FEMC Final Report

Project Title: Building an Adaptive Management Framework for the Green Mountain National Forest Early Successional Habitat Creation Project

Gieder, K., Dillon, K., Huber, K., Clarfeld L. 2024. Building an Adaptive Management Framework for the Green Mountain National Forest Early Successional Habitat Creation Project: Final Report to the Forest Ecosystem Monitoring Cooperative. DOI: https://doi.org/10.18125/l386vj

Rationale

Vermont forests have gradually matured in recent decades as fire suppression and changing forestry practices have limited disturbances to ecosystems, and Green Mountain National Forest (GMNF) in Vermont is no exception. GMNF includes over 400,000 acres of forest and a wide variety of natural communities. The vast majority of GMNF, however, is dominated by maturing forests between 70 and 120 years old; other natural communities only make up ~2% of GMNF. Many wildlife species, including Ruffed Grouse (*Bonasa umbellus*), American Woodcock (*Scolopax minor*), and a variety of neotropical migratory birds rely on younger, early successional forests for habitat. As forests have matured, many of these species have experienced population declines due to habitat loss.

In 2020, Green Mountain National Forest began a 15-year early successional habitat (ESH) creation project to increase the acreage of regenerating age class forests (0 to 9 years old) to provide habitat for early successional species. They proposed a variety of timber harvest treatments, including clearcut with reserves, shelterwood, group selection, and patch cuts to create temporary openings to serve as ESH. To evaluate the effectiveness of such management actions, they required an effective long-term monitoring framework to gather data at managed sites and assess the impact of these timber harvests on early successional wildlife species.

We implemented remote acoustic monitoring in GMNF and provided an efficient framework for data management and analysis to inform management practices in their ESH creation project. These tools will allow GMNF to more easily monitor species of interest, evaluate responses to management interventions, and use these data to inform future management decisions.

Objectives

This project was undertaken with the following objectives:

1. Build automated models for site colonization, site extinction, and site persistence probabilities for priority wildlife species.
2. Develop a customized open-source spatial decision tool that enables GMNF to better match forest management outcomes to their management objectives.
3. Customize currently existing AMMonitor approach and decision tool to smaller state-, town-, or privately-owned forests, with user-friendly interactions.

Methods and Data Collection

Remote Monitoring

Remote monitoring makes use of autonomous monitoring units (AMUs) to collect large amounts of data consistently over a long period of time. For this project, we employed remote acoustic monitoring to collect audio recordings, which could then be analyzed to detect species of interest in GMNF.

Acoustic monitoring was conducted in 2022 and 2023. A total of 65 Song Meter Micro acoustic monitoring units were purchased. Units were deployed at 44 sites in 2022 and 53 sites in 2023 (Figure 1).

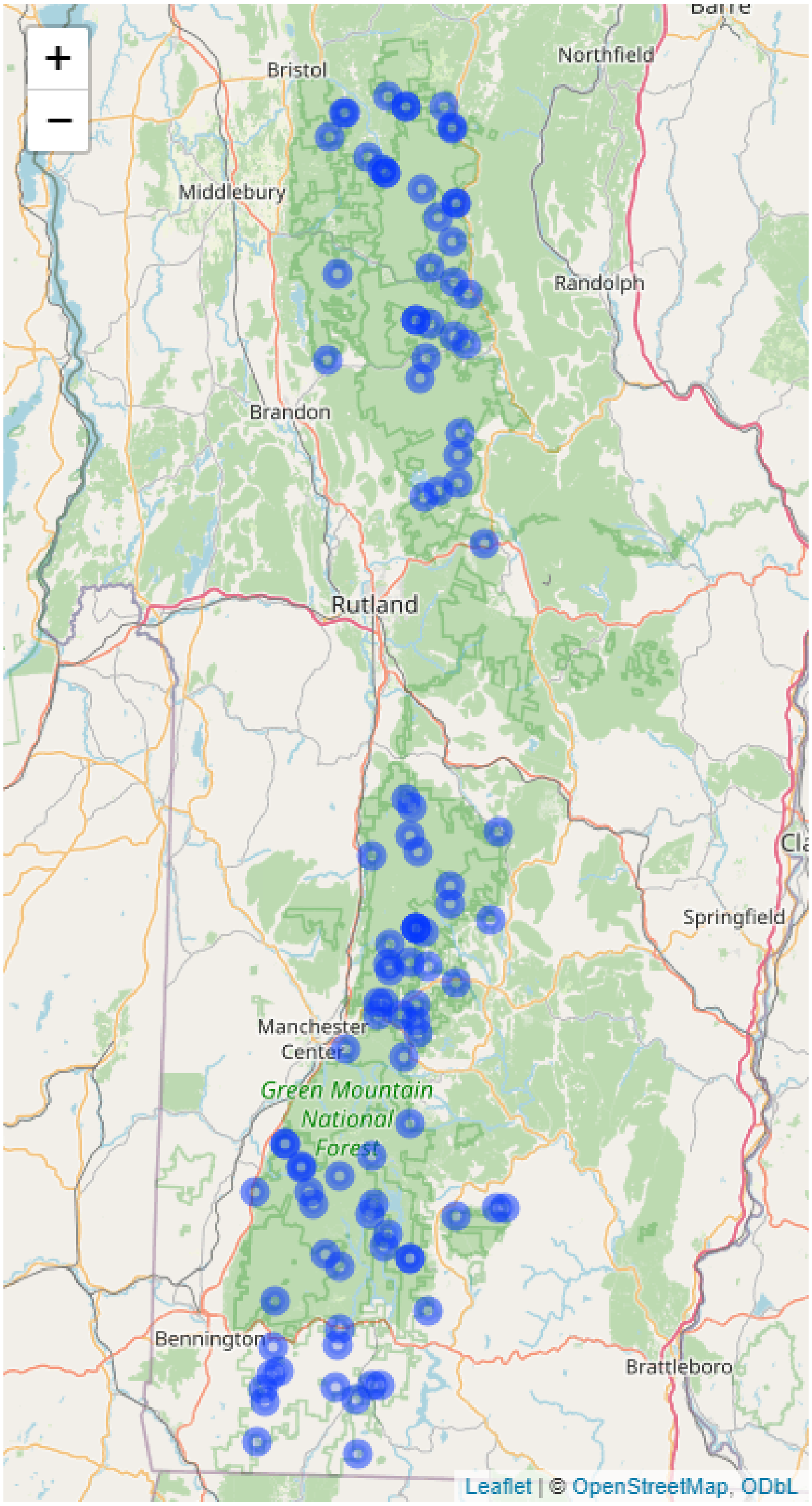

Figure 1: A map of all monitoring locations in Green Mountain National Forest from 2022 and 2023.

Sites were chosen to represent a mixture of habitat conditions in GMNF, with a focus on areas with recent timber harvests. Sites may have any of four kinds of timber harvests: clearcut with reserves, shelterwood, group selection, or patch cuts, or they may have had no timber harvest (control). Additionally, the elevation of each site was recorded, and they were noted as upland or lowland habitats.

Monitoring units were set up at each site between May and June in 2022, and between April and May in 2023. Units were removed between November and December both years. During each winter, monitoring data was downloaded, and units were assessed to ensure they were in operational condition for the following monitoring season. This time frame was chosen to capture data about spring bird migration as well as breeding species.

Each monitoring unit was securely attached to a tree at the study site. A small packet of desiccant was placed inside each unit to prevent water damage. An example monitoring site can be seen in Figure 2.

Figure 2: An example monitoring site in GMNF near a recent cut. Note the Song Meter Micro attached to the tree.

Units were configured using the default settings for a Song Meter Micro: sample rate was set to 24000 Hz and gain was set to 18 dB. Each monitoring unit collected 37 minutes of recordings per day: 1 minute of recording every 10 minutes from 05:00 to 09:00, and 1 minute every 30 minutes from 16:30 to 23:00. This recording schedule was chosen to capture the peak times for birds vocalizing and, more specifically, for Ruffed Grouse drumming. A total of 507,971 recordings were collected across all study sites, representing ~8,950 hours of audio data.

As shown in Figure 3, the number of recordings collected at each location was not entirely consistent. Some monitoring units were damaged during the monitoring season, ran out of battery, or filled up their memory cards, resulting in fewer recordings at these sites.

Additionally, a handful of sites used an experimental recording schedule; they were set to record for longer periods of time over a shorter total time frame with a focus on capturing Ruffed Grouse drumming. These sites therefore had far fewer total recordings, as seen in Figure 3. The data from the experimental recording schedule will be compared to the data collected by the standard recording schedule to assess the detection rate of drumming Ruffed Grouse using these different methods.

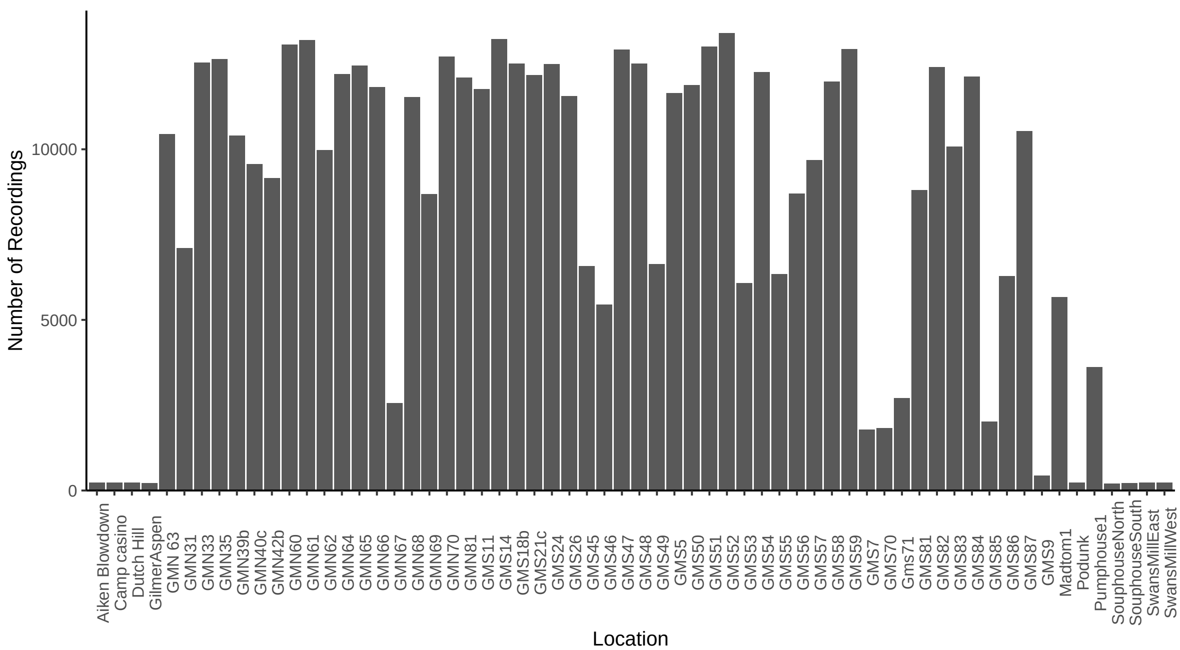

Figure 3: The number of recordings collected at each location in Green Mountain National Forest. Sites with very few recordings, such as the Aiken Blowdown and Souphouse sites, used the experimental recordings schedule.

Recordings were collected consistently throughout the monitoring season (Figure 4). The total number of recordings collected did vary slightly over time. The first and last months of the season had fewer recordings because monitoring units were being deployed and recovered. The number of recordings also tended to decrease slightly over the course of the monitoring season due to equipment malfunctions. In 2023, there was more variation than 2022 (Figure 4). This is partly due to the deployment of the monitoring units on the experimental recording schedule, as mentioned above. These units collected fewer recordings and operated for a shorter period of time than the other units.

