SoftwareX 6 (2017) 271–277

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Software for objective comparison of vocal acoustic features over weeks of audio recording: KLFromRecordingDays

Ken Soderstrom *, Ali Alalawi

Department of Pharmacology and Toxicology, Brody School of Medicine, East Carolina University, Greenville, NC 27834, United States

ARTICLE INFO

Article history:
Received 27 June 2017
Received in revised form 11 September 2017
Accepted 9 October 2017

Keywords:
Vocal learning
Acoustic analysis
Kullback–Leibler Distance
Phonology

ABSTRACT

KLFromRecordingDays allows measurement of Kullback–Leibler (KL) distances between 2D probability distributions of vocal acoustic features. Greater KL distance measures reflect increased phonological divergence across the vocalizations compared. The software has been used to compare *.wav file recordings made by Sound Analysis Recorder 2011 of songbird vocalizations pre- and post-drug and surgical manipulations. Recordings from individual animals in *.wav format are first organized into subdirectories by recording day and then segmented into individual syllables uttered and acoustic features of these syllables using Sound Analysis Pro 2011 (SAP). KLFromRecordingDays uses syllable acoustic feature data output by SAP to a MySQL table to generate and compare "template" (typically pre-treatment) and "target" (typically post-treatment) probability distributions. These distributions are a series of virtual 2D plots of the duration of each syllable (as *x*-axis) to each of 13 other acoustic features measured by SAP for that syllable (as *y*-axes). Differences between "template" and "target" probability distributions for each acoustic feature are determined by calculating KL distance, a measure of divergence of the target 2D distribution pattern from that of the template. KL distances and the mean KL distance across all acoustic features are calculated for each recording day and output to an Excel spreadsheet. Resulting data for individual subjects may then be pooled across treatment groups and graphically summarized and used for statistical comparisons. Because SAP-generated MySQL files are accessed directly, data limits associated with spreadsheet output are avoided, and the totality of vocal output over weeks may be objectively analyzed all at once. The software has been useful for measuring drug effects on songbird vocalizations and assessing recovery from damage to regions of vocal motor cortex. It may be useful in studies employing other species, and as part of speech therapies tracking progress in producing distinct speech sounds in isolation.

Software metadata

Current software version	*1.1*
Permanent link to executables of this version	https://github.com/soderstromk/KLFromRecordingDays/tree/master/cx_freeze/KLFromRecordingDays
Legal Software License	*MIT*
Computing platform / Operating System	*Microsoft Windows, 64-bit*
Installation requirements & dependencies	*Requires Sound Analysis Pro 2011,* http://soundanalysispro.com/ *The utility ParseSAPRecorderWavs is very helpful if Sound Analysis Recorder is used to generate *.wav files,* https://github.com/soderstromk/KLFromRecordingDays/tree/master/cx_freeze/ParseSAPRecorderWavs
If available Link to user manual-if formally published include a reference to the publication in the reference list	*Documentation and manual are incorporated to the manuscript. Installation instructions are included in the README.md file on the GitHub repository,* https://github.com/soderstromk/KLFromRecordingDays/blob/master/README.md
Support email for questions	soderstromk@ecu.edu

* Corresponding author.
E-mail address: soderstromk@ecu.edu (K. Soderstrom).
Abbreviations: SAR, Sound Analysis Recorder 2011; SAP, Sound Analysis Pro 2011

http://dx.doi.org/10.1016/j.softx.2017.10.003

Code metadata

Current Code version	*1.1*
Permanent link to code / repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-17-00050
Legal Code License	*MIT*
Code Versioning system used	*git*
Software Code Language used	*Python 3.5*
Compilation requirements, Operating environments & dependencies	*Requires Microsoft Windows, 64-bit*
If available Link to developer documentation / manual	*Documentation and manual are incorporated to the manuscript*
Support email for questions	soderstromk@ecu.edu

