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

Symes L, Sugai LSMS, Gottesman B, Pitzrick M, Wood C

The sound files to accompany this guide are available at the following link:
https://tinyurl.com/BirdNETsounds

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. 2023. Acoustic analysis with BirdNET and (almost) no coding: practical instructions.

DOI: 10.5281/zenodo.8357176

1. BirdNET Analyzer Installation 2

2. Applying BirdNET to sound files 5
Part 1: Applying BirdNET in Raven 5
Part 2: Applying the BirdNET model using the BirdNET GUI 8

3. Threshold setting in BirdNET 11
Part 1: Exporting clips across a range of Confidence Scores 11
Part 2: Validating Clips using Raven Selection Review 14
Part 3: Logistic Regression 20

4. Custom BirdNET Model Building 23
Part 1: Train your own model in the BirdNET GUI 24
Part 2: Apply your model 25

Appendix A: Downloading R 28

Appendix B: Glossary 29

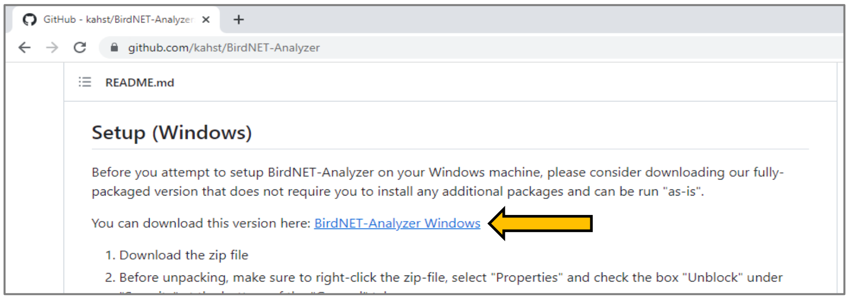
1. BirdNET Analyzer Installation

Video: https://youtu.be/SWhvkWjzamw

Platform: Windows only

BirdNET analyzer is a desktop tool for analyzing audio files with the BirdNET algorithm. It takes as input a folder or folder tree of .wav or .flac files and provides as output tables containing BirdNET predictions in the Raven sound software format. The Analyzer also provides users with a GUI tool for incorporating additional training data and detection classes.

1. Visit the webpage for the BirdNET Analyzer (https://github.com/kahst/BirdNET-Analyzer) or download the BirdNET Analyzer directly by clicking here BirdNET-Analyzer Windows. Depending on the speed of your internet connection, this download may take a few minutes.


GitHub - kahst/BirdNET-Analyzer
github.com/kahst/BirdNET-Analyzer
README.md

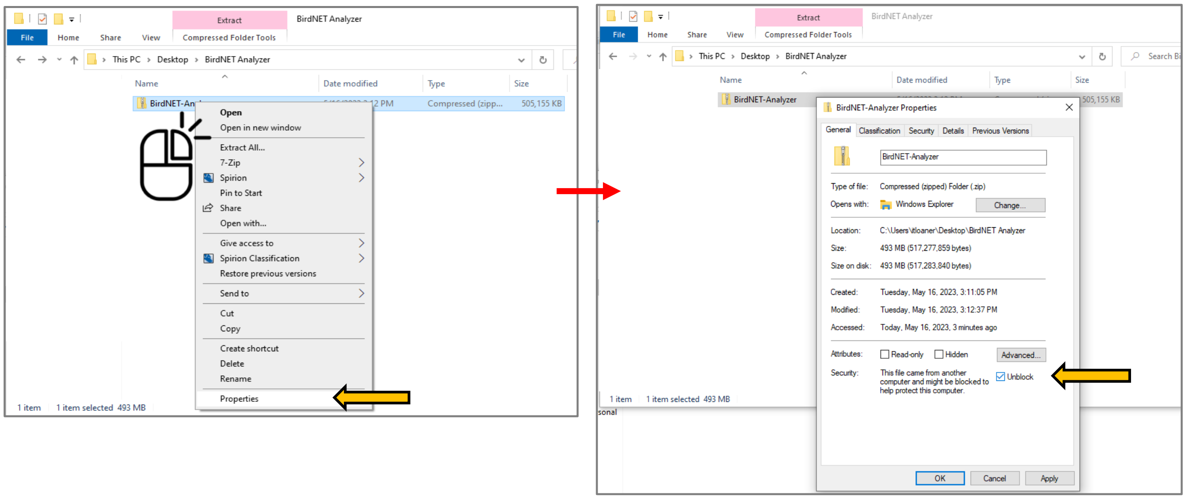
Setup (Windows)

Before you attempt to setup BirdNET-Analyzer on your Windows machine, please consider downloading our fully-packaged version that does not require you to install any additional packages and can be run "as-is".

You can download this version here: BirdNET-Analyzer Windows

1. Download the zip file
2. Before unpacking, make sure to right-click the zip-file, select "Properties" and check the box "Unblock" under

2. Find the download zipped folder (generally in the download folder).
3. While the folder is still zipped, right click on the folder and select 'Properties.' Then, select 'Unblock' and click 'OK.'



BirdNET-Analyzer Properties

General | Classification | Security | Details | Previous Versions

BirdNET-Analyzer

Type of file: Compressed (zipped) Folder (.zip)

Opens with: Windows Explorer [Change...]

Location: C:\Users\itloaner\Desktop\BirdNET Analyzer

Size: 493 MB (517,277,859 bytes)

Size on disk: 493 MB (517,283,840 bytes)

Created: Tuesday, May 16, 2023, 3:11:05 PM

Modified: Tuesday, May 16, 2023, 3:12:37 PM

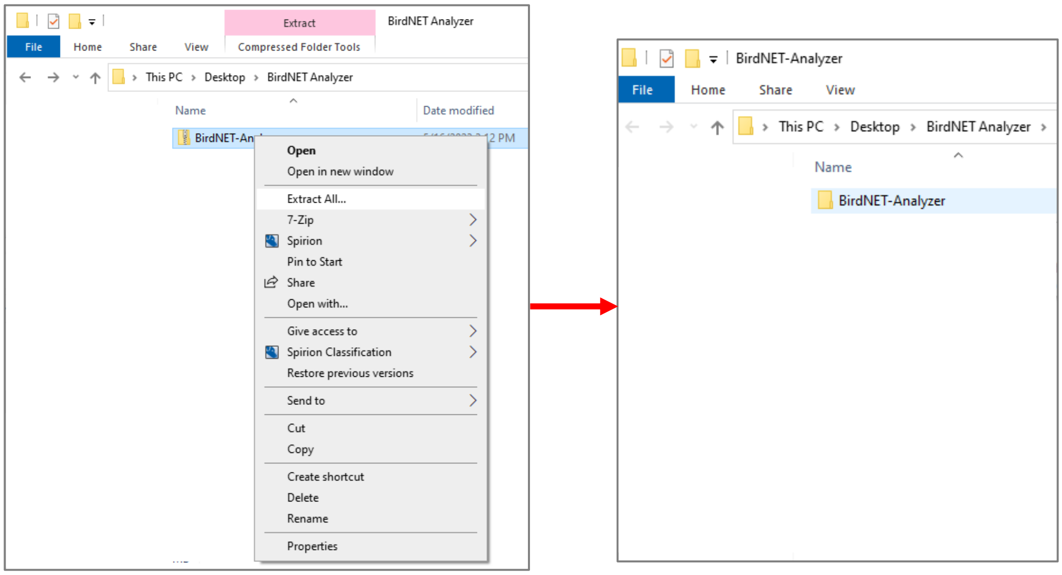
Accessed: Today, May 16, 2023, 3 minutes ago

Attributes: □ Read-only □ Hidden [Advanced...]

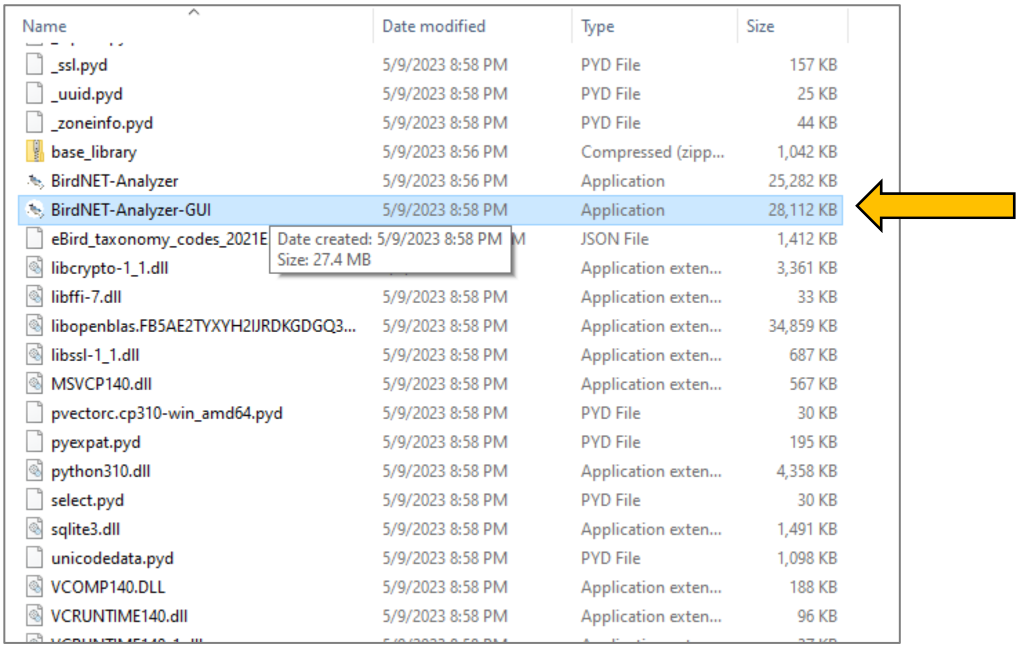
Security: This file came from another computer and might be blocked to help protect this computer. ☑ Unblock

[OK] [Cancel] [Apply]