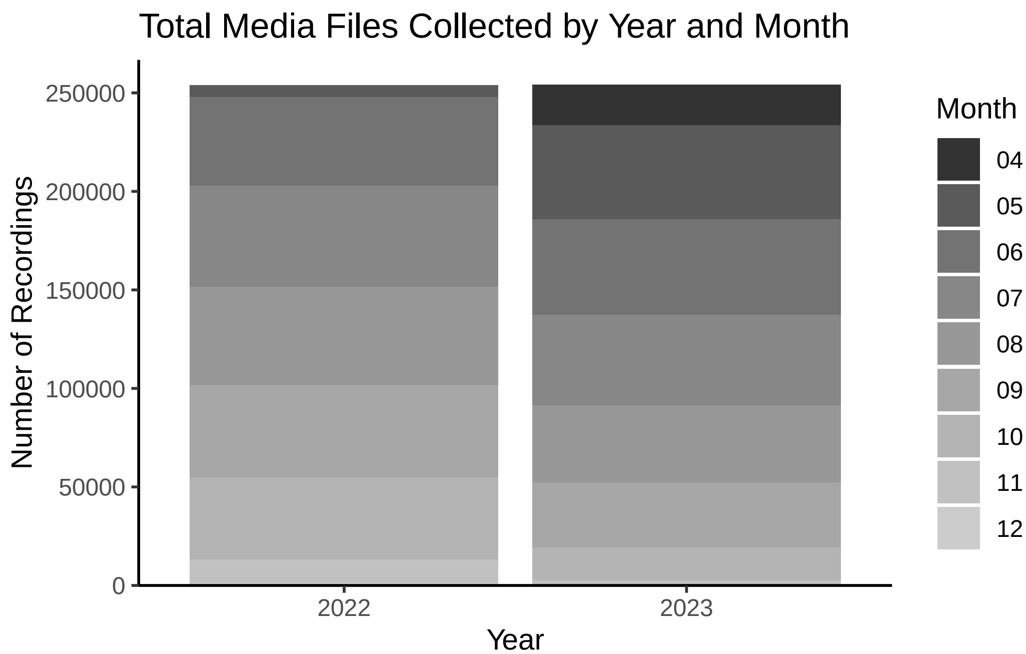

Figure 4: The number of recordings collected by month in Green Mountain National Forest in 2022 and 2023.

Site visits were tracked using ArcGIS Survey123. Survey123 allows users to make forms which can be used to easily record data in the field, then conveniently stores these data for future use. Visits to sites were recorded when monitoring units were set up ("set") and when they were removed at the end of the season ("pull"). In total, 294 visits were made to monitoring sites across both years.

All data collected for this project were stored in a database created using the R package AMMonitor. AMMonitor utilizes a SQLite database to store remote monitoring data and allow for easy analysis (Clarfeld *et al.* 2024).

Automated Species Detection

Long-term remote monitoring can produce extremely large sets of data which would be cumbersome and time consuming to manually review. Advances in machine learning have provided powerful models which can be used to detect species in these large datasets. Humans can then review these detections, rather than the entire dataset. We employed BirdNET, a convolutional neural network created by the Cornell Lab of Ornithology, to analyze the collected recordings. BirdNET is able to identify over 3,000 bird species from audio recordings, as well as a handful of other vocal taxa such as frogs (Kahl *et al.* 2021).

BirdNET can be quite effective at identifying bird vocalizations in recordings, but results are dependent on verification of detections and the confidence thresholds employed. A confidence threshold represents how certain the BirdNET model is that it has correctly identified a species in a recording. Previous studies evaluating BirdNET's performance have found that a threshold of 0.5 can be a useful starting point for verification, while a threshold of 0.7-0.8 may be more suitable for analysis in most studies. The appropriate threshold can vary significantly across species however, and verification of detections is a valuable tool in determining an appropriate value for a given study (Pérez-Granados 2023).

Verifications of BirdNET detections are currently underway. Without a robust set of verifications to evaluate BirdNET's performance for this project, we opted for a conservative confidence threshold of 0.9. Using a high confidence threshold increases the chances that detections are true positive results (a species is present and correctly identified), at the cost of potentially including false negatives as well (a species is present, but not identified by the model). As such, these results should be considered preliminary.

Overall, BirdNET produced 3,983,248 detections including 166 species across all of the available recordings. After filtering the data using a confidence threshold of 0.9, BirdNET produced 261,219 detections including 123 different species. For a full list of species detected by BirdNET at the 0.9 confidence threshold, see Appendix A.

GMNF is particularly interested in a subset of bird species closely associated with early succesional forests, which may benefit from their management actions. Additionally, the control sites in this project offered the opportunity to monitor for species of conservation concern which prefer large areas of mature forest. A list of these focal species can be found in Table 1.

Table 1: Focal species for this monitoring project in GMNF.

Common Name	Scientific Name
American Woodcock	Scolopax minor
Black-throated Blue Warbler	Setophaga caerulescens
Blue-winged Warbler	Vermivora cyanoptera
Brown Thrasher	Toxostoma rufum
Canada Warbler	Cardellina canadensis
Chestnut-sided Warbler	Setophaga pensylvanica
Eastern Towhee	Pipilo erythrophthalmus
Field Sparrow	Spizella pusilla
Golden-winged Warbler	Vermivora chrysoptera
Northern Goshawk	Accipiter atricapillus
Olive-sided Flycatcher	Contopus cooperi
Prairie Warbler	Setophaga discolor
Ruffed Grouse	Bonasa umbellus
Swainson's Thrush	Catharus ustulatus
Wood Thrush	Hylocichla mustelina

Due to the large amount of data collected through remote monitoring and analysis with BirdNET, results will be summarized primarily through these focal species. For more information on each species and why it was chosen for this project, see Appendix B.

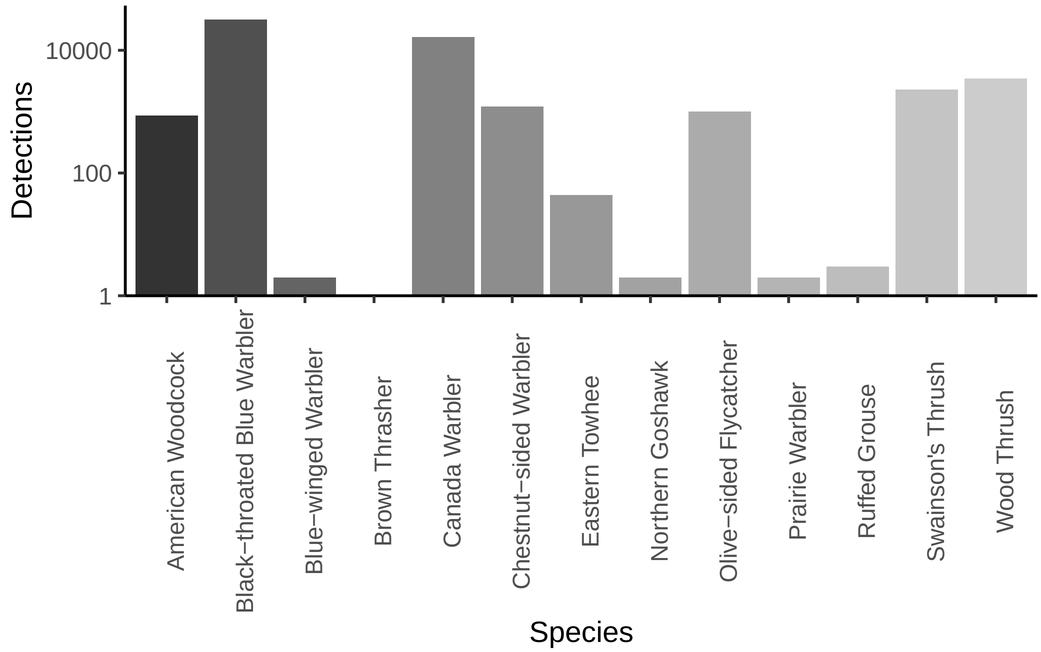

Figure 5: Total BirdNET detections of 13 focal species in Green Mountain National Forest using a confidence threshold of 0.9, scaled logarithmically. Golden-winged Warbler and Field Sparrow have been excluded because no individuals were detected at this confidence threshold.

As shown in Figure 5, BirdNET produced a large number of detections for some focal species at a high confidence threshold, such as Black-throated Blue Warbler and Canada Warbler. Other species were not detected at all at this threshold, such as Golden-winged Warbler and Field Sparrow, or there were very few detections.

Ruffed Grouse was one of these species with very few BirdNET detections. Previous studies have found that BirdNET struggles to accurately detect Ruffed Grouse drumming. As such, we also employed a model trained to specifically identify Ruffed Grouse drumming (Lapp *et al.* 2023).

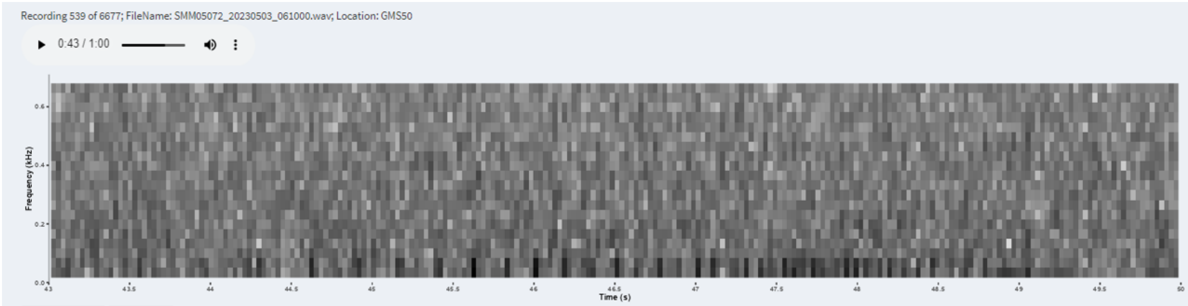

Figure 6: A spectrogram showing the sound of a Ruffed Grouse drumming in GMNF. Note the small, dark bars representing the sound near the bottom of the spectrogram.

The grouse model was run on all recordings collected before May 15th and earlier than 10:00 AM, the expected time for Ruffed Grouse to be drumming in Vermont. We detected Ruffed Grouse drumming using this model from mid-April through mid-May (Figure 7). This model was only run on data up to May 15th, thus it is very possible that grouse were still drumming after this date but not represented in this dataset. The distribution of detections is also rather irregular. This pattern could be a result of a smaller sample size or other unforeseen variables, such as weather, which could have impacted our ability to detect Ruffed Grouse drumming each day.

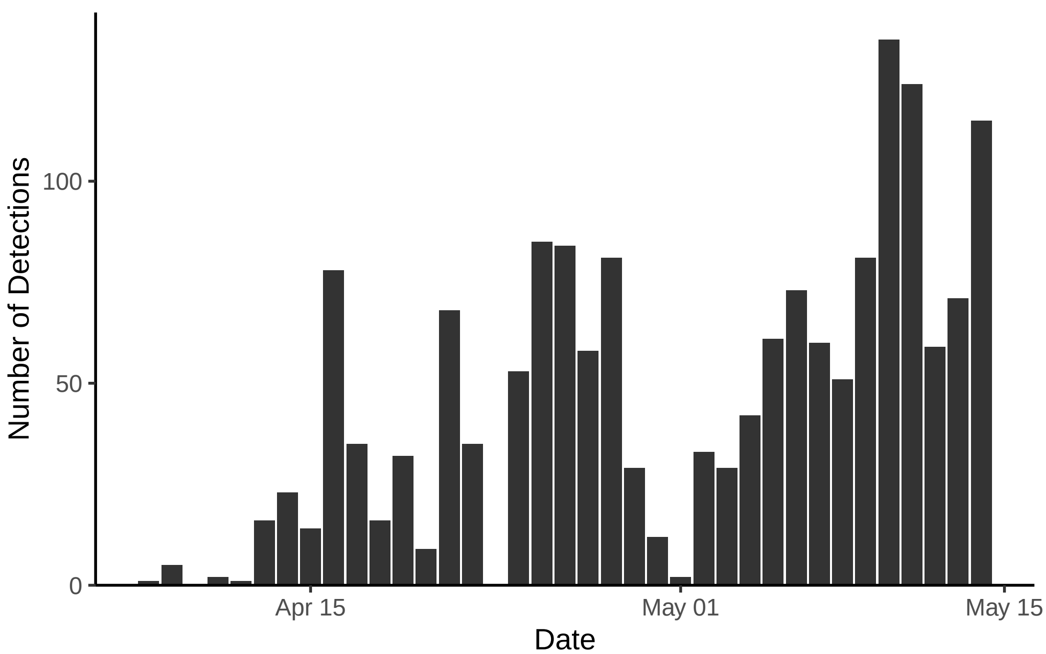

Figure 7: Number of detections of drumming Ruffed Grouse using the Ruffed Grouse model over time in Green Mountain National Forest.

