



Acoustic analysis with BirdNET and (almost) no coding: practical instructions

Laurel Symes, Larissa Sugai, Ben Gottesman, Michael Pitzrick, Connor Wood, Russell Charif

Last revised: 2024-08-04

To cite the use of the BirdNET model in a scientific publication, please use the following reference:

Kahl, S., Wood, C. M., Eibl, M., & Klinck, H. (2021). BirdNET: A deep learning solution for avian diversity monitoring. Ecological Informatics, 61, 101236.

To cite the use of additional guidance or suggestions from this document, please cite:

Symes L, Sugai LSMS, Gottesman B, Pitzrick M, Wood C, Charif, R. 2024. Acoustic analysis with BirdNET and (almost) no coding: practical instructions.

DOI: 10.5281/zenodo.8357176

Contents

1 Introduction 2
1.1 About BirdNET 2
1.2 About this document 2
1.3 Ways to use BirdNET 3
1.4 How BirdNET works (overview) 4
2 Applying the BirdNET model in Raven 6
3 Applying the BirdNET model using the BirdNET GUI 12
4 Threshold setting in BirdNET 15
4.1 Exporting clips across a range of Confidence Scores 16
4.2 Validating clips using Raven Selection Review 20
4.3 Logistic Regression 36
5 Building a custom BirdNET model 40
5.1 Example: training a custom model in the BirdNET GUI 41
5.2 Apply your model 44
5.3 Examining the output of a custom BirdNET model 46
5.4 Improving the performance of a custom BirdNET model 48
6 Literature cited 50
7 Glossary 51

1 Introduction

1.1 About BirdNET

BirdNET is an open-source deep artificial neural network model for identifying bird vocalizations to species in recorded or real-time audio data. BirdNET is designed to enable analysis of large audio data sets in support of biodiversity assessment and monitoring. BirdNET has been deployed on various hardware and operating systems such as Arduino microcontrollers, the Raspberry Pi, smartphones, web browsers, workstation PCs, and even cloud services.

As of August 2024, BirdNET (v2.4) has been trained on sounds of >6,000 bird species worldwide. BirdNET now also enables users to create customized models by adding training data for additional sound classes.

Development of BirdNET is an ongoing project of the [K. Lisa Yang Center for Conservation Bioacoustics](#) at the [Cornell Lab of Ornithology](#) in collaboration with [Chemnitz University of Technology](#). Current development efforts focus on improving the scope and quality of the BirdNET model, including non-avian sounds, and on cross-platform interfaces to make BirdNET more easily accessible to users with no background in computer science.

1.2 About this document

This document is intended primarily for scientists and conservation practitioners using passive acoustic monitoring (PAM) to assess biodiversity in natural environments. No specific knowledge of computer science or machine learning is assumed. Some acquaintance with Raven Pro software for visualizing and annotating sound files is helpful, but not essential. For those who have never used Raven, the introductory tutorial videos [here](#) may be helpful.

This document covers:

- A conceptual overview of how BirdNET works, and the ways currently available to use BirdNET without writing any code.
- Using BirdNET via Raven's Machine Learning Detector interface or via the BirdNET Analyzer Graphical User Interface (GUI; currently available only under Windows).
- How to set threshold prediction values for BirdNET's confidence scores to ensure that BirdNET species predictions have a specified minimum probability of being correct.
- How to build a custom BirdNET model, for example to add sound classes that are not currently included in the base model.

- Tips on how to improve the performance of custom BirdNET models.

Readers are advised to note the revision date above and be aware that some parts of this document may be out of date when you are reading it due to software updates. Text and screenshots in this document are based on BirdNET model v2.4, and BirdNET Analyzer GUI v1.2.0.

1.3 Ways to use BirdNET

There are currently three ways to use the BirdNET model to identify and label bird sounds in single sound files or large sets of many files:

- **Raven Pro Machine Learning Detector interface:** [Raven Pro 1.6.5](#) by default includes v2.2 of the BirdNET model, which can recognize >3000 species of birds. As of August 2024, BirdNET in Raven is available under Windows, and under macOS running on Intel CPU chips. Analysis in Raven Pro works well for small numbers of files, and when you wish to immediately visualize results. Using BirdNET via Raven is discussed in Section 2 of this document.
- **BirdNET Analyzer GUI:** The standalone BirdNET Analyzer Graphical User Interface (GUI) is available for v2.4 of the BirdNET model, which can recognize > 6000 species of birds worldwide. For most applications, the Analyzer GUI is the most convenient way to run the BirdNET model on large numbers of sound files. As of July 2024, the BirdNET Analyzer GUI is available only under Windows; a version for macOS is currently in development.¹ Using the BirdNET Analyzer GUI is discussed in Section 3 of this document.
- **Python command line interface (CLI):** Users who are comfortable working with the Python command line can access all capabilities of the BirdNET Analyzer that way. Using BirdNET via Python is outside the scope of this document; more information is available at the [BirdNET Analyzer GitHub repository](#).

¹ The BirdNET GUI can be run on a Windows virtual machine under [Parallels](#) on macOS.

	Operating System (hardware)				Documentation
	Windows	macOS (Intel CPU)	macOS (Apple M1)	macOS (Apple M2)	
BirdNET via Raven ML Detector interface	•	•	x	x	this document
BirdNET Analyzer (GUI)	•	x	x	x	this document
BirdNET Analyzer (Python command line)	•	?	•	•	GitHub Repository
	• = yes	x = no	? = uncertain		

Figure 1. Availability of BirdNET interfaces on various platforms as of July 2024.

1.4 How BirdNET works (overview)

This section provides a high-level practical overview of how BirdNET processes sound files to detect and label the presence of bird sounds.

Target species lists

Because BirdNET has been trained to recognize sounds from thousands of species of birds from around the world, it's usually helpful to provide BirdNET a list of *target species* that you would like it to identify in a given recording. In some cases you may only be interested in finding one or a few species. But even if you want BirdNET to identify all species present in a recording, these will be only a small subset of the thousands of species in the BirdNET model, depending on the location and time of year when the recording was made. Narrowing the scope of target species can reduce the likelihood of BirdNET misidentifying sounds. The details of how to tell BirdNET what species to look for are different depending on whether you're using Raven or the BirdNET Analyzer GUI, and are discussed in the corresponding sections below.

Input sound file format

By default, BirdNET operates on single-channel sound files recorded with an audio sampling rate of 48 kHz. Files saved in any standard audio format can be processed. Files recorded at non-default sample rates are automatically re-sampled to 48 kHz. For files that contain more than one audio channel, BirdNET merges the channels before processing.

Segmented processing, confidence scores, and prediction labeling

BirdNET processes recordings in 3-second segments. For each segment, the BirdNET model computes a *confidence score* for each species in the target species list. Confidence

scores, which range between 0 and 1, indicate the relative likelihood that a particular recording segment contains a sound from a given target species. For each target species, higher confidence scores indicate a higher probability that the segment contains a sound from that species. However, the confidence score for a species is not, strictly speaking, the probability that a sound from that species is present in the segment. More information about the relationship between confidence scores and probabilities is provided in Section 4, *Threshold setting in BirdNET*.

When the confidence score for a target species exceeds a specified threshold, BirdNET outputs a *prediction label* for that species in that segment. Because the confidence scores for more than one species may exceed the threshold, BirdNET can predict multiple species in a single 3-second processing segment. For each prediction label, BirdNET also outputs its associated confidence score. Species prediction labels from BirdNET are also called *detections*.



Warning: Confusing terminology

*If you intend to use acoustic detection events from BirdNET for occupancy modeling of species presence, be aware that the term **detection** has a specialized definition in the context of occupancy modeling that is different from its usage in the context of automated signal detection by algorithms like BirdNET. Importantly, a single acoustic detection event does not necessarily equate to a species detection in occupancy modeling, and multiple acoustic detection events may be consolidated into a single species detection.*

Output format

For each sound file that BirdNET processes, it creates a Raven selection table containing one selection for each species prediction. Each 3-second selection corresponds to one BirdNET processing segment in the original sound file, and includes annotation columns containing the prediction's species label and confidence score. When BirdNET predicts sounds from multiple species in a single 3-second segment, it creates one selection for each prediction.

When you run BirdNET using Raven's machine learning detector interface, the output selection table is created and appears in Raven immediately. When using the BirdNET Analyzer GUI, selection tables are saved as tab-delimited text files that can then be opened for review in Raven, or processed further using other software.

2 Applying the BirdNET model in Raven

Analysis in Raven Pro works well for small numbers of files, and when you wish to immediately visualize results. For processing large batches of files, the BirdNET Analyzer GUI is more efficient.



Bug Warning

Do not use Raven Pro's Batch Detection option with BirdNET (or any of the other available Learning Detectors). Due to a bug, Raven may unpredictably skip sound files in the input directory.

This guide provides basic instructions for running BirdNET on single sound files in Raven. More extended instructions and options are available [here](#).

Raven Pro 1.6.5 comes with v2.2 of the BirdNET model (trained on > 3,000 species) pre-installed. To install BirdNET v2.4 (>6,00 species) into Raven, follow the instructions [here](#).

System requirements: Raven Pro 1.6.5 running under Windows or under macOS on an Intel processor.



BirdNET is not currently (August 2024) available in Raven under macOS on Apple Silicon (M1 and M2) chips.

1. Open Raven Pro (Version 1.6.5).
2. In Raven, choose File > Open to open the recording(s) that you wish to analyze, or drag the recording into the Raven window.
3. To access the BirdNET detector, choose Tools > Detector > Learning Detector.
4. In the Choose Detector Input dialog that appears (Figure 2), you will be asked whether you want to use the waveform or spectrogram as input. For BirdNET, you will use the waveform as input. Select this view in the right column, then select the arrow in the middle of the window to move the waveform view into the left column (under Required Inputs).

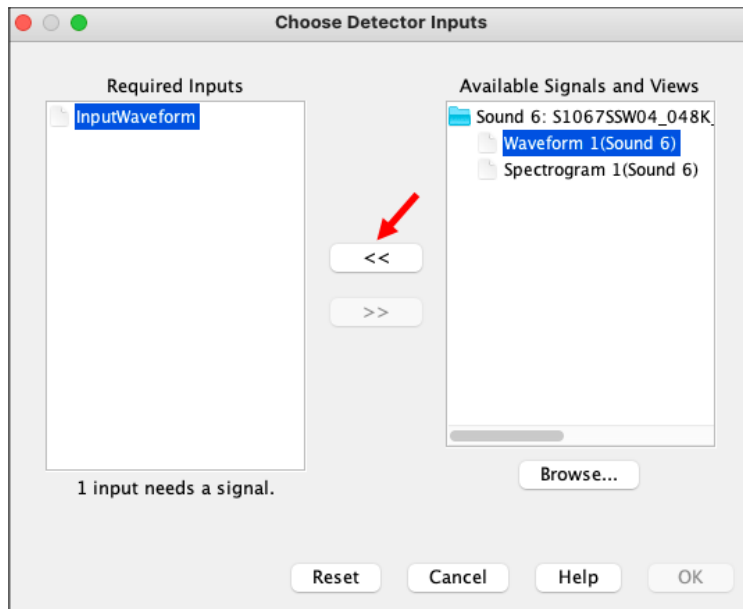


Figure 2. Choosing detector inputs.

5. Click OK.
6. The next dialog will ask you to “Configure the Machine Learning Detector” (Figure 3). On the Inputs tab, use the “Select Model” menu to choose the BirdNET model from the list of options. (If you installed BirdNET v2.4, there will be two BirdNET options available-- “BirdNET_GLOBAL_3K_V2.2” and “BirdNET_GLOBAL_6K_V2.4_Model_Raven”.) The other choices in this tab (Model Signature, Model Input and Model Output) will then auto-complete.

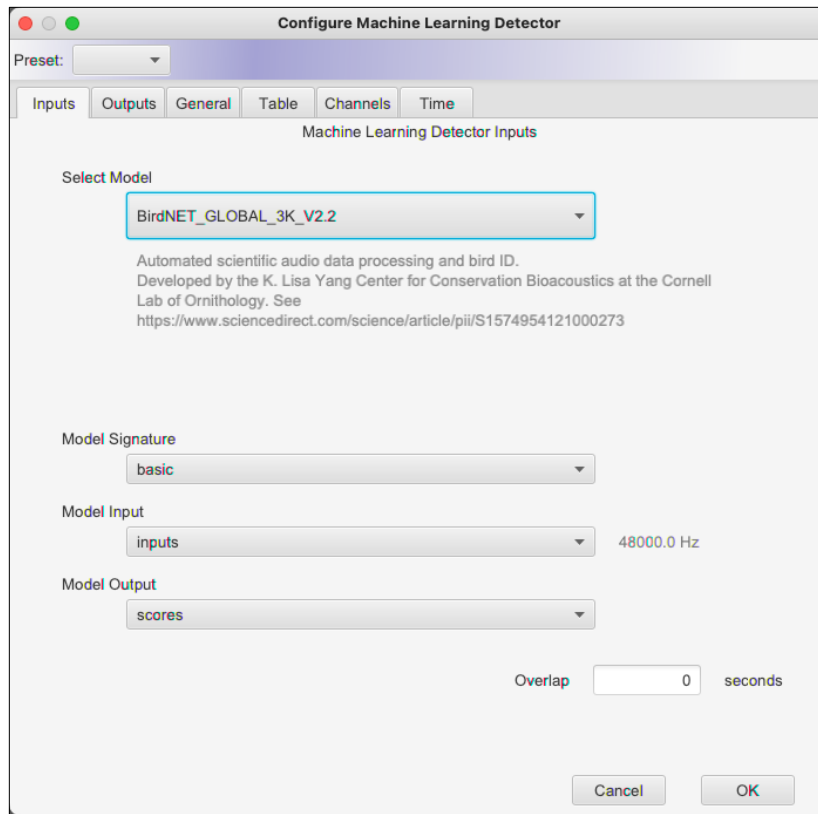


Figure 3. Selecting the BirdNET machine learning model.

7. Go to the “Outputs” tab to define the list of target species that you would like to detect (Figure 4). Select a species list using the “Output Class File” menu. This menu includes the global BirdNET species list, *birds_global.csv*, as well as several regional species lists such as *birds_europe.csv*, *birds_south_east_asia.csv*, and *birds_us_north_east.csv*. Using a regional species list that matches the recording location of your data will provide the best results, since this prevents BirdNET from mis-labeling sounds as similar-sounding species from far away. The “Label File” menu allows you to configure how you want the predictions to appear in the output selection table (common names across a few languages, eBird codes, etc.).

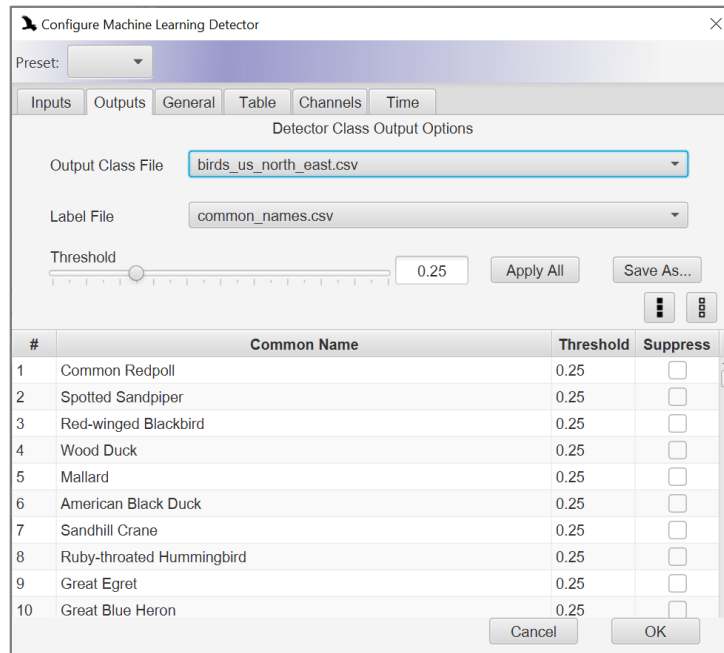


Figure 4. Configuring detector class outputs.

- You can optionally further refine the target species list by suppressing predictions for species that you don't want included in BirdNET's output. There are two common use cases in which you might want to suppress predictions for some species:

Detect all species in a recording

The output class files that are included for BirdNET by default when Raven 1.6.5 is installed are for large geographic regions (e.g., Europe, Southeast Asia, Northeast United States). Because of smaller-scale variation in species ranges and habitats, recordings from a particular site are unlikely to contain all the species listed in any of the output class files. In many cases the extra species listed in a class file that would not be found at your site (for example, shorebirds in a forest) don't result in false detections by BirdNET, so there is no need to suppress those detections. However, if you find in reviewing BirdNET's output that it is predicting species that you're sure are not present in your recording, you can re-run the analysis with predictions suppressed for those species. Simply click the small box next to the species in the "Suppress" column to hide predictions for a particular species.