4. Then, right click again, select extract file, and choose where you want to save it.



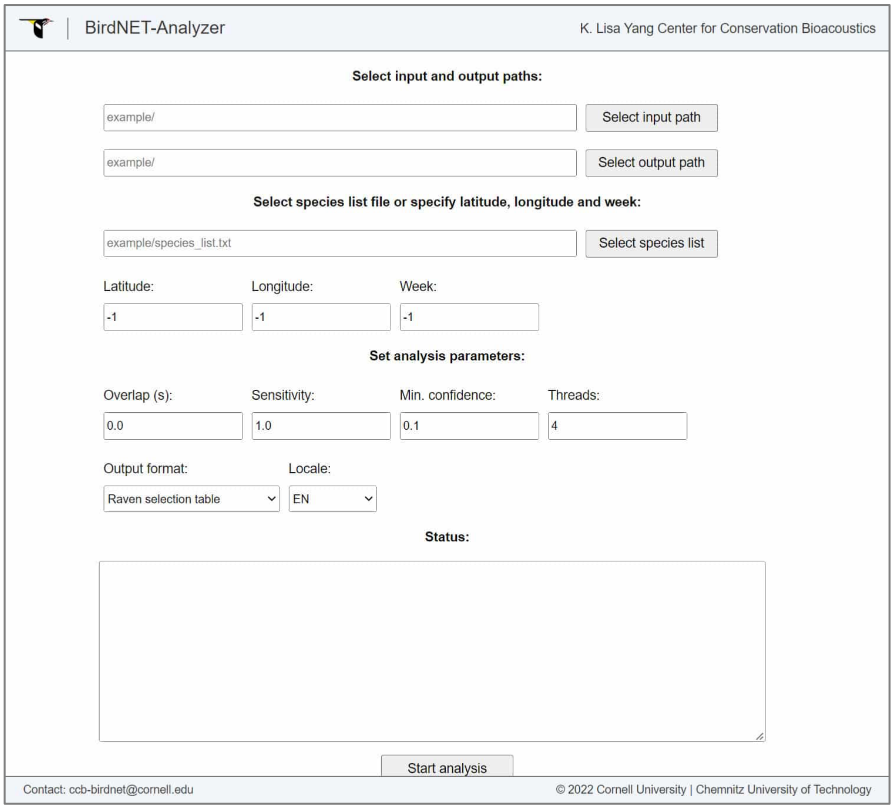
5. Then access the BirdNET-Analyzer folder and scroll down to the BirdNET-Analyzer-GUI and double click to open.



Name	Date modified	Type	Size
_ssl.pyd	5/9/2023 8:58 PM	PYD File	157 KB
_uuid.pyd	5/9/2023 8:58 PM	PYD File	25 KB
_zoneinfo.pyd	5/9/2023 8:58 PM	PYD File	44 KB
base_library	5/9/2023 8:56 PM	Compressed (zipp...	1,042 KB
BirdNET-Analyzer	5/9/2023 8:56 PM	Application	25,282 KB
BirdNET-Analyzer-GUI	5/9/2023 8:58 PM	Application	28,112 KB
eBird_taxonomy_codes_2021E	M	JSON File	1,412 KB
libcrypto-1_1.dll		Application exten...	3,361 KB
libffi-7.dll	5/9/2023 8:58 PM	Application exten...	33 KB
libopenblas.FB5AE2TYXYH2IJRDKGDGQ3...	5/9/2023 8:58 PM	Application exten...	34,859 KB
libssl-1_1.dll	5/9/2023 8:58 PM	Application exten...	687 KB
MSVCP140.dll	5/9/2023 8:58 PM	Application exten...	567 KB
pvectorc.cp310-win_amd64.pyd	5/9/2023 8:58 PM	PYD File	30 KB
pyexpat.pyd	5/9/2023 8:58 PM	PYD File	195 KB
python310.dll	5/9/2023 8:58 PM	Application exten...	4,358 KB
select.pyd	5/9/2023 8:58 PM	PYD File	30 KB
sqlite3.dll	5/9/2023 8:58 PM	Application exten...	1,491 KB
unicodedata.pyd	5/9/2023 8:58 PM	PYD File	1,098 KB
VCOMP140.DLL	5/9/2023 8:58 PM	Application exten...	188 KB
VCRUNTIME140.dll	5/9/2023 8:58 PM	Application exten...	96 KB

Date created: 5/9/2023 8:58 PM
Size: 27.4 MB

6. Wait for the program to open (may take a minute).



BirdNET-Analyzer

K. Lisa Yang Center for Conservation Bioacoustics

Select input and output paths:

example/

example/

Select species list file or specify latitude, longitude and week:

example/species_list.txt

Latitude: -1

Longitude: -1

Week: -1

Set analysis parameters:

Overlap (s): 0.0

Sensitivity: 1.0

Min. confidence: 0.1

Threads: 4

Output format: Raven selection table

Locale: EN

Status:

Contact: ccb-birdnet@cornell.edu

© 2022 Cornell University | Chemnitz University of Technology

7. Required Inputs
    a. Select input path: Choose the folder with your sound files.
    b. Select output path: Choose the folder you want to write selection tables to.

8. Optional Inputs: You can improve your results considerably by reducing the list of bird species BirdNET-Analyzer is looking for by one of two methods.
    a. Select species list: Select a text file listing the bird species you expect to find. The list of all species BirdNET-Analyzer is trained to detect can be found in .\checkpoints\V2.1\BirdNET_GLOBAL_2K_V2.1_Labels.txt. However, you can reduce the number of false detections by removing species that do not occur in your area or which you are not currently interested in finding.
    b. Enter the latitude, longitude, and week (1 – 52). BirdNET-Analyzer will create a virtual species list for you that is appropriate for your geographic location and the time of year. If you aren't sure what your latitude and longitude are, see **How to Get Latitude and Longitude from Google Maps**.

2. Applying BirdNET to sound files

There are currently two ways to apply BirdNET to analyze sound files without the need to write computer code. The first approach is to analyze files within Raven Pro. Analysis in Raven Pro works well for small numbers of files, and when you wish to immediately visualize results. The second approach is to analyze files using the BirdNET GUI program. The BirdNET GUI is effective for processing large batches of files.

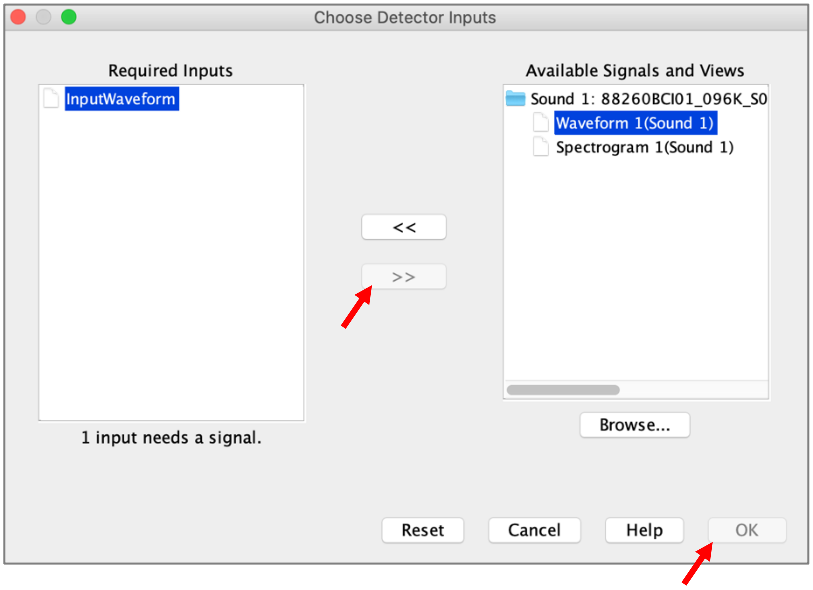
Part 1: Applying BirdNET in Raven

Software requirements: Raven Pro 1.6.5

Hardware requirements: Windows or Mac with M1 chip. As of July 2023, there has not yet been a release of the libraries needed to run the machine learning models on Macs with M2 chips.

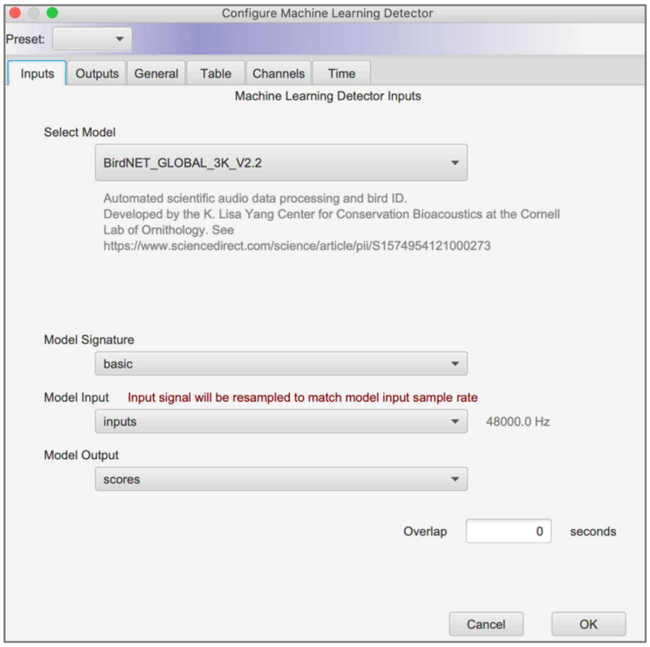
A quickstart guide to the Machine Learning Detector features in Raven can be found in https://ravensoundsoftware.com/knowledge-base/learning-detector/

1. Open the Raven Pro software (Version 1.6.5).
2. In Raven, use the File -> Open menu to open the recording(s) that you wish to analyze, or drag the recording into the Raven window.
3. To access the BirdNET detector, go to Tools -> Detector -> Learning detector.
4. Now, 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, then select the arrow in the middle of the window to move the waveform view into the left panel (under Required Inputs).
5. Select Ok.



1 input needs a signal.

6. The next menu will ask you to "Configure the Machine Learning Detector". In "Select Model", select the BirdNET_GLOBAL_3K_V2.2 model from the list of options and the other choices (Model Signature, Model Input and Model Output) will auto complete.



