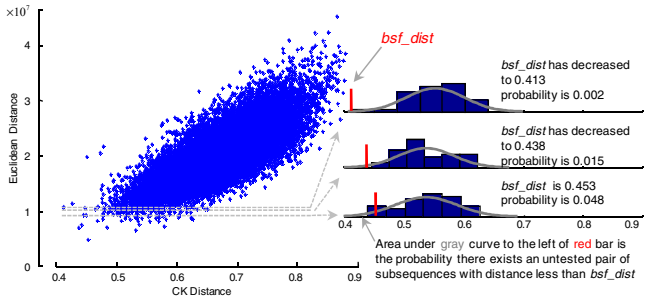

Figure 6. *left*) The empirical relationship between Euclidean and CK distance. *right*) As we search in Euclidean order (the y-axis order), from bottom to top. The *best-so-far* distance moves leftwards and the mean of the Gaussian distribution moves rightwards.

Concretely, we compute d, the CK distance for the ε items below the *lowest* dash-line in Figure 6.*left* to form the histogram shown at the bottom right of Figure 6. Here ε is a small number, enough to learn a Gaussian (we use $\varepsilon = 50$).

$$d_k = dist(Z_{((k-1)\cdot\varepsilon+1:k\cdot\varepsilon,1)}, Z_{((k-1)\cdot\varepsilon+1:k\cdot\varepsilon,2)}) \quad (3)$$

This property of the distance distribution can be realized by a Gaussian process (GP). The probability vector $\{\varphi_k\}$ is drawn from a GP as $\varphi_k \sim N(\mu_k, \sigma_k^2)$, where μ_k is the mean of and variance σ_k^2, shown as the gray "bell" curve. For example, the *best-so-far* distance decreases from 0.453 to 0.413 and the corresponding P_d(*best-so-far*) of the distance distribution changes from 0.048 to 0.002 as shown in Figure 6.*right*. The area below the gray curve, left of the *best-so-far* marker, is the probability that there exists an untested pair of subsections with distance less than the *best-so-far* distance. If P_d(*best-so-far*) is less than the user threshold (denoted as p) then we simply set the *stopFlag* to be true, and the invoking generic search algorithm will terminate. The formal algorithm of PEAMD is outlined in TABLE V.

TABLE V. PROBABILISTIC EARLY ABANDONING SEARCH

	Procedure *heuristicFunction*(*index*, *PEAMD*, ***p***, *best-so-far*)
1	Call the procedure Euclidean Distance Measure Order Search in TABLE IV
2	$Z \leftarrow$ heuristicFunction (*index*, *ED*)
3	**for** $k \leftarrow$ 1 to $\|Z\|$ **do**
4	**for** $j \leftarrow$ 1 to ε **do**
5	$d_k \leftarrow$ Compute CK distance of pair k+j-1 based on Z
6	**end for**
7	$d_k \sim N(\mu, \sigma^2)$ // *Build* Gaussian distribution of d_k
8	*prob* $\leftarrow P_d$(*best-so-far*) // CDF of the current best-so-far distance
9	**if** *prob* < ***p*** **then**
10	*stopFlag* $\leftarrow$ **True**
11	**end if**
12	**end for**
13	[idx_i, idx_j, *stopFlag*] $\leftarrow$Return an array containing the candidates' positions and *stopFlag*

Our probabilistic framework makes some assumptions that are strongly empirically warranted (i.e., that "slices" of the cloud of data points in Figure 6.*left* are approximately Gaussian), and some that are less realistic (i.e. the

independence of the "slices"). However, all such assumptions tend to make our algorithm err *only* on the conservative side.

V. EXPERIMENTS

We have designed all our experiments to ensure that they are *very* easy to reproduce. A supporting webpage [19] contains *all* the code, datasets, and raw data spreadsheets used in this work. Moreover, although this work is completely self-contained, the webpage contains additional experiments and video/sound files to allow an interested reader to directly see and hear the motifs discovered and the original source sounds.