Detect one or a few species

If you wish to detect only one or a few target species, start by suppressing all species, and then unsuppress the few species you want to detect. To suppress all species in the current output class file (regional list), click the button with three black dots above the Suppress column. Then scroll through the list of species to find each of your desired target species, and unsuppress each one by clicking on the checkmark in the corresponding Suppress box.

9. Also in the Outputs tab (Figure 4), you can set the global score threshold to use for all species. Only predictions with confidence scores above the specified threshold will be returned. You can change the threshold either by typing into the text box, or moving the slider, then clicking “Apply All”. Remember that BirdNET confidence scores are not probabilities that the prediction is correct. The relationship between confidence scores and probabilities that a prediction is correct is discussed further in Section 4, *Threshold setting in BirdNET*. You can set different confidence thresholds for different species by editing the output class file as described [here](#).
10. To select where the selection tables containing predictions will be saved and how they are named, go to the “Table” tab (Figure 5). Click “Browse” and navigate to the location where you would like the files to be saved.

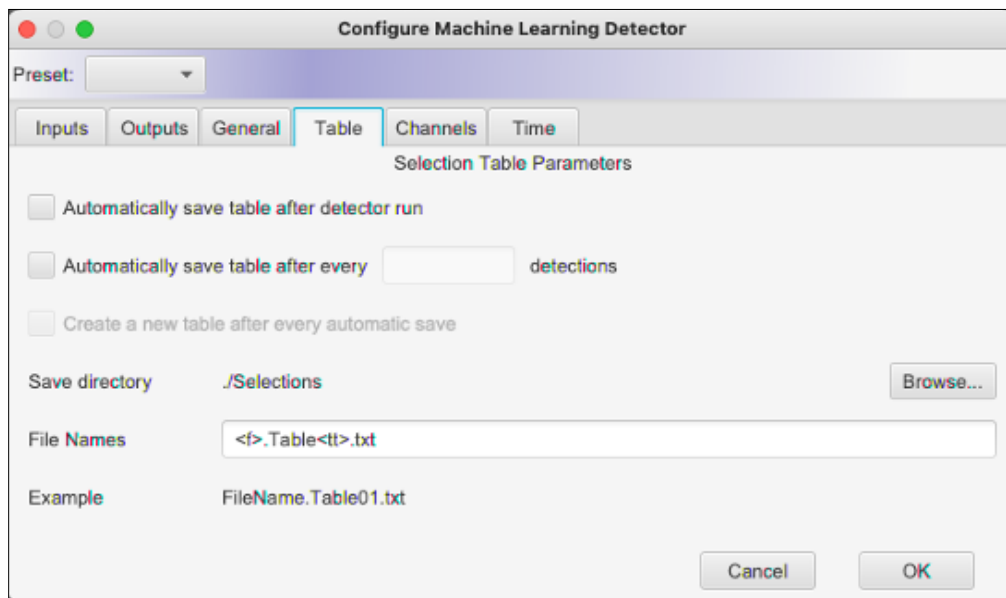


Figure 5. Configuring where and how selection tables are saved.

11. If you wish, you can use other tabs to specify the recording channels that should be analyzed (“Channels” tab), or particular time windows for analysis (“Time” tab).

12. Now click the OK button. The model will begin to run. Note that for several seconds the elapsed time will increase, but it looks like nothing else is happening and the progress bar does not advance. It may take 5-15 seconds for the data processing to begin. Once processing is complete, the progress bar will be full and the progress manager will say “Finished”.
13. Any predictions generated by BirdNET will be shown in the selection table below the sound file (Figure 6) and will be saved in the designated location.

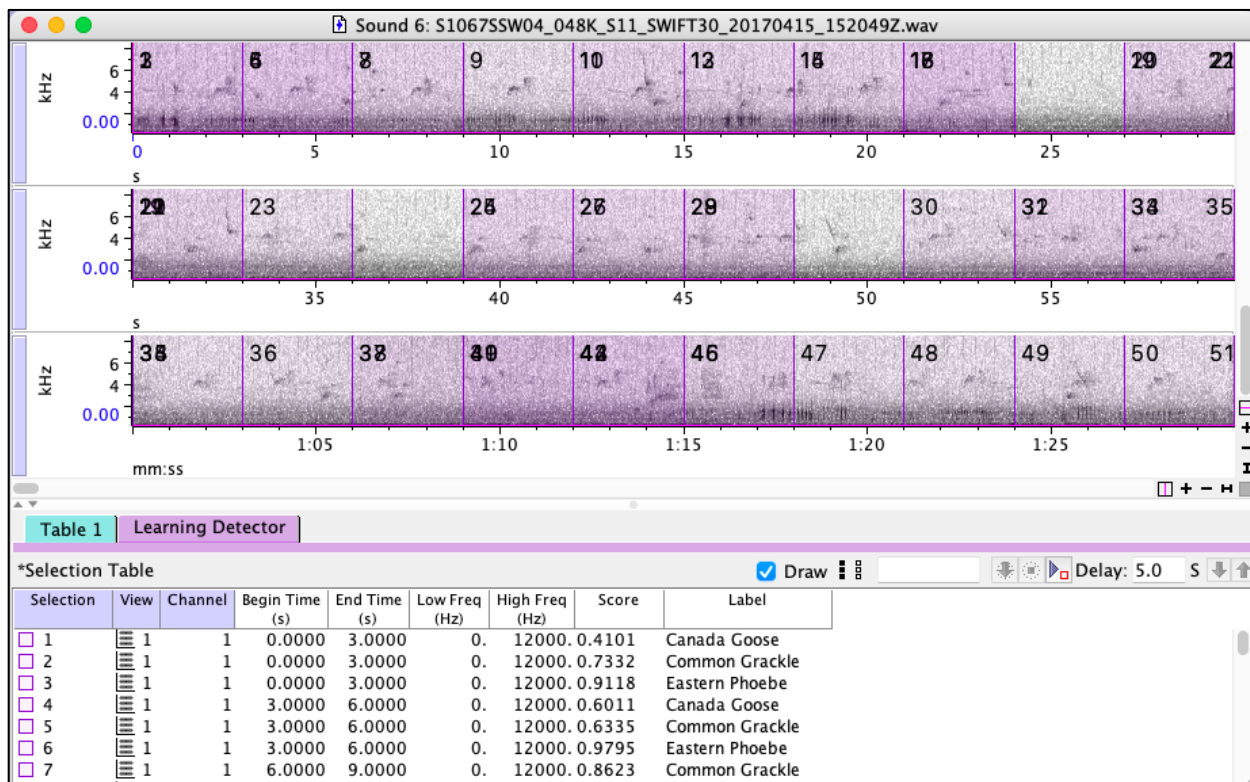


Figure 6. Results of running the BirdNET model on a 1-hour sound file. Only the first 90 seconds are shown in the spectrogram. BirdNET creates one selection for each species that it detects in each 3-second segment, so some segments have multiple selections.

14. Note that each selection is three seconds in length (corresponding to the processing window of BirdNET). The detections will have a “Label” column with the predicted species identification and a “Score” (a value between 0 and 1). For a given species, predictions with higher confidence scores are more likely to be correct than those with lower scores (which are more likely to be background noise or a similar species). Again, remember that *BirdNET confidence scores are not the probability that the prediction is indeed the suggested species*. Some species may be correctly identified when the score is 0.25, while others are rarely correctly identified if the

score is less than 0.99. For additional information, please see Section 4, *Threshold setting in BirdNET*.

3 Applying the BirdNET model using the BirdNET GUI

Background: The BirdNET GUI (Graphical User Interface) provides a user-friendly tool for processing large batches of files.

Hardware required: Computer running Windows.²

Software Required: BirdNET GUI; the GUI version documented here is v1.2.0

1. Begin by downloading and installing the BirdNET GUI as described in this [video](#).
2. Open the BirdNET GUI.
3. To process a batch of files, select the “Batch analysis” tab (Figure 7).

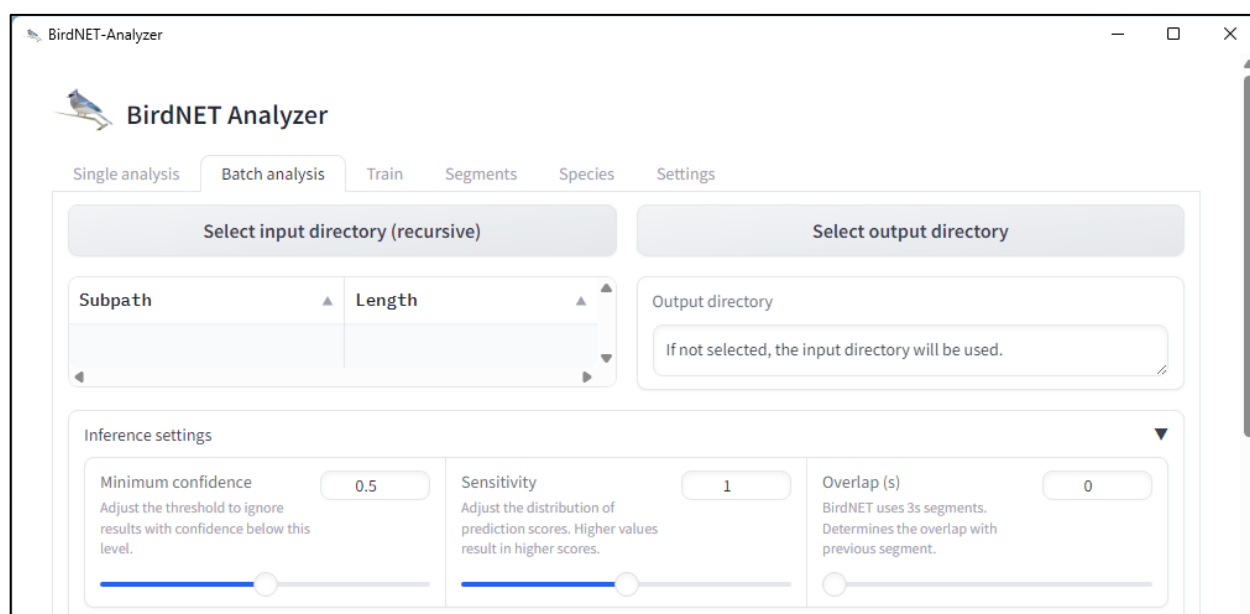


Figure 7. The “Batch analysis” tab in the BirdNET Analyzer GUI.

4. Click the “Select input directory (recursive)” button (Figure 7) and navigate to the directory containing the sound files to be processed.

² The BirdNET Analyzer GUI can be run on a virtual machine running Windows on a Mac using [Parallels](#).

5. Select “Select output directory” and select where prediction files should be saved.
6. Under “Inference settings” (Figure 7), set the “Minimum Confidence” to specify the confidence score threshold to use for all species. Only predictions with confidence scores greater than or equal to this threshold will be returned. You can change the threshold either by typing into the text box, or moving the slider. Remember that *BirdNET confidence score are not probabilities that the prediction is correct*. The relationship between confidence scores and probabilities that a prediction is correct is discussed further in Section 4, *Threshold setting in BirdNET*.
7. For most applications, “Sensitivity” and “Overlap” can be left unchanged.
8. Use the “Species selection” panel (Figure 8; you may have to scroll down in the Batch Analysis tab to see this panel) to determine which target species list BirdNET should use. In most cases, the best option is to select “Species by location” and then specify the latitude and longitude where the sound files were recorded and the week of the year. BirdNET uses data from eBird to construct a list of species likely to occur at your recording location for the selected week of the year. When “Week” is set as -1, all weeks of the year are included.

Species selection

Species list
Filter species that are included in output.

Custom species list

Species by location

Custom classifier

All species

Latitude
Recording location latitude. 0

Longitude
Recording location longitude. 0

Year-round

Week
Specify the week of the year the recording was made, using a simplified system where each month is divided into four weeks. Choose a value from 1 to 48. 1

Location filter threshold
Minimum occurrence probability for a species to be included. 0.03

Figure 8. Target species list selection in BirdNET Analyzer GUI.

9. The “Location filter threshold” parameter (Figure 8) specifies the minimum probability of occurrence at the specified location and week (based on eBird data) required for a species to be included in the target species list.
10. Choose the “Output type” (Figure 9) based on how you plan to work with the prediction files generated by BirdNET. By default, BirdNET creates one output file for each input sound file in whichever format you select (Raven, Audacity, R, CSV, or Kaleidoscope). If you choose Raven selection table and also check the box for “Combine selection tables”, BirdNET creates a selection table named “BirdNET_Results_Selection_Table.txt” (in addition to the table created for each sound file). This combined selection table is suitable for use if you open all of the input sound files together in Raven as a file sequence.

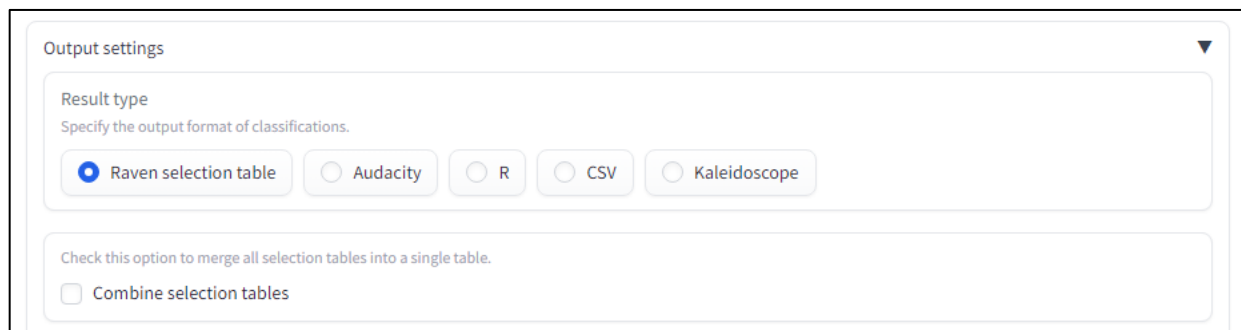


Figure 9. Output format selection.

11. The BirdNET GUI allows you to select the number of threads used in analysis (Figure 10). The maximum possible number of threads is dependent on the number of CPU cores (processors) that your computer has. To find the number of cores, press Ctrl + Shift + Esc to open Task Manager. Select the Performance tab to see how many cores and logical processors your PC has. For fastest performance, set “Threads” to the number of cores available on your machine. High thread values will reduce your computer’s performance on other tasks while BirdNET is processing the data. If you would still like to use this machine for other tasks, set the “Threads” to the number of cores -1 (e.g., if you have a 4 core-computer, choose 3 threads if you will work on other tasks while BirdNET runs).

The image shows a web-based configuration interface for BirdNET. It features three input fields: 'Batch size' (Number of samples to process at the same time) with a value of 1, 'Threads' (Number of CPU threads) with a value of 4, and 'Locale' (Locale for the translated species common names) with a dropdown menu set to 'EN'. Below these fields is a large grey button labeled 'Analyze'.

Figure 10. CPU thread specification.

- Click “Analyze” to begin analysis. As BirdNET processes files, the GUI will show the number of files completed, the elapsed processing time, and a progress bar. (There may be a long delay while BirdNET is preparing the analysis before you see the progress bar.) Analysis time will vary depending on the number, size, and format of files to be processed. Note that the percent value shown by the progress bar (Figure 11) is based on the number of files that BirdNET has *finished* processing. Thus it is normal for the progress bar to remain at 0% until BirdNET has finished processing the first file.

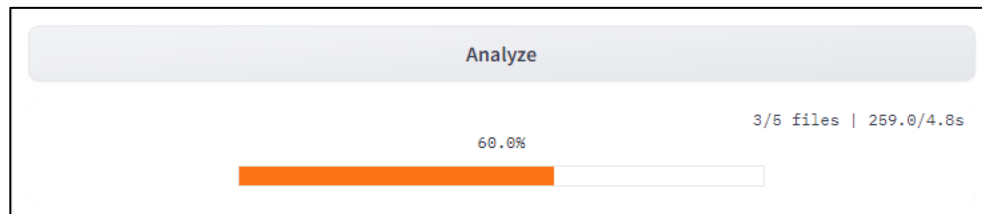


Figure 11. The analysis progress bar.

- When processing is complete, go to the folder that you identified to receive predictions. In this folder, you will find individual prediction files. The predictions can be visualized on the original sound files by opening the sound in Raven and dragging the corresponding prediction file onto the recording. The prediction files can also be processed with R, Python, or other programs.