Machine Learning Detector Inputs

Select Model

BirdNET_GLOBAL_3K_V2.2

Automated scientific audio data processing and bird ID.
Developed by the K. Lisa Yang Center for Conservation Bioacoustics at the Cornell Lab of Ornithology. See https://www.sciencedirect.com/science/article/pii/S1574954121000273

Model Signature: basic

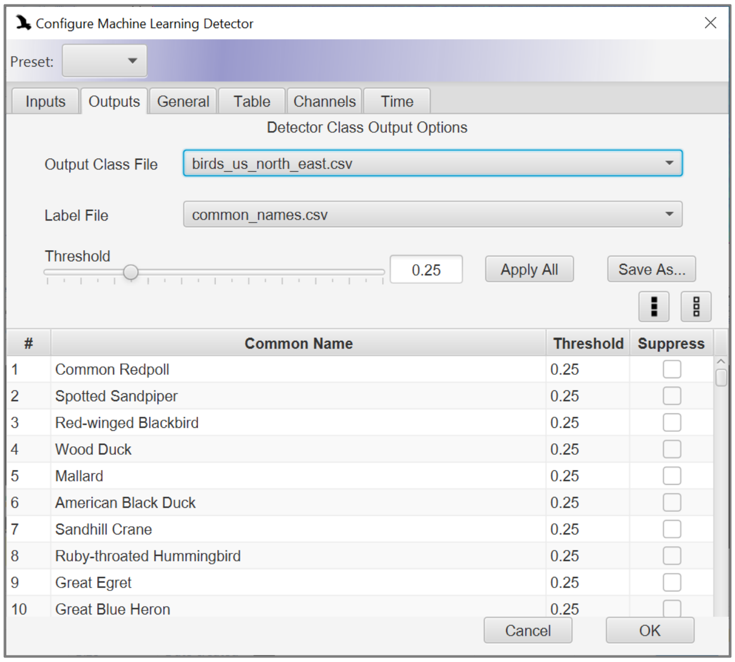
Model Input: inputs — 48000.0 Hz

Input signal will be resampled to match model input sample rate

Model Output: scores

Overlap: 0 seconds

7. Now, go to the "Output" tab to select which species you would like to detect. Select a species list using the "Output Class File" menu. You can select a global species list, but it is often more effective to select a regional species list that matches your location (that way, you don't end up with similar sounding birds from the other side of the world). For this example, select *birds_North_America*.csv as the Output Class File. The "Label File" option allows you to configure how you want the predictions to appear (common names across a few languages, eBird codes, etc).



Detector Class Output Options

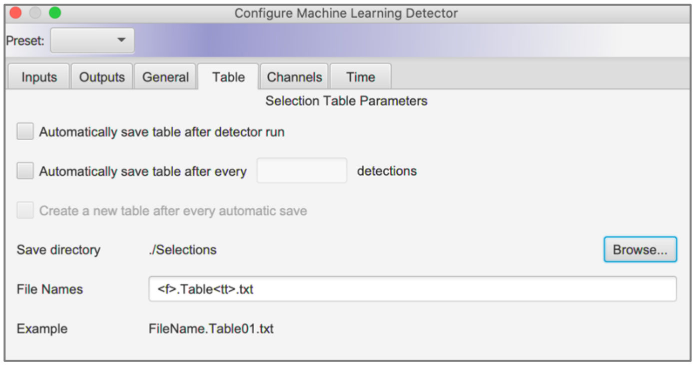
Output Class File: birds_us_north_east.csv

Label File: common_names.csv

Threshold: 0.25

#	Common Name	Threshold	Suppress
1	Common Redpoll	0.25	□
2	Spotted Sandpiper	0.25	□
3	Red-winged Blackbird	0.25	□
4	Wood Duck	0.25	□
5	Mallard	0.25	□
6	American Black Duck	0.25	□
7	Sandhill Crane	0.25	□
8	Ruby-throated Hummingbird	0.25	□
9	Great Egret	0.25	□
10	Great Blue Heron	0.25	□

8. You can also use this window to suppress any species that you don't want predicted, or to suppress all predictions except for a few focal species. Click the small box next to the species in the "Suppress" column to hide predictions for a particular species. You can also use this dialogue window to specify species-level thresholds, where only predictions above the specified threshold will be returned.
9. To select where the predictions will be saved and how they are named, go to the "Table" tab. Click "Browse" and navigate to the location where you would like the files to be saved.
10. 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).
11. Now hit 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.



Selection Table Parameters

□ Automatically save table after detector run

□ Automatically save table after every ____ detections

□ Create a new table after every automatic save

Save directory: ./Selections

File Names: <f>.Table<tt>.txt

Example: FileName.Table01.txt

12. Any detections will be shown in the selection table below the sound file and will be saved in the designated location.
13. Note that each prediction is three seconds in length (corresponding to the processing window of BirdNET). The detections will have a "Label" column with the species identification and a "Score" (a value from zero to one). Note that when predictions have values closer to 1, they are more likely to be correct and when they are closer to 0, they are more likely to be background noise or a similar species. It is important to understand that these numbers are not the probability that the prediction indeed is the suggested species, and that some species may be correctly identified when the Score is 0.3, while others are rarely correctly identified if the score is less than 0.99. For additional information, please see the guide to setting BirdNET score thresholds for individual species.

This guide provides basic instructions. For more extended instructions and options, please see: https://ravensoundsoftware.com/knowledge-base/learning-detector/

Part 2: Applying the BirdNET model using the BirdNET GUI

Software requirements: BirdNET GUI

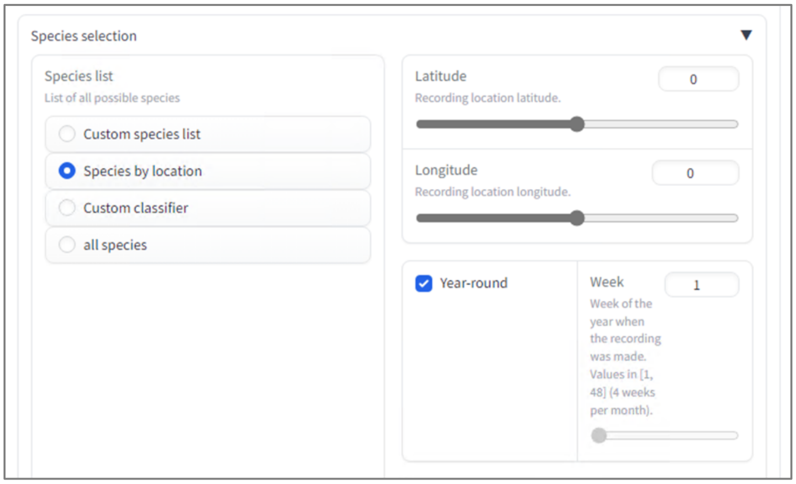
Hardware requirements: Windows computer.

Note: For M1 Macs, BirdNET can be run in batches using the Python command line option (described under Setup masOS: https://github.com/kahst/BirdNET-Analyzer#setup-macos).

The BirdNET GUI (GUI: Graphic User Interface) provides a stripped-down tool for processing large batches of files.

1. Begin by downloading and installing the BirdNET GUI. Installation video available here: https://www.youtube.com/watch?v=SWhvkWjzamw
2. Open the BirdNET GUI.
3. To run a batch of files, select the "Multiple Files" tab.
4. Select the "Select directory (recursive)" button and navigate to the file location.
5. Select "Select output directory" and select where detection files should be saved.
6. For most applications, the "Inference Settings" in the next section can be left unchanged.
7. Use the "Species selection" toggles to determine which species list is used.

8. Under the species list options, you can select “Species by location” and then restrict the latitude and longitude and the week of the year (useful for migratory species). When “Week” is set as -1, all weeks of the year are included.



Species selection ▼

Species list
List of all possible species

○ Custom species list
● Species by location
○ Custom classifier
○ all species

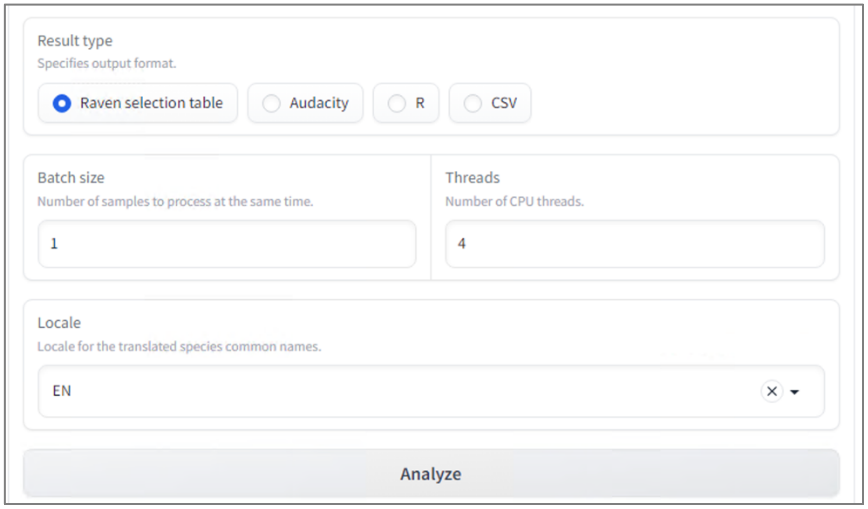
Latitude 0
Recording location latitude.

Longitude 0
Recording location longitude.

☑ Year-round

Week 1
Week of the year when the recording was made. Values in [1, 48] (4 weeks per month).

9. The BirdNET GUI allows you to select the number of threads (CPU cores) used in analysis. The maximum possible number of threads is dependent on the number of cores 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. Note that high thread values will reduce performance on other tasks while the data are processing.



Result type
Specifies output format.

● Raven selection table ○ Audacity ○ R ○ CSV

Batch size
Number of samples to process at the same time.
1

Threads
Number of CPU threads.
4

Locale
Locale for the translated species common names.
EN

10. Select “Analyze” to begin analysis. The BirdNET GUI will show files as they are processed, along with a detection bar. Analysis time will vary depending on the number and size of files processed.

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

3. Threshold setting in BirdNET

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

Background: For some species, even predictions with low BirdNET Confidence Scores are often true positives. In 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.

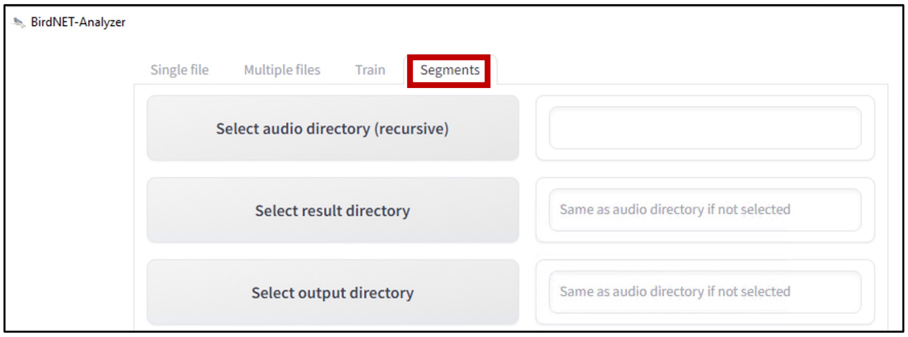
Goal: The goal of this exercise is to convert BirdNET Confidence Scores to the probability that a prediction is a true positive in a given species. This allows us to set Confidence Score thresholds that are customized to a species and location and make statements like "In this dataset, a prediction of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

Process: We will first select clips from 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 outcome (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.