A. Motif Discovery in Human Speech

Human speech is surely the most studied sound source [10]. Recurrences in human speech have implications for studying linguistics, cognitive disorders, and pragmatic applications in indexing speech [24], etc. Thus, we will test our algorithm with a familiar audio book in section V.A.1) and show a comparison with a state-of-the-art speech processing tool in section V.A.2).

1) Motif Discovery in an Audio Book

A famous example of reoccurring text can be found in the book *The Cat in the Hat* by Dr. Seuss [12]. It is an impressive feat of wordplay that this **1629** word book contains only **236** distinct words, and this is suggestive of significant repetition. The experiment shown in Figure 7 demonstrates that our algorithm finds a meaningful motif pair (three seconds long) from an audio performance of story (professional male actor). Note that our algorithm is robust to the fact that one occurrence contains an additional word ("*ball*").

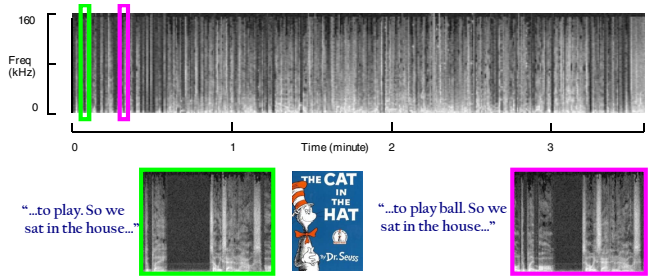

Figure 7. *top*) A performance of *The Cat in the Hat* has a motif three seconds long. *bottom*) A zoom-in of the two occurrences and corresponding sentences [12].

2) Comparison with state-of-the-art Work

The utility of "black-box" CK distance on human speech may be surprising, given that most human speech processing algorithms are highly optimized with domain knowledge of linguistics, phonetics, etc. To further explore this, we attempted to reproduce a result in a recent state-of-the-art work [24]. Here the data is a nine-second snippet of telephone quality audio. We take the same spoken query of word "*California*" as [24], and build the same type of dot-plot, but use the CK distance measure as shown in Figure 8.

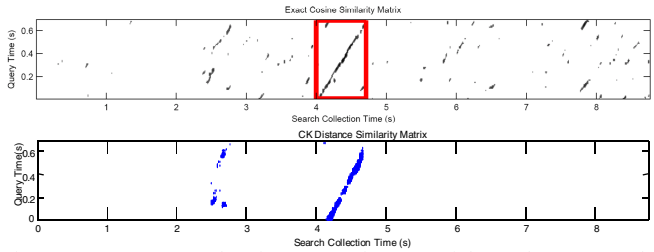

Figure 8. *top*) A screenshot from [24] of a state-of-the-art human speech recognition algorithm correctly matching two utterances of "California" (red box). *bottom*) Our re-creation of the experiment using only the CK distance measure.

While the interpretation of the results is somewhat subjective, our simple approach does seem *at least* competitive with current human speech processing methods without the need for tuning the *nine* parameters used in [24]. Note that we are only comparing on *effectiveness* here; [24] does not make claims on *efficiency*.

B. Motif Discovery in Bird Songs

Complex songs produced by animals (bats, whales, mice, birds) have been receiving increasing attention because summaries of these sounds can be a measure of the health of the ecosystem and its biodiversity. For example, The Long Island Sound Study, a six-year research project, is a notable effort devoted to protecting the environment [38]. Birds, though still a common sight even in cities, are facing threats from habitat reduction. While bird songs have been explored in several research efforts [2][5], like human sound processing, the algorithms tend to be very specialized and parameter-laden. How well can we do with no parameters? We tested our algorithm on audio sequences of the *Common Scimitarbill* from *xeno-canto* [39]. One representative experiment is shown in Figure 9. We obtained similarly intuitive results for many other diverse species. We encourage the interested reader to hear/see them at [19].

