**HethFinder**

--a program for automated selection of Hermit Thrush Song--

**Overview**

HethFinder uses a machine learning model to search through wave files for hermit thrush song and label each song’s introductory note.

Inputs:

* .wav file(s) that contain hermit thrush song
* a hethfinder\_params.csv file with wave files listed and (optional) start and stop times if you prefer to analyze only a portion of the file for faster, more accurate, performance. It is recommended that you preview the spectrogram beforehand and input a region where clear hermit thrush song occurs.

Output: a .txt file that is formatted to be opened in Raven as a selection table. HethFinder will attempt to identify each hermit thrush song by placing a box around the intro note portion of the song.

**Installation**

Currently, HethFinder requires that python be installed on the user’s computer. HethFinder was developed on python version 3.9.12 but other versions are likely to be compatible. Other python libraries need to be installed as well, including tensorflow and librosa (the other imports such as numpy and pandas are sometimes included with a package manager). I recommend using anaconda to manage python and its dependencies. To download anaconda and learn more, go to: <https://www.anaconda.com/download>

**Usage**

1. Setup
   1. Begin with a folder that contains the following files:
      1. the wave file(s) to be analyzed.
      2. the HethFinder.py file.
      3. the hethfinder\_params .csv file.
   2. Open hethfinder\_parameters.csv and enter the .wav filename(s) into the filename column. Enter all filenames to be analyzed. Do not change anything else about this file, such as column names.
   3. (Optional) In corresponding columns, enter a start time and end time for each wave file. If either is left blank, HethFinder will assume that the start time is 0 and the end time is the end of the .wav file. HethFinder will run more quickly (and accurately) if it does not need to evaluate regions that are noisy or have little to no hermit thrush song.
2. Run HethFinder
   1. Open HethFinder.py in whatever code editor you use. Popular options are VSCode, Idle, or Jupyter Labs.
   2. Run HethFinder. It will take approximately as long as the selected recordings. For example, each ten-minute file will take approximately ten minutes to analyze. Set it and forget it! HethFinder will produce a .txt file for each .wav file it analyzes. These can be opened in Raven as selection tables.
3. Review and Manually Edit
   1. Open each new selection table and wave file in Raven. Check the selections for errors and fix them if need be. Depending on the research question and amount of error, editing may or may not be necessary. HethFinder accuracy varies depending on several factors, including signal strength, background noise, song-types, and presence of countersinging (for more information, see accuracy tests below).

**How HethFinder Works**

HethFinder uses a combination of machine learning and post-processing. There are three stages that it runs through on each recording, described briefly below:

***Machine Learning Model***

First, HethFinder moves along the entire spectrogram of the wave file, taking pictures that are ~1.5 seconds long (a little longer than a HETH song) every 46 milliseconds. There is obviously a lot of overlap between these pictures. Imagine a camera with a 1.5s viewport sliding along the spectrogram from left to right rapidly taking pictures. Every snapshot is checked by the machine learning model and assigned a probability from 0 to 1. This number represents how likely it is that the snapshot contains a hermit thrush song *that begins at the left edge of the frame*.

The machine learning model was trained on 36,000 such pictures, half of which were real hermit thrush song and half of which were random non-song pictures from the same recordings. Since there is variability in how accurately and precisely researchers have selected the beginnings of these training songs, there is also variability in how confident the ML model is that it has found the exact beginning of a song. The output from stage one is a list of probabilities corresponding to each 46ms moment of the .wav file.

The graph below and corresponding spectrogram demonstrates approximately 15 seconds of these results. You can see that at around 359s (5:59 on the spectrogram) HethFinder starts to detect a song, then becomes very confident for about half-a-second, then tapers off. This is a typical signal for a real HETH song. Other songs with similar signals can be seen at 355.5s (5:55.5) and at 369s (6:09). The other lesser signals in the image will be discussed below.

