Use of symbolic representations of pitch contours for Orca call indexing using tools from Bioinformatics

Steven Ness
Department of Computer Science
University of Victoria
Canada
sness@uvic.ca

Niko Rebenich
Department of Computer Science
University of Victoria
Canada
niko@uvic.ca

Patrick Gorman
Department of Computer Science
University of Victoria
Canada
pgorman@uvic.ca

Chris W. Pitkin@uvic.ca
Department of Computer Science
University of Victoria
Canada
cwpitkin@uvic.ca

ABSTRACT

The Orchive is a large collection of over 20,000 hours of audio recordings from the OrcaLab research facility located off the northern tip of Vancouver Island. It contains recorded orca vocalizations from the 1980 to the present time and is one of the largest resources of bioacoustic data in the world. We have developed a web-based interface that allows researchers to listen to these recordings, view waveform and spectral representations of the audio, label clips with annotations, and view the results of machine learning classifiers based on automatic audio features extraction.

In this paper we investigate the use of symbolic representations of pitch contours to allow tools from bioinformatics to be used to search large databases of pitch information. We examine the performance of different types of symbolic representations on this dataset.

General Terms

Theory

Keywords

Symbolic Approximation

1. INTRODUCTION

The Orchive is a large archive containing over 20,000 hours of recordings from the Orcalab research station. These recordings were made using a network of hydrophones and originally stored on analog cassette tapes. OrcaLab is a research station on Hanson Island which is located at the north part of Vancouver Island on the west coast of Canada. It has been in continuous operation since 1980. It was designed as a land based station in order to reduce the impact on the orcas under study, as the noise and disturbance from boats affects the orcas in observable but currently unquantified ways. In collaboration with OrcaLab, we have digitized the tapes and have made these recordings available to the scientific community through the Orchive website (http://orchive.cs.uvic.ca).

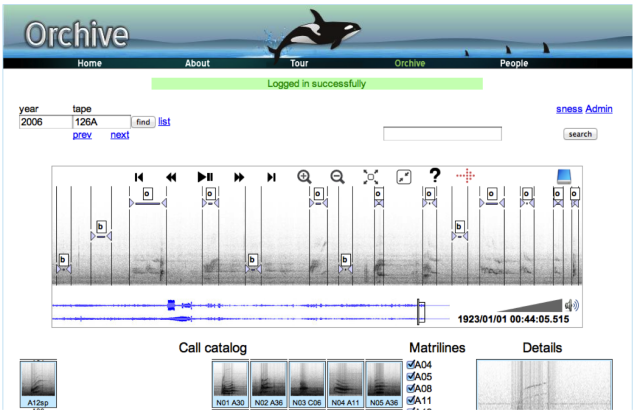

Figure 1: Annotated audio from from the Orchive

Over the past 5 years, a number of orca researchers using our website have added over 18,000 clip annotations to our database. A small section of annotated audio from the Orchive is shown in Figure 1. These clip annotations are of two main types: The first is clips that differentiate background noise from orca calls and from the voice notes of the researchers that collected the data. The second type of clip annotations classify orca vocalizations into different calls. Orcas make three types of vocalizations, echolocation clicks, whistles and pulsed calls. The pulsed calls are highly conserved stereotyped vocalizations which have been classified into a catalog of over 52 different calls by John Ford [8]. Of the 18,000 annotations currently in the Orchive, 3000 are

of these individually classified calls. In addition, we have a curated call catalog containing 384 different recordings of different calls vocalized by a variety of different pods and matrilines.

Many parts of the recordings contain boat noise which makes identifying orca calls both difficult and tiring. In addition, the size of the Orchive makes full human annotation practically impossible. Therefore we have explored machine learning approaches to the task. One data mining task is to segment and label the recordings with the labels background, orca, voice. Another is to subsequently classify the orca calls into the classes specified in the call catalog.

2. INTRODUCTION

Audio feature extraction is the first step in classifying audio using machine learning algorithms. A commonly use audio feature that is often used is an estimate of the fundamental frequency (F0) or pitch, one well known algorithm for this is the Yin algorithm [5]. This algorithm is primarily an autocorrelation based approach, which means that it takes the audio signal and convolves it with itself. The peaks in this convolution then correspond to harmonics in the signal, and with noise free data with harmonics that strictly decrease, the lowest peak is the fundamental frequency. With audio that does not fit this strict definition, there are many cases where the lowest peak is not the fundamental frequency, one example is if odd harmonics are systematically lower than even harmonics, and another is if there is substantial noise in the data. The Yin algorithm makes several modifications to simple autocorrelation to overcome these issues. Another modern pitch detector is SWIPEP [4].