4 Threshold setting in BirdNET

Background: For some species, even predictions with low BirdNET Confidence Scores are often true positives. For other species, even predictions with very high Confidence Scores are often not true positives. Thus, cross-species comparisons based on

Confidence Score thresholds are unlikely to be appropriate. A standardized quantitative means of comparison is needed. Moreover, for many applications, it is important to be able to make statements about the probability that a given prediction is a true positive. For further information about the relationship between BirdNET confidence scores and probabilities, see Wood & Kahl (2024).

Goal: The goal of this exercise is to convert BirdNET Confidence Scores to the probability that a prediction is a true positive for a given species. This allows us to set Confidence Score thresholds that are customized to a species and location and make statements like “In this dataset, a prediction of Species A has at least a 0.95 probability of being a true positive if the Confidence Score is above 0.897.”

Process overview: After running BirdNET on one or more sound files, we will use BirdNET’s Segments feature to extract and save sound clips (short sound files) corresponding to individual 3-second events for which BirdNET generated a species prediction, across a range of confidence scores. We will then use Raven Selection Review to evaluate whether each clip is a true or false positive. Finally, we will use the confidence score of the clip and the evaluation (true/false positive) to fit a logistic regression in R. The logistic regression will allow us to translate Confidence Scores into the probability that a prediction is a true positive.

Software Required: BirdNET GUI (Windows only), Raven 1.6.5 or later, R or comparable statistical software.

4.1 Exporting clips across a range of Confidence Scores

To identify and export audio clips of BirdNET predictions that represent a range of Confidence Scores, we will use the Segments feature of the BirdNET Analyzer GUI (this can also be achieved using the command line interface via the “segments.py” script). These instructions assume that you have already run BirdNET over your recordings and have generated Raven selection tables that contain predictions from the recordings. The process described below will generate clips for all classes represented in the Raven selection tables, with clips for each class saved in a separate directory named with the species label predicted by BirdNET.

1. Open the BirdNET GUI and click on the Segments tab (Figure 12, red box).

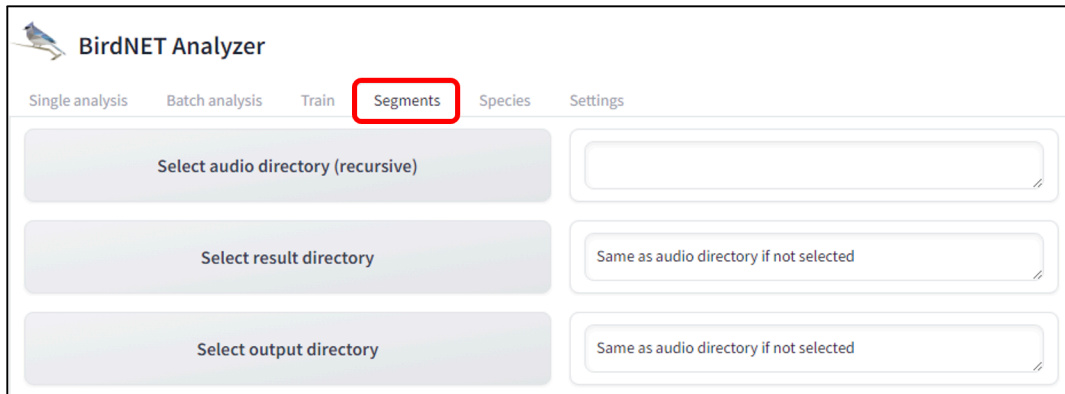


Figure 12. The Segments tab of the BirdNET Analyzer GUI, used for extracting and saving sound clips corresponding to events for which BirdNET has predicted species labels in one or more sound files that have already been analyzed.

2. Click on “Select audio directory (recursive)” and navigate to where the sound recordings are stored.
3. Click “Select result directory” and navigate to the location where you have stored the Raven selection tables that were generated by BirdNET.
4. Then click “Select output directory” and navigate to where you would like the segment clips to be saved.
5. For “Minimum confidence” select the lowest Confidence Score that you would like your evaluation clips to include (Figure 13). If you select a value that is lower than the lowest value that you included in your BirdNET run, you will not receive clips with these low values (the Segments function is only filtering the results that are in your Raven selection tables, not generating new predictions).



Figure 13. Minimum Confidence and maximum number of segments settings on the Segments tab.

6. For “Max number of segments”, indicate the number of randomly selected clips above the specified confidence threshold that you would like to export for each

predicted species label. If you select a value that is higher than the total quantity of BirdNET predictions above that score, you will simply receive all predictions.

Tip: Export all predictions above a threshold

*While there is currently no way to simply specify that you want all predictions above a given threshold (or all predictions in a specific range), you can obtain them by choosing an extremely high number here (i.e., $\gg (1/3) * (\text{total duration of your audio dataset in seconds})$).*

7. For “Sequence length” (a synonym for segment or clip length) indicate how long you would like the clips to be (Figure 14). BirdNET works on three-second clips. However, if your focal signal is more than 3 seconds, you might consider exporting longer clips to provide additional temporal context for human reviewers.

The image shows a screenshot of a software interface with two input fields. The first field is labeled 'Sequence length' with a subtitle 'Length of extracted segments in seconds.' and contains the number '3'. The second field is labeled 'Threads' with a subtitle 'Number of CPU threads.' and contains the number '4'. Both fields have a light gray border and rounded corners.

Figure 14. Specifying Sequence (segment) length, and CPU usage (Threads), on the Segments tab of the BirdNET GUI.

8. As with the initial BirdNET analysis, you can specify the number of threads to be used during the segment export process. Exporting audio clips based on previously generated predictions is much faster than the initial prediction process, so choice of thread number here typically has less impact on total run time than it does in the initial analysis. More information about specifying number of threads to be used is provided in Section 3, *Applying the BirdNET model using the BirdNET GUI* above.
9. Finally, click on “Extract Segments”. When the extraction process is complete, BirdNET will have created a folder inside of the output folder for each species for which predictions were found in the input (Figure 15).

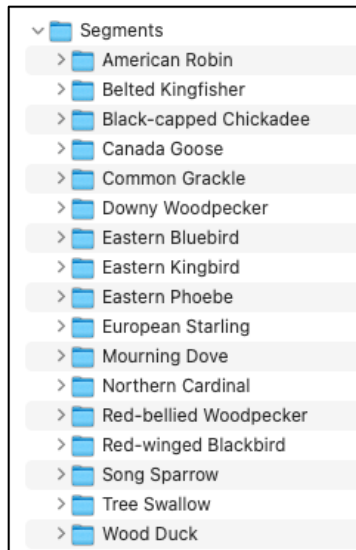


Figure 15. Contents of a folder named “Segments” that was specified as the Output directory for BirdNET’s Segments export process, given a set of sound files for which 17 species had previously been predicted. BirdNET creates a separate folder for each species prediction present in the input.

10. Within each species folder, there will be a series of short extracted audio files for the given species (Figure 16).

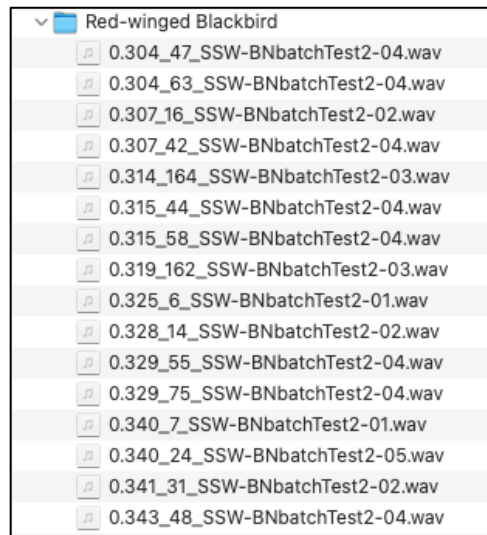


Figure 16. Content of a single species folder generated by BirdNET’s Segments export tool. The decimal number at the start of each file name is the prediction confidence score associated with that audio segment.

Clip file names begin with the confidence score that BirdNET reported for that segment, and include the name of the original audio file from which the clip was

extracted. Files name also include a unique segment ID number generated by BirdNET to disambiguate names in case two different clips from the same file were predicted with the same confidence score (Figure 17).

score audio source file
0.526_165_SSW-BNbatchTest2-03.wav
segment

Figure 17. Filename format for exported audio clip files. The first five characters represent BirdNET’s prediction confidence score for this clip.

4.2 Validating clips using Raven Selection Review

We will use Raven *selection review* to review all of the clips exported by BirdNET for a given species. Selection review is a tool within Raven that displays spectrograms of selections within an open sound as a matrix of small thumbnail images. Individual selections can optionally be played back for aural review, and annotated using a single keystroke. To validate BirdNET’s predictions, each clip will be annotated to indicate whether or not it includes a sound of the target species as predicted by BirdNET (i.e., whether the prediction is true or false).

Open the exported clips as a sound file sequence

1. Open all of the clips exported for a single target species by selecting File > Open. Navigate to the folder containing the exported clips for your selected target species, select all clips, then click OK. You can also open the full set of clips by highlighting all of them in your file manager and dragging them into Raven.
2. In Raven’s Configure New Sound Window dialog (Figure 18), choose “Open entire sound” and “Open as file sequence in one window” (these are the default selections), then click OK.

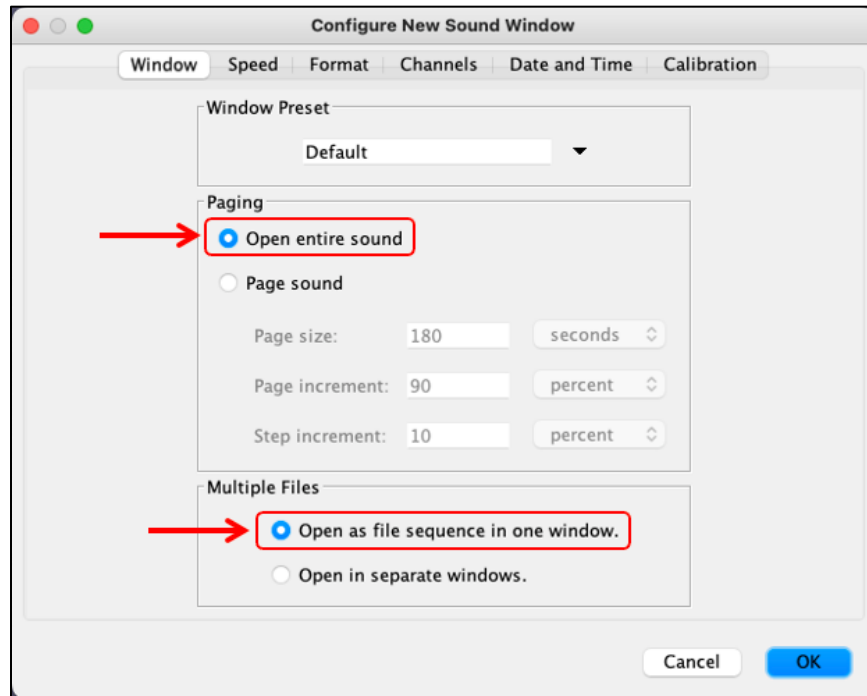


Figure 18. Raven's Configure New Sound Window dialog.

When the clips are opened, they open as a single *sound file sequence*. A sound file sequence is a series of individual sound files that are treated by Raven as one continuous sound. By default, Raven displays the boundaries between the individual clip files with dashed vertical green lines (Figure 19).

When Raven opens a sound file sequence, the individual files are ordered based on an alphanumeric sorting of their filenames. Since the clip files extracted by BirdNET are named beginning with their corresponding confidence scores, the resulting sound file sequence in Raven is sorted from the lowest-confidence clips at the beginning of the sequence to highest-confidence clips at the end.

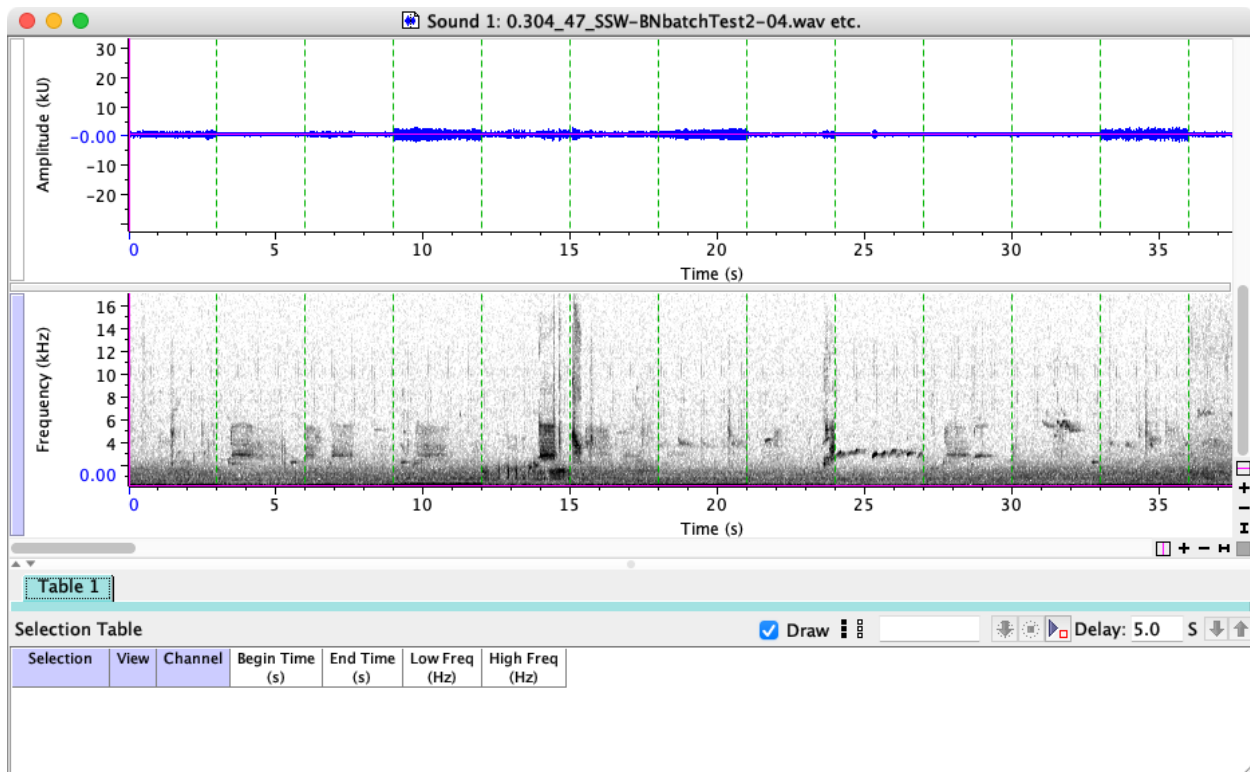


Figure 19. A series of prediction clips exported by BirdNET, opened as a sound file sequence in Raven. The dashed vertical green lines indicate boundaries between clip files. (The green file boundary markers may be difficult to see with some spectrogram colormaps.)

Create selections corresponding to the boundaries of individual clips

1. In order to use Raven’s selection review feature, you will need to create selections corresponding to the individual sound clips in the sequence. To do this, click on Table > Generate selections.
2. In the Generate Selections dialog (Figure 20), under Method, choose “One Selection Per File”. For “Create selections in:”, choose Table 1, then click OK. The selection boundaries and ID numbers will be displayed in the waveform and spectrogram views, and entries for each selection will be shown in the selection table (Figure 21).

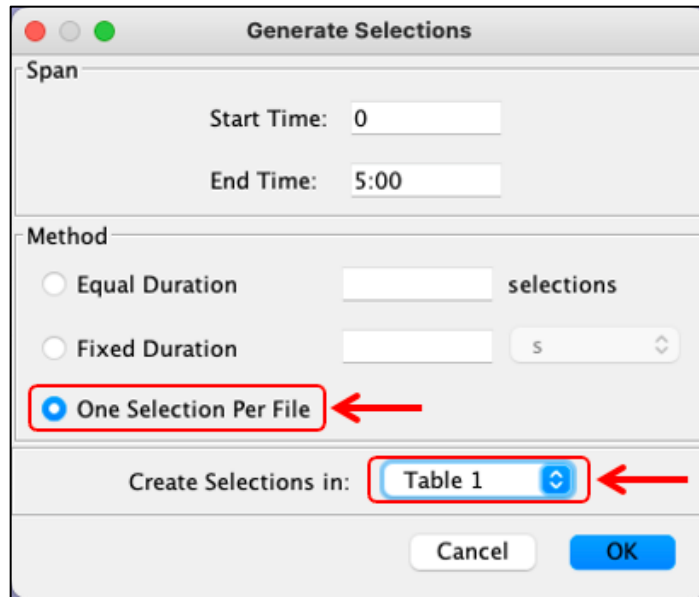


Figure 20. The Generate Selections dialog.