The Ruffed Grouse model produced an additional 1,673 detections of Ruffed Grouse. Unlike BirdNET, this model does not produce a confidence value. We opted to filter data based on the total number of detections at a location instead. This approach is intended to both filter out false positives and retain locations Ruffed Grouse are likely using as breeding habitat. As such, we only included locations with at least 10 detections of Ruffed Grouse (Figure 8).

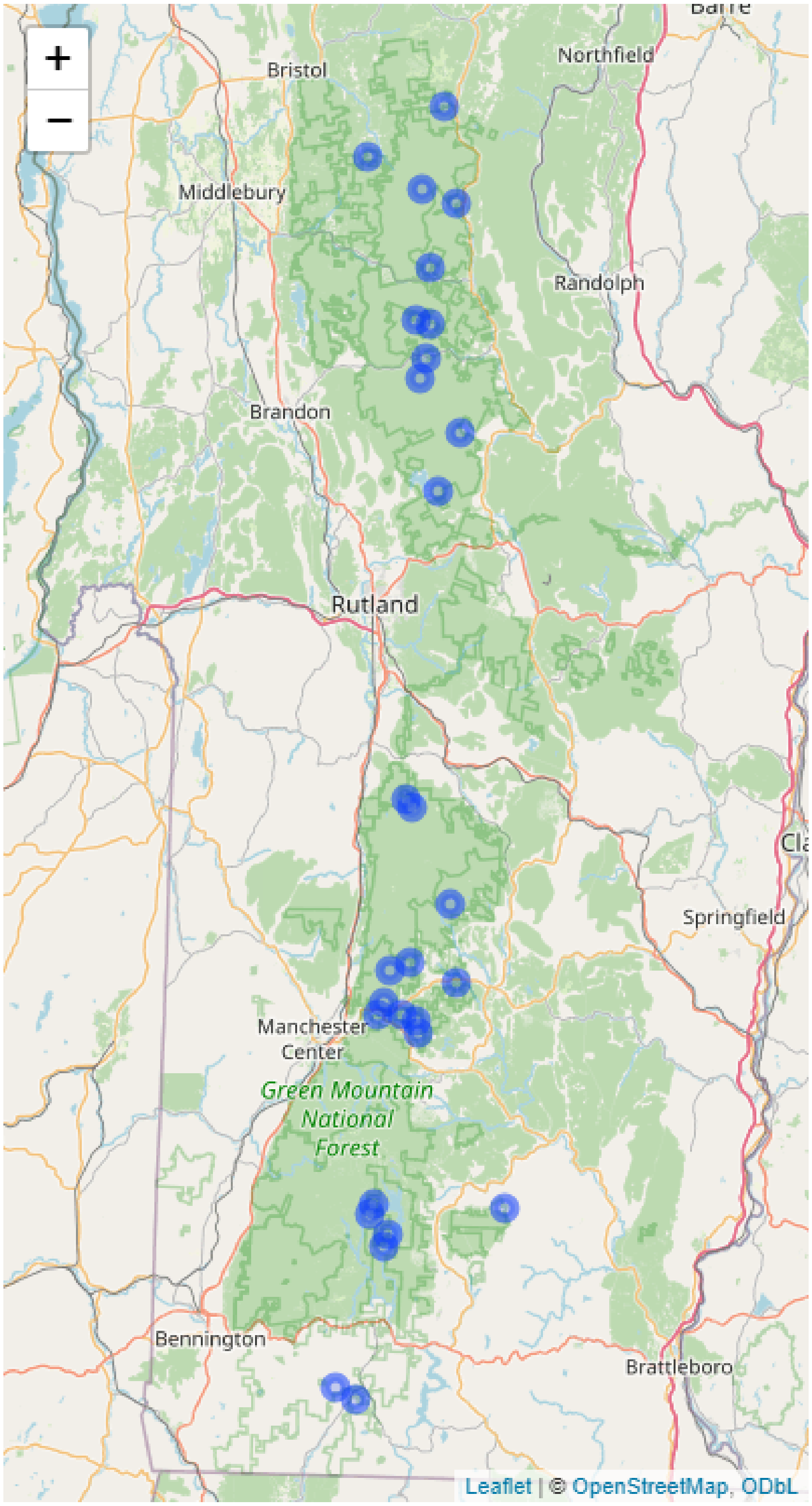

Figure 8: A map of all locations where Ruffed Grouse were detected at least 10 times.

Results and Outcomes of Objectives

1. Build automated models for site colonization, site extinction, and site persistence probabilities for priority wildlife species.

Population and occupancy models such as these can be valuable tools for wildlife managers because they allow them to account for imperfect detection of wildlife species and evaluate the impacts of their management actions on species of interest. Using the collected data on study sites and where species were detected, models can predict where species are likely to be present but undetected, and how they may respond to future management interventions.

We employed AMMonitor to more easily collect and organize our large amount of remote monitoring data and conduct these kinds of analyses. In this case, we would be looking to create multi-season occupancy models for priority species. We are well positioned to do so, once enough data has been verified to produce a robust presence/absence matrix for our focal species at study sites. Furthermore, the associated R package OccupancyTuts (Donovan *et al.* 2023) has been developed and provides a series of tutorials on occupancy modeling in R and using remote monitoring data to construct these models.

2. Develop a customized open-source spatial decision tool that enables GMNF to better match forest management outcomes to their management objectives.

We developed an R function utilizing integer programming and the R package lpSolve (Berkelaar *et al.* 2023) to help GMNF select sites to best achieve their forest management outcomes. Integer programming in this case requires a presence/absence table containing detection data for species at each included location. The user can then set an objective variable to be maximized or minimized, as well as other constraints. These goals and constraints can represent both ecological concerns, such as preserving a set of focal species, and logistical concerns, such as the cost of conserving a site or how accessible the site is for monitoring and management purposes.

Our R function `siteSelection()` uses location and species detection information from an AMMonitor database. Data for variables such as 'cost' or 'accessibility' for each location can be added to the 'locations' table of the database using the AMMonitor function `dbAddUserCol()`. To run `siteSelection()`, the user can set an objective and various constraints to meet their needs and subset the data by species, location, or date range if desired. The function returns information from the database for the portfolio of selected locations, presence/absence tables used in the analysis, and a map of the selected sites. If no solution exists for the given data, objective, and constraints, there will be no output object.

The minimum arguments to run `siteSelection()` are an open database connection and an objective object (whether to maximize or minimize a variable that exists as a column in the 'locations' table of the database). This function's many additional optional arguments can be adapted to increase complexity and match a variety of management objectives:

- **con**: An open database connection
- **objective**: A named vector of length 1. The name should be a column in the locations table (added with `dbAddUserCol()`), and the value should be 'min' or 'max' to specify whether the output portfolio should minimize or maximize this column's values
- **constraints**: Optional additional constraints. A named vector, with names specifying a column in the locations table (added with `dbAddUserCol()`), of minimum or maximum allowed value per constraint, with direction to be specified in the constraints_dir argument
- **constraints_dir**: Vector of direction of constraint caps in order of constraints. For example, if accessibility must be greater than 10, and cost must be at or below 20, enter c('>', "<=')
- **data_source**: 'annotations' or 'modeloutputs', where to get the data to populate the presence-absence table. Default is 'annotations'.
- **verified**: TRUE or FALSE, should the annotations or modeloutputs used as the data source be verified. Requires data in the verifications table.
- **includeNA**: TRUE or FALSE, should modeloutputs with NA in the value_num column be included in the analysis
- **modeloutput_threshold**: If data_source is 'modeloutputs', the threshold to count a model output as a detection.
- **included_locations**: A vector of locations to include in the analysis. Default includes all locations
- **min_survey_length**: Minimum number of days a site must be surveyed (calculated by media dates) to be included in the analysis. Default is 14 days
- **included_taxa**: A vector of taxa to include in the analysis. Default will include all taxa in the data_source table.
- **survey_date_range**: The range of dates (in YYYY-MM-DD format) as a vector, c('start', 'end') to subset media annotations for detection data
- **locations_per_species**: A vector in order of included_taxa- at how many of the selected locations must each species be present? Default is 1 for all included taxa. Output will be empty if all species are not present in at least this number of sites.
- **map_database**: Name of map database for plotting locations. Options are "world" or "usa" or "state" or "county"
- **regions**: Vector of regions within the map_database for plotting (e.g., c("vermont"))

The function can be used with either camera or audio data, or both. The 'data_source' argument specifies whether data should be used from 'annotations' (human taggers) or 'modeloutputs' (machine learning detection models). Users may specify a minimum survey length at each site ('min_survey_length', for which the default is 14 days of media at each site), and a 'modeloutput_threshold' for which to count each modeloutput as a detection or not.

The 'locations_per_species' argument allows users to specify how many sites within the locations portfolio must contain each of the 'included_taxa'. For example, users might use this argument in a case where the goal is to provide a portfolio of sites that minimizes cost but also contains at least 5 sites with Ruffed Grouse and 3 sites with American Woodcock.

A few examples of how this decision tool can be used in GMNF have been included under Objective 3. The code for the `siteSelection()` function can be found in Appendix D.

3. Customize currently existing AMMonitor approach and decision tool to smaller state-, town-, or privately-owned forests, with user-friendly interactions.

The AMMonitor approach was designed to be able to address ecological problems of various scales. The recent release of AMMonitor 2.0.0 includes a user-friendly graphical interface and a robust set of tutorials to make it easy to use, even for those with little to no experience with R programming (Clarfeld *et al.* 2024). These updates make the AMMonitor approach more accessible to those managing smaller forests and projects, while also offering the structure and capacity for much larger projects such as GMNF's. For more information on AMMonitor, see Appendix C.

The decision tool can similarly be used to solve problems of different scales. An analysis can be carried out with only a few sites and species, or with a large set of potential locations and species as seen in GMNF. Additionally, the user is able to specify their own constraints, meaning the tool can easily be adapted to address a wide variety of practical considerations. While there is not currently a graphical user interface for the decision tool, we do intend to include one with a later release of AMMonitor, or potentially as part of a larger complementary R package with a wide variety of decision tools. As such, we have included the R code for the current decision tool in Appendix D of this report.

This decision tool can also be used to evaluate a much smaller selection of sites and species of interest. To illustrate how the decision tool works, let's consider an example using the AMMonitor demo database. This database includes 3 sites and a small selection of species, making it a useful stand in for a much smaller scale conservation project.

In this example, the monitoring organization wants to conserve all of the species they detected. Additionally, they want the total accessibility of the selected sites to have a value of at least 4 to ensure the sites are easy to reach for continued monitoring. They want to achieve these goals for the lowest cost possible. By inputting these objectives and constraints into the site selection function, we get the following results, summarized for clarity:

```
Number of species and other constraints included: 9

Number of locations included: 3

Objective: Minimize COST

Total objective value: 3

Selected sites: locationA, locationB, locationC
```

In this scenario, all three monitoring sites were selected for conservation. If we look at the provided presence-absence table, as shown in Table 2, each site had at least one species which was only present at that site. Therefore, every site must be conserved to conserve all species; further reducing costs would require removing at least one species from consideration.

Table 2: A presence-absence table of species at all selected locations in the demo database. A 1 indicates presence and a 0 indicates absence. Sites were selected to minimize cost, preserve all species, and ensure accessibility was at least 4.

	Species							
Location	Deer	Bear	Raccoon	Coyote	Hare	Moose	BTNW	OVEN
locationA	1	1	1	1	0	0	1	1
locationB	1	1	0	0	1	0	0	0
locationC	1	0	0	0	0	1	0	0

Now, let's look at a few possible examples for GMNF. First, let's consider a situation in which GMNF wants to continue using acoustic monitoring and managing sites for all of the species detected during this project. They also want to minimize the cost of managing these sites.