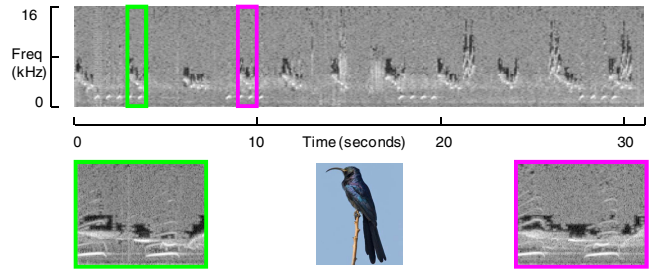

Figure 9. *top*) A 31-second excerpt of a two-minute audio performance of a Common Scimitarbill. *bottom*) A zoom-in of the two one-second long audio motif occurrences.

C. Motif Discovery in Music Data

Algorithms for automatic discovery of repeated patterns in music data can be very useful; they have a number of applications for content-based retrieval, indexing, and audio-thumbnailing (summarization) [3][21][27]. In the absence of formal benchmarks for music motif discovery, we will reproduce an experiment in a highly cited paper [3].

We attempted to find motifs in André Bourvil's song *C'était bien* [6]. As with [3], we set the motif length to three seconds and as shown in Figure 10 we discover a motif

phase *Et c'était bien*, the song title itself. In contrast to the string-matching techniques on a derived symbolic representation used by [6] and almost all music motif efforts, we do not need to extract explicit features or tune any parameters. As before, the same algorithm works on mice and men, on birds and whales, with no human intervention.

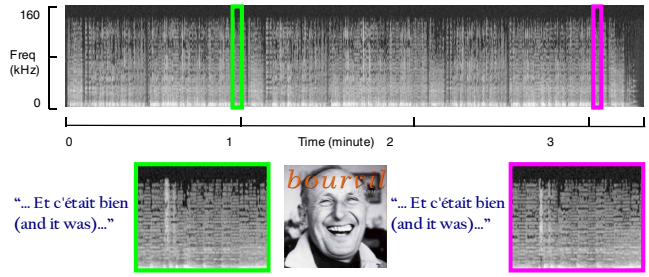

Figure 10. *top*) A performance of Bourvil's song *C'était bien* has a three-second motif. *bottom*) A zoom-in of the two motif occurrences and corresponding lyrics.

As a sanity check, we tested to see if *music* motif algorithms could be made to work for our bird/mice/whale data (to be fair, no one has claimed they might). After all, biologists do speak of mice courtship *songs* [15], bird *choruses*, and whale *melodies*, etc. However, in spite of significant effort, we could not make the music motif algorithms work for any biological datasets [19].

D. Motif Discovery in Mice Vocalization

Mice have been extensively used as genetic models of human disease for almost four decades. They can produce ultrasonic vocalizations, inaudible to humans. These vocalizations are important to researchers who study *human* pathologies by testing the effects of manipulating homologue genes in *mice* [40]. Analyzing vocal behaviors of mice models in this manner has lead to the discovery of the genetic cause of Autism [16], and has shown great promise for the study of Alzheimer's disease [30].

We applied our audio motif discovery algorithm to various subsets of the mice vocalization dataset studied in [40]. One such result is shown in Figure 1. We find that we can obtain similar results to [40] (and [15]) but *without* the need for explicitly extracting syllables, a painstaking and time consuming step. This experiment speaks volumes to the generalizability of our algorithm.

We show the *actionability* of audio motif discovery by showing that motifs, once discovered, can be used to test for changes in vocal repertoire that may be attributable to genes that were deliberately *deleted* (in genetics parlance "*knocked out*" or "*KO*") from the mouse genome.

We obtained six hours of vocalizations recorded during courtship/mating of various pairs of mice (only males vocalize). These sessions were annotated by the mice behaviors, from the set: {`Defensive` (*D*), `Ejaculate` (*E*), `Grooming` (*G*), `Intromission` (*I*), `Mounting` (*M*), `No Contact` (*N*), `Rooting` (*R*) and `Sniffing` (*S*)}. Neuroscience researchers at University of California, Riverside want to know if vocal repertoire or frequency during these behaviors differ for different mice genomes. Below we hint at the answer to this question.