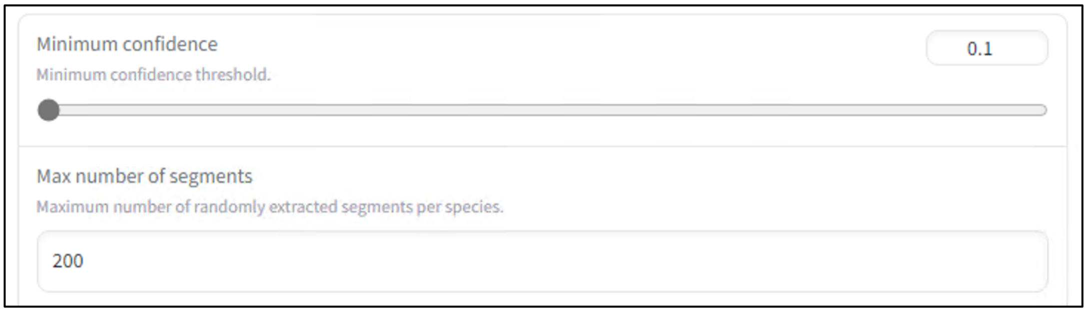
Part 1: Exporting clips across a range of Confidence Scores

To identify and export BirdNET predictions that represent a range of Confidence Scores, we will use the BirdNET GUI (this can also be achieved via 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.

1. Open the BirdNET GUI and navigate to the Segments tab (indicated with a red box below).


BirdNET-Analyzer
Single file
Multiple files
Train
Segments
Select audio directory (recursive)
Select result directory
Same as audio directory if not selected
Select output directory
Same as audio directory if not selected

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 are generated by BirdNET.
4. Then click "Select output directory" and navigate to where you would like the clips to be saved.
5. For "Minimum confidence" select the lowest Confidence Score that you would like your evaluation clips to include. 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).



Minimum confidence

Minimum confidence threshold.

0.1

Max number of segments

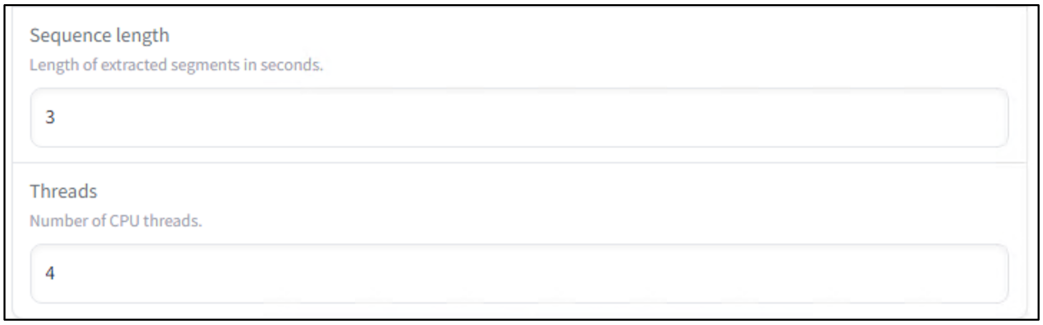
Maximum number of randomly extracted segments per species.

200

6. For "Max number of segments", indicate the number of clips that you would like to export. If you select a value that is lower than the total quantity of BirdNET predictions above that score, you will simply receive all predictions.

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

7. For "Sequence length" indicate how long you would like the segments to be. BirdNET works on three-second clips. However, if your focal signal is more than 3 seconds, you might consider using longer segments to provide additional temporal context for human reviewers.

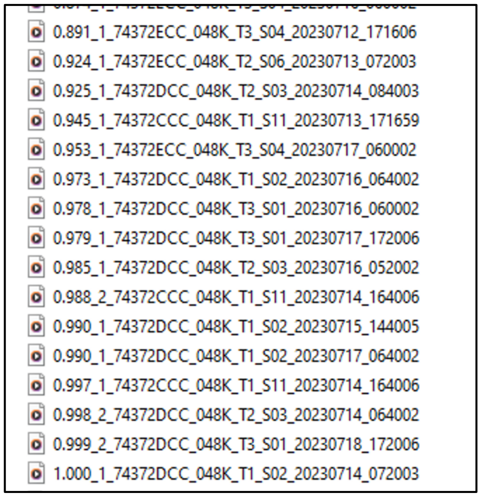


Sequence length
Length of extracted segments in seconds.
3

Threads
Number of CPU threads.
4

8. The viable values of "Threads" is dependent on the number of cores 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. Factors like sample rate and recording duration per file can influence the maximum number of threads; some guess-and-check work might be necessary to identify the maximum number. However, generating these predictions does not take long compared to the initial analysis required to generate the selection tables.

9. Finally, click on "Extract Segments", then check the identified output folder to ensure that the clips have been generated and that the clips begin with the Confidence Score (see below).



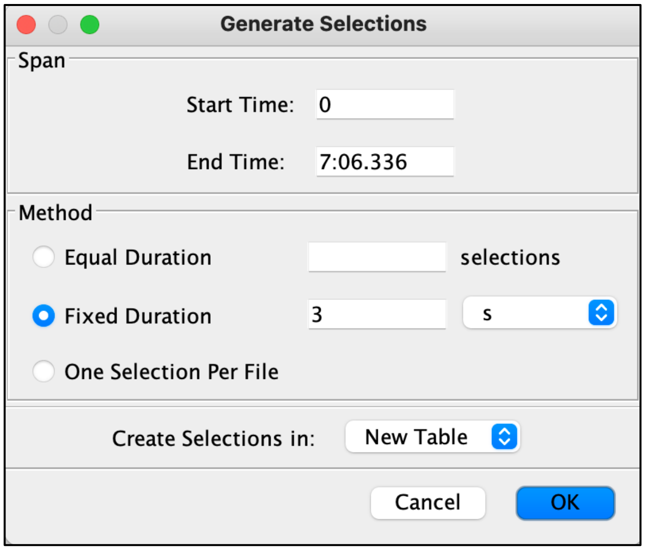
```
0.891_1_74372ECC_048K_T3_S04_20230712_171606
0.924_1_74372ECC_048K_T2_S06_20230713_072003
0.925_1_74372DCC_048K_T2_S03_20230714_084003
0.945_1_74372CCC_048K_T1_S11_20230713_171659
0.953_1_74372ECC_048K_T3_S04_20230717_060002
0.973_1_74372DCC_048K_T1_S02_20230716_064002
0.978_1_74372DCC_048K_T3_S01_20230716_060002
0.979_1_74372DCC_048K_T3_S01_20230717_172006
0.985_1_74372DCC_048K_T2_S03_20230716_052002
0.988_2_74372CCC_048K_T1_S11_20230714_164006
0.990_1_74372DCC_048K_T1_S02_20230715_144005
0.990_1_74372DCC_048K_T1_S02_20230717_064002
0.997_1_74372CCC_048K_T1_S11_20230714_164006
0.998_2_74372DCC_048K_T2_S03_20230714_064002
0.999_2_74372DCC_048K_T3_S01_20230718_172006
1.000_1_74372DCC_048K_T1_S02_20230714_072003
```

Part 2: Validating Clips using Raven Selection Review

To review and validate the individual clips, we will open all of them in Raven, add an annotation column, and use the annotation column to indicate whether each clip is a true or false prediction.

1. To validate the clips, open all clips by selecting File -> Open. Select all clips and click OK. You can also open the full set of clips by highlighting all of them and dragging them into Raven.

2. When the clips are opened, they open as a single sound file. You can generate the boundaries by clicking on Table -> Generate selections. You will see the following dialogue box:



Generate Selections

Span

Start Time: 0

End Time: 7:06.336

Method

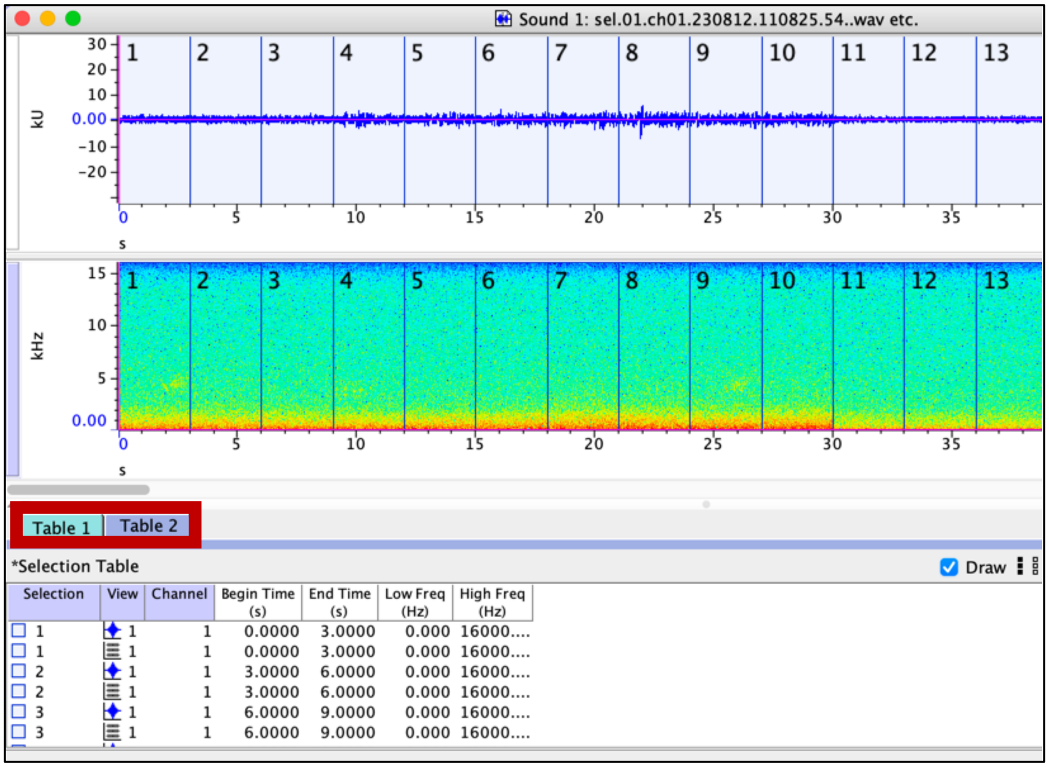
○ Equal Duration [ ] selections

● Fixed Duration 3 s

○ One Selection Per File

Create Selections in: New Table

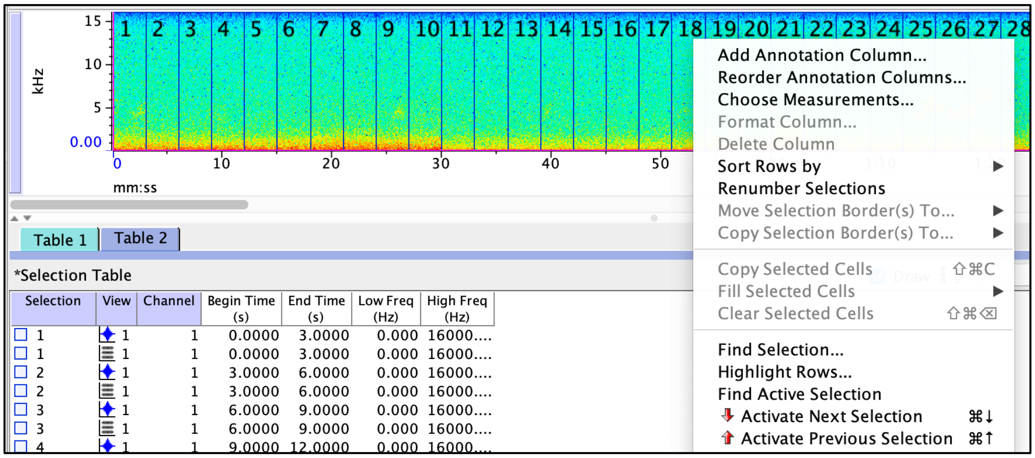
3. To validate each individual event, you will need to re-create boundaries between the clips. By default, each clip that “Segments” exports is three seconds long. To create boundaries between three second events, select “Fixed Duration” and type in “3” seconds. Now click OK and you will see three second events in the selection table (shown below).



*Selection Table

Selection	View	Channel	Begin Time (s)	End Time (s)	Low Freq (Hz)	High Freq (Hz)
□ 1	1	1	0.0000	3.0000	0.000	16000....
□ 1	1	1	0.0000	3.0000	0.000	16000....
□ 2	1	1	3.0000	6.0000	0.000	16000....
□ 2	1	1	3.0000	6.0000	0.000	16000....
□ 3	1	1	6.0000	9.0000	0.000	16000....
□ 3	1	1	6.0000	9.0000	0.000	16000....

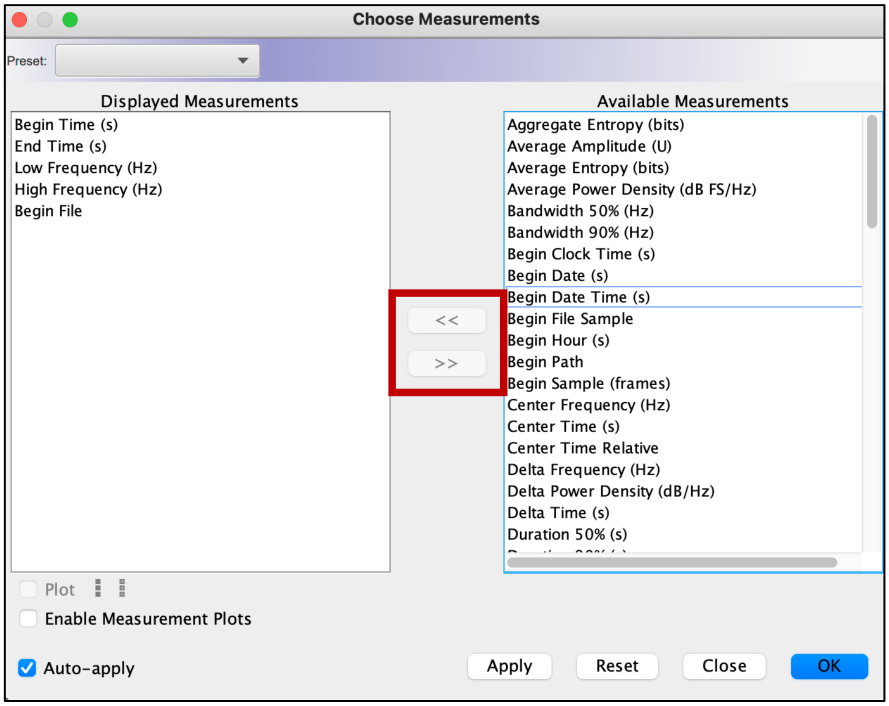
4. Now you will add the name of the file to the selection table as well so that you can match true positives and false positives to their original file and confidence threshold. Right-click on the column names of the selection table (on Mac, control + click). The select "Choose Measurements" (the third option on the menu).



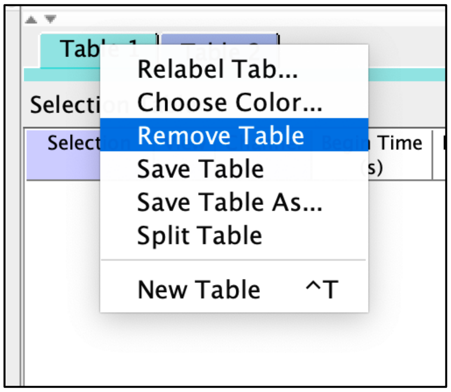
*Selection Table

Selection	View	Channel	Begin Time (s)	End Time (s)	Low Freq (Hz)	High Freq (Hz)
□ 1	1	1	0.0000	3.0000	0.000	16000....
□ 1	1	1	0.0000	3.0000	0.000	16000....
□ 2	1	1	3.0000	6.0000	0.000	16000....
□ 2	1	1	3.0000	6.0000	0.000	16000....
□ 3	1	1	6.0000	9.0000	0.000	16000....
□ 3	1	1	6.0000	9.0000	0.000	16000....
□ 4	1	1	9.0000	12.0000	0.000	16000....

5. From the list of available measurements, select "Begin File". Then, use the arrow in the middle of the dialogue box to move the measurement to the Displayed Measurements on the left side of the window (arrow shown in red box below).

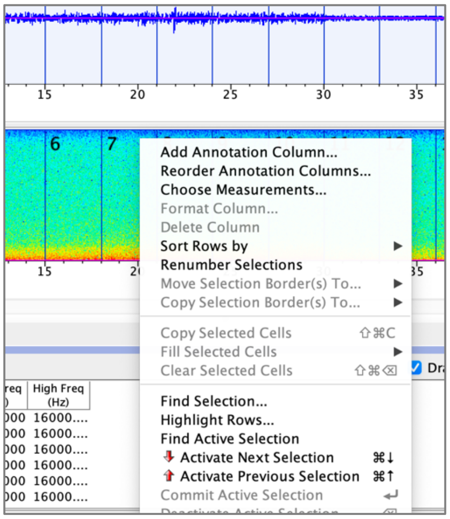


6. Note that you now have two tables (shown with the red box two pictures above). To prevent confusion, remove Table 1. Right-click (on Mac, control + click) on the tab for Table 1 and then select "Remove Table."

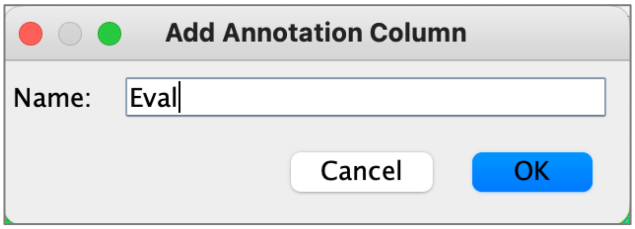


7. Next, we will create an Annotation column. Right-click on the column names of the selection table (on Mac, control + click).

8. In the menu that opens, the top option is "Add Annotation Column." Select this option and name your annotation column. For this example, we will use "Eval" as the column name to indicate that it is the evaluation of the clip (true positive or false positive).



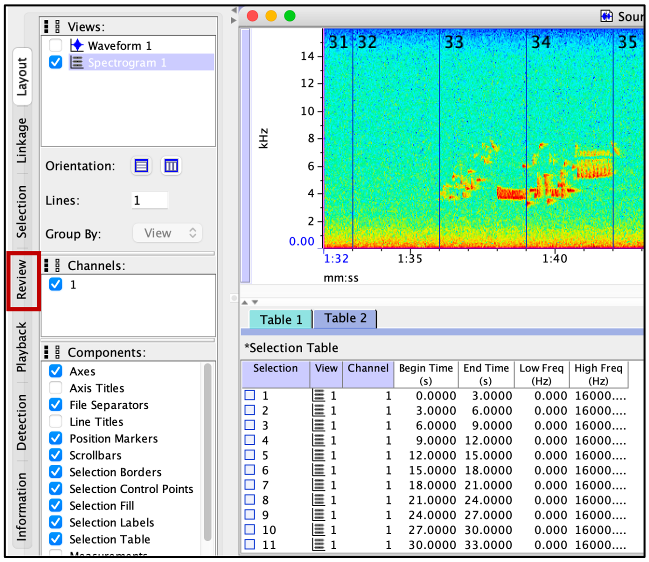
req)	High Freq (Hz)
000	16000....
000	16000....
000	16000....
000	16000....
000	16000....
000	16000....