In this case, we'll tell the function to minimize cost and include all species detected by BirdNET at a confidence threshold of 0.9 to match our earlier analysis. Additionally, we'll set the includeNA argument to true to include the outputs from the Ruffed Grouse model. Lastly, we do not currently have estimates of how costly it would be to manage each individual site. As such, the cost of every site has been set to 1. This means the decision tool will functionally try to include all detected species while choosing as few sites as possible.

```
Number of species and other constraints included: 123

Number of locations included: 60

Objective: Minimize COST

Total objective value: 18

Selected sites: GMN40c, GMN42b, GMN62, GMN65, GMN66, GMS21c, GMS46, GMS48,
GMS50, GMS54, GMS58, GMS59, GMS81, GMS84, GMS87, Madtom1, SouphouseNorth,
SwansMillEast
```

The decision tool identified a total of 18 sites to conserve all 123 species detected at a confidence threshold of 0.9. The selected sites are show in Figure 9.

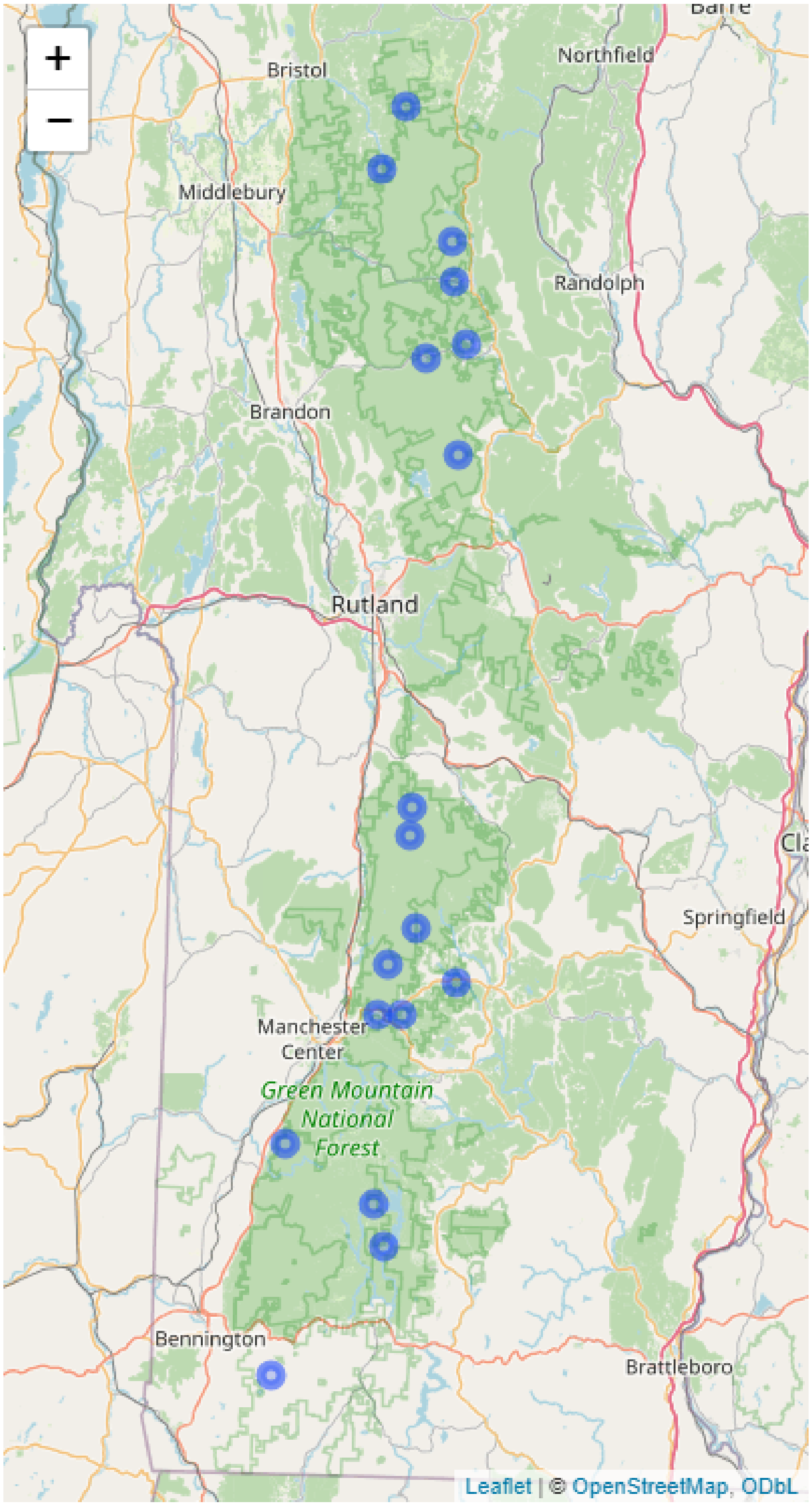

Figure 9: A map of sites selected by the decision tool when trying to conserve all species detected at a confidence threshold of 0.9 while minimizing cost.

Now, let's consider a more complicated management scenario. GMNF is concerned with the ease of accessing sites, and they have a budget for managing these sites. Instead of considering all species detected, they want to focus on sites where their focal species are present. They want the site selection function to provide a possible list of sites which they could easily manage for these early successional species while staying within their budget.

To accomplish this, we tell the function to maximize accessibility. We also provide a list of GMNF's focal species and set a constraint of cost <= 10 to represent their budget. The function is set to include NA values for the grouse model and otherwise use a confidence threshold of 0.9, as before.

Because we do not currently have a metric to measure accessibility at each site, a value of 2 was assigned to all sites. Golden-winged Warbler and Field Sparrow were excluded from this analysis because there were no high confidence detections of either species.

```
Number of species and other constraints included: 14

Number of locations included: 60

Objective: Maximize ACCESSIBILITY

Total objective value: 20

Selected sites: GMN62, GMS46, GMS87, Madtom1, Podunk, Pumphouse1, SouphouseNorth,
SouphouseSouth, SwansMillEast, SwansMillWest
```

In this scenario, the function chose 10 sites, matching the provided budget, with a total accessibility score of 20 which includes all detected focal species. These sites are shown in Figure 10.

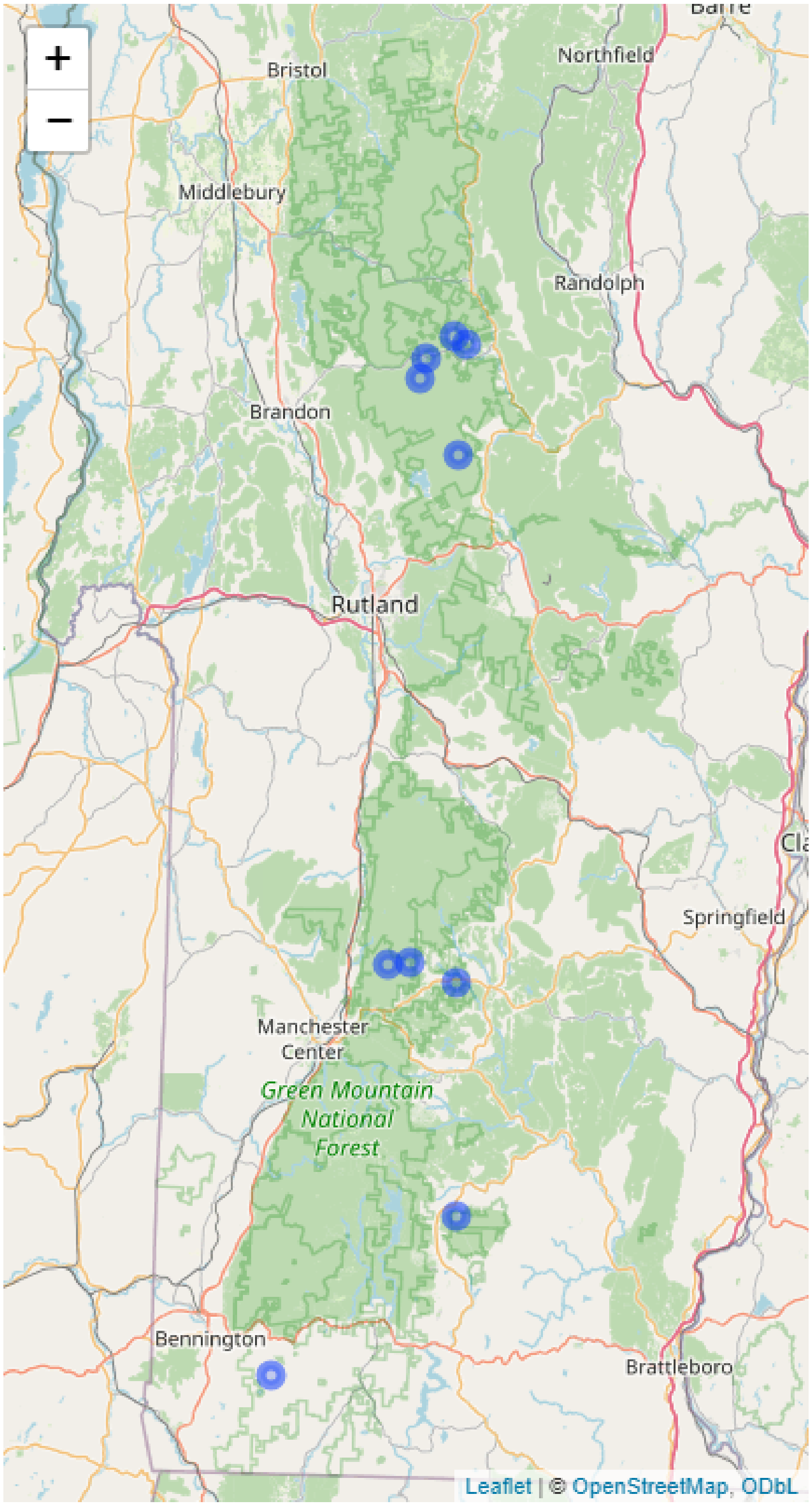

Figure 10: A map of sites selected by the decision tool when trying to conserve a set of focal species at accessible sites within a budget.

Next Steps and Future Work

The collected data provide a solid foundation for a variety of analyses, and we intend to move forward with more complex occupancy modeling and decision analysis. Once more BirdNET detections have been verified, we can use those verifications to identify more specific confidence thresholds suitable for this project. Furthermore, we can derive useful data about our sites and the surrounding area using GIS and spatial analysis. Combined, these data and the verified BirdNET outputs can be used to produce robust occupancy models for species of interest in GMNF. These models can then be used to estimate where species of interest likely occur and predict their responses to potential management actions in GMNF.

Furthermore, we intend to collect more meaningful data to define our objectives and constraints in the decision tool. For example, in this project, the "accessibility" constraint may be defined by the distance from existing roads in GMNF, as sites closer to roads would be easier to reach for the purpose of timber harvests. This decision tool will also be expanded upon into a larger R package focused on decision analysis and will work alongside the existing AMMonitor framework. Combined with the information provided by occupancy modeling, these tools can help GMNF make informed management decisions and efficiently achieve their goals.

These developments are being funded by an additional grant, and we are actively testing and refining our techniques. We expect to release further details and results in peer-reviewed publications in 2025 and 2026.

Data

Data will be released under the AMMonitor USGS ScienceBase community, and a subset of sites will be released under the Vermont Fish and Wildlife Department.

GMNF Project: https://www.sciencebase.gov/catalog/item/664cd939d34e6f297dfc2bd4

Vermont Fish and Wildlife Department Project: https://www.sciencebase.gov/catalog/item/6639428cd34e7789083967b0

AMMonitor ScienceBase Community: https://www.sciencebase.gov/catalog/item/6188c0c4d34ec04fc9c4f7a4

Acknowledgements

This project has been a collaboration between many partners, and we'd like to recognize The Ruffed Grouse Society, Green Mountain National Forest, the Vermont Fish and Wildlife Department, and the Vermont Cooperative Fish and Wildlife Research Unit for their contributions to this project. Of course, we'd also like to thank the Forest Ecosystem Monitoring Cooperative for their financial support.

Appendices

Appendix A: All Detected Species

Table 3, Table 4, and Table 5 contain a list of all species detected by BirdNET using a confidence threshold of 0.9, sorted by total number of detections. Detections are separated into frogs, mammals, and birds.

Table 3: A table of all frog species detected by BirdNET at a confidence threshold of 0.9, organized by total number of detections.

Species	Detections
Spring Peeper	112
Green Frog	14
Pickerel Frog	6
Wood Frog	4
Gray Treefrog	2

Table 4: A table of all mammal species detected by BirdNET at a confidence threshold of 0.9, organized by total number of detections.