Mel-Frequency Cepstral Coefficients [12] (MFCC) have been widely used for this purpose. MFCCs have also been used in bioacoustics, and have been used to classify insect sounds [9], birds [10] and orca calls [15]. We have investigated the use of MFCC values for classifying calls from the orca call catalog, and results using these with a variety of machine learning techniques are described below. In future work we would like to try to use MFCC or other forms of spectral data as input to our symbolic approximation algorithm.

Another type of audio feature extraction that is promising is features based on models of the auditory cortex [13]. These algorithms model the properties of the cochlea and peripheral nervous system[14], and have at their core an adaptive filterbank coupled to a triggered pulse model [19]. However, one issue with these systems is that instead of a 1-dimensional (waveform) or 2-dimensional (spectral), they lead to a 3-dimensional dataset in which 2-D audio images change over time, which leads to a very large amount of data. Approaches using vector quantization could be used as an input to our symbolic approximation algorithm in future work.

3. TIME SERIES QUANTIZATION

The transcription of time series data into a character sequence that can be consumed by bioinformatics sequence alignment tools is a nontrivial task. An attempt towards this goal is Symbolic Aggregate Approximation (SAX) which was proposed by Lin et al. in [11]. The underlying idea of SAX is to parse a time series using a sliding window and to generate a character sequence that approximates the signal's normalized slope using a technique called Piecewise Aggregate Approximation (PAA). SAX is most useful in cases where the data is not on an absolute scale. In order to investigate the utility of SAX on this dataset, we plotted the fundamental frequency curves of a number of examples of calls to each other, two of these plots are shown in Figure 2 and Figure 3. From these we can see that the absolute pitch of these calls is well conserved, a result that has been previously observed [8]. It is of interest to note that the N05 call voiced by the A35 matriline is of a lower pitch than the others, this matriline is in the A4 pod, while the calls by the A12 and A36 matrilines are of more similar pitch to one another. This information could be used to help classify which pod is vocalizing a particular call.

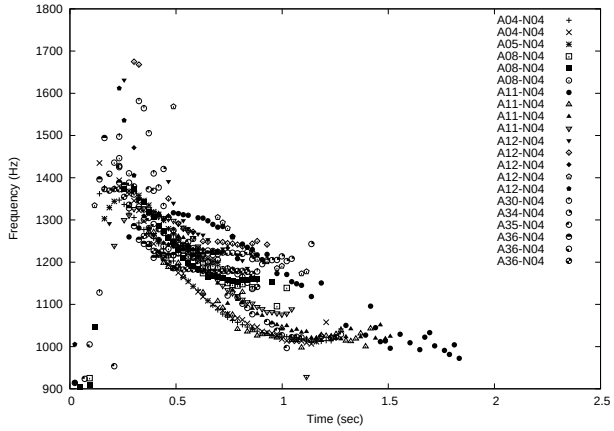

Figure 2: F0 contour for 21 examples of the N04 call.

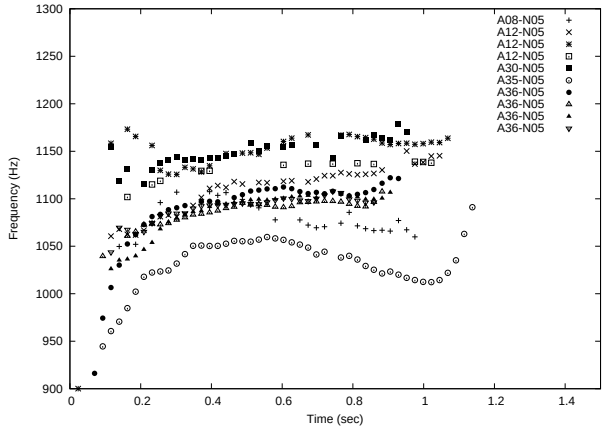

Figure 3: F0 Contour for 10 examples of N05 call.

