

A FAST AND AUTOMATED PIPELINE FOR TIME-DOMAIN BIOACOUSTIC ANALYSIS

Alex Townsend, Peter Beerli, Anke Meyer-Baese, Dennis E. Slice †
Florida State University, Department of Scientific Computing



Abstract

Speed, efficiency and automation are extremely important for modern field researchers who are increasingly using computer tools in their work. Bioacoustics has traditionally involved very computationally expensive and very complex techniques which, while highly accurate, can be very time consuming and can require a lot of computational power which may not be easily available to researchers in the field. Another aspect of modern software-based analysis pipelines is the need for end-to-end automation. Most systems require a good amount of scripting, parameter selection and tuning and pre-processing work from the end-user. This work presents a new approach to bioacoustics analysis which allows for full end-to-end automation and a reduction in processing complexity and computational power requirements by using a strictly time-domain and shape-driven approach.

Overview

- Step 1: Pipeline takes in raw files of individual vocalizations
- Step 2: An $O(n)$ Algorithm is used to extract pulses from vocalizations
- Step 3: Pulses are organized into ordinal groups
- Step 4: Grouped Pulses are Normalized and aligned by Maximum Cross-Correlation
- Step 5: Grouped Pulses are compared via Pairwise Distance Matrices
- Step 6: Pulse Group Distance Matrices are assimilated by Simple Averaging
- Step 7: Assimilated Consensus Distance Matrix is clustered by Neighbor-Joining

Advantages

- **Unsupervised:** This approach uses clustering and does not require training data. As a result, it is very good for situations where available data or resources may be limited.
- **Transparent:** Hierarchical Clustering, Cross-Correlation and this work's simplified version of the Pan-Tompkins Algorithm are all very well understood processes which do not suffer from the kind of interpretability issues Neural Networks often do.

Pipeline Workflow

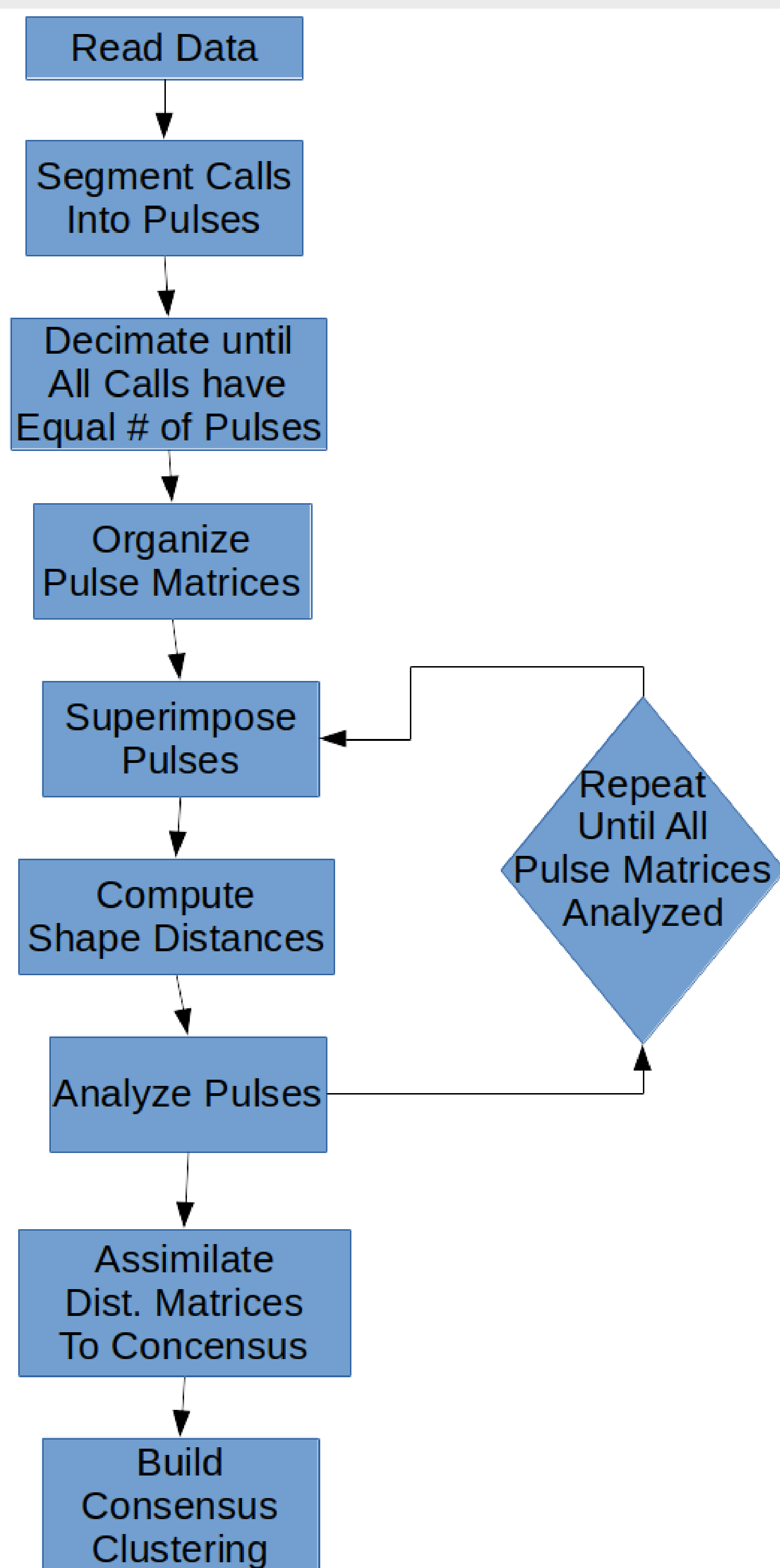


Figure 1: A block-flow diagram of the steps of the pipeline.