A graph of a graph

Description automatically generated with medium confidence

A greyscale shot of a white background

Description automatically generated with medium confidence

***Post-Processing***

As can be seen in the graph above, stage one does not provide enough information to know precisely and accurately where HETH songs are. It merely provides moment-to-moment probabilities. The post-processing stage takes this list of probabilities and turns it into a list of timestamps. It considers the magnitude and duration of the plateaus and spikes seen on the graph above, and then the timing of these plateaus and spikes.

For example, consider the small spike just before 358s (5:58). In theory, this could be an atypical HETH song-type, or a HETH song with noise overlapping, or a particularly faint HETH song, or maybe a vocalization from a different species of bird. The signal strength alone does not provide enough information for HethFinder to decide which option is best. However, since a hermit thrush typically sings at a steady rate, HethFinder calculates the median song rate on the recording, and gives higher priority to signals that occur roughly that distance from other strong signals. Since this small spike at 358s (5:58) occurs only a second before the strong signal at 359s (5:59), HethFinder decides it is too rapid for typical hermit thrush singing, so it correctly discards that signal.

However, the signal at 362s (6:02) is further away, closer to the median song rate, so it is correctly included as HETH song. The tiny signal at 364s (6:04) is also discarded because the signal is simply too small, regardless of timing. On the spectrogram, we can see that this is a very faint vocalization that may even be a countersinging hermit thrush in the background.

The output of stage 2 is a list of timestamps corresponding to when HETH songs are likely to occur.

***Intro-note Selection***

If everything has gone well, each timestamp from stage 2 should be within ~500ms of an actual HETH song. The purpose of stage 3 is to narrow the temporal precision and find the frequency of the intro note. Like at stage 1, HethFinder again takes overlapping pictures. This time, it uses a smaller camera frame to sweep and filter the local region for the most acoustically active, hermit-thrush-like area, which should correspond to the post-introductory portion of the song.

It then uses an even smaller, short-but-wide camera frame to sweep up and down the area preceding the post-introductory portion, searching for an intro note. This is typically a long horizontal line on a spectrogram. HethFinder prioritizes the loudest line and the leftmost line, striking a balance between the two if they conflict. Once it chooses a line, it outputs the timestamp at the beginning of the line and the pitch frequency of the middle of the line.

The output of stage three is a raven selection table with timestamps and frequencies corresponding to each selected song.

**Accuracy Tests**

At each of the three steps listed above, there are assumptions and guesses. Because of these guesses, HethFinder varies in accuracy depending on context. Therefore, it is up to the user to decide when it would be beneficial to use HethFinder and when it would be better to analyze recordings manually. The following tests were performed to give the user a sense of what to expect from HethFinder depending on three factors: focal bird signal strength, heterospecific interference levels (noise), and conspecific interference levels (countersinging). Each recording was visually scanned before analysis and designated a 3-part score, such as “2A3”:

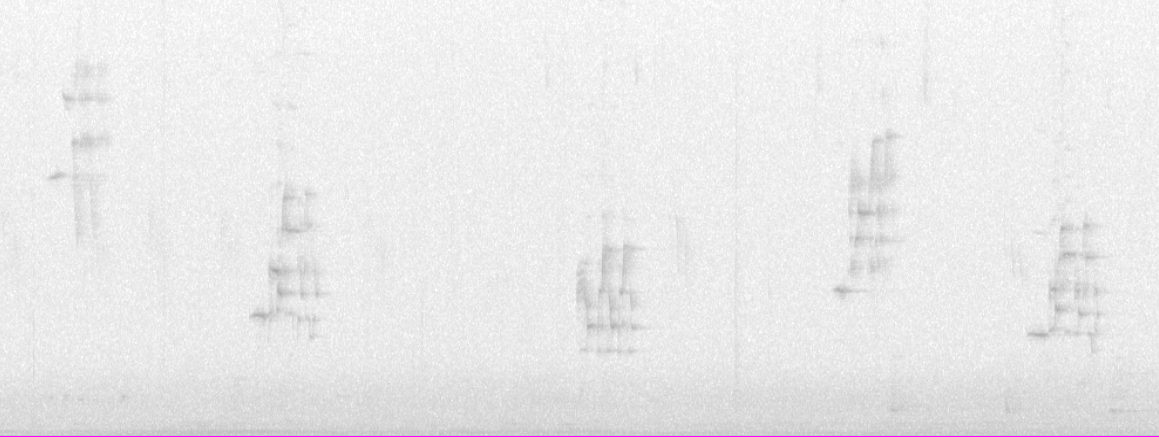
Slot 1) The first digit is from 1-3, with 1 being a strong signal of the focal bird and 3 being faint. In the example above, the ‘2’ suggests that the focal bird signal is moderate.