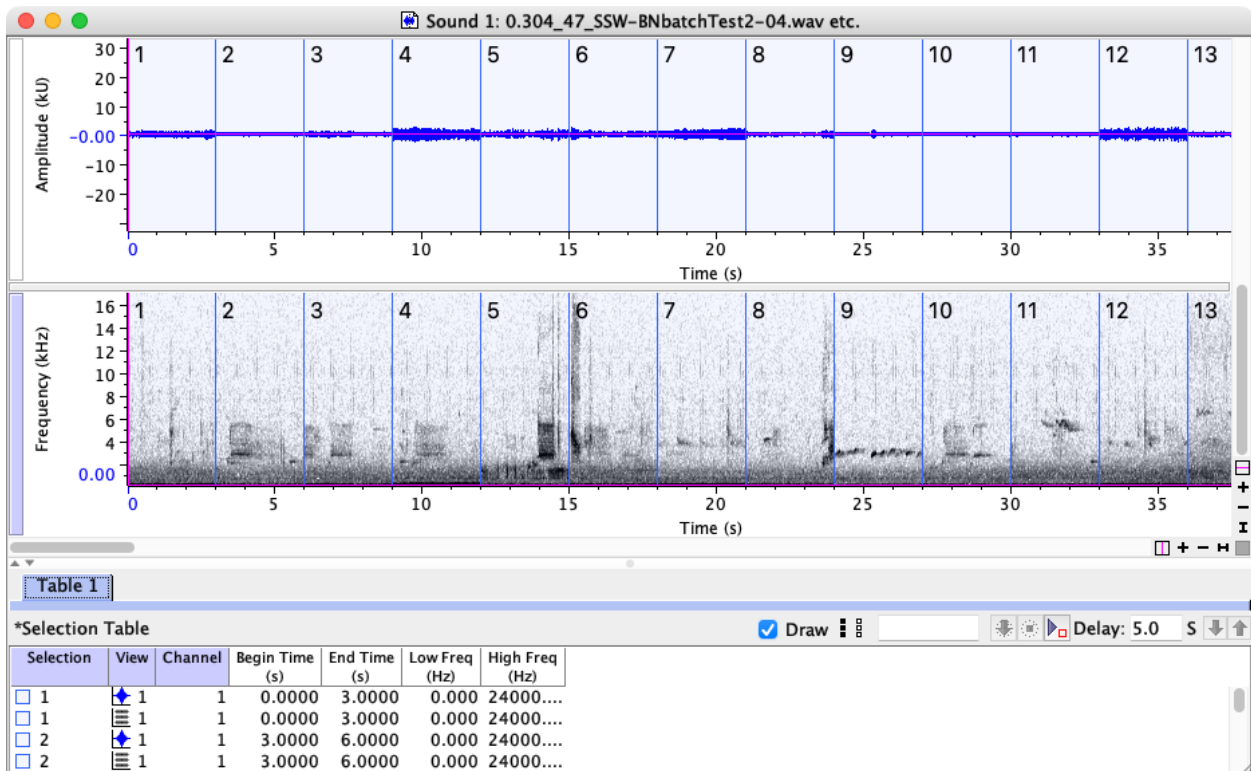


Figure 21. The same clip file sequence shown in Figure 19 after generating selections aligned with the file boundaries.

Add the names of the clip files to the selection table

When the selection table is later exported to R for calculating the logistic regression (after the selections have been annotated as true or false predictions), each selection will need to be associated with its respective BirdNET confidence score. Recall that the confidence score for each clip file is included at the start of the file name generated by BirdNET during the extraction process (Figure 16). Including the names of the clip files in the selection table will enable us to extract the confidence score for each prediction later in R.

1. To add the names of the clip files to the selection table so that the confidence scores will be available later in R, right-click (Windows) or ctrl-click (macOS) on any of the column names of the selection table. Then select “Choose Measurements” (Figure 22).

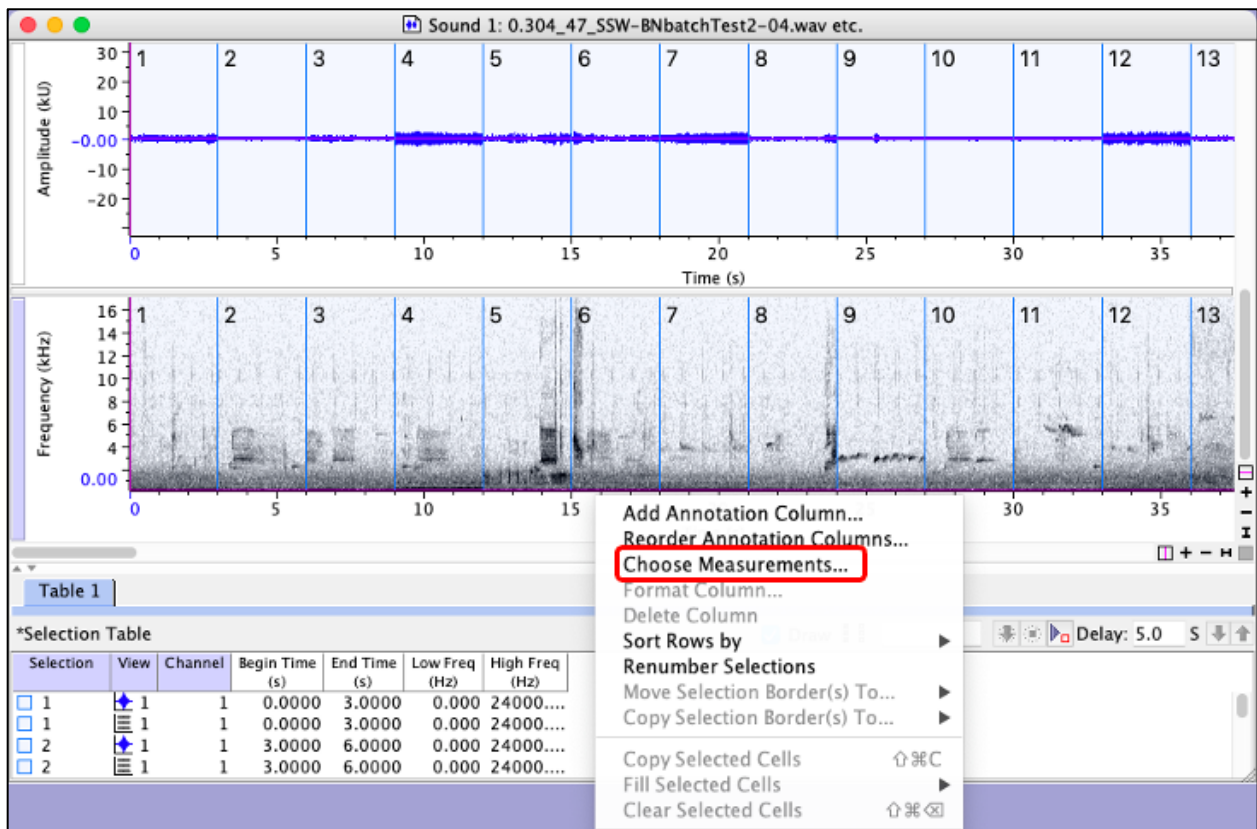


Figure 22. Accessing the Choose Measurements dialog for the selection table.

2. From the list of available measurements in the right column of the Choose Measurements dialog, select “Begin File” (Figure 23). Next, click the arrow in the

middle of the dialogue box to move the measurement to the list of Displayed Measurements in the left column, then click OK.

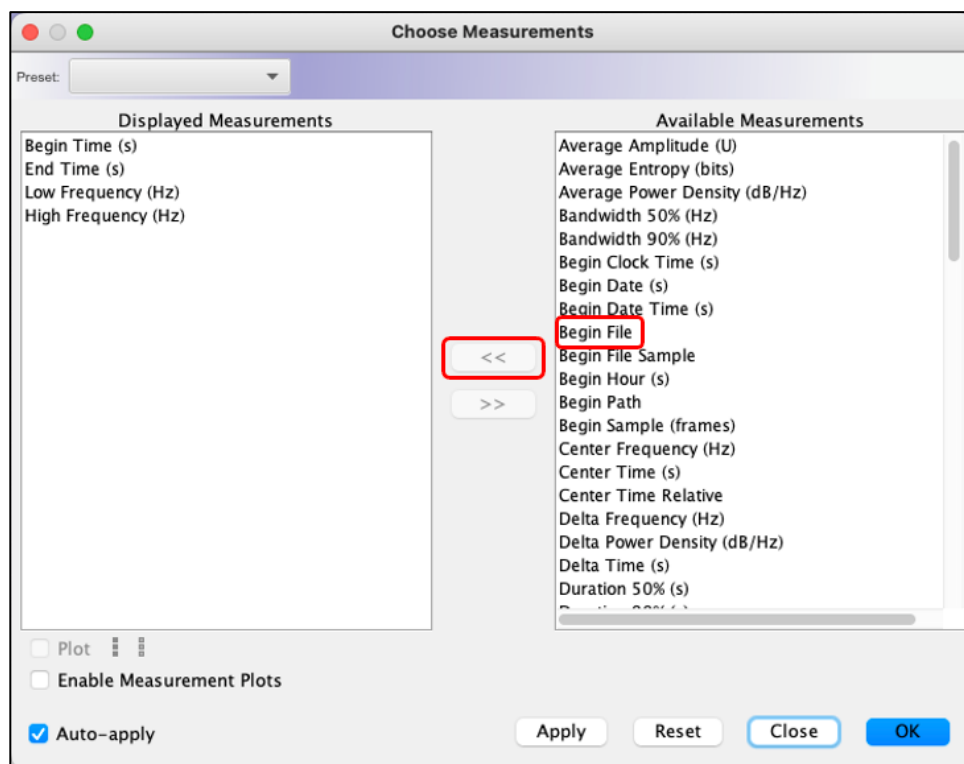


Figure 23. The Choose Measurements dialog.

Add a column for the true/false annotations to be applied in selection review

1. Next, we will create an annotation column, which will be used to add true/false annotations to each selection during the selection review process. Right-click (Windows) or ctrl-click (macOS) on the column names of the selection table.
2. In the menu that appears, click “Add Annotation Column” (the top item in the menu), and name your annotation column (Figure 24). For this example, we will use “Valid” as the column name to indicate that it is the validation of the clip (true prediction or false prediction). Then click OK.

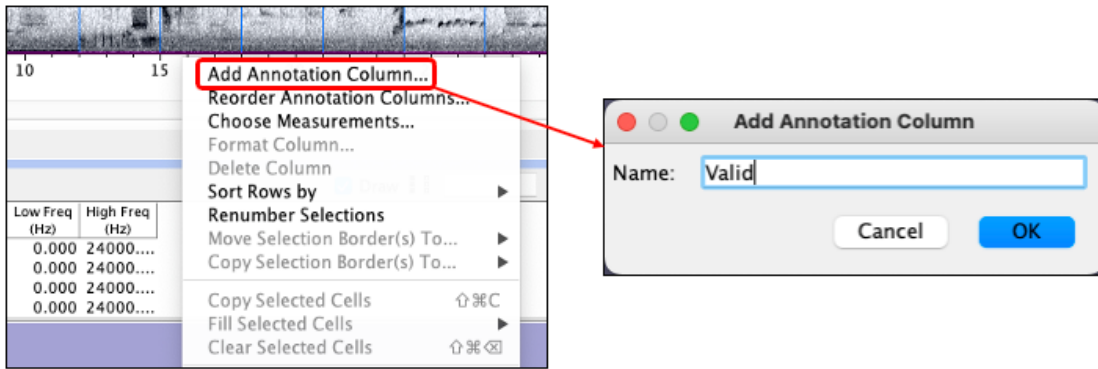


Figure 24. Adding an annotation column.

Configure selection review preferences

Before launching selection review, open the Raven Preferences dialog by choosing Edit > Preferences. Check to ensure that the three highlighted preferences are set as shown in Figure 25.

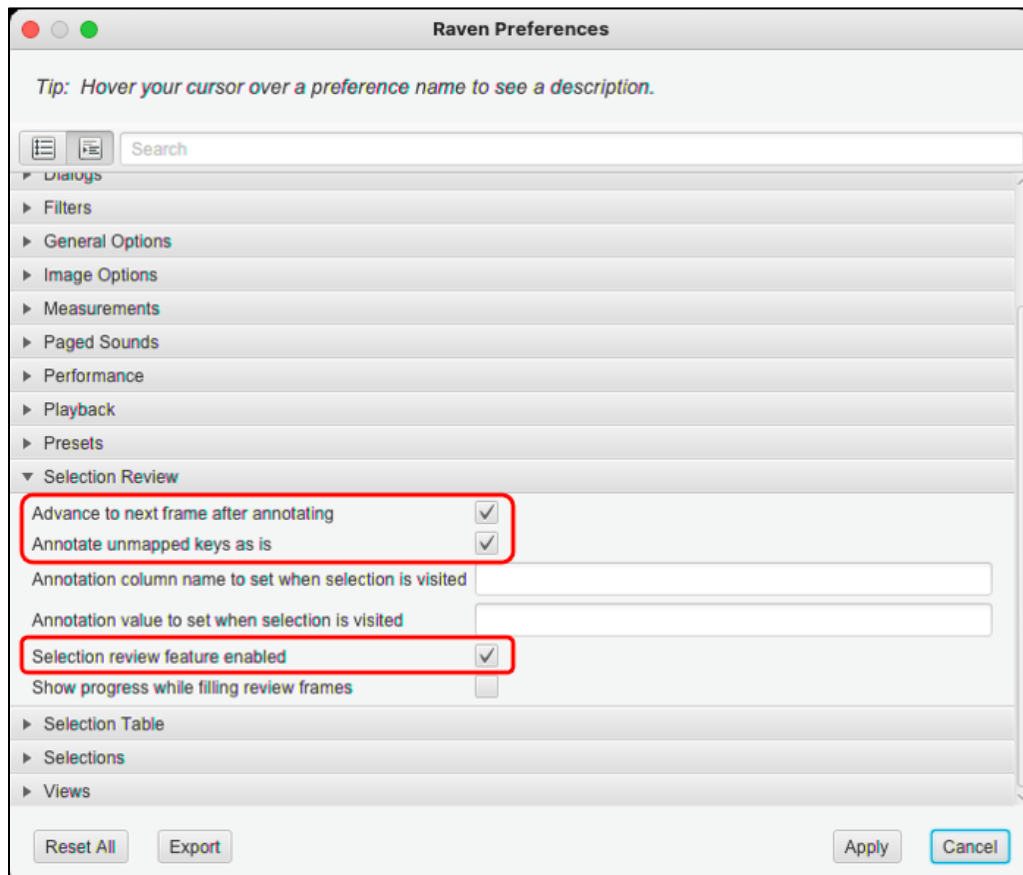


Figure 25. The Raven Preferences dialog.

Activate and configure selection review

1. Next, we will use the Raven Selection Review feature to review and validate the clips. Access this feature by clicking on the “Review” tab on the left side of the Raven window (Figure 26). This will open the Selection Review configuration panel in the side panel to the left of the signal window (Figure 27).

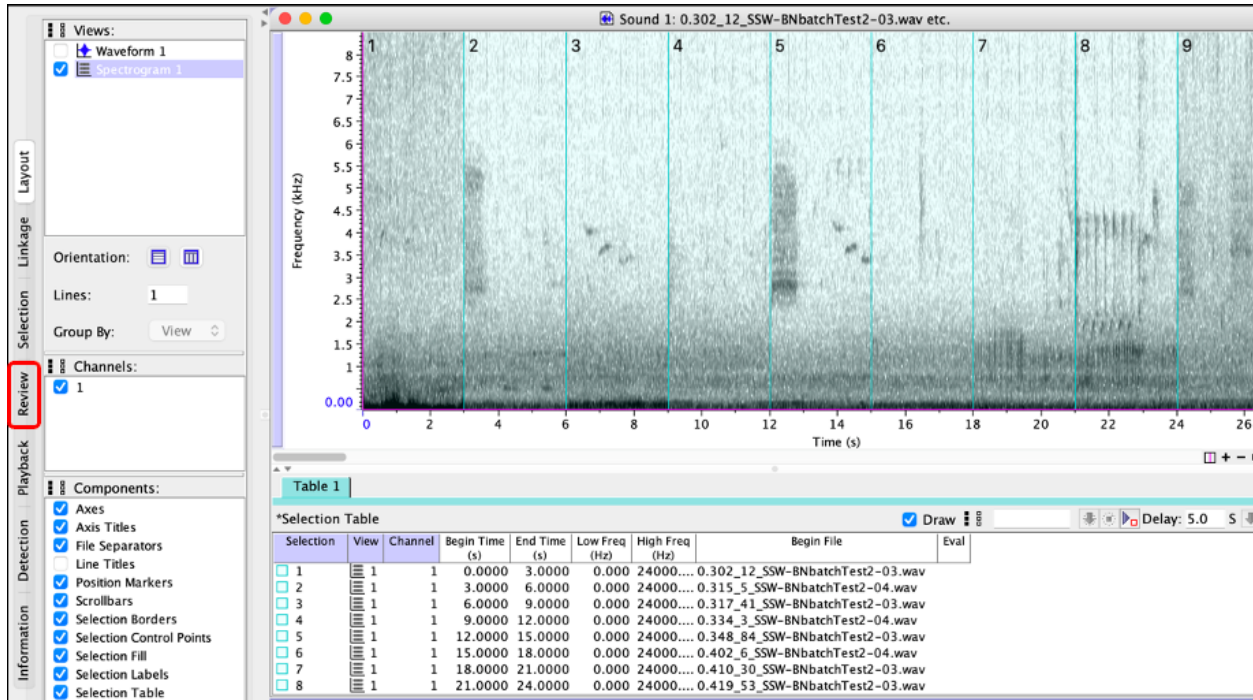


Figure 26. To activate selection review, click the tab labeled Review (red box) in the left side panel. This screenshot is from the macOS version of Raven; on Windows, the tab labels (Review, Layout, etc.) are displayed at the bottom of the side panel.