Species	Detections
Eastern Chipmunk	192
Red Squirrel	48

Table 5: A table of all bird species detected by BirdNET at a confidence threshold of 0.9, sorted by number of detections.

Species	Detections
Winter Wren	35717
Hermit Thrush	34457
Black-throated Blue Warbler	31829
Black-throated Green Warbler	23550
Ovenbird	20168
Canada Warbler	16512
Blackburnian Warbler	9216
Blue-headed Vireo	8312
Mourning Warbler	7697
Red-breasted Nuthatch	5724
Brown Creeper	5345
Hairy Woodpecker	5205
Blue Jay	5028
Veery	4789
Scarlet Tanager	3703
Wood Thrush	3448
Barred Owl	3248
Cedar Waxwing	2565
Swainson's Thrush	2284
Black-and-white Warbler	2253
Magnolia Warbler	2095
Rose-breasted Grosbeak	1828
White-throated Sparrow	1666
Least Flycatcher	1542
Black-capped Chickadee	1413
Golden-crowned Kinglet	1383
Pileated Woodpecker	1321
Yellow-bellied Sapsucker	1317
Chestnut-sided Warbler	1212

Species	Detections
White-breasted Nuthatch	1108
Olive-sided Flycatcher	1002
Eastern Wood-Pewee	886
American Woodcock	862
Broad-winged Hawk	850
Indigo Bunting	824
Northern Waterthrush	674
Dark-eyed Junco	621
Downy Woodpecker	589
American Crow	581
Alder Flycatcher	576
Cerulean Warbler	526
Common Raven	483
Northern Saw-whet Owl	446
Northern Flicker	408
Red-eyed Vireo	399
Nashville Warbler	387
Eastern Whip-poor-will	347
Eastern Phoebe	315
Common Yellowthroat	299
Blackpoll Warbler	279
Wild Turkey	274
Belted Kingfisher	273
Black-billed Cuckoo	251
Ruby-crowned Kinglet	244
Yellow-bellied Flycatcher	181
American Redstart	173
Evening Grosbeak	171
Northern Parula	164

Species	Detections
American Robin	153
Merlin	139
Canada Goose	129
Bay-breasted Warbler	123
Yellow-rumped Warbler	115
Song Sparrow	98
Common Grackle	96
Swamp Sparrow	96
Sharp-shinned Hawk	82
Purple Finch	78
Common Loon	76
White-crowned Sparrow	66
Chimney Swift	65
Eastern Kingbird	58
Tufted Titmouse	52
Eastern Towhee	44
Great Crested Flycatcher	40
American Pipit	32
American Goldfinch	30
Pine Grosbeak	24
Mourning Dove	21
Eastern Bluebird	20
Pine Siskin	19
Tennessee Warbler	18
Yellow-throated Vireo	18
Tree Swallow	17
Common Nighthawk	10
Cooper's Hawk	10
Pine Warbler	9

Species	Detections
Gray Catbird	8
Solitary Sandpiper	8
Cape May Warbler	6
Chipping Sparrow	6
Mallard	6
Great Horned Owl	5
Ruby-throated Hummingbird	5
Common Redpoll	4
House Wren	3
Palm Warbler	3
Ruffed Grouse	3
Wood Duck	3
Barn Swallow	2
Blue-winged Warbler	2
Northern Goshawk	2
Prairie Warbler	2
Rusty Blackbird	2
White-winged Crossbill	2
American Bittern	1
American Kestrel	1
Blue-gray Gnatcatcher	1
Brown Thrasher	1
Fox Sparrow	1
Great Blue Heron	1
Northern Cardinal	1
Red-bellied Woodpecker	1
Snow Bunting	1
Snow Goose	1
Wilson's Snipe	1

Appendix B: Focal Species

Focal species were selected for this project based on the expert opinion of GMNF and the Vermont Fish and Wildlife Department. The ESH creation project was designed to benefit a wide variety of birds which rely on ESH during the breeding season, ranging from game birds such as Ruffed Grouse and American Woodcock to migratory songbirds such as Eastern Towhee (*Pipilo erythrophthalmus*), Brown Thrasher (*Toxostoma rufum*), and various species of warblers. Additionally, this project offered the opportunity to monitor for some species of conservation concern which prefer older forests at the control sites, such as Northern Goshawk (*Accipiter atricapillus*) and Wood Thrush (*Hylocichla mustelina*).

American Woodcock

American Woodcock are a unique species with complex habitat requirements. They favor young forests which provide a matrix of openings and cover to serve as display areas, feeding grounds, and nesting habitat (McAuley *et al.* 2020). Woodcock populations have declined in portions of their range, including the east and midwest, since the 1960s due to habitat loss, and the maturation of forests in the east could reduce the quantity and quality of woodcock habitat (Renfrew 2013; McAuley *et al.* 2020). Therefore, American Woodcock are listed a medium priority Species of Greatest Conservation Need, as continued management of forests to provide ESH is key to maintaining populations (Renfrew 2013).

In total, BirdNET produced 862 detections of American Woodcock across 19 sites at a confidence threshold of 0.9.

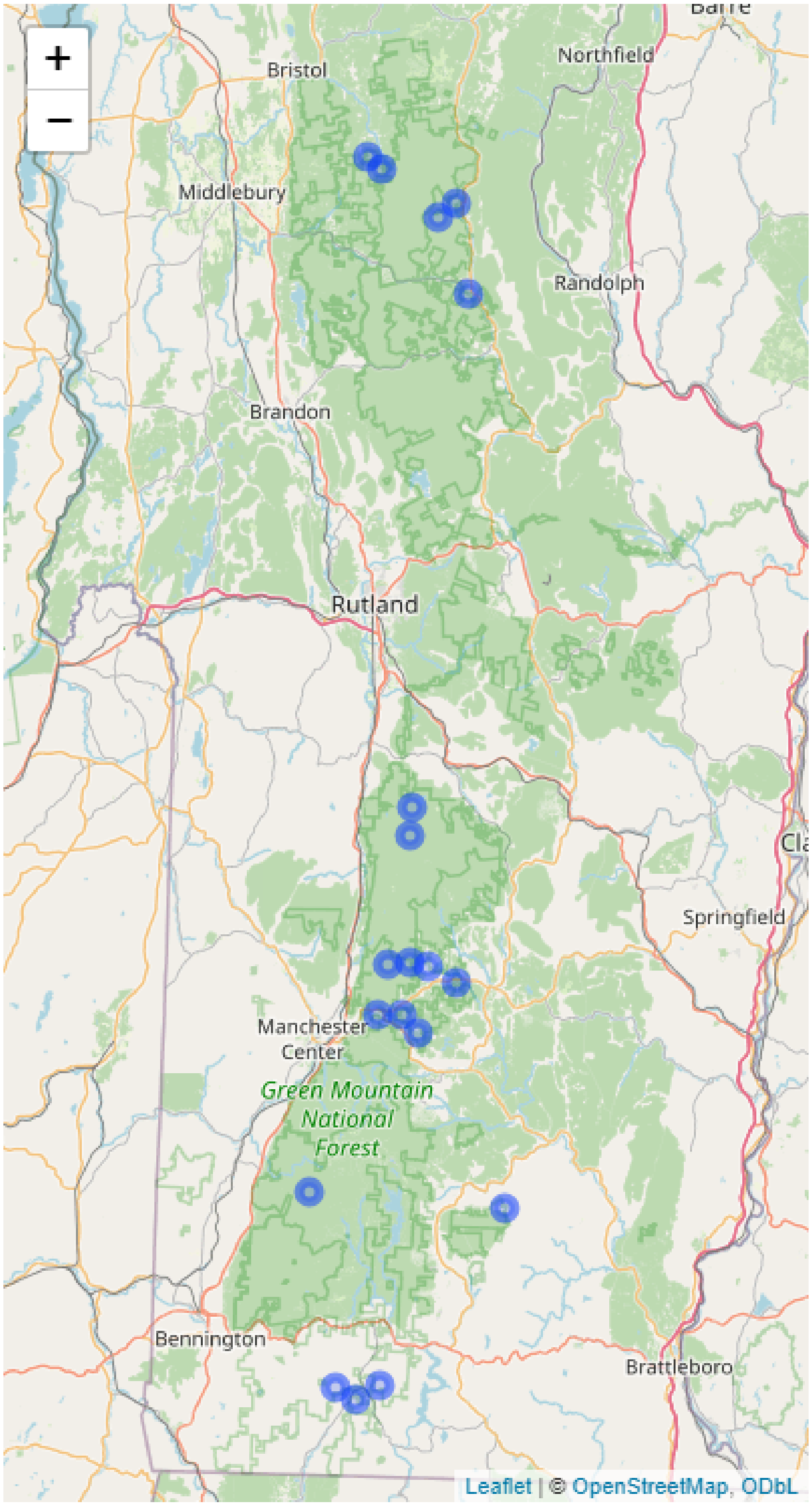

Figure 11: A map of sites where American Woodcocks were detected by BirdNET using a confidence threshold of 0.9.

Black-throated Blue Warbler

Black-throated Blue Warblers (*Setophaga caerulescens*) are a common resident of Vermont's mature forests. This species is well studied and easy to detect, making it an easy species to survey over long periods of time. Black-throated Blue Warbler populations have actually increased in Vermont, and the continued maturation of forests is expected to benefit them. Potential habitat loss on their wintering grounds could threaten Black-throated Blue Warbler populations. Large numbers of these birds breed in Vermont forests, so Vermont included them as a Species of Greatest Conservation Need as a responsibility species in hopes of detecting potential future population declines (Renfrew 2013). These factors make them an ideal candidate for long-term monitoring at our control sites within GMNF.

In total, BirdNET produced 31,829 detections of Black-throated Blue Warbler across 58 sites at a confidence threshold of 0.9.

Blue-winged Warbler

Blue-winged Warblers (*Vermivora cyanoptera*) have only bred in Vermont since the mid-1900s. The clearing of eastern forests allowed them to expand their range northward, and they have regularly bred in abandoned farmlands and early successional forests since (Renfrew 2013). Blue-winged Warblers will inhabit a variety of early to mid successional habitats, but prefer areas with substantial herbaceous and shrub cover and few large trees (Gill *et al.* 2020). Limited available habitat in Vermont supports a small population, and continued habitat management will likely be necessary for Blue-winged Warblers to persist in the state. Therefore, they are listed as a Species of Greatest Conservation Need in Vermont (Renfrew 2013).

Additionally, Blue-winged Warblers are difficult to monitor using remote acoustic monitoring, as they regularly hybridize with Golden-winged Warblers, and Blue-winged Warblers will occasionally sing a typical Golden-winged Warbler song (Gill *et al.* 2020). As such, we have to assume that a Blue-winged Warbler detection represents an actual Blue-winged Warbler, and not a hybrid or Golden-winged Warbler.

In total, BirdNET produced 2 detections of Blue-winged Warbler at 1 site, SouphouseNorth, at a confidence threshold of 0.9.

Brown Thrasher

Brown Thrashers (*Toxostoma rufum*) are very vocal birds found in dense shrublands. They prefer shrub-dominated habitats and are less likely to be found in other types of ESH dominated by saplings. Additionally, Brown Thrashers are territorial and therefore generally found in low densities. Reforestation in Vermont has driven population declines, particularly in mountainous regions of the state, and these declines were mirrored in other northern parts of their range. Vermont considers them as a Species of Greatest Conservation Need, and management interventions to create and maintain shrubland are expected to benefit the species (Renfrew 2013).

In total, BirdNET produced 1 detection of Brown Thrasher at 1 site, GMS46, at a confidence threshold of 0.9.

Canada Warbler

Canada Warblers (*Cardellina canadensis*) are vibrant songbirds with a tendency to conceal themselves in the thick undergrowth of their preferred habitats. Unlike many of the ESH dependent species this project focuses on, Canada Warblers have more specific habitat preferences. They tend to breed in forests with a dense understory and structural complexity, including a variety of wetlands, regenerating timber cuts, and naturally occurring gaps in blocks of older forest. Canada Warblers have declined throughout their range in recent years and are considered a Species of Greatest Conservation Need, but the cause of their decline is less clear. These declines do not cleanly correlate with the maturation of Vermont's forests, and various habitat types seem to serve as effective breeding grounds (Renfrew 2013). Continued monitoring of this species may be crucial to understanding population declines and therefore implementing effective management strategies.