Pulse Extraction

- One particularly important issue in this workflow and bioacoustic analyses in general is Pulse Extraction. Most modern methods use very complex methods which combine time and frequency domain representations to find pulses. These have the downside of being very parameter-heavy and very sensitive to mistuned parameters.
- In this paper, we have taken a simpler $O(n)$, time-domain only approach which takes inspiration from the Pan-Tompkins Algorithm for finding QRS complexes in EKG signals. The method we use takes the absolute value of the call to transform the data to strictly positive values, then we use a sliding window to look for locations where the average deviates from background noise.

Pulse Extraction

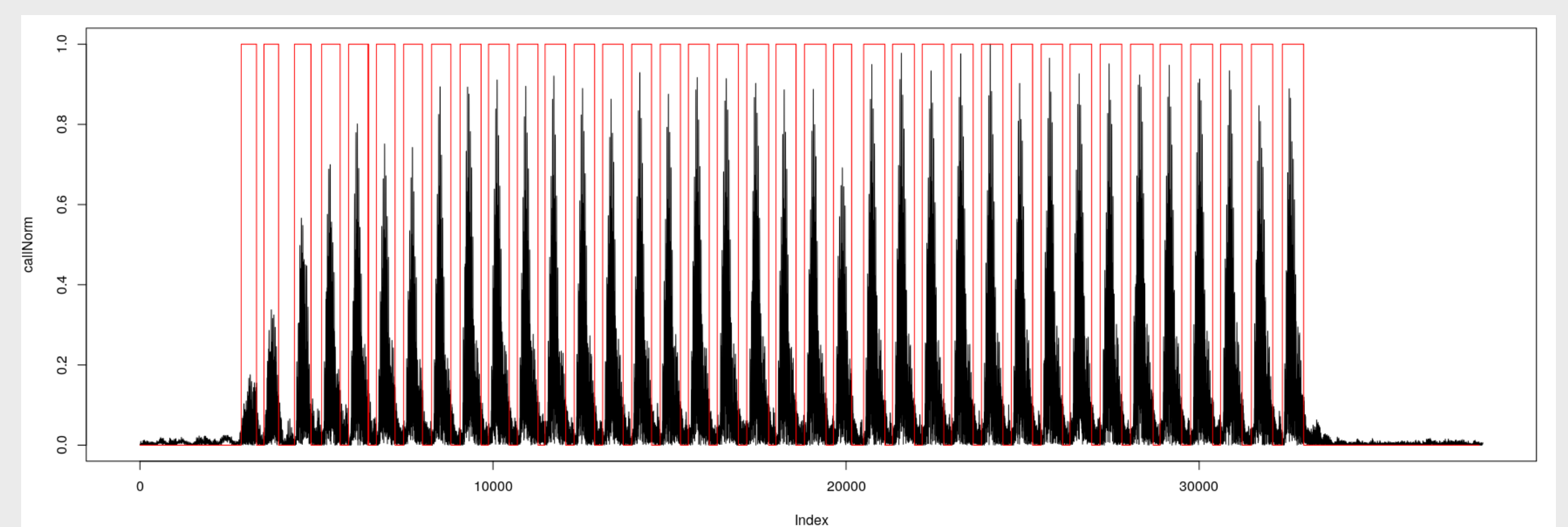


Figure 2: A demonstration of the Simplified Pan-Tompkins Algorithm finding Pulses.

Efficiency and Scalability

- Pulse Extraction is $O(n)$ but embarrassingly parallel.
- Pulse Alignment is $O(n^2 \log n)$ in the worst case but embarrassingly parallel.
- Distance Matrix Calculation is $O(n^2)$ but parallelizable.
- Distance Matrix Assimilation is $O(n^3)$ in the worst case.
- Clustering is $O(n^3)$ in the worst case.

Refs

- 1 Cormen, Thomas H., et al. Introduction to algorithms. MIT press, 2009.
- 2 Pan, Jiapu, and Willis J. Tompkins. "A real-time QRS detection algorithm." IEEE transactions on biomedical engineering 3 (1985): 230-236.
- 3 Stiber, Michael D, Stiber, Bilin Zhang and Larson, Eric C. Signal Computing: Digital Signals in the Software Doman. 2020.
- 4 Studier, James A., and Karl J. Keppler. "A note on the neighbor-joining algorithm of Saitou and Nei." Molecular biology and evolution 5.6 (1988): 729-731.
- 5 Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. "Data mining introduction." (2006).
- 6 Zelditch, Miriam Leah, Donald L. Swiderski, and H. David Sheets. Geometric morphometrics for biologists: a primer. academic press, 2012.

Assumptions and Pre-Processing

- Each file of input data must not contain more than one individual vocalization
- The vocalizations are assumed to have a discrete pulse-like structure
- Missing data is automatically excised so we assume that there are no empty files.