Name ideas: RMDCS (raven mediated detection and classification system)

LFDCS: 1. Not optimized to reduce FPR, optimized to go fast

2. only on mac

3. interfaced through terminal, which makes it hard to look under the hood and tweak actual program

Mastor\_detecter

1. Optimized to reduce FPR, much slower computation
2. On PC, could probably port to mac
3. R script is interpretable and customizable by any researcher familiar with the language.

Autodetection tools are popular in passive acoustics to streamline manual analysis. Detector performance is commonly compared via TPR and FPR, but factors such as run time, ease of setup and use, and platform compatibility are also important considerations to performance. The LFDCS is a premier analysis tool in the field. It was designed to be compatible with near real-time detection on wave gliders so it is very computationally efficient. The trade off is that the learning method of the LFDCS (quadratic discriminant function analysis) only utilizes four features that do not contain information that allows for the discrimination of noise that can resemble positives for the target species. This can result in high FPR, particularly with environments and recorders that experience regular instrument noise, which is the case with AURAL recorders in the arctic. High FPR results in longer analysis time and more expensive analysis.

Despite not being optimized for this tradeoff from the perspective of acoustics labs working on archival data, the LFDCS is often used for this purpose. The LFCDS is also a Mac only implementation and interfaced through the command line, so ubiquity and customization for analysts is limited.

We submit an alternative approach for machine assisted analysis of low frequency sounds, optimized for efficient analysis time on archival data. This approach relies on a representative library of both the positives and negatives of a pitch tracker on ground truth reference data. It extracts measurements of each known and putative call to build and compare with random forest models. As you supply positive and negative detections, the models not only learn the identifying features of true positives, but also that of consistent types of false positives. In this way it ‘learns from its mistakes’. This is a flexible architecture that has been successfully applied to right whale upcalls and gunshots, as well as being able to discriminate from a variety of consistent sources of false positives due to noise. It is designed to be compatible with any stereotyped, distinct call type, and resilient in a variety of acoustic environments.

The detector is associated with Raven Pro 1.5 as part of its core functionality. The detector uses Raven Pro 1.5 Band Limited Energy Detectors (BLEDs) through API, and selection table outputs are formatted to be reviewed in Raven. The use of Raven software, as well as an implementation of the R language, make Mastor\_detecter comfortable to interface with and customize for scientists in the field.

Intro:

AURAL recorders in the arctic region can suffer from high levels intermittent ‘mooring noise’, making implementation of autodetectors challenging due to increased FPR. Popular autodetectors such as the LFDCS (which uses QDFA on 4 features: ) and Ishmael (which uses spectrogram correlation) are not resilient against false positives produced by intermittent noise as they don’t have the necessary features and learning system to distinguish noise from positive calls. High FPR hurts the implementation of an autodetector as it either 1. increases the time of an analyst to verify detections or 2. reduces TPR due to having to set a lower threshold for time efficient analysis.

Mastor\_detecter was created as a solution to the high FPR issue encountered with existing autodetectors on noisy arctic data. While the idea to model true positives from a band limited energy detector against null data is not new (flightcallr, Mellinger 2004), mastor\_detector is the first detector that models the true and false positives of a pitch tracker result to identify and weed out consistent sources of false positives from the results of a pitch tracker. This approach makes high TPR analysis feasible despite the high intermittent ‘mooring noise’ common to our AURAL detectors. As the models are spontaneously generated from a set of features in the ground truth library, this method should be applicable to other regions that experience high levels of intermittent interference on their recorders.

Mastor\_detecter has similar use limitations to existing pitch tracking detectors- it can be used to identify stereotyped, distinct calls that preferably feature FM, but is vulnerable to anything that will mask or obscure the shape of calls. It will not produce reliable tracks in acoustic environments with heavy/complex biotic environments (bowhead, bearded seal regimes). It cannot use species presence or patterning of calls as a criteria for detection, so manual verification is required in cases where the call type in question may be attributable to multiple species (ie NPRW upcalls and bowhead, humpback, bearded seal upsweeps).

Mastor\_detecter is easy to use, but hard to set up. Detectors must be created manually in Raven, and criteria for pitch tracking must be coded in R language. Mastor\_detector will never be an executable program, but rather a flexible infrastructure that can be customized to various call types. Due to high runtimes, it is not meant to be applicable to real time or near real time detection but an efficient method for reducing the analyst time needed for boxing desired calls within archival acoustic data.

We have successfully applied Mastor\_detector to right whale upcalls and gunshots and demonstrate results for these applications, as well as provide guidance for its implementation to any distinct, stereotyped signal of interest.

Methods:

Construction and design:

We identified the Raven BLED detectors as a good starting place to assess the viability of applying energy detectors to our data, given the comfortability of the Raven interface. Upon finding an R implementation to call this algorithm from API, we decided to continue developing around Raven and the R language. It was determined that in environments where the amplitude of mooring noise was greater than that of individual calls, wide band limited energy detectors worked poorly for pulling out faint calls, whereas more narrowband detectors were more likely to hit pieces of faint calls. With this knowledge, we pursued using a ‘suite’ of narrowband BLEDs throughout the frequency range of the desired call. Using the suite of detectors allowed for an additional ‘pitch-tracking’ filter, by comparing the frequency and time of each hit to assess FM of the signal. Using a simple pitch tracker garnered too many FPs to make analysis feasible, so a machine learning solution using RF models inspired by the detector design in (flightcallr) was implemented to help weed out FPs. Image analysis was later introduced to improve the classification performance of the RF models. Parallelization is implemented while running the pitch tracking algorithm, extracting features, and building the models, and is highly recommended to improve run time. The infrastructure for these parallelized sections is in theory portable to Azure.

There are many parameters that allow you to control and fine tune the script (parameters, supplemental). There are too many parameters to optimize automatically, so optimization is best fine tuned by hand.

Building