In total, BirdNET produced 16,512 detections of Canada Warbler across 27 sites at a confidence threshold of 0.9.

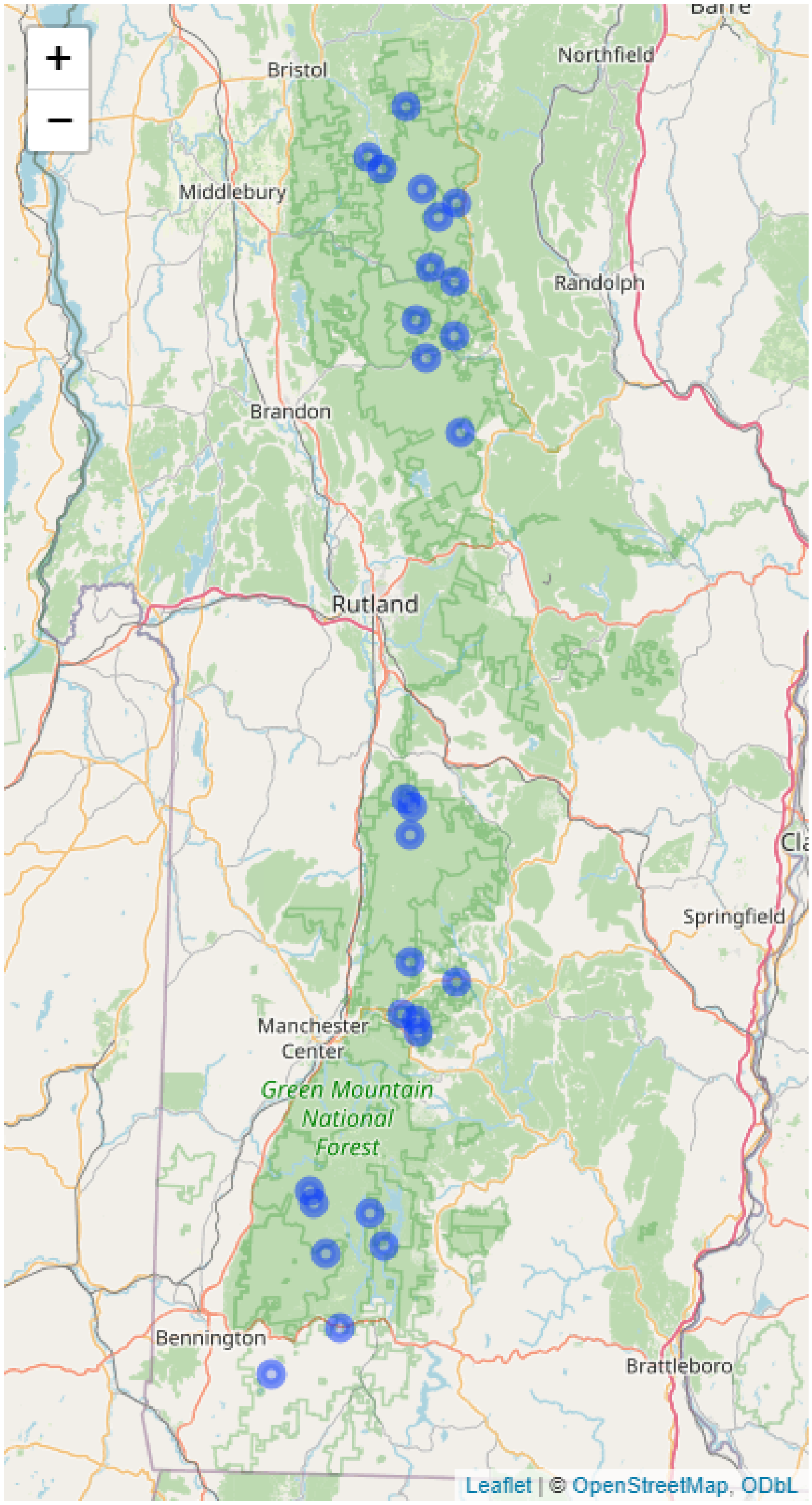

Figure 12: A map of sites where Canada Warblers were detected by BirdNET using a confidence threshold of 0.9.

Chestnut-sided Warbler

Chestnut-sided Warblers (*Setophaga pensylvanica*) were once an uncommon sight in Vermont, but they were quick to colonize the early successional habitats creating by clearing northeastern forests and the abandonment of farms. Today they are common throughout Vermont, inhabiting early successional forests, shrublands, and edges created by human activities. Despite their success, Chestnut-sided Warblers are listed as a medium priority Species of Greatest Conservation Need in Vermont due to concerns about forest maturation. This species quickly responds to changes in the landscape and readily colonizes new forest gaps, such as those created by timber harvests, making it an ideal species for this monitoring project (Renfrew 2013).

In total, BirdNET produced 1,212 detections of Chestnut-sided Warbler across 18 sites at a confidence threshold of 0.9.

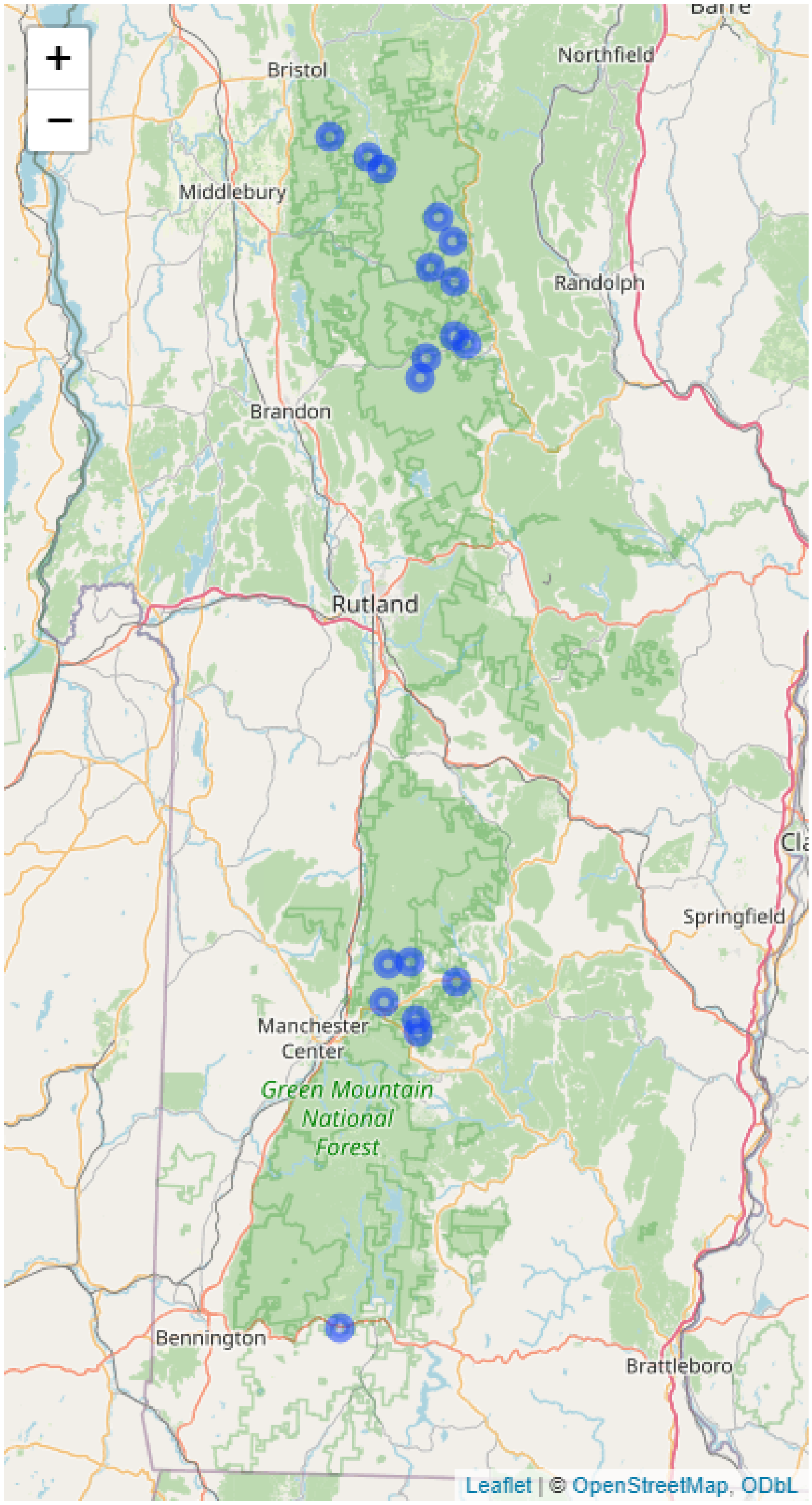

Figure 13: A map of sites where Chestnut-sided Warblers were detected by BirdNET using a confidence threshold of 0.9.

Eastern Towhee

Eastern Towhees (*Pipilo erythrothalamus*) are in a unique and tenuous position in Vermont. These large sparrows favor habitat with a developed and maintained shrub layer; they are much less common in ESH dominated by saplings instead. Eastern Towhees, like many early successional species, benefited from the clearing of Vermont forests and subsequent abandonment of farms. As forests have matured, they suffered significant habitat loss and declined drastically, making them one of the most rapidly declining birds in Vermont starting in the 1980s. These declines have been the most dramatic in the northeastern portions of their range, leading Vermont to designate them a Species of Greatest Conservation Need. Generally, Eastern Towhees respond favorably to habitat management which promotes or maintains a shrub layer, making them a valuable species for this monitoring project (Renfrew 2013).

In total, BirdNET produced 44 detections of Eastern Towhee across 14 sites at a confidence threshold of 0.9.

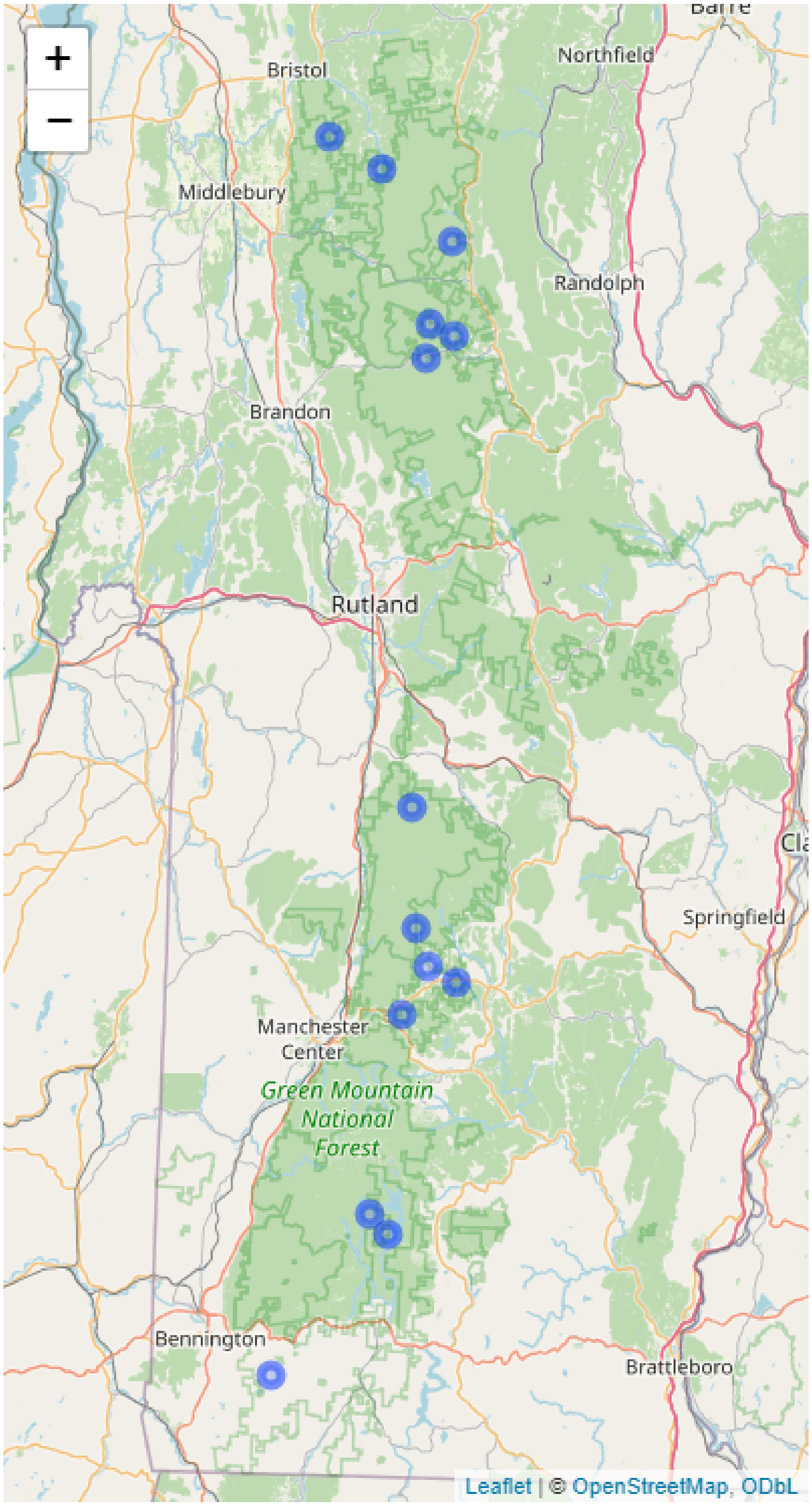

Figure 14: A map of sites where Eastern Towhees were detected by BirdNET using a confidence threshold of 0.9.