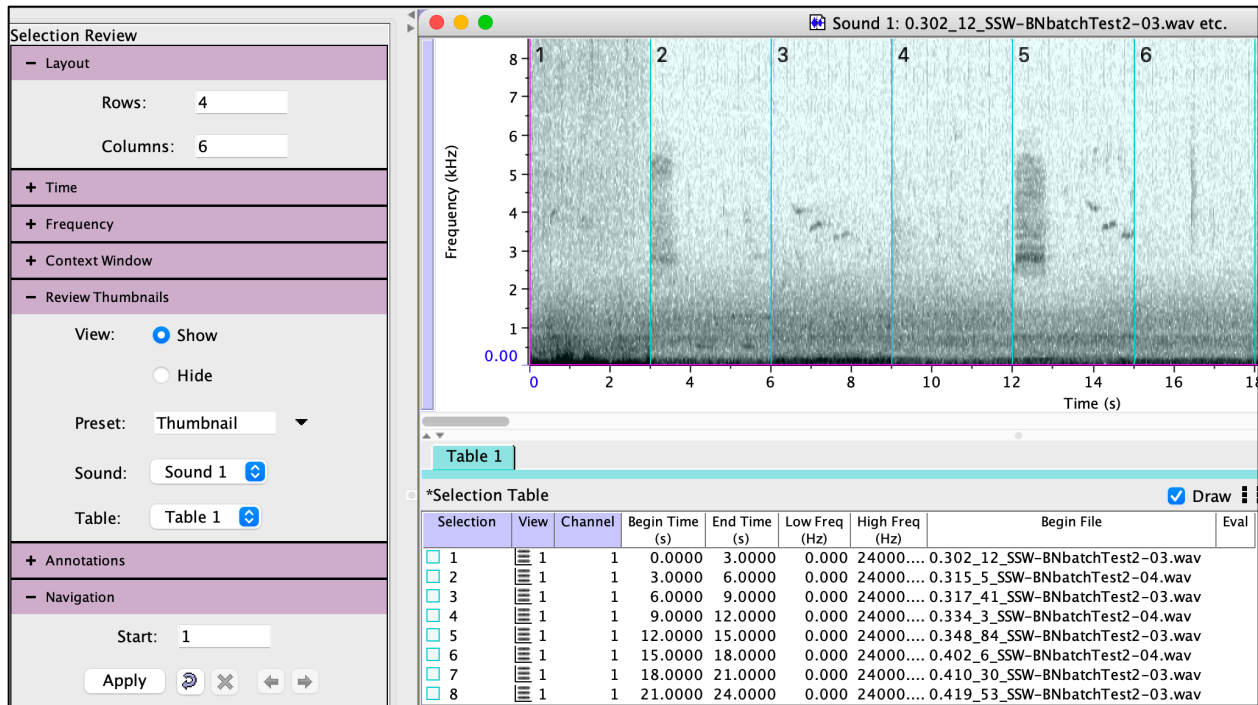



Figure 27. The Selection Review configuration panel (left). Each tab (Layout, Time, Frequency, etc.) can be expanded to review settings by clicking on + in its title or collapsed by clicking -.

2. The Selection Review configuration panel contains seven sections or tabs (Figure 27). Clicking on the + next to the tab's title reveals various settings to configure. For purposes of validating BirdNET's exported clips, the following settings are recommended:
 - **Layout:** The number of thumbnail spectrograms displayed at once is the product of the number of rows and columns entered here. Larger values will enable you to process many clips more quickly, but the individual spectrograms should be large enough to enable you to see signals clearly. If you're working on a laptop, 4 rows and 6 columns should work well; if you're working on a large monitor, at least 6 rows and 8 columns might be preferable.
 - **Time:** Choose Time Scale: Selection and Play: Visible.
 - **Frequency:** Set Freq Pad: 0 Hz, Freq Scale: $[max]/\text{Hz}$ where max is slightly above the maximum frequency of the sounds of your target species.
 - **Context Window:** set to OFF (unchecked).
 - **Review Thumbnails:** Set View: Show, Preset: Thumbnail, Sound: Sound 1, Table: Table 1.

- **Annotations:** Set Filter: Valid, Show: OFF (unchecked), Annotate: Valid
 - **Navigation:** Start: 1
3. Click “Apply” (at the bottom left of the configuration panel).

 If Raven displays an error message that says that there are no selections to display, check whether the selections that you created are in Table 1. If they are in Table 2, then change the Table setting to Table 2 under the Review Thumbnails tab.

4. Raven will then create and display a matrix of small spectrograms, one for each clip (Figure 28). Seeing the spectrograms side by side can be helpful for visually separating true and false positives.

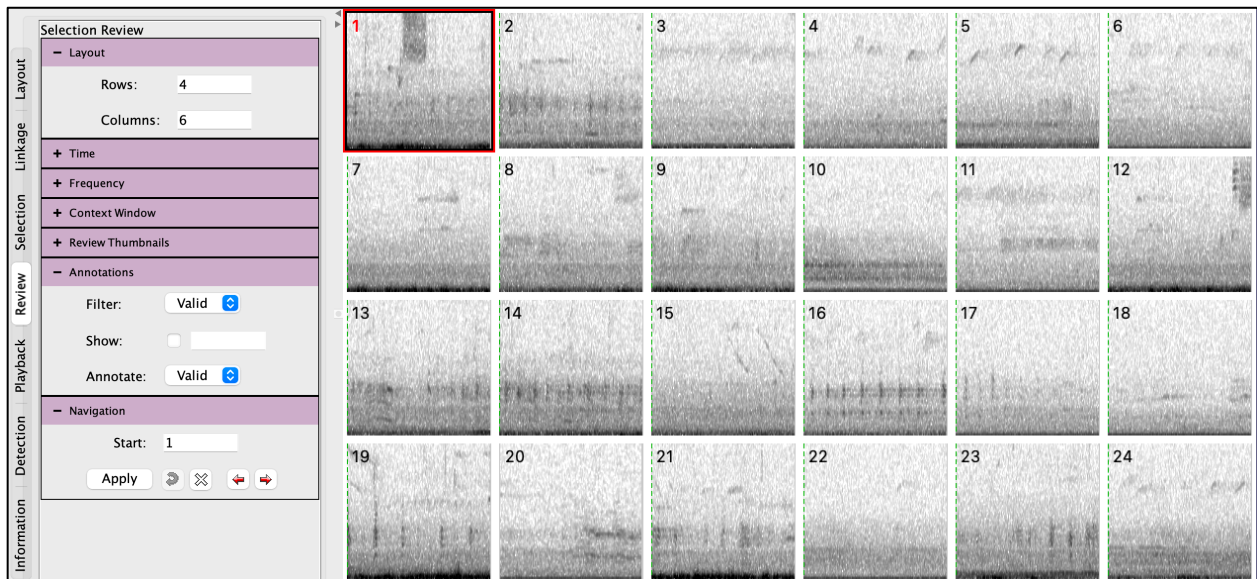


Figure 28. The Selection Review thumbnail matrix.

Configure selection labels

Selection labels in Raven are text strings that are displayed as overlays on selections in spectrogram and waveform views. By default, a selection’s label displays its selection ID (as shown in Figure 28), but other information can optionally be displayed as well. In this section we will configure the selection labels of the thumbnail spectrograms to display the Valid code (1 or 0 to represent true or false predictions) that you assign, in addition to the selection ID. The steps in this section are not essential, but will make the selection review process easier.

1. To access the Configure Selection Labels dialog, right-click (Windows) or ctrl-click (macOS) on the first thumbnail, and choose “Configure Selection Labels...” from the menu that appears (Figure 29).

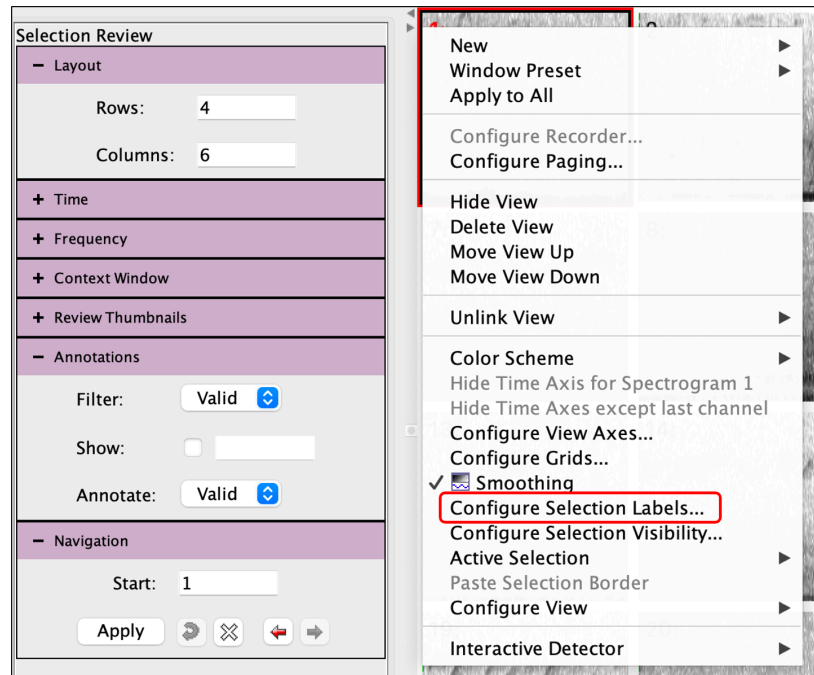


Figure 29. The Configure Selection Labels... menu item is found in the context menu displayed by right-clicking (Windows) or ctrl-clicking (macOS) on any one of the thumbnail spectrograms.

2. In the Configure Selection Labels dialog (Figure 30), (1) select the Valid annotation label in the list of Available Items, then (2) click the left arrow button to move it to the list of Displayed Items on the left. When multiple items from a selection table (in this case the selection ID and the Valid annotation) are displayed in a selection label, you can choose to have them separated by a string of characters, such as one or more spaces, or a colon, slash, or other characters that you choose. (3) Type your preferred separator string into the Item separator field, then click OK.

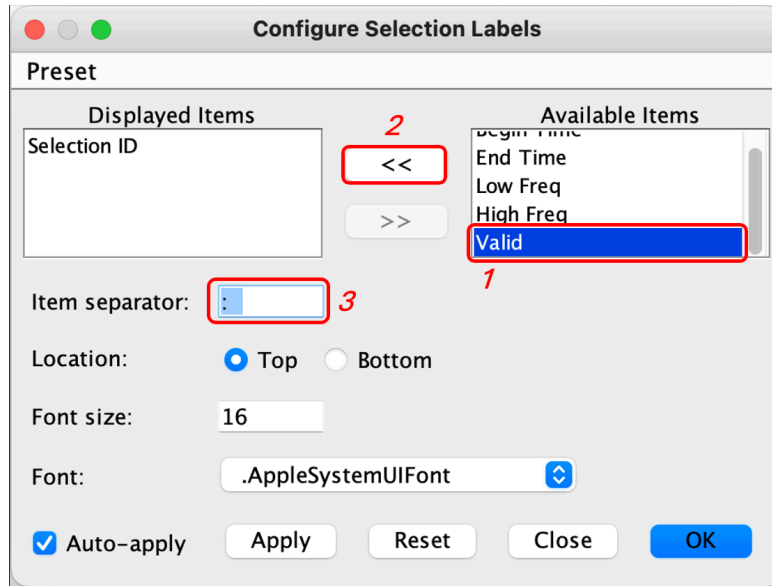


Figure 30. The Configure Selection Labels dialog. In this case, the item separator is a colon followed by two spaces.

3. To make the new label configuration apply to all thumbnails (not just the one you had highlighted) you need to update the Thumbnail window preset³ and re-draw the thumbnails. To do this, (1) right-click (Windows) or ctrl-click (macOS) on the same thumbnail, and choose Window Preset > Save “Thumbnail” (Figure 31). Then (2) click the Apply button at the bottom of the selection review side panel (Figure 31).

³ Window presets in Raven provide a convenient way to save many aspects of how sound windows are configured, including selection labels. Because all of the thumbnail spectrograms shown in selection review use a preset named Thumbnail, updating that preset after changing the fields displayed in selection labels for one thumbnail automatically updates the selection configuration of selection labels for all thumbnails when they are re-drawn.

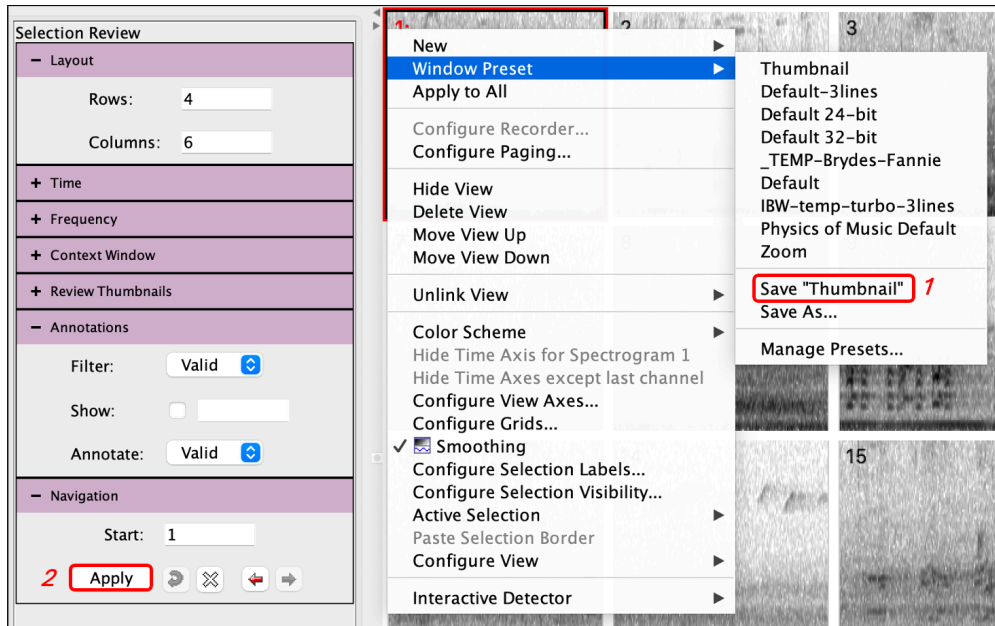


Figure 31. Updating the Thumbnail window preset.

Review and annotate all selections

You're now ready to review and annotate all of the clips exported by BirdNET following the steps in Section 0. The goal of this process is to assign an annotation (1 or 0) to each selection (corresponding to the exported clip files) to indicate whether BirdNET's species prediction for that clip is true or false. Later, these true/false annotations will be used in conjunction with BirdNET's confidence score for each prediction, to build a logistic regression model relating a prediction's confidence score to the probability that the prediction is correct.

Tip: Know all your target species sounds

When validating BirdNET predictions, it's important that you be able to reliably recognize all sounds of the target species, not just the most common or conspicuous sounds. This is because, for many species, BirdNET has been trained on multiple sound types. If you only recognize the most distinctive or well-known sounds (for example territorial advertisement songs), you may mistakenly consider a prediction false if the clip contains a less familiar sound (for example, alarm or contact calls).

1. Activate one of the thumbnails by clicking on it. Raven displays a red border around that thumbnail to indicate that it is now the *active selection*. To play the active selection, double-click its spectrogram, or press the spacebar on the keyboard. To

annotate the active selection as either a true or false prediction of the target species, simply press the 1 (for true) or 0 key. If you configured selection labels to show the value of the Valid annotation field as described above, you will immediately see the 0 or 1 you typed displayed on the thumbnail (Figure 32).