Name: Eval

9. After naming the column, select OK.

10. Now, 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 (indicated by the red box below).



Views:
☐ Waveform 1
☑ Spectrogram 1

Orientation:

Lines: 1

Group By: View

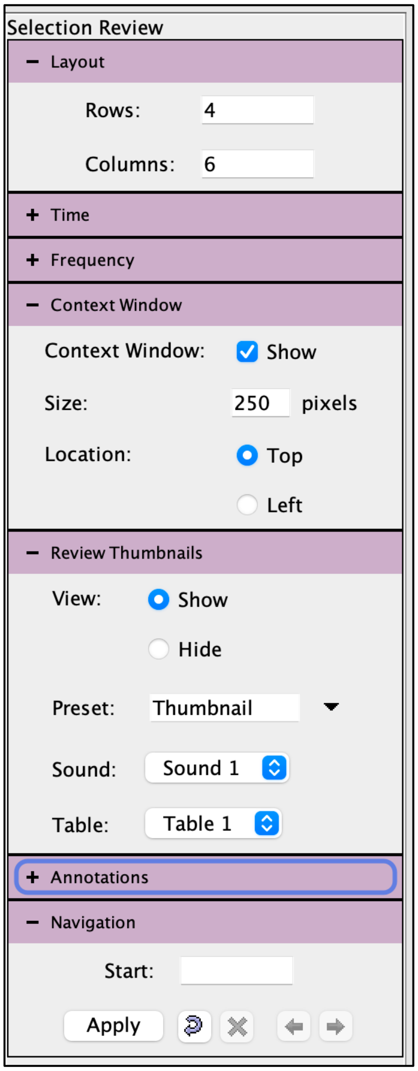
Channels:
☑ 1

Components:
☑ Axes
☐ Axis Titles
☑ File Separators
☐ Line Titles
☑ Position Markers
☑ Scrollbars
☑ Selection Borders
☑ Selection Control Points
☑ Selection Fill
☑ Selection Labels
☑ Selection Table

*Selection Table

Selection	View	Channel	Begin Time (s)	End Time (s)	Low Freq (Hz)	High Freq (Hz)
☐ 1	1	1	0.0000	3.0000	0.000	16000....
☐ 2	1	1	3.0000	6.0000	0.000	16000....
☐ 3	1	1	6.0000	9.0000	0.000	16000....
☐ 4	1	1	9.0000	12.0000	0.000	16000....
☐ 5	1	1	12.0000	15.0000	0.000	16000....
☐ 6	1	1	15.0000	18.0000	0.000	16000....
☐ 7	1	1	18.0000	21.0000	0.000	16000....
☐ 8	1	1	21.0000	24.0000	0.000	16000....
☐ 9	1	1	24.0000	27.0000	0.000	16000....
☐ 10	1	1	27.0000	30.0000	0.000	16000....
☐ 11	1	1	30.0000	33.0000	0.000	16000....

11. For now, we will keep the default settings and click "Apply" (at the bottom left of the Review window). If you receive an error that says that there are no selections to display, ensure that you have removed the empty Table 1.



Selection Review

− Layout

Rows: 4

Columns: 6

+ Time

+ Frequency

− Context Window

Context Window: ☑ Show

Size: 250 pixels

Location: ● Top ○ Left

− Review Thumbnails

View: ● Show ○ Hide

Preset: Thumbnail

Sound: Sound 1

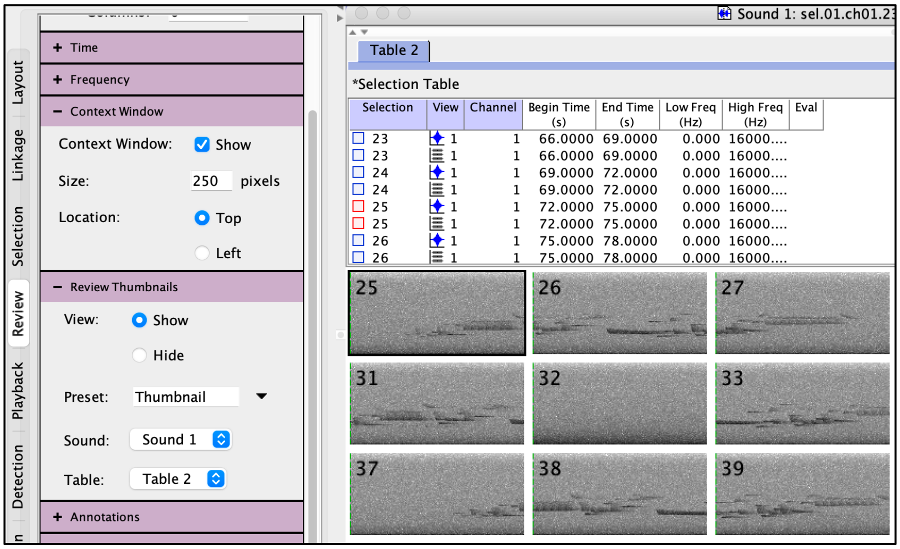
Table: Table 1

+ Annotations

− Navigation

Start:

12. Raven will then create small spectrograms for each clip. Seeing the spectrograms side by side can be helpful for visually separating true and false positives.



Context Window: ☑ Show

Size: 250 pixels

Location: ● Top ○ Left

View: ● Show ○ Hide

Preset: Thumbnail

Sound: Sound 1

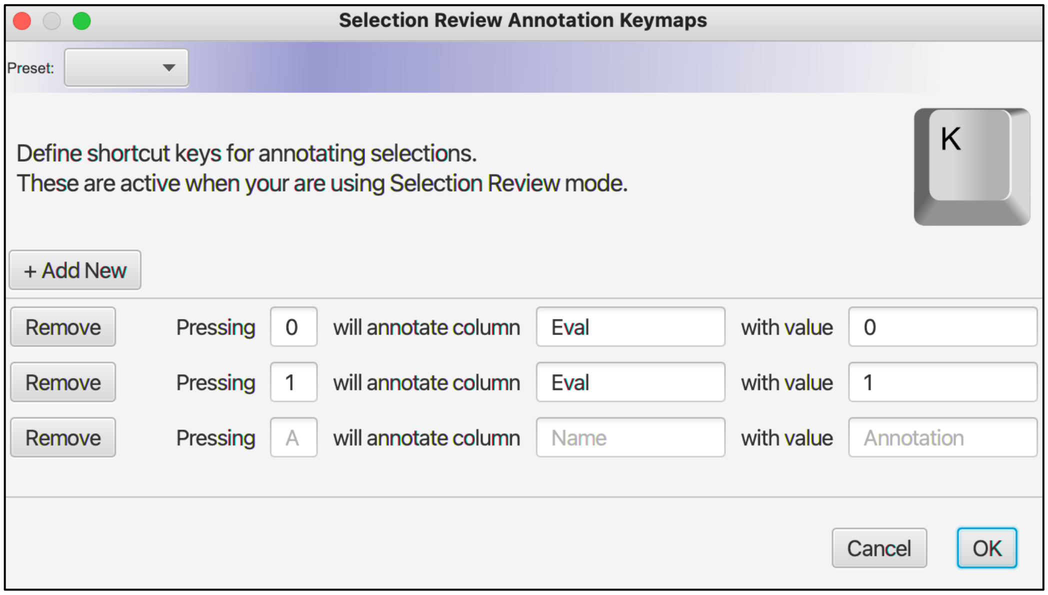
Table: Table 2

*Selection Table

Selection	View	Channel	Begin Time (s)	End Time (s)	Low Freq (Hz)	High Freq (Hz)	Eval
□ 23	1	1	66.0000	69.0000	0.000	16000....	
□ 23	1	1	66.0000	69.0000	0.000	16000....	
□ 24	1	1	69.0000	72.0000	0.000	16000....	
□ 24	1	1	69.0000	72.0000	0.000	16000....	
□ 25	1	1	72.0000	75.0000	0.000	16000....	
□ 25	1	1	72.0000	75.0000	0.000	16000....	
□ 26	1	1	75.0000	78.0000	0.000	16000....	
□ 26	1	1	75.0000	78.0000	0.000	16000....	

13. To add annotations on each clip, set up the annotation keymap. To access the keymap, use Edit -> Annotation Keymap. Choose convenient keystrokes and map values to show in the evaluation column, as shown below. It is helpful to choose short values (without spaces or other special characters) so that the labels can later be read easily by other programs. It is best to use 1 for correct and 0 for incorrect (as you'll see in Part 3) but any two codes will work. Now select OK.

Note: It's tempting to want to add a third option for "unsure". This is not inherently problematic, but it 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, it's wrong."



Selection Review Annotation Keymaps

Preset:

Define shortcut keys for annotating selections.
These are active when your are using Selection Review mode.

Pressing 0 will annotate column Eval with value 0

Pressing 1 will annotate column Eval with value 1

Pressing A will annotate column Name with value Annotation

14. Highlight one of the spectrogram panels and add an annotation by pressing one of your mapped keys. When you press the mapped key, the associated text will be written to the Eval column.

15. When you try to move past the final spectrogram in the spectrogram review pane, it will advance to the next page of images.

For a more extended description of Selection Review, please see the Selection Review and Annotation article on the Raven Knowledge Base.

Part 3: Logistic Regression

1. We will use logistic regression to determine how Confidence Thresholds translate 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 from the name of each clip file. 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).

2. To extract this information and plot the logistic regression, we will use the R software package. If you don't have R but would like to download it, see the Appendix at the end of this document. R is not necessary though, other software packages can be used here as perform this function as well.

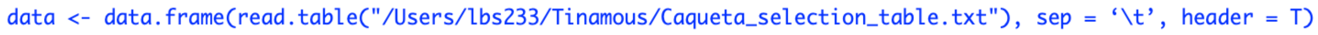
3. Import your annotated selection table, either by providing its file path:

```
> data <-
data.frame(read.table(‘you/file/path/selection_table.txt’), sep =
‘\t’, header = T)
```

or by manually selecting it:

```
> data <- data.frame(read.table(file.choose(), sep = ‘\t’, header =
T))
```

Example:



```
data <- data.frame(read.table("/Users/lbs233/Tinamous/Caqueta_selection_table.txt"), sep = ‘\t’, header = T)
```

4. Next, we will need to extract the Confidence Score from the ‘Begin.File’ column and add it as a new column:

```
> data$Score <- substr(data$Begin.File, 1, 5)
```

5. Now we fit a logistic regression model in which the binary outcome (correct or incorrect) is a function of BirdNET Confidence score:

```
> model <- glm(Eval ~ Score, family = “binomial”, data = data)
```

Note: This code will not work if you have coded ‘Eval’ as anything other than 1 for correct and 0 for incorrect. If you have used other labels, you will need to convert your labels to this format.