This approach is certainly valid if slope changes within a given window characterize the signal well and the signal is

fairly free of noise. We therefore have examined SAX for a possible avenue to transcribe Orca vocalizations. We determined, however, that considering windowed slope fragments of Orca vocalizations does produce sequences that are of lower quality than those produced by our transcription technique which we refer to as FTSQ (Fundamental Frequency Time Series Quantization). The FTSQ procedure is outlined in detail below. We do not provide classification results for SAX sequences because the majority of SAX sequences were too long in order to be processed by our dynamic programming based alignment algorithm and hence a direct comparison can't be impartial. Further, we observed that the fundamental frequency of Orca vocalizations carries a lot of information, that is, it seems to be stable, and within a certain range that is unique to a large portion of the vocalizations in our call catalog. Since SAX is normalizing the signal slope within a sliding window it cannot take advantage of this important feature.

3.1 Signal Pre-processing and FTSQ

Fundamental Frequency Time Series Quantization is a time series quantization approach that transcribes an audio time series based on an estimate of it's fundamental frequency (F_0).

In general it is not feasible to analyze Orca vocalizations as a raw time series, since these signals are a complex mixture of sinusoids and noise (see Figure 4). Figure 5 shows a spectrogram representation of the N01 and N09 Orca vocalizations. When comparing Figures 4 and 5 it is easy to see that the frequency components of the audio signal reveals much more about the structure of the signal than a raw audio time series by itself.

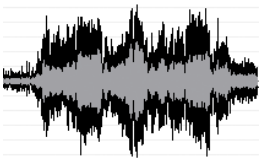

Figure 4: Raw audio signal of Orca vocalization N01.

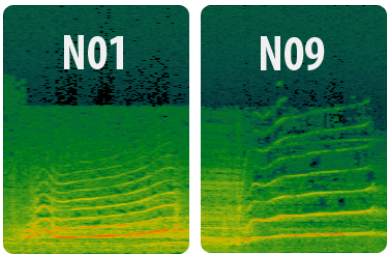

Figure 5: Frequency spectrum of Orca vocalizations N01 and N09.

Quantizing the fundamental frequency of an audio signal is am approach that makes use of the underlying structure in the frequency bands of the audio signal. Loosely speaking one may think of the fundamental frequency as the pitch of a sound signal. For our FTSQ technique we use Yin [6] a robust fundamental frequency estimation algorithm developed by Cheveigne et al. Processing the audio waveform from Figure 4 with Yin yields the fundamental frequency (in octaves relative to 440 Hz) shown in the top plot of Figure 6. The two plots below the fundamental frequency show the aperiodicity and the period-smoothed instantaneous power of the signal; they give estimate in the confidence on the approximation of F_0 and are used to identify meaningful regions of interest within the F_0 signal.

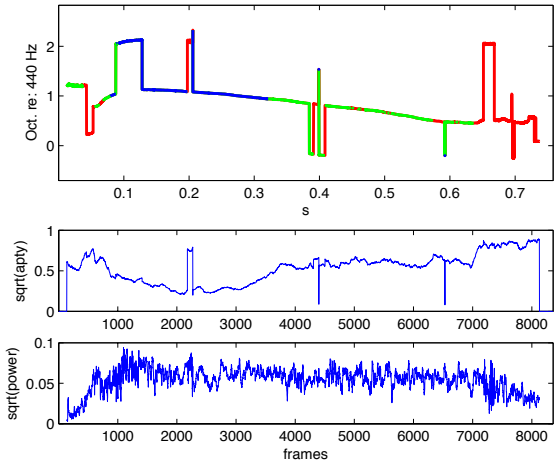

Figure 6: Frequency spectrum of Orca vocalizations N01 and N09

A further issue that is encountered commonly in the process of determining a pitch contour for an audio recording is the optimization of all the parameters of the pitch detection algorithm to perform well on the particular dataset that is being used. There are many such parameters for each different algorithm, the important ones in the Yin algorithm include the window size and hop size of the FFT, the high and low frequency cutoffs, the high and low frequencies around which the pitches will be wrapped using modulo arithmetic, the tolerance of the Yin algorithm which determines which peaks in the autocorrelation will be used for the pitch determination, the window size for the median filter, and the number of histogram bins in which to divide the frequency range into. In our previous work, this was done by hand by plotting points using MATLAB or Gnuplot, which can be a long and labour intensive process.