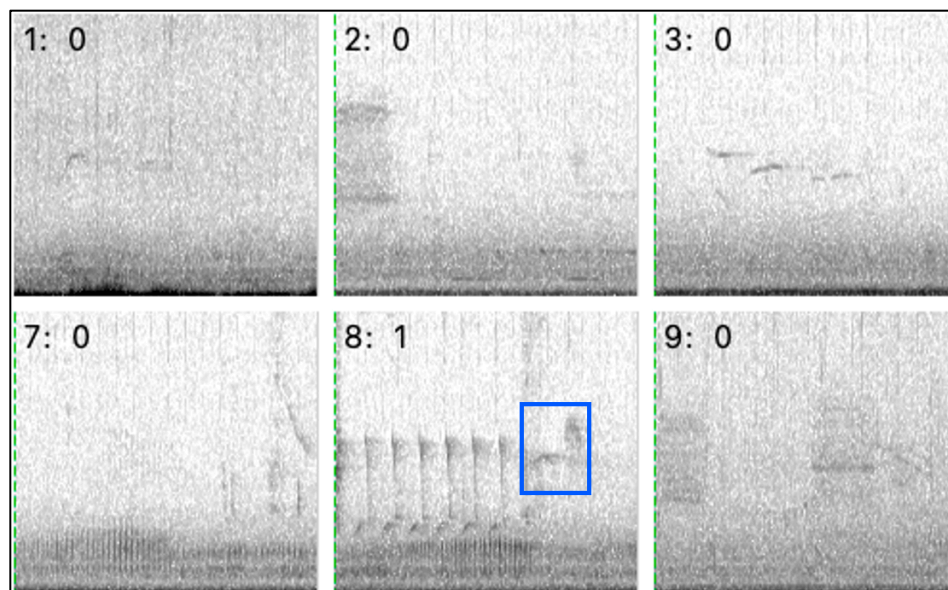


Figure 32. Six clips that have been annotated for the presence (1) or absence (0) of sounds of the target species, Eastern Phoebe. The one Eastern Phoebe song actually present in any of these clips is highlighted in blue, in selection 8.

Tip: Avoid “unsure” annotations

It’s not uncommon to encounter clips where you’re uncertain of whether or not a sound of the target species is truly present (due to distant sounds, overlapping noise, etc.). It’s tempting to want to add a third “unsure” annotation label for such cases. However, a three-category classification scheme is not compatible with logistic regression, which requires a binary response variable. A simple (and hopefully helpful) way to think about ambiguous BirdNET predictions is whether you would want them in your final dataset being treated as an animal observation. The answer is often “no”, meaning that a good rule is “if you aren’t certain it’s right, label it wrong.”

2. As soon as you annotate a selection, Raven activates the next selection for you to play or annotate.

3. When you enter an annotation for the last thumbnail on a page, Raven automatically advances to the next page of thumbnails, and activates the first clip on the page.
4. You can use the page navigation controls in the selection review sidebar (Figure 33) to jump ahead or back in the selection review process.

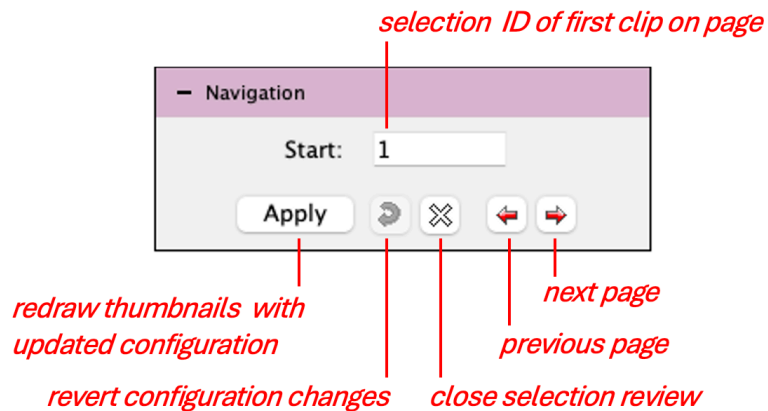


Figure 33. Page navigation controls for Selection Review.

5. (Optional) If many of the thumbnails on a page are easily recognized as true based on their spectrograms (without needing to listen to them), you can speed up the annotation process by highlighting them all, then annotating them in bulk with a single keystroke. To highlight multiple thumbnails, hold down the Ctrl (Windows, Linux) or command (⌘, macOS) key while clicking on each of the thumbnails of interest. Each highlighted thumbnail is then indicated by a black border (Figure 34). Once you have highlighted multiple thumbnails, pressing the desired annotation key (1 or 0) will apply that annotation to all the highlighted selections.

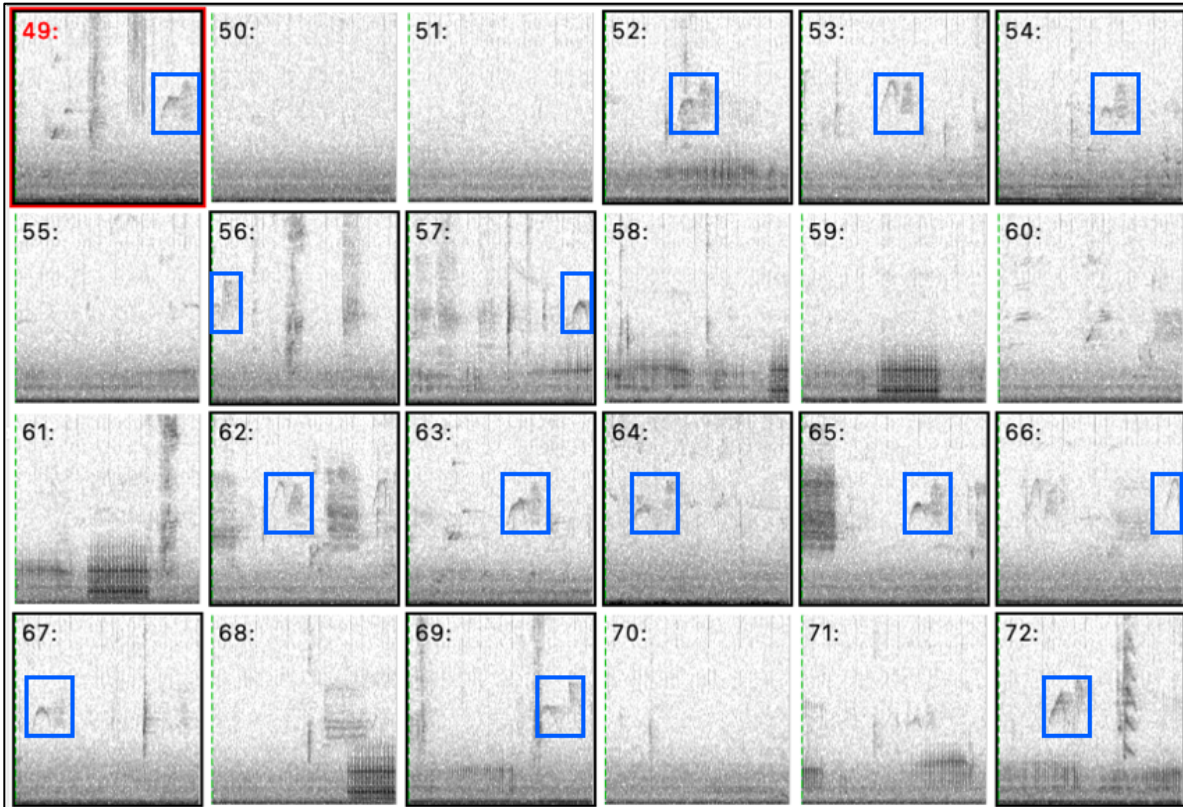


Figure 34. Highlighting multiple thumbnails for annotation with a single keystroke. In this example, all clips shown here were predicted by BirdNET to contain sounds of eastern phoebe. For this illustration, all of the true phoebe songs are outlined in blue. The thumbnails outlined in black were highlighted by *ctrl*-clicking (Windows, Linux) or *command*-clicking (macOS). A single annotation keystroke (1 in this case, since these are all true predictions) annotates all 14 clips simultaneously.

Close selection review and save the selection table

1. Once you have annotated all of the clips, you can exit selection review mode by clicking the X button in the Navigation tab of the side panel (Figure 33). The thumbnails will disappear and be replaced by a normal sound window, showing the selection table below the spectrogram (Figure 35). Note that the Valid annotation column now displays the 1/0 annotations that you applied in selection review mode.

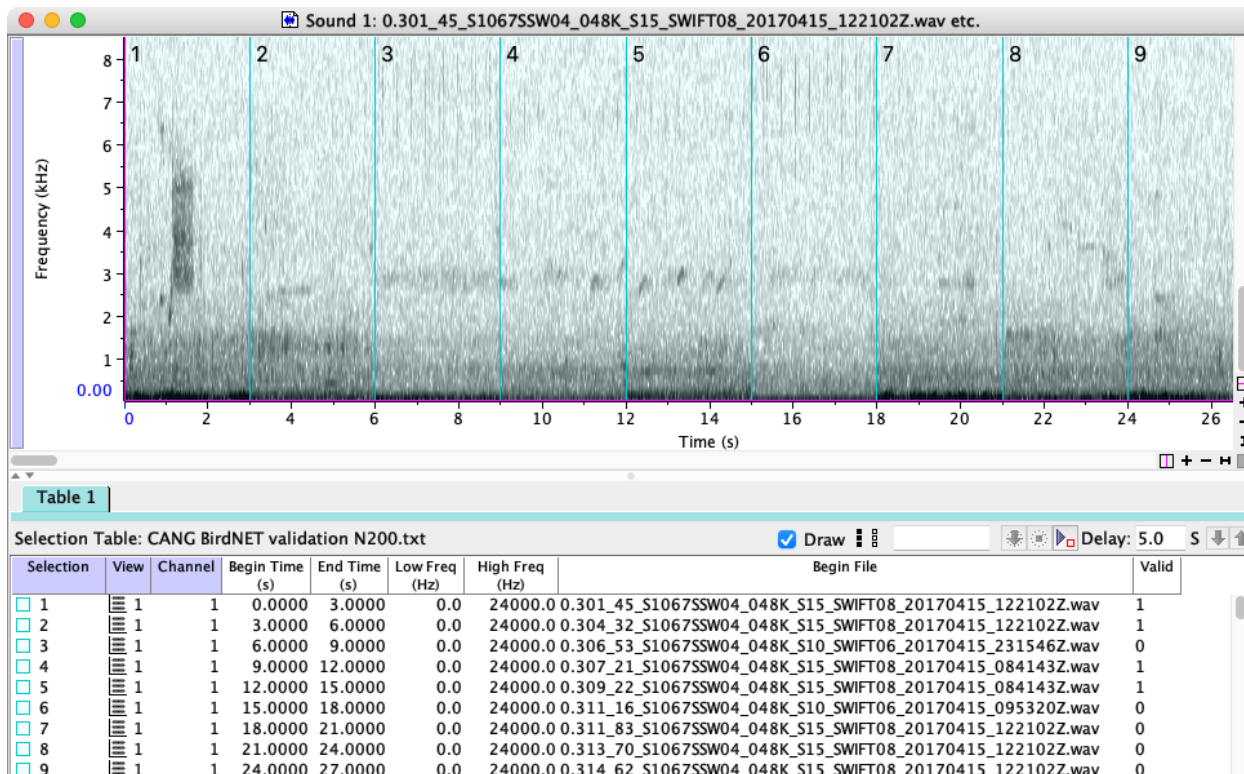


Figure 35. The sound window after exiting from Selection Review mode. Note the 0/1 annotations in the Valid column.

2. Save the selection table by choosing File > Save Selection Table. Raven saves selection tables in tab-delimited text format, which is easily imported into R or other statistics or spreadsheet programs.
3. Before moving on to the next section, it's a good idea to make sure you didn't miss annotating any selections. An easy way to do that is to scroll through the selection table and look for any row in which the Valid column is blank (or contains a character other than 0 or 1). If you find a selection that's missing an annotation, click on that row of the table to highlight the selection in the spectrogram view. Once the selection is highlighted, you can play it by clicking the Play (▶) button in the toolbar, or choosing Play > Play. To add or change the annotation, just double-click in the Valid cell for that row and type the new annotation. If you add or change any annotations, be sure to save the selection table again.

4.3 Logistic Regression

We will now use logistic regression to determine how BirdNET confidence scores correspond to the probability that a prediction is a true or false positive. To fit this

regression, we will need to extract two pieces of information for each row in the selection table saved in the validation procedure using Raven as described in the previous section. The first is the Confidence Score of the clip (the x-axis of the logistic regression); and the second is whether the prediction was a true positive or false positive (the y-axis of the logistic regression). Recall that the confidence score is contained in the file name of each clip that was exported by BirdNET (Figure 17). The indication of whether each clip was rated as true (1) or false (0) is in the annotation column called Valid.

To extract this information and plot the logistic regression, we will use the R software package. Instructions for downloading and installing R and RStudio (which provides a user-friendly GUI for R) are available [here](#). (You can also perform logistic regression in some other statistics package if you prefer, but the detailed instruction below are for performing the regression in R.)

1. Import your annotated selection table, either by providing its file path:

```
> # import a specified selection table:  
> table <- read.table("you/filePath/selection_table.txt",  
  sep = "\t", header = TRUE)
```

or by manually selecting it:

```
> # import a selection table chosen via a file selection dialog:  
> table <- read.table(file.choose(), sep = "\t", header = TRUE)
```

Example:

```
> table <- read.table("/Users/rac8/Raven Pro 1.6/Selections/cangoo  
  validation N200 clips.txt", header = TRUE, sep = "\t")
```

2. Recall that the first 5 characters of the filename of each clip represent the confidence score of BirdNET's species prediction for that clip (Figure 17). We will now extract the confidence score from the 'Begin.File' column and add it as a new column, converting it from a string to a numeric value:

```
> # extract the numeric score from the file name:  
> table$Score <- as.numeric(substr(table$Begin.File, 1, 5))
```

3. Now we fit a logistic regression model in which the binary outcome (true or false, represented by 1 or 0 values in the 'Valid' column) is a function of BirdNET confidence score:

```
> # fit the logistic regression model:  
> model <- glm(Valid ~ Score, family = "binomial", data = table)
```



Note: This code will not work if you have coded 'Valid' as anything other than 1 for true and 0 for false. If you have used other labels, you will need to convert your labels to this format.

Optional but recommended: plot the data points, and the modeled relationship between score and probability of a correct prediction:

```
# plot the individual points (x = Score, y = Valid):  
> plot(Valid~Score, data=table, xlim=c(0,1), pch=16,  
      col=rgb(0,0,0,.3))  
  
> # add the modeled regression relationship (a curve):  
> curve(predict(model, data.frame(Score = x), type = "response"),  
      add = TRUE, col = "red", lwd = 2)
```

Figure 36 shows an example of a logistic regression plot with real data from validating a set of 200 BirdNET predictions.

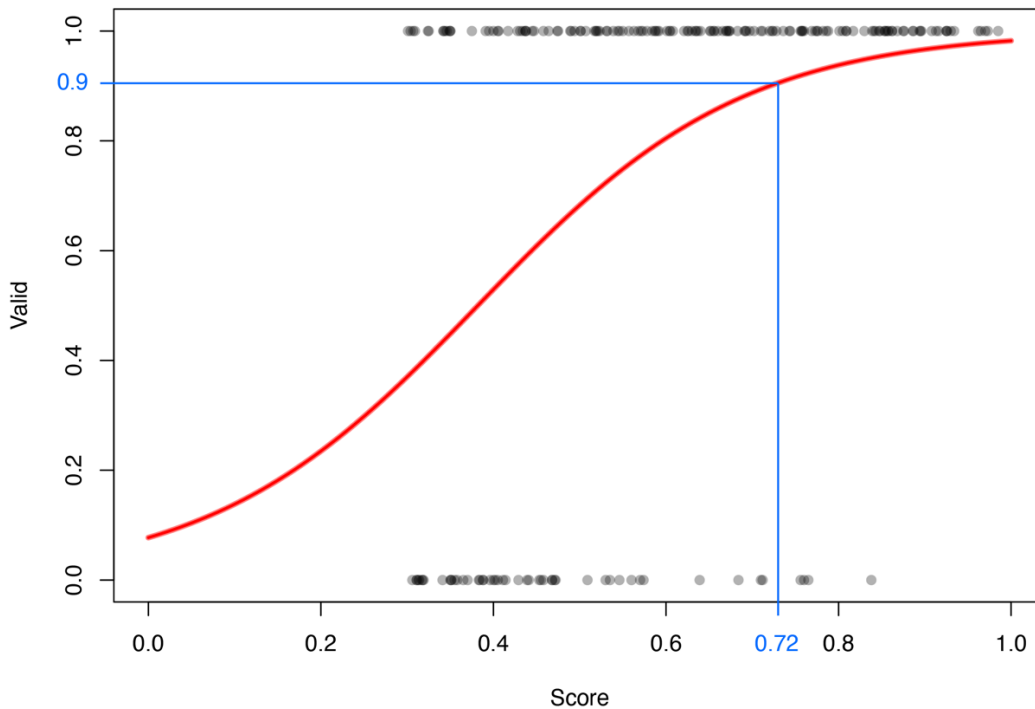


Figure 36. Sample logistic regression relating the probability of a prediction being correct (Valid) to BirdNET’s confidence score. The data are from manual validation of 200 clips predicted to contain vocalizations of Canada goose (*Branta canadensis*), extracted from 19 1-hour audio files recorded on one day at eight different recording sites within approximately 700 m of each other near the Cornell Lab of Ornithology. The blue overlay indicates that to achieve a probability ≥ 0.9 that a prediction is correct for this dataset, a score threshold of 0.72 is required (see below for code to calculate this value).