Optional but recommended: plot the relationship between score and outcome. First, generate a sequence of hypothetical BirdNET scores across the full range of values:

```
> prediction_values <- seq(0, 1, 0.01)
> predictions <- predict(model, list(Score = prediction_values),
type = ‘r’
```

Next, plot your observations (as points) and the predicted regression relationship (a curve)

```
> plot(Eval~Score, data=data, xlim=c(0,1), pch=16,
col=rgb(0,0,0,.3))
>lines(predictions~prediction_values, lwd=4, col=rgb(1,0,0,.5))
```

6. You can now select any particular probability and see what BirdNET Confidence Score is required to achieve it. The logistic regression equation will always give you an answer, you’ll need to interpret it carefully. For example, some probabilities may be unattainable (e.g., a BirdNET Confidence Score beyond [0, 1] is required).

```
> p <- 0.9 # your desired p (probability of true positive)
> score_threshold <- (log(p/(1-p))-model$coefficients[1]) /
model$coefficients[2]
```

4. Custom BirdNET Model Building

Goal: In this exercise, you will learn how to add additional classes to BirdNET to create a customized model (for birds, bats, gunshots, chainsaws, or other signals).

Software requirements: Raven Pro to create training data and review detections, BirdNET Analyzer GUI to select and build model.

Hardware requirements: As of July 2023, the BirdNET Analyzer GUI only works on Windows computers.

To create a custom BirdNET model, you will require a training dataset. A training dataset should be composed of short sound files s containing a vocalization/call of interest. For each class to be detected by the model, clips should be combined in a folder with the name of the class (generally a species name, call type, or anthropogenic sound). For instance, one folder for species 1 calls, another for species 2 calls, and so on. In addition to the detection classes, a folder containing all other non-target sounds contained in your PAM dataset should be created and named as 'other', 'background', or 'noise'. It is important to give exactly the 'other' name so BirdNET can process it and learn to avoid them.

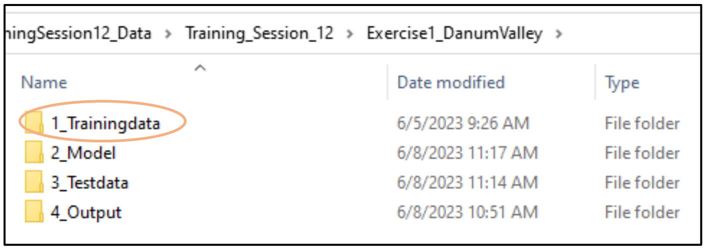
Dataset description: This dataset comes from the Danum Valley Conservation Area in Sabah, Malaysia and was collected by Dena Clink and Abdul Hamid Ahmad. The full annotated dataset will soon be publicly available at the following DOI: https://zenodo.org/record/8356807. If you want to use or distribute the data, please use the following citation:

Clink, Dena Jane, & Hamid Ahmad, Abdul. (2023). A labelled dataset of the loud calls of four vertebrates collected using passive acoustic monitoring in Malaysian Borneo (1.0.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.8356807

Part 1: Train your own model in the BirdNET GUI

Many of the species in Danum Valley are not represented in BirdNET. Today, we will build a model for the northern grey gibbon, the rhinoceros hornbill, and the helmeted hornbill.

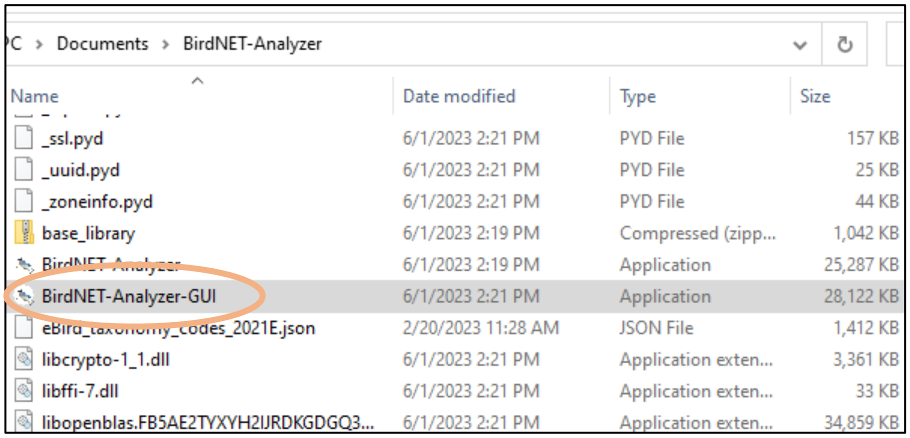
We suggest the following folder organization to create a custom BirdNET model:



ingSession12_Data › Training_Session_12 › Exercise1_DanumValley ›

Name	Date modified	Type
1_Trainingdata	6/5/2023 9:26 AM	File folder
2_Model	6/8/2023 11:17 AM	File folder
3_Testdata	6/8/2023 11:14 AM	File folder
4_Output	6/8/2023 10:51 AM	File folder

1: Open the BirdNET GUI by clicking “BirdNET-Analyzer-GUI”.



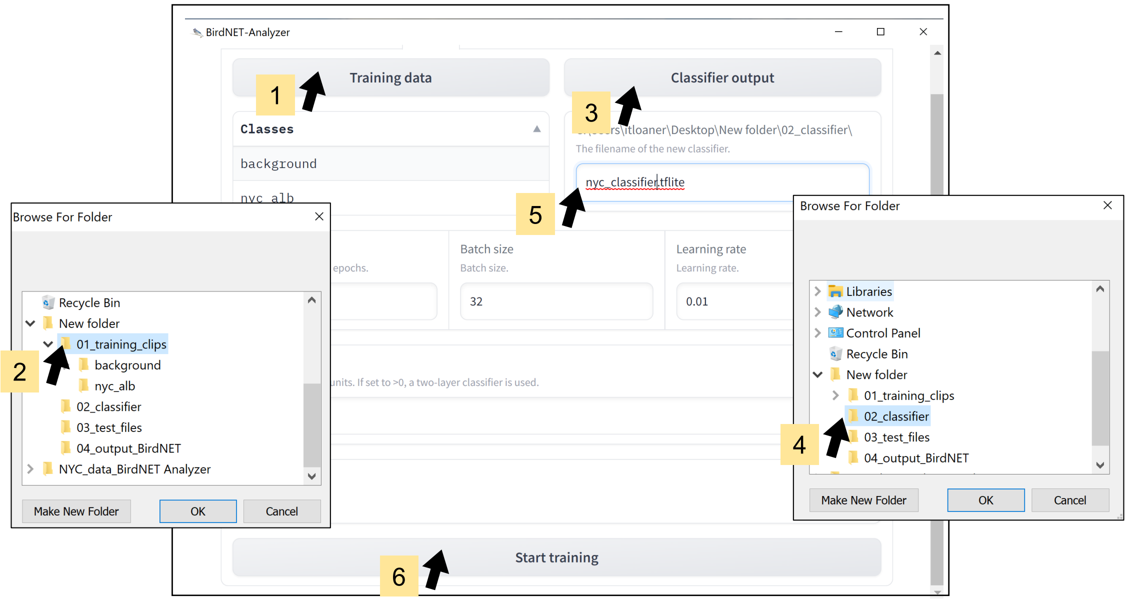
C › Documents › BirdNET-Analyzer

Name	Date modified	Type	Size
_ssl.pyd	6/1/2023 2:21 PM	PYD File	157 KB
_uuid.pyd	6/1/2023 2:21 PM	PYD File	25 KB
_zoneinfo.pyd	6/1/2023 2:21 PM	PYD File	44 KB
base_library	6/1/2023 2:19 PM	Compressed (zipp...	1,042 KB
BirdNET-Analyzer	6/1/2023 2:19 PM	Application	25,287 KB
BirdNET-Analyzer-GUI	6/1/2023 2:21 PM	Application	28,122 KB
eBird_taxonomy_codes_2021E.json	2/20/2023 11:28 AM	JSON File	1,412 KB
libcrypto-1_1.dll	6/1/2023 2:21 PM	Application exten...	3,361 KB
libffi-7.dll	6/1/2023 2:21 PM	Application exten...	33 KB
libopenblas.FB5AE2TYXYH2IJRDKGDGQ3...	6/1/2023 2:21 PM	Application exten...	34,859 KB

1. Go to the “Train” window.
2. Click the button “Training data” and choose the folder which contains the subfolders for each sound type. In the BirdNET GUI, you should now see the four classes listed under “Training data” (“gibbon.female”, “hornbill.helmeted”, “hornbill.rhino”, “noise”).
3. Click the button “Classifier output” to choose where to save this model and select a new name, keeping the ”.tflite” extension.
4. The BirdNET GUI has several model parameters that you can adjust, 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 use situations.

5. Click “Start Training”.



BirdNET-Analyzer

Training data

Classes

background

nyc_alb

Classifier output

...rs\itloaner\Desktop\New folder\02_classifier\

The filename of the new classifier.

nyc_classifier.tflite

epochs.

Batch size

Batch size.

32

Learning rate

Learning rate.

0.01

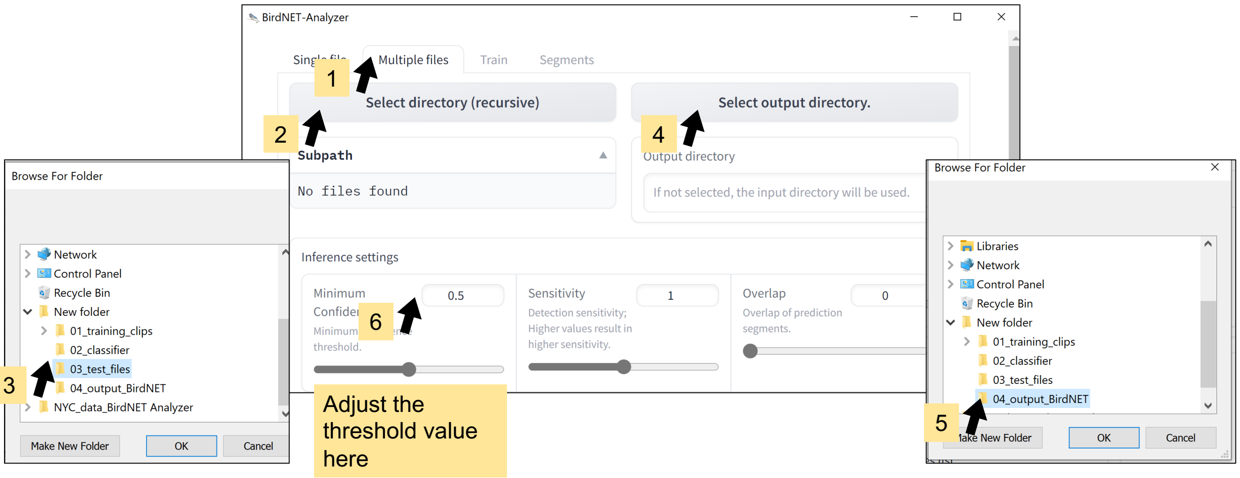
units. If set to >0, a two-layer classifier is used.