Slot 2) The letter in the second slot refers to background noise/interference, with A being clean or minimal background noise and C being lots of background noise.

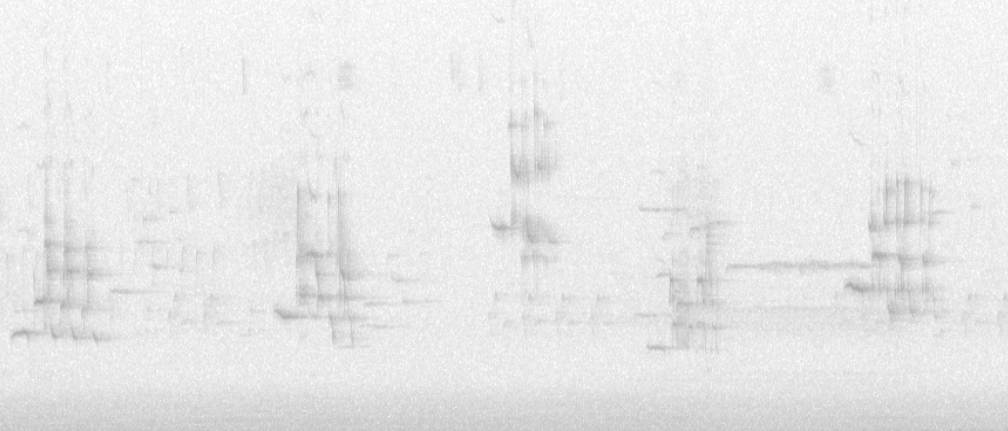
Slot 3) The third slot is another digit from 1-3 and refers to conspecific signals or countersinging. The 3 in the above example suggests that there is a faint conspecific in the background of the recording. In cases where there is no countersinging present, the third digit is discarded.

In summary, the designation 2A3 refers to a recording with a moderate focal signal, very little background noise, and some faint conspecific singing in the background. Here are some sample spectrograms with their designations for reference:

1A (strong signal, little-to-no background sounds)



1B2 (strong signal, some background sounds, a moderate countersinger)



2B (a moderate signal, moderate background sounds)



3C (weak signal, lots of background sounds)



HethFinder has been tested in various combinations of these conditions and the two most relevant performance variables have been measured in each case:

**True positive rate (TPR):**

\*Note: ‘correctly identified’ means that the selection was within 0.2 seconds of the start of the actual intro note (as measured by a human inspecting the spectrogram), and within 1000 Hz of the median pitch of the actual intro note.

**False positive rate (FPR):**

Important information about the recordings used in the following tests:

* All recordings were from birds that HethFinder’s machine learning model was *not* trained on. Using the birds from the training data would give HethFinder an unrealistic advantage.
* All the recordings were from automated field recorders passively recording hermit thrush territories. A better signal-to-noise ratio (and correspondingly better results) would be expected from directional microphones pointed at singing birds.