We applied our algorithm to the data and found many instances of motif shown in Figure 11.*top*. Having discovered this motif, we used a sliding window to calculate its density over time. As shown in Figure 11.*middle*, this particular motif occurs about 4.1 times more frequently during `Sniffing` than during `Rooting` for this particular strain of *KO* mice. Moreover, because we are able to automate this process (most similar research efforts resort to manual counting [15][30][40]) we can automatically search through a large space of motifs × behaviors × genomes, scoring the frequency differences by significant tests.

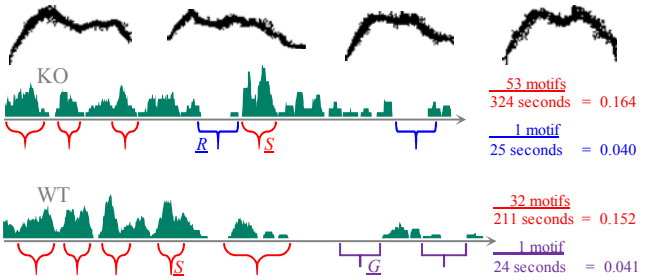

Figure 11. *top*) Sample instances of a motif discovered from mice vocalizations by applying our algorithm (*middle*) Comparing the number of motifs during *S* and *R* behaviors for a sample recording of *KO* mice vocalization. *bottom*) Comparing the number of motifs during *S* and *G* behaviors for a sample recording of WT mice vocalization.

In Figure 11.*bottom*, we show another example of a similarly significant contrasting pattern, this time in WT (wild type) mice. In this case we noted a dearth of the motif during `Grooming`. Note in [19] we show that the same algorithm that finds motifs in mice, expressed in about the 40 to 110 kHz range, also works for whales, expressed in a completely disjoint range of about 20 Hz to 24 kHz. The only difference in the two experiments was the suggested length of the motif was increased for the much larger whales, as suggested by allometry of vocal production [11].

E. Scalability of Audio Motif Discovery

After demonstrating the *utility* of our algorithm, we now show the *scalability* of finding audio motifs with an example of human speech data, a ten-minute performance of "The Raven" as shown in Figure 12.

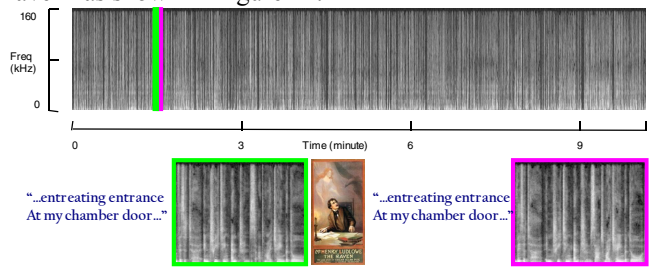

Figure 12. *top*) "The Raven" has a motif of length seven seconds. *bottom*) A zoom-in of the two occurrences and the corresponding text.

We compared the four algorithms (brute-force, random, Euclidean distance reordering heuristic, and PEAMD search) shown in Figure 13. The brute-force search takes **twenty-two** hours. Random search takes the same time, but converges more quickly, so if "anytime" interrupted after just **fourteen** minutes it would have already converged on the correct result [1][28].

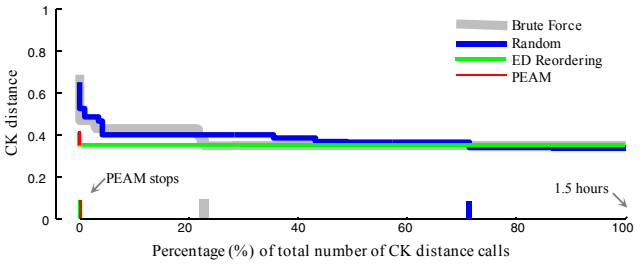

Figure 13. A comparison of efficiency of four algorithms normalized to the 100% time taken for brute-force search.