4. You can now use the logistic regression model to compute the BirdNET confidence score required to achieve a specified probability that the predicted species label is correct. For example, the following code computes the score (`score_threshold`) required to achieve a probability of ≥ 0.9 that the prediction is correct:

```
> # compute score threshold for probability p of correct
  prediction:
> p <- 0.9 # your desired p (probability of true positive)
> score_threshold <- (log(p/(1-p))-model$coefficients[1]) /
  model$coefficients[2]
```

For the data and model shown in Figure 36, this code computes a `score_threshold` of 0.72, as indicated by the blue lines shown in Figure 36.



The logistic regression equation will always compute a score threshold value to achieve the specified probability. However, in some cases, that threshold may exceed the maximum possible confidence score of 1. In such cases, the result indicates that the specified probability of a correct prediction is not attainable.

5 Building a custom BirdNET model

Background: The BirdNET Analyzer enables you to customize the BirdNET model to recognize additional classes of sounds that are not included by default (for example to add sounds of additional bird and non-bird animal species, gunshots, chainsaws, or other signals). Ghani et al. (2023) provide conceptual background and illustrations of how such *transfer learning* can be effective with modest amounts of training data. Further background and a demo of much of what's covered here is also presented in this [video](#).

Developing a custom BirdNET model that performs acceptably well is typically an iterative process. It's not unusual to find that the first version of a custom model misses many of the target sounds present in a recording, labels many non-target sounds as targets, or repeatedly confuses two or more target classes. Section 5.4 discusses strategies for improving model performance.

Software required: Raven Pro to create training data and review detections, BirdNET Analyzer GUI to select and build model. (As of March 2024, the BirdNET Analyzer GUI only works on Windows computers.⁴)

Training data: To create a custom BirdNET model, you will require a training dataset. A training dataset consists of a set of short sound files (clips), each containing a sound from a *target* or *non-target* class. Target sounds are ones that you want BirdNET to recognize, such as those of a given species or a type of non-biological sound (for example, gunshots or chainsaws). Non-target sounds are ones that you want BirdNET to ignore, such as non-target animal species or call types, or background noise. The sound clips for each target class should be grouped together in a single folder with the name of the class (typically a species name, call type, or anthropogenic sound). For example, a training dataset might contain a folder named 'species 1', another named 'species 2',

⁴ The BirdNET Analyzer GUI can be run on a virtual machine running Windows on a Mac using [Parallels](#).

another named ‘chainsaw’, and so on. Non-target sounds should be grouped together in a folder named ‘other’, ‘background’, or ‘noise’. BirdNET treats those three folder names (and only those) as equivalent, so you can use whichever one seems most appropriate for your application.

Tip: How many training examples do I need?

The number of training examples needed for each sound class to achieve an acceptable level of performance can vary widely, and depends on many factors including variability of the sound classes, what other sounds are present in the recordings to be analyzed, and the goals of the specific project. Although quite small training datasets (like the demo dataset used here) can provide some ability to correctly recognize sounds, larger training sets are always desirable. If possible, start with around 100 examples per class. If the resulting model performs poorly, increase the number of training examples if possible. Also see Section 5.4 below for further suggestions for improving model performance.

5.1 Example: training a custom model in the BirdNET GUI

In this section we illustrate how to train and apply a custom BirdNET model using a small data set of bird and primate vocalization extracted from PAM recordings made in the Danum Valley Conservation Area in Sabah, Malaysia. Many of the species in Danum Valley are not represented in BirdNET.

Sample dataset: You can download a small sample training dataset from the Danum Valley Conservation Area [here](#).⁵ This dataset, collected by Dena Clink and Abdul Hamid Ahmad, contains two folders, ‘Trainingdata’ and ‘Testdata’.

The Trainingdata folder contains folders with sound clips from three target classes (one primate and two bird species) and one non-target class: the northern grey gibbon (*Hylobates funereus*, folder ‘gibbon.female’), the helmeted hornbill (*Rhinoplax vigil*, ‘hornbill.helmeted’), and the rhinoceros hornbill (*Buceros rhinoceros*, ‘hornbill.rhino’),

⁵ If you want to use or distribute this sample dataset, please use the following citation:

Clink, Dena Jane, & Hamid Ahmad, Abdul. (2024). Dataset from Danum Valley Conservation Area, Sabah, Malaysia for BirdNET transfer learning exercise (1.0.0) [Data set]. Zenodo. <https://zenodo.org/records/10790619>

and ‘noise’. Each training folder contains 10 individual training clips for the corresponding sound class.

The ‘Testdata’ folder contains one 2-hour sound file and a corresponding Raven selection table containing manual annotations of ‘gibbon.female’, ‘hornbill.helmeted’ and ‘hornbill.rhino’. After you have trained a new model, you can evaluate its performance by running it on the ‘Testdata’ audio file and comparing its output to the human-generated selections in the manual selection table, as described in Section 5.3 below.

1. Download the zipped archive of the sample dataset [here](#), and unzip the archive file.

We suggest the following folder organization to create a custom BirdNET model:

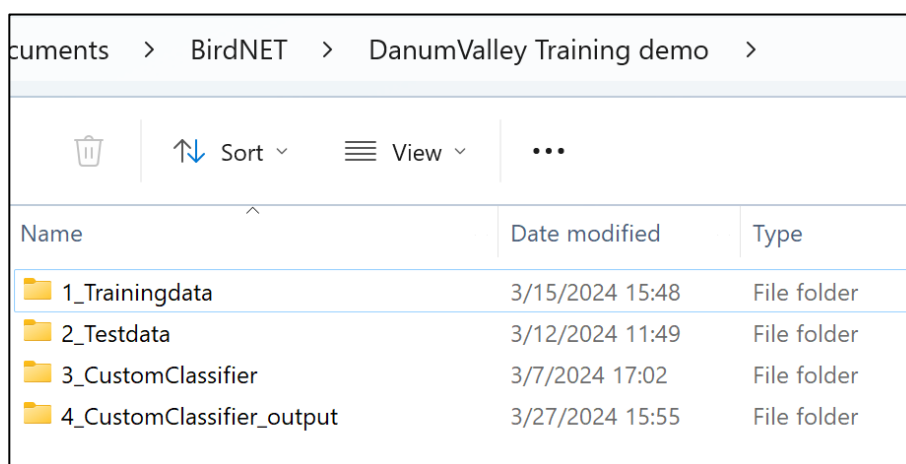


Figure 37. Suggested folder organization for custom model training exercise. In this case, the first two folders were downloaded from the [Zenodo repository](#).

2. Launch BirdNET-Analyzer-GUI by double-clicking on its icon.
3. Go to the “Train” tab (Figure 38A).
4. Click the “Training data” button (Figure 38B) and choose the folder that contains the subfolders for each sound type (Figure 38C). You should now see the four classes listed under the “Training data” button (“gibbon.female”, “hornbill.helmeted”, “hornbill.rhino”, “noise”).
5. Click the “Classifier output” button to choose where to save this model and select a new name (Figure 38D, E).
6. Choose the output format for the newly trained model (Figure 38G). If the model will be used within the BirdNET Analyzer GUI, choose ‘tflite’. To use the model in

Raven, as described Section 2 above, choose ‘raven’, or choose ‘both’ to save the model in both formats.

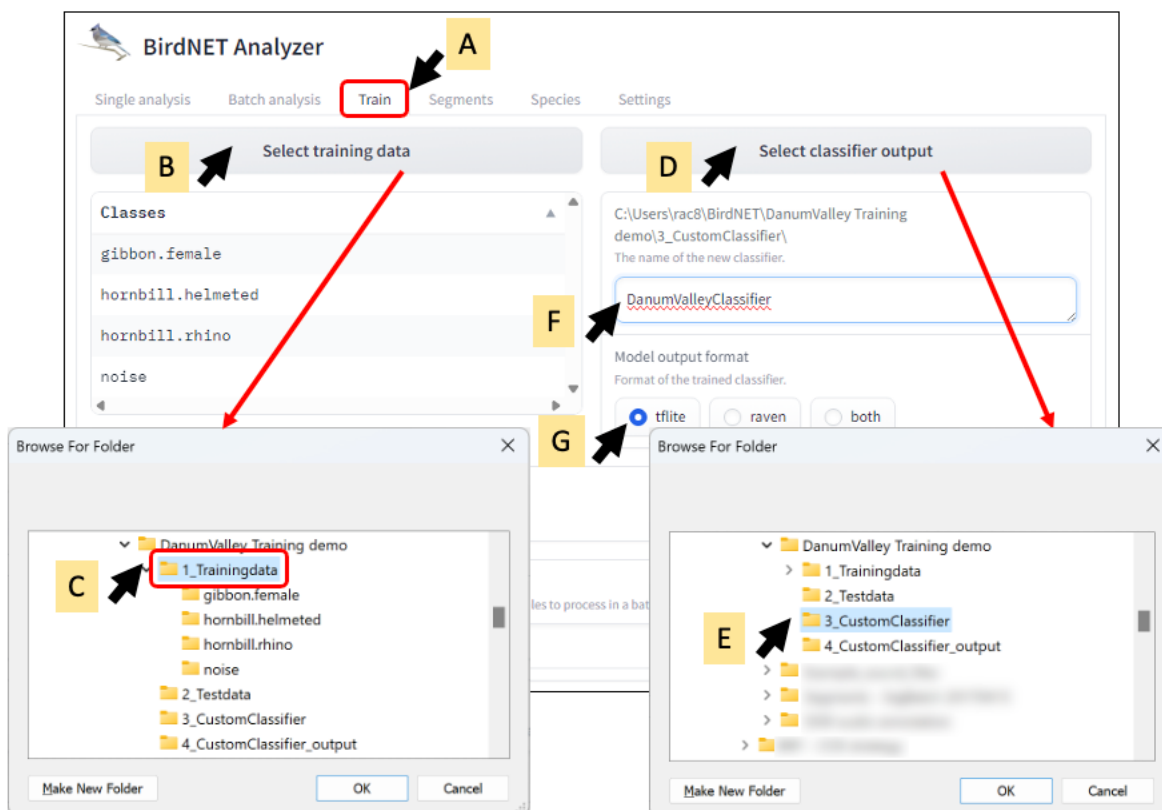


Figure 38. Configuring training for a custom BirdNET classifier.

7. The BirdNET GUI has several training parameters that you can adjust (accessible when you scroll down in the Train tab of the GUI below the view in Figure 38), including the number of epochs, the batch size, the learning rate, and the number of hidden units. We recommend not changing these parameters at first, as they are appropriate for most situations.
8. Scroll to the bottom of the Train tab, and click “Start Training”.
9. You will see a progress bar labeled “Loading data & building classifier”. At first, it will look like nothing is happening. Once it starts, it will likely take a few minutes, and longer than the estimated time. When it is finished, the progress bar will be replaced by a figure showing how the precision improved over the training process. When this process is complete, you will have a trained machine learning model that is ready to identify sounds.

5.2 Apply your model

Running a new model in the BirdNET GUI

1. Click on the “Multiple Files” tab (Figure 39A).

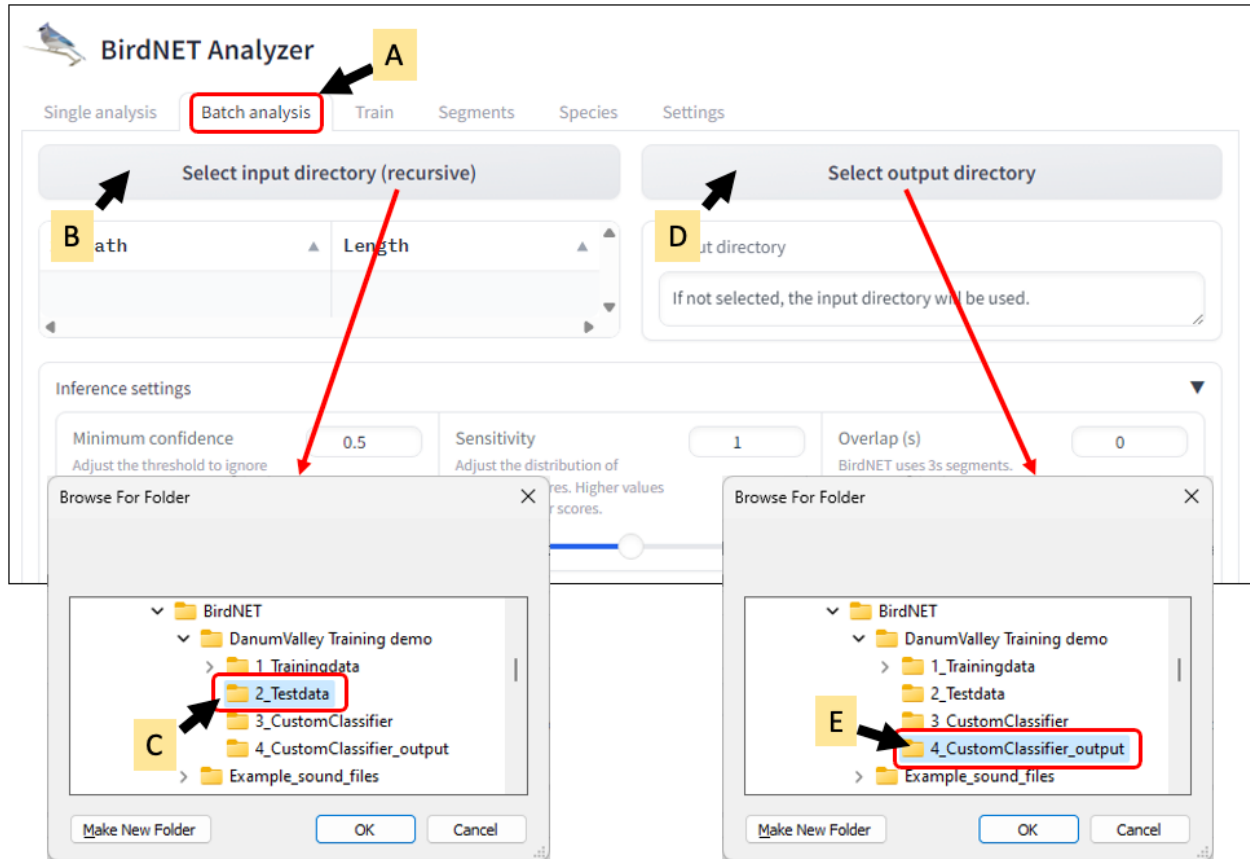


Figure 39. Configuring input and output directories for running a model.

2. Click “Select directory (recursive)” (Figure 39B) and select the folder that contains the test file (Figure 39C). This can also be a parent folder that contains subfolders.
3. Click “Select output directory” (Figure 39D), then choose the folder where you would like to save your output (Figure 39E).
4. For your first iteration of testing the newly trained model, we recommend using the defaults for the inference settings. They can be adjusted in subsequent iterations if needed to improve model performance.
5. Next, scroll down to the “Species selection” section (Figure 40). This is where you can select your own model to run in the BirdNET GUI. Choose “Custom classifier”

(Figure 40A), then click “Select classifier”(Figure 40B). Choose the classifier that you trained earlier (Figure 40C).