For this project, we added custom visualization tools to our OpenMIR platform to view the original audio as a spectrogram, to allow the user to listen to this audio, to show a pitch contour with dynamic controls over these various parameters, and an energy display that shows the RMS energy of the audio signal. This interface is shown in Figure 7. This software allowed us to quickly iterate over a large number of parameters, and to determine the optimal parameters for the Yin algorithm. The most important of these was the median filter size, and a median filter of size 11 is shown in Figure 7.

3.1.1 Octave Errors

The output of Yin does not always give a proper estimate of F_0. Indeed, the F_0 signal in Figure 6 displays several estimation errors that we herein refer to as *octave errors*. Octave

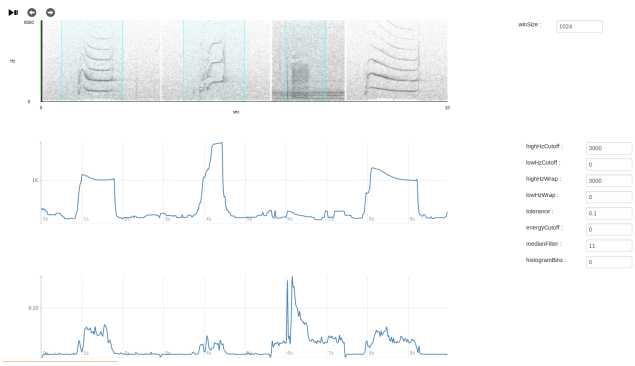

Figure 7: Spectrogram, pitch and RMS view of a small portion of the Orchive catalog viewed with the OpenMIR interface.

errors manifest themselves as sudden jumps in a multiple of the core F_0 frequency. Figure 6 displays multiple octave errors the first one at about 0.1 seconds. When quantizing the F_0 signal into a letter representation over a finite alphabet. Octave errors result in mis-mapped letters in the output sequence. To a certain degree we are able to handle octave errors by crafting a custom substitution matrix in our sequence alignment algorithm. However, doing so ultimately lowers the confidence in our alignment score and might confuse legitimate frequency jumps with octave errors and as a result would not penalize mismatches appropriately. Therefore, we have developed a technique that can automatically correct for the majority of octave errors and as a net effect produces a more accurate letter representation.

We first run the F_0 signal obtained from Yin through a median filter, which smoothes the signal and eliminates small transients that occur due to noise in the original signal. We then threshold the aperiodicity and the period-smoothed instantaneous power to obtain F_0 signal regions at which the F_0 signal is estimated with high confidence. After downsampling the signal we obtain the representation shown in Figure 8.

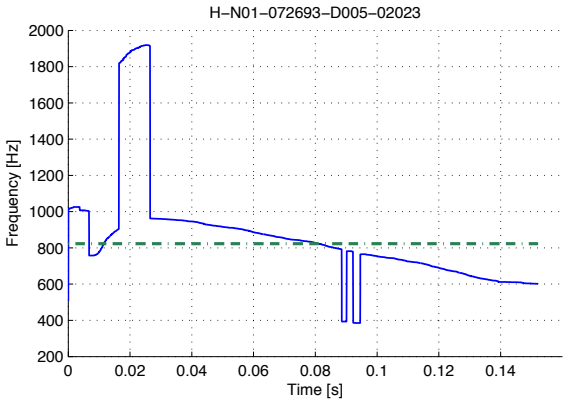

Figure 8: Thresholded and down-sampled F_0 signal of a N01 call.

The green line represents the median of the original F_0 signal drawn in blue. The octave errors are clearly identifiable as such in Figure 9 which shows versions of the F_0 signal one octave above (purple) and on octave below (red) the original (blue). F_0 signal. To correct for octave errors we simply piece the portions of the three versions of the F_0 signal together according to which ever signal is closest to a slightly upwards biased median of the original signal (blue). This yields the black F_0 curve wich shows the now automatically corrected version of the F_0 signal.

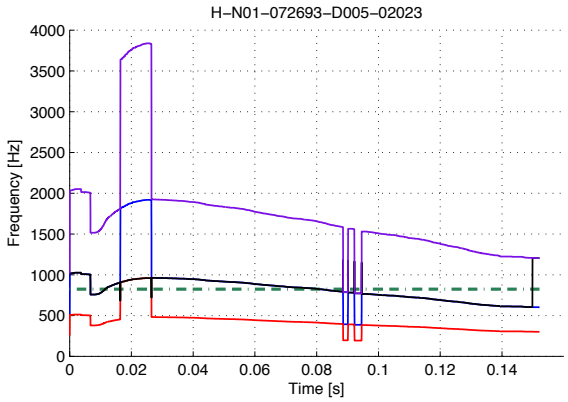

Figure 9: Octave errors within a F_0 signal of a N01 call.