Results:

**Table 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bird | Quality | False Positive Rate (%) | True Positive Rate (%) | # of Songs | Duration (min) |
| Cottage1 | 1A | 0 | 96.7 | 212 | 15 |
| Frye6 | 1A3 | 1.27 | 92.36 | 157 | 11 |
| Cottage1 | 1B | 5.2 | 96.2 | 500 | 37 |
| Salmon1 | 1B | 5.77 | 80.77 | 104 | 7 |
| Cottage1 | 1B | 0 | 100 | 102 | 7 |
| Frye11 | 1B | 4.17 | 83.33 | 48 | 5 |
| Glebe3 | 1B2 | 4.41 | 94.12 | 136 | 10 |
| Frye11 | 2A | 3.33 | 88.33 | 60 | 8 |
| Frye6 | 2A2 | 0 | 86.57 | 67 | 5 |
| Salmon1 | 2B | 5.88 | 76.47 | 85 | 5 |
| Frye11 | 2B | 14.47 | 67.11 | 152 | 16 |
| Frye11 | 2C | 9.47 | 75.15 | 169 | 18 |
| Cottage1 | 3B | 3.26 | 88.04 | 92 | 7 |
| Glebe3 | 3C | 10.07 | 63.31 | 139 | 12 |

A graph of a bar chart

Description automatically generated with medium confidence

**Figure 1**

As can be seen in the above graph, there is considerable variation in both FPR and TPR. This variation matters. The difference between a TPR of 100% and 63% is the difference between a researcher quickly scanning for any errors or painstakingly adjusting every third selection. Considering that it takes at least twice as long for a researcher to adjust a selection as it does for them to create a selection, there is an accuracy threshold where HethFinder is no longer worth using. Based on limited experience, I estimate that this threshold is somewhere between a TPR of 70% and 80% (the red dotted lines in the graph).

Since many of the samples above have a TPR near this threshold, it is important to know when it is worth using HethFinder. It would be frustrating to have a computer spend hours running HethFinder only to find that the resulting selection tables are worse than no selection tables at all. The results above suggest that as the focal bird signal diminishes (left to right) and background noise increases (left to right), the TPR (orange) decreases and the FPR (blue) increases. **This suggests that HethFinder is most promising for recordings with a large signal-to-noise ratio.**

In the few samples tested, countersinging did not have much effect. This is likely because the countersingers were much quieter than the focal bird in the samples tested. A recording with a designation such as 1A1 (loud focal bird and loud countersinging bird) would likely have many missed songs from both birds, depending on their timing and overlap. **HethFinder currently has no algorithm for choosing between birds.**

Another important variable in Figure 1 is individual bird: HethFinder performs differently across individuals. For example, it consistently performs much better on Cottage1 than Salmon1, regardless of the variations in signal and noise (see Table 1). In such cases, there are usually one or two atypical song-types that repeatedly cause most of the errors. Usually in these cases, a piece of the post-introductory portion of the song is mistakenly selected as the intro note. HethFinder identifies the song but struggles to locate the intro note. So, birds with strange songs have poorer results.

Current recommendations:

1. Confidently use HethFinder with recordings that have a strong signal and little background noise. Selection tables may require manual editing, but this should be much less work than manually creating a selection table.
2. Recordings with a faint signal and/or lots of background noise may not be worth it.
3. If a bird has atypical song-types, expect errors on those songs, but it is likely still worth using HethFinder (assuming a strong signal-to-noise ratio).

**Contact and Additional Info:**

Contact: HethFinder was created by Luke McLean (MA candidate) for the Sean Roach Laboratory at the University of New Brunswick – Saint John Campus. Inquiries about the software can be sent to [lmclean@unb.ca](mailto:lmclean@unb.ca)

Future Versions: It is my hope that I will continue to improve HethFinder as I use it and find time to work on it. As the above accuracy tests demonstrate, there is certainly room for improvement! I also intend to publish the code on GitHub. If you’d like to help or have questions, don’t hesitate to contact me.

HethSorter: Finding songs is usually only the first part of analyzing hermit thrush song. After HethFinder finds the songs, HethSorter sorts them into labelled song-types. So, if you have songs that need labelling, be sure to check out HethFinder’s best friend and companion, HethSorter.