6. Choose your “Result type”. Here we will select “Raven selection table”.

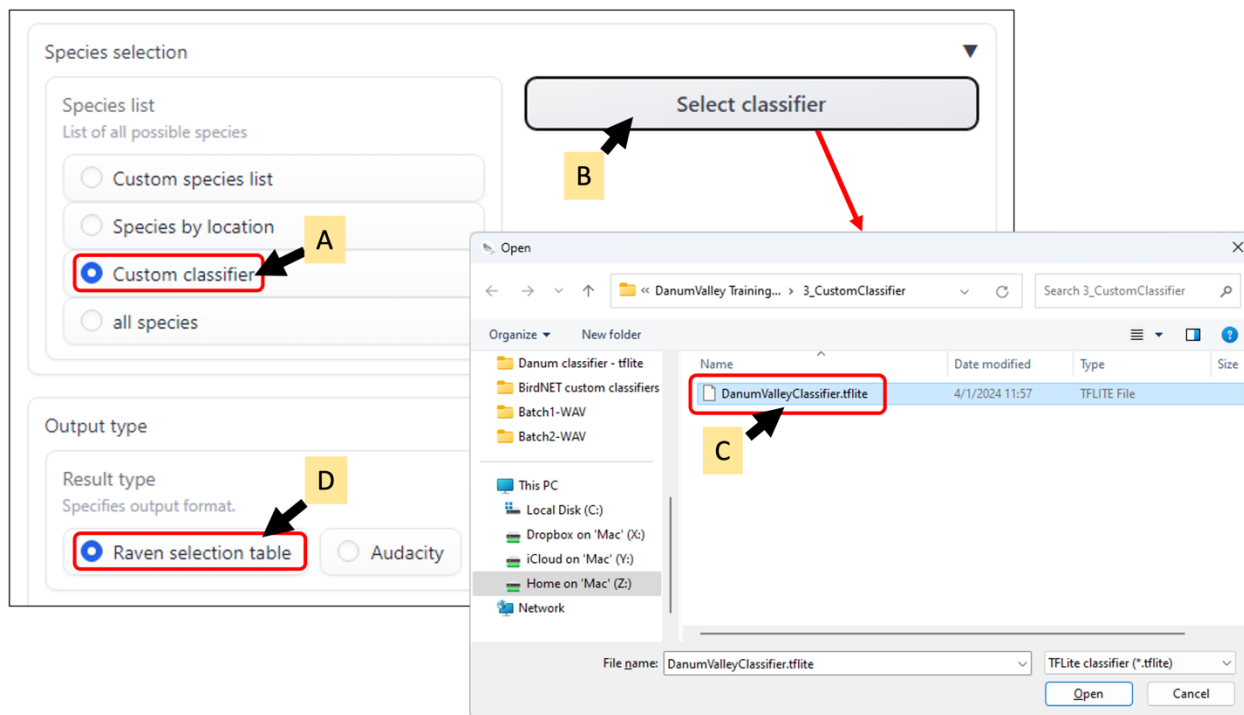


Figure 40. Selecting a custom classifier and output format.

7. Click “Analyze”. While the analysis is running, a continuously updated elapsed time counter will be displayed below the “Analyze” button (Figure 41). When the analysis is complete, BirdNET displays the name(s) of the file(s) processed and the “Execution” result (Figure 42). An execution result of “true” indicates that the file was successfully processed.

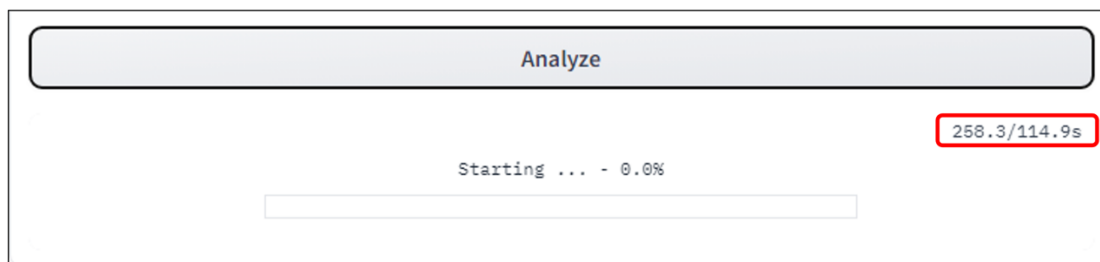


Figure 41. BirdNET displays the elapsed time (in this case, 258.3 seconds) for the analysis while files are being processed.

File	Execution
S19_20180214_080003_resampled.wav	true

Figure 42. BirdNET displays the name(s) of all file(s) processed when the analysis is complete.

Running a new model in Raven Pro

If you saved the newly trained model in Raven format, you will need to move or copy the folder containing the new model into the Raven “Models” in order for Raven to run the model.

1. Find the folder created by BirdNET containing the Raven version of the new model (which consists of several files and folders). The folder will be located in the directory you chose using the “Classifier output” button when you configured the training run, and will have the name you specified.
2. Move or copy the model folder into the folder named “Models” in your Raven folder. On Windows, your Raven folder is located by default at `C:\Program Files\Raven Pro 1.6\`. On macOS, your Raven folder is located at `/Users/<yourUserName>/Raven Pro 1.6`.
3. Follow the instructions in Section 2, *Applying the BirdNET model in Raven*, but choose your newly trained model (instead of the default BirdNET model) in Figure 3.

5.3 Examining the output of a custom BirdNET model

To review the output of your custom BirdNET model, open the Danum Valley test sound file (named “S19_20180214_080003_resampled.wav”) in Raven, and then open the selection table generated by your custom BirdNET model.


Comparing BirdNET predictions to manual annotations

To compare BirdNET’s predicted species labels with manually created annotations, first create a second selection table tab by right-clicking on the existing “Table 1” tab, then choosing “New Table”. Then open the manually created selection table by dragging and dropping its file icon onto the empty table tab, or by choosing File > Open Selection Table... . For the Danum Valley test data, this file is named “S19_20180214_080003.Table.1.selections.txt”. Selection bounds from the two selection

tables will be displayed on the spectrogram in different colors corresponding to the colors of the tabs of the tables (Figure 43).

Displaying annotation labels

If the BirdNET model you're working with has been trained to predict and label multiple sound classes, it can be helpful to have the label associated with each selection displayed directly on the spectrogram. You can do this by configuring Raven's *selection labels* to display the contents of a given annotation column, in addition to the selection ID, which is displayed by default. Instructions for configuring selection labels are provided [here](#).



When BirdNET creates a series of contiguous selections, their selection labels may overlap and be difficult to read (Figure 43). Zooming in on the spectrogram time axis, and choosing a smaller font for selection labels can reduce this problem.

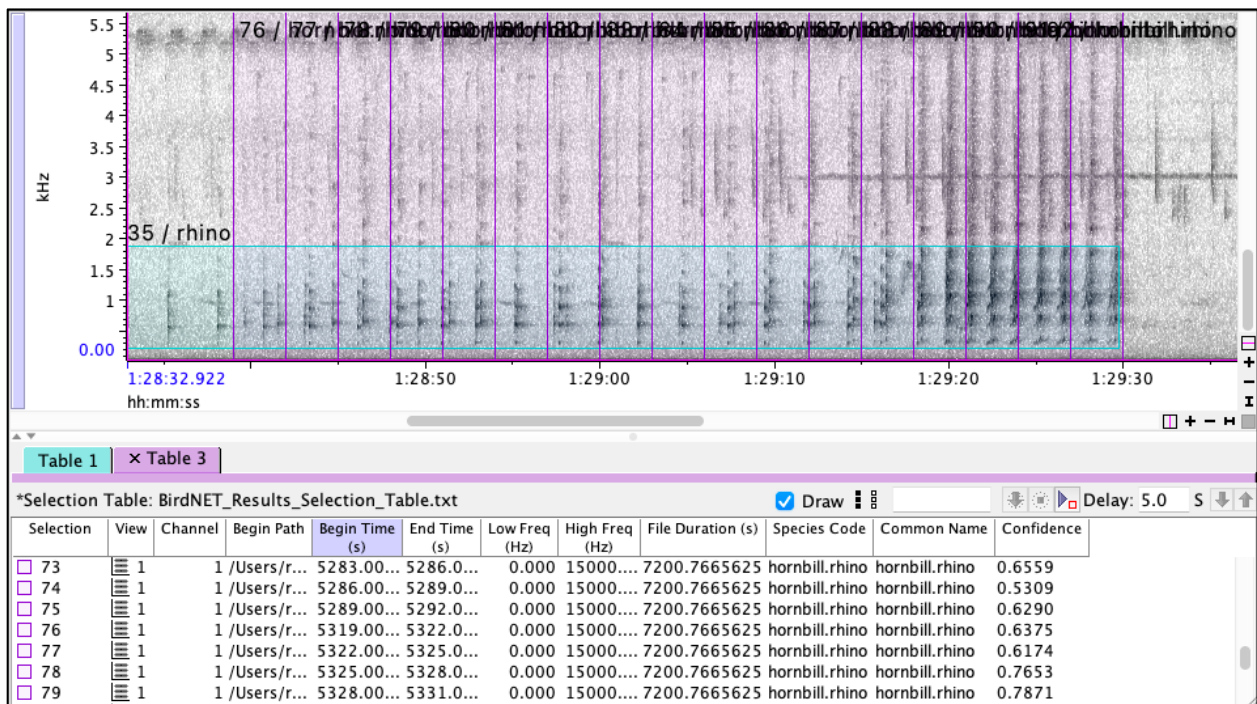


Figure 43. Comparing predictions from a custom BirdNET model to manually created annotations. The selection outlined in turquoise from approximately 200 to 1800 Hz was manually created by a human to label a sequence of calls by a rhinoceros hornbill (Species Code 'hornbill.rhino'); the series of 3-second selections outlined in purple spanning the entire frequency range were labeled as rhinoceros hornbill by a custom BirdNET classifier.

5.4 Improving the performance of a custom BirdNET model

Prediction, true and false

Any classification process (henceforth *classifier*, including human judgement) labels each input item (e.g., each segment of a continuous sound stream) as belonging to or more specific target classes, and leaves non-targets unlabeled. When the classifier correctly labels an input item as a target, we call this a true positive (TP) prediction.

All classifiers are vulnerable to two types of errors. *False positive (FP)* predictions, also known as Type I errors or false alarms, occur when the classifier erroneously labels a sound. *False negative (FN)* predictions, also known as Type II errors or missed detections, occur when the classifier fails to label a target sound that is present.

If we take human judgements of presence or absence of target sounds as our nominal “ground truth” (even though it is of course imperfect) when using an automated classification algorithm such as BirdNET, FP errors can be identified by human review of the classifier output. FN errors can only be identified by comparing the classifier output with complete human annotation of a sample of the input data to find targets that were missed by the classifier.

All classifiers inevitably exhibit a tradeoff between the rate of FP and FN errors. For example, a specific BirdNET model can be adjusted to reduce the rate of FP errors (e.g., by raising the confidence threshold parameter), but doing so will increase the rate of FN errors (missed detections). Choosing an optimal balance between the two types of error depends on the goals and priorities of a specific study.

In the following sections, we briefly describe approaches to improving the performance of a BirdNET model by reducing FP or FN error rates. In some cases performance can be improved simply by adjusting the Threshold parameter of an existing model; in other cases, improvements may require retraining the model with additional data.

Reducing false positives

There are several possible approaches to reducing the number of false positives in BirdNET’s output. Which approach is likely to be most effective depends on the specific cause of most of the errors. Here we describe three common scenarios.

1. Many FPs from a specific type of non-target sound

Sometimes BirdNET consistently mistakes a specific common non-target sound for a similar target, resulting in many false positives. For example, if you’re aiming to detect a particular bird species, and the recording contains many instances of a similar frog

sound, BirdNET might repeatedly mistake the frog sound for the target bird sound. In this case, the frog sound would be considered a *confounder* (or *confounding class*).

In this type of situation, try retraining the model with an additional folder of “negative examples” for a given target. A negative example folder must have a name that begins with ‘-’ and otherwise exactly matches the name of another folder containing examples of a particular target. For example if you have a training folder named `myTargetBird`, you would create a new folder named `-myTargetBird`, then populate that folder with examples of the confounding class taken from BirdNET’s mislabelled outputs. Then retrain the model.

During training, negative training classes help BirdNET learn to more effectively discriminate between the target and one or more specific confounders.

2. Many FPs from two target sound classes being confused with each other

If BirdNET frequently mistakes sounds from one target class for another target class, try adding the FP events labeled by the initial model to the training folders for their correct class, then retrain the model.

3. Many FPs from diverse non-target sounds

If a BirdNET model generates many FP detections that are from variable and diverse sources (rather than a single confounder as in case 1 above, or from misclassifying sounds from one target as being from another target, as in case 2), try re-running the same model with a higher threshold. In many cases, this will reduce the number of FPs. However it may also reduce the number of true positive detections (i.e., increase the number of FN errors). Whether this tradeoff is acceptable will vary depending on the particular model and the goals of your project.

If raising the threshold does not provide adequate improvement in performance, add the FP detections from the initial model to the “noise”, “background” or “other” training folder, then retrain the model. Adding a greater diversity of manually selected sounds to those folders may also help.

Reducing the number of missed target sounds (FNs)

If a BirdNET model is failing to detect too many target sounds (i.e., high FN rate), try lowering the threshold and re-running the model. In many cases, this results in more targets being detected. However, it may also yield more FP detections. Once again, whether this tradeoff is acceptable will vary depending on the particular model and the goals of your project.

If adjusting the threshold does not provide adequate improvement in performance, retrain the model, adding TP detections from the previous model to the training data. Adding additional manually selected examples of the target classes may also help, particularly if those examples represent more of the variation in the target class(es), due either to natural variation in the produced sounds, or to variation in acoustic propagation conditions or background noise.

6 Literature cited

- Ghani, B., Denton, T., Kahl, S., & Klinck, H. (2023). Global birdsong embeddings enable superior transfer learning for bioacoustic classification. *Scientific Reports*, *13*(1), 22876. <https://doi.org/10.1038/s41598-023-49989-z>
- Wood, C. M., & Kahl, S. (2024). Guidelines for appropriate use of BirdNET scores and other detector outputs. *Journal of Ornithology*. <https://doi.org/10.1007/s10336-024-02144-5>

7 Glossary

Batch size: BirdNET training parameter: the number of samples processed before the model is updated.

Classifier: Type of machine learning model that labels input data as belonging to one or more classes (e.g. Species 1, Species 2, or Species 3). Classifiers can perform either single-label or multi-label classification, depending on whether the input data can belong to only one class, or multiple classes. BirdNET is a multi-label classifier, because a given sound segment can contain sounds from multiple species.

Confidence score: A numerical value between 0 and 1, indicating the relative likelihood that a particular recording segment contains a sound from a given target species. Higher confidence scores for a given species indicate a higher probability that the segment contains a sound from that species. However, the confidence score for a species is not, strictly speaking, the probability that a sound from that species is present in the segment. The relationship between a prediction's confidence score and the probability that the prediction is correct varies among species, and may also vary among recordings made under different conditions (e.g., different sites or seasons). See also *Prediction*.

Detection: In the context of signal classification by BirdNET or other algorithms, a prediction for which the associated confidence score exceeds a specified threshold. Detections are always binary, whereas predictions have associated confidence scores. (Note that the term *detection* also has a specific but different meaning in the context of occupancy modeling, a common application of passive acoustic monitoring data.) See also *Prediction*, *Confidence Score*.

False negative: Incorrect non-prediction by a model.

False positive: Incorrect prediction by a model.

Learning rate: A BirdNET training parameter that effects how model weights are calculated.

Machine learning⁶: A subfield of artificial intelligence (AI) focused on the development of algorithms and statistical models that enable computers to perform specific tasks without explicit instructions. Instead of being programmed with step-by-step

⁶ This definition provided by ChatGPT, a large language model that is itself a product of machine learning.

instructions, machine learning algorithms use patterns and inference from data to make decisions and predictions.

Number of epochs: A BirdNET training parameter: the number of complete passes through the training dataset during the training process.

Number of hidden units: A BirdNET training parameter: related to the architecture of a neural network, which is the type of machine learning algorithm used by BirdNET.

Precision: Percentage of predictions that are correct.

Prediction: A BirdNET output indicating the presence of a target species sound in a given 3-second processing segment. BirdNET outputs a prediction for a given target species when the confidence score for the target's presence exceeds a specified threshold. BirdNET's output for each prediction includes a label, and the associated confidence score. See also *Confidence score*, *Detection*.

Recall: Percentage of true events that are found (predicted).

Sound class: The categories of sound that BirdNET is trained to recognize. This includes sounds of individual target species, and may also include classes of sounds that BirdNET is trained to ignore.

Target species: Any species that BirdNET is trained to recognize by sound.

Test dataset: A dataset used to evaluate the performance of a machine learning model. In our case, annotated soundfiles.

Training dataset: A dataset used to teach a machine learning model. For BirdNET, a training dataset consists of sound clips sorted into folders for target sound types (typically sounds of target species or other sounds of interest, such as classes of anthropogenic sounds or distinctive noise sources), and folders for non-target sound types to be ignored.

Transfer learning: A machine learning method where we use a trained model and repurpose it for a related task.

True negative: Correct non-prediction by a model for a given input item.

True positive: Correct prediction by a model for a given input item.