Euclidean ordered search converges to the optimal motif in a few minutes, and about a minute later PEAMD (allowing a 1 in 10,000 chance of a non-optimal answer) is confident enough to abandon the search and return the correct answer. Thus, using the PEAMD we can search audio files that are on the order of *hours* in real time. Examining *day*-long audio recordings does require more than a day (unless there are large time periods of silence, cf. Section III).

We have *informally* shown the accuracy of our algorithm in an intuitive fashion. For example in the ability to find the chorus of a poem/song (cf. Figure 7, Figure 10 and Figure 12). However, to *formally* evaluate the accuracy of the PEAMD algorithm, we compute the ratio of the motif distance returned by the brute-force linear search algorithm over the motif distance algorithm returned by our algorithm. Numbers approaching *one* indicate that there is little difference in the two algorithms output. As we can see in the second column in TABLE VI. this is strongly the case.

In addition, we compute the *speedup* of the PEAMD algorithm compared to the brute-force algorithm for all the datasets we tested. The results are shown in the third column of TABLE VI.

TABLE VI. COMPARISON BETWEEN PEAMD AND BRUTE-FORCE

Dataset (Figure Number)	$\frac{BF\_dist}{PEAMD\_dist}$	$\frac{time\ BF}{time\ PEAMD}$
A Dream within a Dream (5)	0.94	4,100
Mice vocalization (1,11)	0.92	179
The Cat in the Hat (7)	1.00	6,591
Scimitarbill (9)	1.00	267
C'était bien (10)	1.00	605
The Raven (12)	1.00	525

The results show a significant speedup for our PEAMD algorithm across diverse audio archives. Moreover, in four out of six cases the results returned are *identical* to the brute-force algorithm.

It is worth considering the two cases where our algorithm failed to return the optimal answer; in particular we can ask how badly did we fail? The answer can be seen directly in Figure 5. Here the two motifs are *very* slightly misaligned. One begins with "*See or seem...*" and the other begins "*ee or seem...*". Thus the answer is semantically correct. Similar remarks can be made for the mice vocalization dataset. Due to the page limitations, we refer to [19] for more scalability analyses and experiments.

F. *Sensitivity of User-Choice(Motif Length)*

The results shown in previous sections demonstrate the efficiency and effectiveness of our algorithm in finding motifs for a *given* user-defined length. However, the reader may wonder how critical this user choice is. Clearly, motifs can exist on different scales, for example repeated *words*, and repeated *phrases* in speech. However, it would be very undesirable if the results returned were very sensitive to tiny changes in this user choice.

It is hard to make any strong claims about this issue, as one could construct an artificial dataset for which the motifs of length w-□, w, and w + □ are disjoint. However, on real data we generally find that our algorithm will report the same essential concept when the motif length is within [w-σ, w+σ] for values of σ which are a significant fraction of w.

For example, let us revisit the *The Cat in the Hat* dataset. The motif length used in Section V.A was three seconds; however, we found that our algorithm allows us to set σ anywhere in the range of [1.7 sec, 3.9 sec] (-43%~ 30%) to obtain motifs that correspond to the same basic phrase. This result suggests a simple way to explore a dataset for which one poor intuition about possible motif lengths. We can simply set w to be a small number and find motifs of length w, 2w, 4w, 8w, etc. The efficiency of the PEAMD algorithm makes such iterative doubling search tenable.

VI. CONCLUSION AND FUTURE WORK

In this work we introduced a scalable and extremely general framework for finding audio motifs. We have demonstrated the utility of audio motifs analysis in diverse domains including music, human speech, mice vocalizations, and bird songs. By comparisons to existing work (Figure 8, Figure 10) we have shown that the representative power of our general purpose distance measure is typically competitive with domain specialized measures. While there is no obvious rival strawman to compare to in terms of *efficiency*, we have shown that by using probabilistic early abandoning we can examine most realistic length scientific recordings in much less than real time.

For brevity we have hinted at the *utility* of audio motifs only in the mice genetics domain; however, in data types as diverse as text, DNA, time series, and video, motif discovery is often leveraged for diverse types of analyses [20][25][27]. We believe this work has the potential to enable analogous analyses for audio.