Field Sparrow

Like Eastern Towhees, Field Sparrows (*Spizella pusilla*) favor shrublands, though they may inhabit a greater variety of early successional habitats. The clearing of forests made Field Sparrows a common species in Vermont, and their populations have predictably declined as forests have reclaimed the landscape and matured. These long term declines led Vermont to list them as a Species of Greatest Conservation Need, and available habitat is a major limiting factor on Field Sparrow populations in the state. This species readily colonizes newly available habitat, making it a strong candidate for this monitoring project (Renfrew 2013).

BirdNET did not produce any detections of Field Sparrows at a confidence threshold of 0.9.

Golden-winged Warbler

In many ways, Golden-winged Warblers (*Vermivora chrysoptera*) are similar to the closely related Blue-winged Warbler. They similarly expanded their range into Vermont as forest cover decreased and farms were abandoned (Renfrew 2013). Golden-winged Warblers, however, have seen their populations decline significantly as forests have regenerated. They seem to favor earlier successional habitat than Blue-winged Warblers, with very limited numbers of large trees and a high density of shrubs and herbaceous vegetation. These conditions are often created by clearcuts and similar forest management practices. Hybridization has also likely contributed to declines. Golden-winged Warblers are considered a Species of Greatest Conservation Need in Vermont and listed as Near Threatened (Renfrew 2013; Confer *et al.* 2020).

Golden-winged Warblers introduce the same monitoring problems as Blue-winged Warblers; Golden-winged Warblers may sing typical Blue-winged Warbler songs and hybrids are always a possibility (Confer *et al.* 2020). As such, we must assume that a Golden-winged Warbler song indicates the presence of a Golden-winged Warbler without additional information.

BirdNET did not produce any detections of Golden-winged Warbler at a confidence threshold of 0.9.

Northern Goshawk

Northern Goshawks (*Accipiter atricapillus*) are a sparsely distributed and sensitive species of hawk in Vermont. They nest in various types of mature forests and are sensitive to human disturbance, including large timber harvests. Additionally, they prey primarily on species which rely on ESH such as Ruffed Grouse and Snowshow Hare. Goshawks seem to prefer nesting locations near small forest openings, including natural gaps and woodland trails. Human disturbances, habitat fragmentation, and declining prey populations have driven population declines in the northeast, and Vermont lists them as a Species of Greatest Conservation Need (Renfrew 2013; Squires & Reynolds 2023). This project may offer a unique opportunity to monitor for Northern Goshawks, as control sites are set in the mature forests they use for nesting, while treatment sites may benefit their favored prey, so long as these sites are far enough away from active nests.

Northern Goshawk was recently split in American Goshawk and Eurasian Goshawk. BirdNET still referred to them as “Northern Goshawk” at the time of our analysis, so we have opted to use that common name, but updated the scientific name to reflect the current scientific name of American Goshawk.

In total, BirdNET produced 2 detections of Northern Goshawk across 2 sites, GMS24 and GMN62, at a confidence threshold of 0.9.

Olive-sided Flycatcher

Olive-sided Flycatchers (*Contopus cooperi*) are a patchily distributed and imperiled species in Vermont. They primarily inhabit gaps in coniferous forests, including wetlands and gaps created by natural disturbances or timber harvests. The presence of remaining trees or snags is essential, as they use these perches for feeding. In Vermont, they are most common in the far northeastern portion of the state, but also inhabit the Green Mountains (Renfrew 2013). Olive-sided Flycatchers have experienced significant population declines throughout their range; they are listed as Near Threatened and considered a Species of Greatest Conservation Need in Vermont (Renfrew 2013; Altman & Sallabanks 2020). The cause of population declines remains unclear. Consistent declines throughout their breeding range suggest the cause may lie in their nonbreeding habitat in South and Central America. Alternatively, some studies suggest that timber harvests actually create ecological traps for Olive-sided Flycatchers and fuel population declines, rather than providing quality habitat (Altman & Sallabanks 2020). As a result, continued monitoring of this species may be crucial to conservation efforts.

In total, BirdNET produced 1,002 detections of Olive-sided Flycatcher across 5 sites at a confidence threshold of 0.9. These detections occurred consistently from early June through early July (Figure 15). Olive-sided Flycatchers pair and nest in June in Vermont, suggesting that at least some of these sites were likely used as breeding habitat (Altman & Sallabanks 2020).

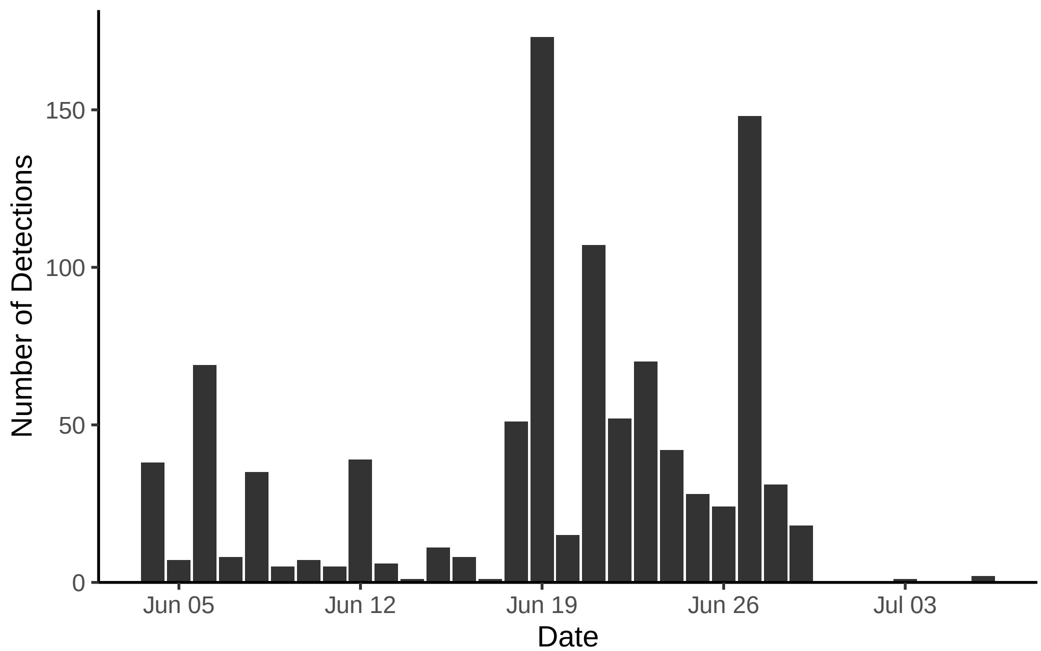

Figure 15: The number of Detections of Olive-sided Flycatchers over time in Green Mountain National Forest.

Prairie Warbler

Prairie Warblers (*Setophaga discolor*) are another relatively recent addition to Vermont's bird communities, with their first confirmed breeding record in 1971. They favor early successional habitat with more scattered shrubs, unlike many other early successional species which favor dense shrub cover. This preference for open shrubland may make Prairie Warblers even more sensitive to forest maturation than our other focal species, and populations have declined significantly in much of its range, particularly the northeastern US. It is considered a Species of Greatest Conservation Need, and management interventions which create these open, early successional habitats can successfully increase local populations (Renfrew 2013).

In total, BirdNET produced 2 detections of Prairie Warbler at 1 site, SouphouseNorth, at a confidence threshold of 0.9.

Ruffed Grouse

Ruffed Grouse are closely associated with early successional forests, especially those dominated by aspens (*Populus* sp.). Forest management practices in the east, including fire control and opposition to clearcutting, have resulted in the maturation of forests and therefore less suitable habitat for Ruffed Grouse (Rusch *et al.* 2020). This habitat loss has fueled a significant population decline in the eastern US. Restoration strategies which create ESH can be effective at restoring and maintaining grouse populations, especially in large block of forest (Porter & Jarzyna 2013; Rusch *et al.* 2020). Ruffed Grouse are considered a medium priority Species of Greatest Conservation Need in Vermont, and guidelines focus on creating and maintaining high quality habitat to sustain grouse populations (Renfrew 2013).

For details on Ruffed Grouse detections in this project, see the main Methods and Data Collection section of this report.

Swainson's Thrush

Swainson's Thrushes (*Catharus ustalatus*) inhabit specific habitats in Vermont driven by a variety of factors. They primarily breed in the boreal forests of the far northeastern corner of the state and higher elevation forests in the Green Mountains, near the transition from hardwood to coniferous forests (Renfrew 2013). Furthermore, they seem to prefer mixed-age forests and may utilize early successional habitat and mature forests with adequate shrub cover (Renfrew 2013; Mack & Yong 2020). These habitats generally fall between the high elevation forests inhabited by Bicknell's Thrush and the lower elevation forests dominated by Hermit Thrush. Swainson's Thrush populations have declined in Vermont, but nearby areas such as Ontario have actually seen populations increase (Renfrew 2013). Continued monitoring of this species could help managers and conservationists better understand this species and respond to potential declines, especially in the face of developing threats such as climate change, which may impact their higher elevation habitats.

In total, BirdNET produced 2,284 detections of Swainson's Thrush across 35 sites at a confidence threshold of 0.9.