3.2 Log-Normal Signal Quantization

When transcribing the F_0 signal into a letter representation suitable for sequence alignment tools it is natural to ask how these letters should be assigned over the range of possible F_0 frequency values. A naive approach would be to map F_0 frequencies to letters in a linear fashion. However, while this approach does work well in practice, we investigated if a non-linear mapping might potentially be more appropriate. For this purpose we plotted the distribution of F_0 frequencies over time for the whole catalog of Orca vocalizations in Figure 10. The dynamic range of F_0 incorporates frequencies from about 80 to 2400Hz, with three high density bands at around 250, 700 and 1200Hz. Ideally one would therefore quantize the F_0 signal using a three-modal distribution. However, for simplicity we chose a log-normal distribution (see Figure 11), which provides us with a fine resolution for the more common F_0 frequencies and quantizes high F_0 frequencies at a coarser level.

The result of log-normal quantization of the F_0 trace in Figure 9 is provided in Figure 12. Finally, the resulting letter sequence is given by `LLHHJJKKKKKKKKKKKJJJJJJJJI-IIIIIHHHHHHHGGGGGFFFFFF` which can readily be consumed by a dynamic programming sequence alignment algorithm.

4. ALPHABETIC SEQUENCE REPRESENTATION

In order for sequence comparison algorithms to work well, it is necessary that the combination of the letters and the scoring matrix are compatible and output meaningful results. In the two sections below, we show the result of quantizing the signal using the nonlinear histogram approach for two of the more common calls vocalized by A-clan Northern Resident whales. The first is the very common N4 call, which consists

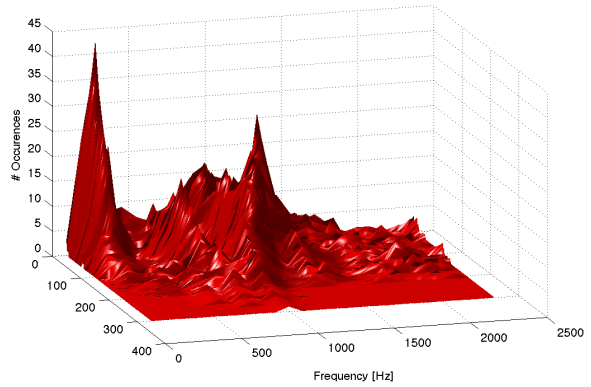

Figure 10: Distribution of F_0 frequencies of call catalog.

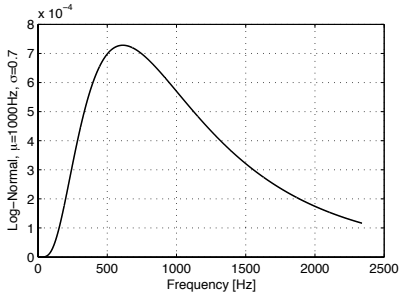

Figure 11: Log-normal distribution for quantization

of a quick up swing in pitch, followed by a downswing and then a constant tone. In it we can clearly see long regions of the repeated letters, P,O,N,M,L,K. Even from a quick visual inspection these repeated letters show that our method of converting frequencies into a discrete alphabet is promising.

```
N04 A04 HIKNOOOOOPPPPOOOOOOOOOOOONNNNNNNNNNNMMMMMMMMMMMLLLLLLLLLLLLLLLLLLLLKKKKKKKKKKKKKK
N04 A04 IJNOPPPPPPPPPPPOOOOOOOOOONNNNNNNNMMMMMMMMMMMLLLLLLLLLLLLLLLLLLLLKKKKKKKKKKKKKK
N04 A05 NLKLMNOOOOOOOOOOOOOOOOOOOOOOOOOONNNNNNNNNNNNNNNNNNNMHGQ
N04 A08 RILMNNOOOOOOOOIIOOOOOOOOOONNNNNNNNNNNNNNNNNNNNMMMMMMMMMMMMMMMMMMMLLLLLLLMLLLLL
N04 A08 SIIJJKNPQQRPIIOOOOOOOONNNNNNNNNNNNNNNMMMMMMMMMMMMMMMMMMMMMMMMMDCCCMMM
```