We have claimed that our method is essentially parameter free. The reader might object to this claim, noting for example that the algorithm that converts audio to a spectrogram representation requires several parameters to be set. This is true, but in most cases the best parameters have been determined by the community decades ago. For example, virtually all mouse researchers truncate below 20 Hz and above 100 kHz [14][30][40]. The best parameters for *human* vocalization research are even better understood [23].

In future work, we hope to leverage off the Minimum Description Length principle to automatically find the natural length for motifs, thus removing the need for this user input, the only true parameter we need to set. However

we note that even here, as we showed in Section V.F, our algorithm is not particularly sensitive to this setting.

Finally, we note that we have made all code and data freely available in perpetuity so others can confirm, use, and extend our work [19].

ACKNOWLEDGMENT

We would like to thank the Cornell Lab of Ornithology and *xeno-canto* for sharing their data used in [32][39], Khaleel Razak for their help with the mice vocalization dataset, and the financial support for our research provided by NSF IIS-1161997 and FRAXA Research Foundation.

REFERENCES

[1] I. Assent, P. Kranen, C. Baldauf, T. Seidl. AnyOut: Anytime Outlier Detection on Streaming Data. *DASFAA* (1) 2012: 228-242.

[2] S. E. Anderson, A. S. Dave, and D. Margoliash. Template-based automatic recognition of birdsong syllables from continuous recorderings. *Acoustic Society of America Journal*, 100: 1209-19, Aug 1996.

[3] J.-J. Aucouturier and M. Sandler. Finding repeating patterns in acoustic musical signals: applications for audio thumbnailing. In *AES 22nd Int' Conference*, 2002.

[4] S. Baluja, M. Covell. Waveprint Efficient wavelet-based audio fingerprinting. *Pattern Recognition* 41, 3467-80, 2008.

[5] R. Bardeli. Similarity search in animal sound databases. *IEEE Trans on Multimedia*, vol. 11, no. 1, pp. 68–76, 2009.

[6] Bourvil. C'etait bien (*le petit bal perdu*). Lyrics: R. Nyel, Music: G. Verlor, *Editions Bagatelle*. 1961.

[7] N. A. Butinov, Y. Knorozov. Preliminary Report on the Study of the Written Language of Easter Island. *Journal of the Polynesian Society* 66 (1): 5–17, 1957.

[8] B. J. L. Campana, E. J. Keogh. A Compression Based Distance Measure for Texture. *SDM*: 850-861, 2010.

[9] P. Cano, E. Battle, T. Kalker, J. Haitsma. A review of audio fingerprinting. *The Journal of VLSI Signal Processing*, pp. 271-284, 2005.

[10] R. Cole, J. Mariani, H. Uszkoreit, G. B. Varile, A. Zaenen, A. Zampolli. Survey of the State of the Art in Human Language Technology. *Cambridge University Press*, 1998.

[11] K. P. Dial, E. Greene, and D. J. Irschick. Allometry of behavior. *Trends in Ecology and Evolution* 23:394–401.

[12] Dr. Seuss. The Cat in the Hat. ISBN 0-394-80001-X, *Random House*, 1957.

[13] P. Fussell. Poetic Meter and Poetic Form. *Random House*. 1965.

[14] C. M. Glaze, T. W. Troyer. Behavioral Measurements of a Temporally Precise Moto Code for Birdsong. *Journal of Neuroscience*, 27(29): 7631-7639, July 18, 2007.

[15] J. M. S. Grimsley, J. J. M. Monaghan, J. J. Wenstrup. Development of Social Vocalizations in Mice. *PLoS ONE* 6(3): e17460, 2007.

[16] R. J. Hagerman, et.al. Advances in the Treatment of Fragile X Syndrome. *Pediatrics* Vol. 123 No.1, January, pp 378–390 2009.

[17] J. Haitsma, T. Kalker. A Highly Robust Audio Fingerprinting System. *In Proceedings of International Conference on Music Information Retrieval*, 2002.

[18] Y. Hao, B. J. L. Campana, E. J. Keogh. Monitoring and Mining Insect Sound in Visual Space. *SIAM SDM*, 2012: pp 792-803.