1. Introduction

Speech disorders afflict approximately 4% of children by the time they reach six years of age, and are often secondary to other more severe developmental disorders including intractable childhood epilepsy and autism [1]. These disorders are typically treated with speech-language physical therapy. Effective pharmacological interventions remain unavailable, possibly due to a lack of appropriate pre-clinical animal models. Very few animals learn a form of vocal communication in a manner similar to humans. Songbirds are among this small group of vocal learners. Thus, we have begun to develop a songbird, the zebra finch, as a laboratory animal model to evaluate drugs for effects to improve vocal learning and recovery from CNS damage. This model depends upon development of methods to objectively compare quality of vocalizations pre- and post-treatment over the course of weeks. Excellent software exists for recording and analyzing acoustic features of animal vocalizations (e.g. Sound Analysis Recorder and Sound Analysis Pro 2011 [SAP, [2]]), and good methods for comparing phonetic quality of vocalizations by KL distance measures have been described [3]. The goal in developing the current software was to integrate these approaches in a manner allowing efficient analysis of the totality of vocalizations produced over experiments lasting several weeks.

2. Problems and background

2.1. Problems solved

The software described solves the problem of applying KL distance methods of quantifying phonology (developed by [3]) to the literally millions of acoustic feature measures calculated by SAP that are derived from every syllable recorded from animals over a multi-week experimental period. The analysis is accomplished objectively, from audio recording to final output of KL distance measures.

2.2. Background

Sound Analysis Pro (SAP) is powerful, free, open source software developed to study songbird vocal development [2]. It includes modules that allow both recording and management of recordings in *.wav format, as well as analysis software that automatically segments sounds into the individual syllables uttered and analyses their spectral structure through calculation of 14 acoustic features (syllable duration, mean amplitude, mean pitch, mean FM, mean AM[2], mean entropy, mean goodness of pitch, mean mean frequency, pitch variance, FM variance, entropy variance, goodness of pitch variance, mean frequency variance, AM variance). SAP uses the open source MySQL relational database system to manage data. SAP provides for the export of these data in either native MySQL table or Excel spreadsheet formats. Others have developed an excellent approach to use SAP spreadsheet output to statistically compare changes in songbird phonology over time using KL distance measures [3,4]. These methods have already been successfully applied to studying effects of surgical manipulation of brain regions important to vocal learning and motor production [4–6].

In using songbirds as a preclinical animal model to screen drugs for effects on vocalizations, our initial intent was to apply the excellent SongSeq software already developed and made freely available by [4] (https://www.math.fsu.edu/~bertram/software/birdsong/JNM_12/, last accessed 9/5/2017). This software calculates KL distances in a manner similar to the KLFromRecordingDays program that we have developed. In addition, it features a powerful method to distinguish individual syllables and analyze the degree to which they are produced in a consistent sequence within songbird vocalizations. We continue to use this sequence analysis feature of SongSeq in our project.

In the case of SongSeq's KL distance measurement capability, two issues made the feature impractical for our work. The first issue is related to SongSeq's production of KL distance measures one probability distribution at a time. That is, when using syllable duration as an *x*-axis, only one of the remaining 13 acoustic parameters may be used as the *y*-axis to generate a distribution that is compared across days. Additional parameters require separate, sequential analyses that, due to the scale of our project, were time-consuming and created problems with organizing and summarizing results. The second and more significant problem we encountered in applying SongSeq to generate KL distance measures is attributable to its use of SAP-generated spreadsheets for data input. These spreadsheets are of the older *.xls file format limited to a maximum of 65,536 rows. As acoustic measures from each syllable processed by SAP occupies a spreadsheet row, and it is not uncommon to have hundreds of thousands of syllables produced during a single day of recording, in order to objectively analyze data from every syllable uttered we needed to establish KL distance measurement software that directly accesses the MySQL database system employed by SAP. The size of MySQL database tables are limited only by the memory resources of the computer used.

Given the open source nature of Python and availability of PyMySQL and openpyxl packages to interface with MySQL and Excel respectively, we chose this programming language (version 3.5) for development of our application. The software was developed to run on a 64-bit Windows system.

3. Software framework

3.1. Software architecture

The software is controlled by a graphical user interface derived from the Tkinter Python package (see Fig. 1). This interface collects information from the user including: The directory to store the resulting Excel spreadsheet of KL distance measures; the SAP-produced syllable table to analyze; the recording day(s) to use as the "template" (remaining days will be used as the "target"); the number of divisions to divide each axis of virtual 2D plots into (explained below); the minimum number of syllables required