Even more promising were the results for another call, N5, which consists of a constant tone. These calls are from the A36 matriline of orcas, which now consists of only two whales, the brothers A37 and A46, although the grandmother whale, A12 often now associates with this matriline

Data Set	Global Accuracy
Original	64%
Octave Removal - Linear	81%
Octave Removal - NonLinear 900	80%
Octave Removal - NonLinear 1200	81%
J48	58%
Naive Bayes	65%
SVM	75%

Table 1: Global accuracy for different methods of converting a Yin pitch contour into an alphabet using FTSQ

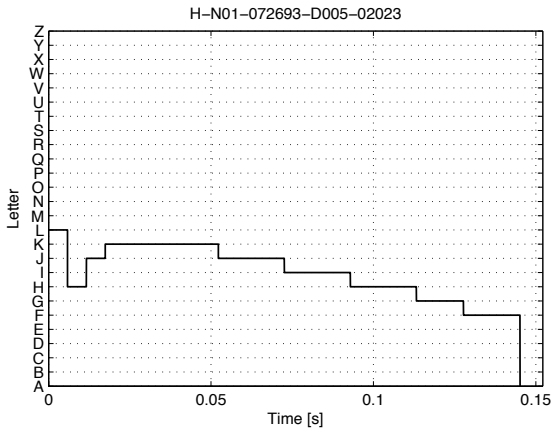

Figure 12: F_0 signal of N01 call quantized to a finite alphabet

Call	Accuracy
N47	0.400
N01	0.969
N12	0.583
N05	0.785
N04	1.000
N03	0.75
N09	0.772

Table 2: Global accuracy for different call types using Octave Removal NonLinear 1200 parameters with the FTSQ algorithm.

after A34, her daughter's matriline, grew large in size. From these calls, we can see that the frequency represented by the letter L is very constant throughout the entire call, and is a clear indication that this is an N5 call.

```
N05,A36,A36-N05-070806-D012-13913,IIJJJKKKKKLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
N05,A36,A36-N05-070806-D012-13917,JJJKKLLFLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
N05,A36,A36-N05-070806-D012-13921,IIJJKKKKKKKKLLLLLLLLLLLLLGLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
N05,A36,A36-N05-071506-D017-11311,IQHJKLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
```

5. FUTURE WORK

From these promising results from using Yin for pitch estimation, a nonlinear histogram binning procedure, and a non-linear histogram binning approach, we would like to investigate changes to all three of these sections of our algorithm.

We first wish to investigate the use of other pitch determination algorithms, primarily the SWIPEP algorithm [4] which was used in previous work with audio from chant traditions [15]. There are many other pitch determination methods that we wish to investigate as well, including a convolutive probabilistic method from Benetos and Dixon [3], and a multi-tier approach by Sun [18].

We also wish to investigate the use of histogram based methods for deriving a histogram directly from the audio, as investigated in [15]. This approach uses a series of gaussians to build up an approximation of the frequency content of the sound directly from the fundamental frequencies in a range

of audio. From Figure 10, we can clearly see three different lobes of frequency, and a quantization approach that takes this into account would be very useful.

In early work on this paper, we investigated another method of turning audio into an alphabet, namely, clustering Mel-Frequency Cepstral Coefficients (MFCC) into distinct clusters and using these clusters as the letters for our alphabet. Unfortunately the time complexity of clustering algorithms depends heavily on the size of the number of clusters, and thus the size of the alphabet. For k-means clustering the complexity is $O(n^{dk+1}log(n))$, where d is the size of the input vector, k is the number of clusters, and n is the number of input vectors. The MFCC algorithm typically gives vectors of size 13, but can give a vector of size 26 or more. For an alphabet of size 20, the exponent is of order $13*20+1 = 261$, which caused problems in terms of length of time. With further reflection we may be able to find another way of doing this clustering, either with a reduced set of input vectors, or by finding some way to reduce the size of this exponent.

Alternatively, we could use algorithms from multidimensional sequential pattern mining, such as that described by Pinto et al. [17]. These algorithms take as input a multidimensional representation of a sequence, containing many parallel sequences, and look for patterns in this sequence. We feel this might be a good approach for mining data directly from MFCC or other related spectral representations of audio.