[19] *Audio Motif Discovery Webpage*. *https://sites.google.com/site/audiomotif*

[20] C. Herley. ARGOS: Automatically Extracting Repeating Objects from Multimedia Streams. *IEEE Transactions on multimedia*, Vol.8, No.1, February 2006.

[21] J.-L. Hsu, C.-C. Liu, A. L. P .Chen. Discovering nontrivial repeating patterns in music data. *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 311-25, Sep. 2001.

[22] B. Hu, T. Rakthanmanon, B. J. L. Campana, A. Mueen, E. J. Keogh. Image Mining of Historical Manuscripts to Establish Provenance. pp 804-815. *SDM* 2012.

[23] A. Jansen, K. Church, H. Hermansky. Towards Spoken Term Discovery at Scale with Zero Resources. *INTERSPEECH*, 1676-1679, 2010.

[24] A. Jansen, B. V. Durme. Indexing Raw Acoustic Features for Scalable Zero Resource Search. *INTERSPEECH*, 2012.

[25] H. Jiang, T. Lin, H.-J. Zhang. Video segmentation with the assistance of audio content analysis. *In Proc. ICME*, New York, 2000.

[26] D. C. Karnopp. Random Search Techniques for Optimization Problems. *Automatica*, 1963.

[27] Y. Ke, D. Hoiem, R. Sukthankar. Computer Vision for Music Identification. *Proc. Computer Vision and Pattern Recognition, (CVPR)*, pp. 597-604, 2005.

[28] P. Kranen, M. Hassani, T. Seidl: BT*- An Advanced Algorithm for Anytime Classification. *SSDBM*: 298-315, 2012.

[29] D. G. Lowe. Distinctive Image Features from Scale Invariant Key Point. *International Journal of Computer Vision*, vol.60, pp. 91-110, 2004.

[30] C. Menuet, Y. Cazals, C. Gestreau, P. Borghgraef, L. Gielis, et al. (2011) Age-Related Impairment of Ultrasonic Vocalization in Tau.P301L Mice: Possible Implication for Progressive Language Disorders. *PloS ONE* Jan; 6(10).

[31] A. Mueen, E. J. Keogh, Q. Zhu, S. Cash, M. Brandon Westover. Exact Discovery of Time Series Motifs. *SDM* 2009: 473-484

[32] Macaulay Library, Cornell Lab of Ornithology *www.macaulaylibrary.org/index.do*

[33] S. Pfeiffer, S. Fischer, W. Effelsberg. Automatic Audio Content Analysis. *ACM Multimedia* 1996.

[34] J. C. Ross, Vinutha T. P., P. Rao. Detecting Melodic Motifs from Audio for Hindustani Classical Music. *ISMIR*, 2012.

[35] M. L. Scattoni, S. U. Gandhy, L. Ricceri, J. N. Crawley. Unusual Repertoire of Vocalizations in the BTBR T+tf/J Mouse Model of Autism. *PLoS ONE* 3: e3067, 2008.

[36] V. M. Trifa, L. Girod, T. Collier, D. T. Blumstein, C. E. Taylor. Automated Wildlife Monitoring Using Self- Sensor Networks Deployed in Natural Habits. *AROB* 2007.

[37] A. Vedaldi. *www.vlfeat.org/~vedaldi/index.html*, 2011

[38] L. Wahle. *Plants and Animals of Long Island Sound*. Sea Grant, CT-SG-90-11. 1990.

[39] Xeno-canto. *http://www.xeno-canto.org/*

[40] J. Zakaria, S. Rotschafer, A. Mueen, K. Razak, E. Keogh. Mining Massive Archives of Mice Sounds with Symbolized Representations. *SIAM SDM*, 2012. pp 588-599.

[41] Y. Zhang, S. Rajagopalan, M. Salman. A Practical Approach for Belt Slip Detection in Automotive Electric Power Generation and Storage System. *In Aerospace Conference*, IEEE pp.1-7, 2010.