Start training

6. You will see a progress bar that says “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, you will see a figure showing how the precision improves over the training process. When this process is complete, you will have a trained machine learning model that is ready to identify sounds.

Part 2: Apply your model

1. To run the model on your sound files, go to the tab “Multiple Files”.



2. Click "Select directory (recursive)" and select the folder that contains the test file. This can also be a parent folder that contains subfolders.

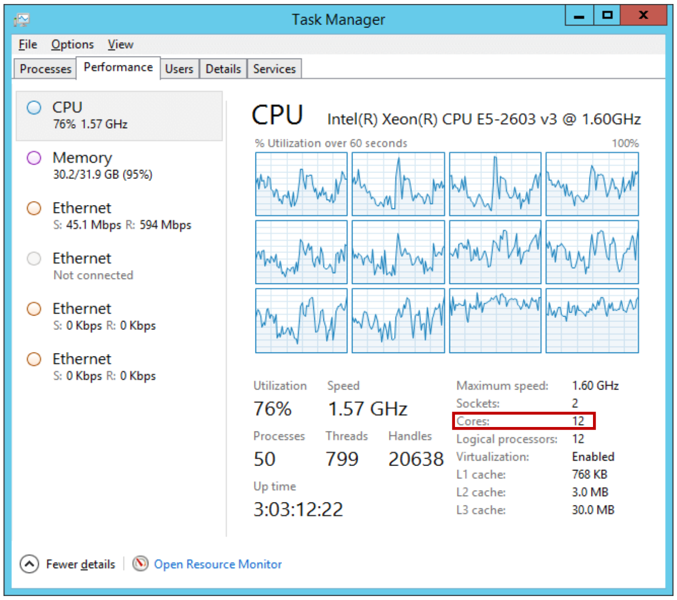
3. Click "Select output directory" to choose the folder where you would like to save your output.

4. You can use the defaults for the inference settings. These will work for most cases.

5. Next, scroll to the "Species selection" section. This is where you can select your own model to run in the BirdNET GUI. Click "Custom classifier" and then "Select classifier". Choose the classifier that you would like to use.

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

7. The "Batch Size" and "Threads" affect how fast the model will execute on your data. The "Threads" is dependent on the number of cores 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.



CPU 76% 1.57 GHz
Memory 30.2/31.9 GB (95%)
Ethernet S: 45.1 Mbps R: 594 Mbps
Ethernet Not connected
Ethernet S: 0 Kbps R: 0 Kbps
Ethernet S: 0 Kbps R: 0 Kbps

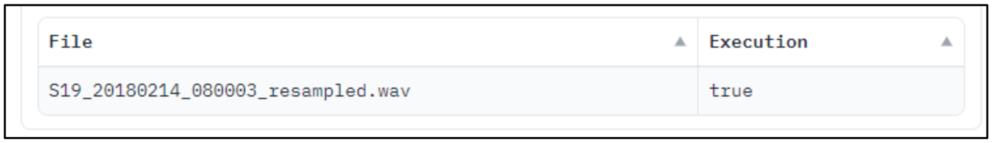
CPU Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz

Utilization 76%
Speed 1.57 GHz
Processes 50
Threads 799
Handles 20638
Up time 3:03:12:22

Maximum speed:	1.60 GHz
Sockets:	2
Cores:	12
Logical processors:	12
Virtualization:	Enabled
L1 cache:	768 KB
L2 cache:	3.0 MB
L3 cache:	30.0 MB

Look up the number of cores on your computer and set the “Threads” accordingly. For fastest performance, set “Threads” to the number of cores available on your machine, though you will not be able to do other intensive work on your computer when running the model. 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).

8. Click “Analyze”. It should take between 5-10 minutes to run. You may see “Starting…0.0%” on the progress bar for several minutes before it progresses. When it is finished, you will see.

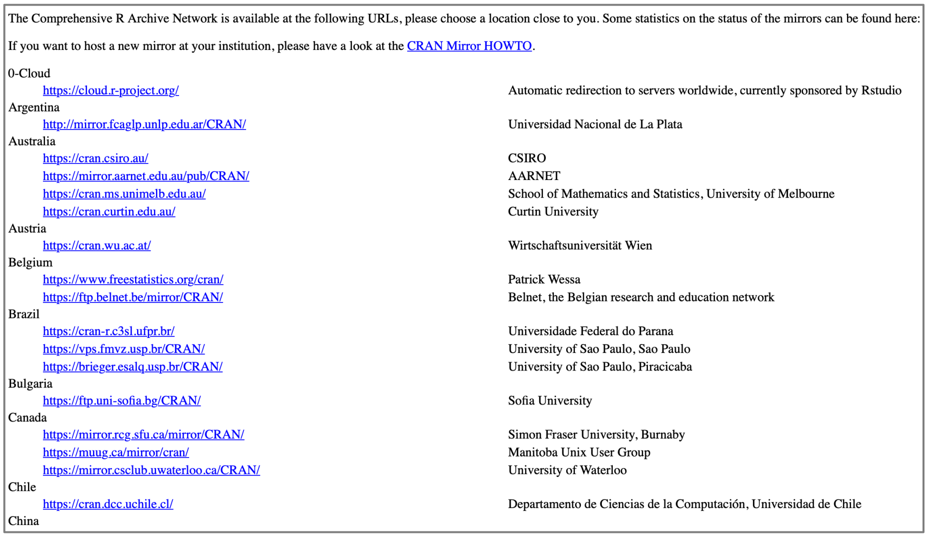


File	Execution
S19_20180214_080003_resampled.wav	true

Appendix A: Downloading R

1. Download the R software package here: https://cran.r-project.org/mirrors.html.

2. R downloads are passed through various "hubs" around the world to facilitate convenient access. Select a hub near you.

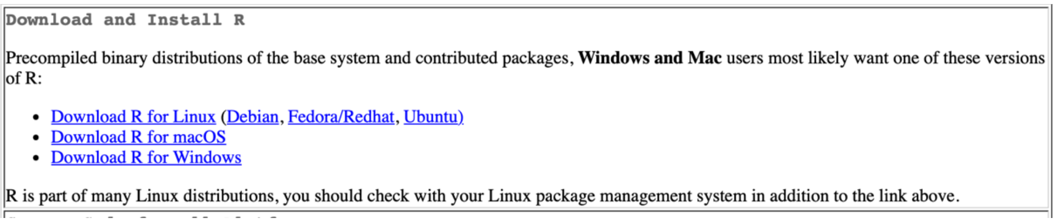


The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here:

If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

0-Cloud	
https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Argentina	
http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	
https://cran.csiro.au/	CSIRO
https://mirror.aarnet.edu.au/pub/CRAN/	AARNET
https://cran.ms.unimelb.edu.au/	School of Mathematics and Statistics, University of Melbourne
https://cran.curtin.edu.au/	Curtin University
Austria	
https://cran.wu.ac.at/	Wirtschaftsuniversität Wien
Belgium	
https://www.freestatistics.org/cran/	Patrick Wessa
https://ftp.belnet.be/mirror/CRAN/	Belnet, the Belgian research and education network
Brazil	
https://cran-r.c3sl.ufpr.br/	Universidade Federal do Parana
https://vps.fmvz.usp.br/CRAN/	University of Sao Paulo, Sao Paulo
https://brieger.esalq.usp.br/CRAN/	University of Sao Paulo, Piracicaba
Bulgaria	
https://ftp.uni-sofia.bg/CRAN/	Sofia University
Canada	
https://mirror.rcg.sfu.ca/mirror/CRAN/	Simon Fraser University, Burnaby
https://muug.ca/mirror/cran/	Manitoba Unix User Group
https://mirror.csclub.uwaterloo.ca/CRAN/	University of Waterloo
Chile	
https://cran.dcc.uchile.cl/	Departamento de Ciencias de la Computación, Universidad de Chile
China	

3. On the next page, select the appropriate download for the type of computer that you are using.



Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

4. Follow the installation steps and open R.

Appendix B: Glossary

Prediction: a BirdNET output for any given species that has not been reviewed or validated by a human analyst. BirdNET expresses its confidence in these predictions with a “confidence score” that is scaled from 0 to 1, which is not a probability despite the resemblance to one. For example, “We analyzed 1,000 hours of audio with BirdNET, which yielded 235 predictions for our target species.”

Observation: a BirdNET prediction that has been verified as correct by a qualified analyst. For example, “We reviewed 100 BirdNET predictions for this species and determined that 87 were indeed observations of our target species, while 13 were incorrect.”

Detection: a BirdNET-based observation (or prediction with associated probability that it is correct) used specifically within the context of occupancy modeling such that a given species is treated as “present” during a survey interval (i.e., secondary sampling period). We differentiate “observation” and “detection” because the latter is a well-established term in the field of occupancy modeling, a common application of passive acoustic monitoring data. Importantly, a single observation does not necessarily equate to a detection, and multiple observations may be consolidated into a single detection. For example, “We subdivided our 15-day survey into three consecutive five-day secondary sampling periods. There were six BirdNET observations of our target species in the first sampling period, zero in the second, and four in the third such that we had two detections across our three sampling periods.”

Training dataset: dataset used to teach a machine learning model. In our case, annotated sound clips of target sounds.

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

Classifier: type of machine learning model that sorts data into one or more classes (i.e. Species 1, Species 2, or Species 3). Depending on the speaker or software, the terms “model” and “classifier” are both commonly used.

Sound class: the sounds that the machine learning model is learning to classify. Consists of target sounds and the “other” categories.

Machine learning: Machine learning gives computers the ability to learn from data to find patterns and make predictions.

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

Precision: Percentage of detections that are correct.

Recall: Percentage of true events that are found.

True positive: Correct detection by the model

False positive: Incorrect detection by the model

True negative: Correct non-detection by the model

False negative: Incorrect non-detection by the model

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

Batch size: the number of samples processed before the model is updated

Learning rate: Affects how model weights are calculated

Number of hidden units: related to the architecture of a neural network, which is a type of machine learning algorithm