We also want to investigate the use of other tools from string mining and bioinformatics in this problem domain. The first algorithm that we wish to try is the BLAST [1] algorithm, however we have doubts about the applicability of this algorithm to our problem domain, because instead of looking for small sequence fragments that are identical or closely related, we are more looking for large regions of similar sequence.

Instead, we see great hope in the use of other algorithms for mining data in sequential patterns, such as those described by Dietterich [7]. The PrefixSpan algorithm [16] is also a relevant algorithm for this work which mines sequential patterns. The SPAM algorithm [2] uses a bitmap representation for sequences, and would also be useful to investigate for this dataset.

6. CONCLUSION

In this work we investigated the possibility of converting audio of the vocalizations of Orcas into a discrete alphabet, thus making it amenable to be studied by the large variety of bioinformatics and string comparison tools extant in the scientific community.

In the first section of this work, we used the Yin algorithm to determine the fundamental frequency of the audio of a curated call catalog of orca vocalizations. By adjusting various parameters of the Yin algorithm using software custom designed for this project in the OpenMIR suite, we were able to obtain pitch contours of this audio.

We then took the extracted pitch contours and used an algorithm that first did a median filtering on the audio followed by a novel algorithm that replaced parts of the pitch contour that were subject to octave errors and output a pitch contour that had many fewer such octave errors.

A series of experiments looking at different ways of dividing the frequency range of this audio into histogram bins was then performed. We first tried a naive approach that simply divided the entire frequency range into equal sized bins, but found that we obtained improved performance when we used a non-linear frequency histogram.

This output of this histogram binned pitch contour was turned it into letters, and used a local alignment algorithm to align these. In other work Dynamic Time Warping is typically used, an algorithm that takes in real valued numbers and does a global alignment. We found that for this data, a local alignment outperformed a global alignment. From this local alignment, we obtained quite satisfactory classification results using an accuracy-at-top-1 approach of approximately 80%.

In conclusion, we see the technique of using the tools for bioinformatics on the audio data of orca calls to be promising. We have converted the entire 18,000 hours of data currently in the Orchive into pitch contours and are working on evaluating new string mining algorithms on this dataset.

7. REFERENCES

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, Oct. 1990.

[2] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 429–435, 2002.

[3] E. Benetos and S. Dixon. A temporally-constrained convolutive probabilistic model for pitch detection. In *WASPAA*, pages 133–136, 2011.

[4] A. Camacho. *A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music*. PhD thesis, University of Florida, 2007.

[5] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[6] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[7] T. G. Dietterich. Machine learning for sequential data: A review. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30. Springer-Verlag, 2002.

[8] J. Ford. A catalogue of underwater calls produced by killer whales (orcinus orca) in british columbia. Technical Report 633, Canadian Data Report of Fisheries and Aquatic Science, 1987.

[9] Z. Le-Qing. Insect sound recognition based on mfcc and pnn. In *Proc. of the 2011 Int. Conf. on*

Multimedia and Signal Processing, CMSP '11, pages 42–46, 2011.

[10] C. Lee, C. Lien, and R. Huang. Automatic recognition of birdsongs using mel-frequency cepstral coefficients and vector quantization. In *IMECS*, pages 331–335, 2006.

[11] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, DMKD '03, pages 2–11, New York, NY, USA, 2003. ACM.

[12] B. Logan. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[13] R. Lyon. A computational model of filtering, detection, and compression in the cochlea. In *ICASSP*, pages 1282 – 1285, 1982.

[14] R. F. Lyon, A. G. Katsiamis, and E. M. Drakakis. History and future of auditory filter models. In *ISCAS*, pages 3809–3812, 2010.

[15] S. Ness, M. Wright, L. Martins, and G. Tzanetakis. Chants and orcas: semi-automatic tools for audio annotation and analysis in niche domains. In *Proc. of the 2nd ACM workshop on Multimedia semantics*, pages 9–16, 2008.

[16] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: the prefixspan approach. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1424–1440, 2004.

[17] H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *Proceedings of the tenth international conference on Information and knowledge management (CIKM '01)*, pages 81–88, 2001.

[18] X. Sun. F0 generation for speech synthesis using a multi-tier approach. In *INTERSPEECH*, 2002.

[19] T. Walters. *Auditory-Based Processing of Communication Sounds*. PhD thesis, University of Cambridge., 2011.