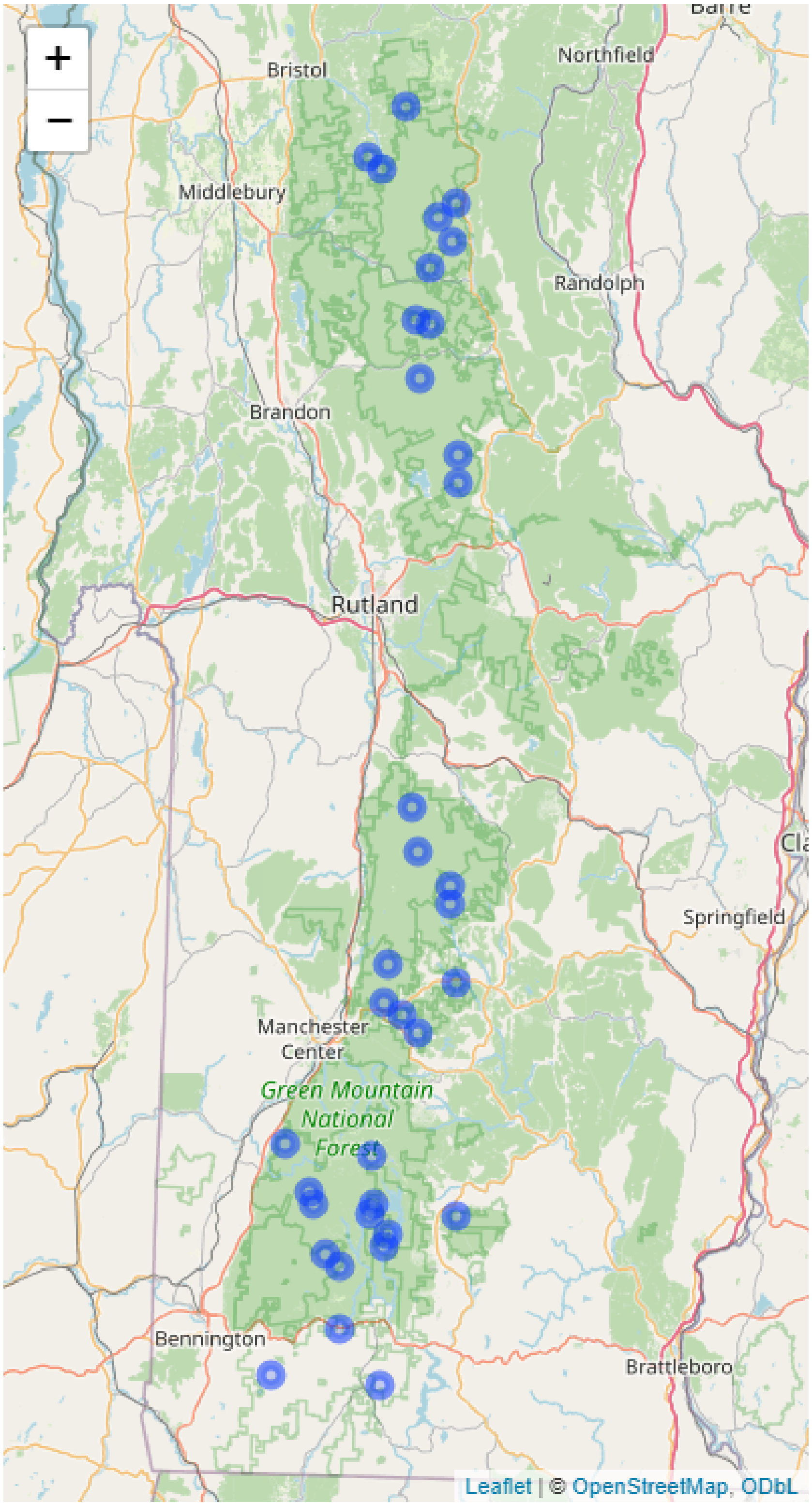

Figure 16: A map of sites where Swainson's Thrushes were detected by BirdNET using a confidence threshold of 0.9.

Wood Thrush

Wood Thrushes (*Hylocichla mustelina*) are a common resident of deciduous forests with a somewhat dense understory and rich layer of leaf litter. The regrowth and maturation of forests in Vermont has provided ample habitat for Wood Thrushes, but populations have still declined significantly in Vermont and across its range (Renfrew 2013). A variety of causes for these declines have been proposed. Wood Thrushes do seem to be area sensitive, preferring large patches of mature forest with few edges significantly altered by human activity, such as paved roads. More gradual forest edges may be beneficial however, as these areas may provide abundant food resources. Pesticides and acid rain may also impact Wood Thrush populations due to their impacts on the calcium-rich invertebrates they feed on during the breeding season (Evans *et al.* 2020). They are considered a Species of Greatest Conservation Need in Vermont and recognized as a species of concern by many national and international organizations as well (Renfrew 2013; Evans *et al.* 2020).

In total, BirdNET produced 3,448 detections of Wood Thrush across 45 sites at a confidence threshold of 0.9.

Appendix C: AMMonitor

AMMonitor, short for "Monitoring for Adaptive Management," is an R package designed to simplify and standardize remote monitoring data collection, allow for rapid analysis, and inform management decisions. It provides a robust and flexible structure which is standardized, allowing it to be easily interpreted and incorporated into other projects, and adaptable, allowing users to shape an AMMonitor project to suit their specific needs. All AMMonitor projects utilize the same basic workflow, as shown below:

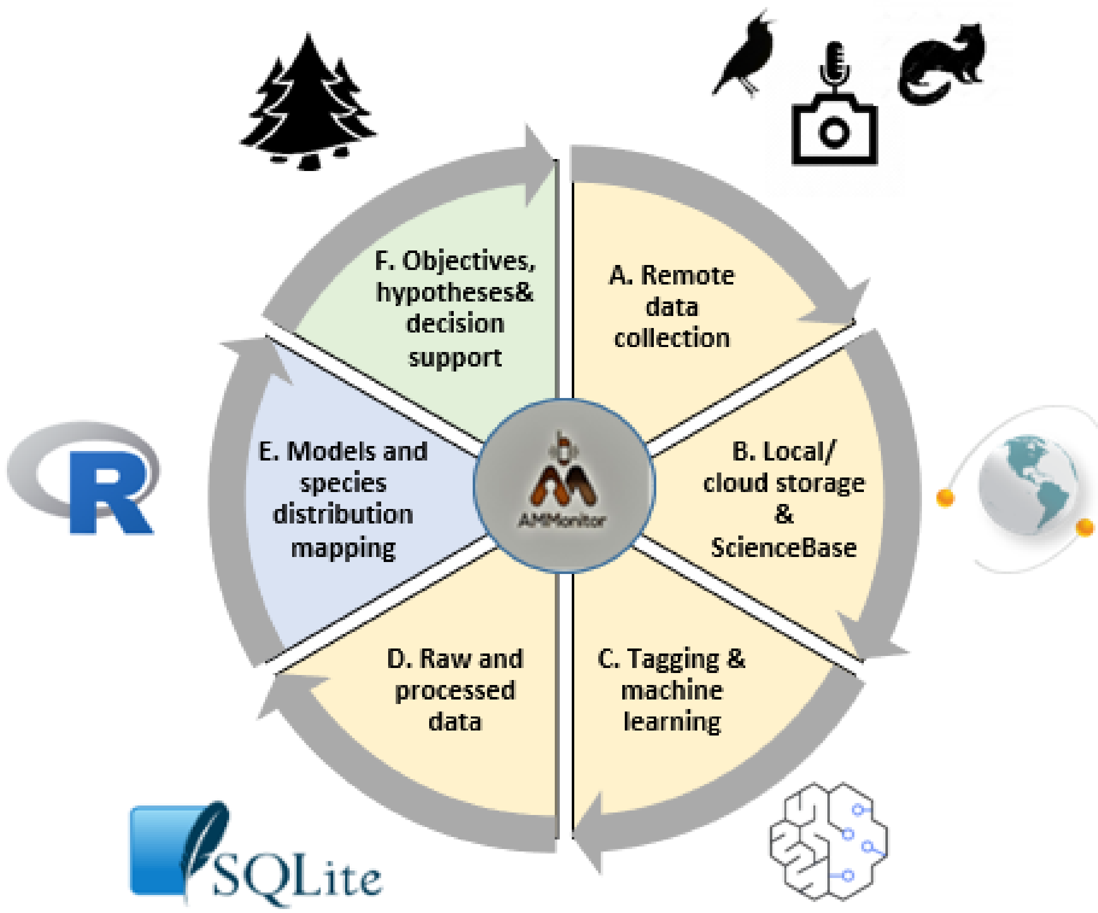

Figure 17: A visual representation of the AMMonitor workflow.

An AMMonitor project uses the following workflow. Note that this workflow is cyclical; each stage of a project informs the next.

A. Data are collected by autonomous monitoring units, such as audio recorders or trail cameras. The type of media collected generally depends on what wildlife taxa a project is interested in monitoring.

B. Collected media are uploaded into some form of storage, either on a local hard drive or in a cloud storage system. If desired, data can be permanently archived in the USGS ScienceBase repository.

C. Media files are reviewed by humans to identify taxa of interest and/or analyzed by machine learning models, such as BirdNET or the grouse model used in this project.

D. Data are stored in the project's SQLite database. This provides a consistent structure for data and metadata, while also allowing the user to add and store additional information relevant to their project.

E. Analyses, such as species occupancy and distribution models, are run using R functions and scripts. AMMonitor not only allows for easy, rapid analysis; it stores metadata to make analyses reproducible and easy to update as new data are gathered.

F. Results can serve as a basis for various decision tools to inform management decisions.

This workflow allows users to continually monitor species and conduct analyses, providing a robust framework for adaptive management and decision making. Additionally, these projects can be easily scaled to suit the needs of the user, from small local projects to large scale, long-term monitoring efforts.

For more information on AMMonitor, or to download the package, visit its USGS GitLab page: https://code.usgs.gov/vtcfwru/ammonitor

Appendix D: siteSelection.R

This appendix contains the code for the site selection function used in this report. The code can be copied into an R script and sourced for use with any AMMonitor database. Additionally, the function contains a help file with details of how each function argument works and useful examples.

```
#' @name siteSelection
#' @title Site selection (decision analysis) for conservation of species in
#' an
#'  adaptive monitoring program
#' @description A versatile decision tool using integer programming to help
#' inform site selection for conservation of species. The function requires
#' two arguments: an open connection to an AMMonitor database, and an objective
#' argument pointing to a column in the 'locations' table that should be
#' minimized or maximized. The default data_source is the 'annotations' table
#' of the database. Users may otherwise select 'modeloutputs' and may specify
#' if the data from either source should be verified or not with the 'verified'
#' argument. Optional additional constraints and other arguments
#' may be implemented for various management objectives.
#'
#' The function returns a list of objects, including the presence-absence data
#' used in the analysis, the portfolio of selected sites that meet objectives
#' and
#' constraints, the total objective value of selected sites, and (if the
#' 'map_database' and 'regions' arguments are given) a map of the selected
```

```
#' sites.
#'
#' See details for more information.
#'
#' @param con An open database connection
#' @param objective A named vector of length 1. The name should be a column in
#' the
#' locations table (added with `dbAddUserCol()`), and the value should be 'min'
#' or 'max' to specify whether the output portfolio should minimize or
#' maximize this
#' column's values
#' @param constraints Optional additional constraints. A named vector, with
#' names specifying a column in the locations table (added with
#' `dbAddUserCol()`),
#' of minimum or maximum allowed value per constraint, with direction
#' to be specified in the constraints_dir argument
#' @param constraints_dir Vector of direction of constraint caps in order
#' of constraints.
#' For example, if accessibility must be greater than 10, and cost must be at or
#' below 20, enter c('>', "<=')
#' @param data_source 'annotations' or 'modeloutputs', where to get the
#' data to populate the presence-absence table. Default is 'annotations'.
#' @param verified TRUE or FALSE, should the annotations or modeloutputs
#' used as the data source be verified. Requires data in the verifications
#' table.
#' @param includeNA TRUE or FALSE, should modeloutputs with NA in the value_num
#' column be included in the analysis
#' @param modeloutput_threshold if data_source is 'modeloutputs', the
#' threshold to count a model output as a
#' detection.
#' @param included_locations A vector of locations to include in the analysis.
#' Default includes all locations
#' @param min_survey_length Minimum number of days a site must be surveyed
#' (calculated by media dates) to be included in the analysis. Default is
#' 14 days
#' @param included_taxa A vector of taxa to include in the analysis. Default
#'  will include all taxa in the data_source table.
#' @param survey_date_range The range of dates (in YYYY-MM-DD format) as a
#' vector,
#'  c('start', 'end') to subset media for detection data
#' @param locations_per_species A vector in order of included taxa- at how
#' many of the selected locations must
#' each species be present? Default is 1 for all included taxa. Output will be
```

```
#' empty if all species are not present in at least this number of sites.
#' @param map_database  Name of map database for plotting locations. Options
#' are
#'  "world" or "usa" or "state" or "county"
#' @param regions  Vector of regions within the map_database for plotting
#' (e.g., c("vermont"))
#'
#' @details There are many optional arguments incuded in this function,
#' which gives users the ability to select for sites meeting a wide variety of
#' constraints.
#'
#' The objective name and constraint names must be columns in the
#' 'locations' table of the database, which can be added with the AMMonitor
#' function `dbAddUserCol()`. These columns might store information for
#' variables such as
#' cost per site, accessibility of the site in time or distance, etc. If
#' the 'constraints' argument is used, the 'constraints_dir' argument
#' must also be
#' implemented to specify the direction of the constraint (if the sites must be
#' above or below each constraint value). See examples below for formatting of
#' these arguments.
#'
#' The
#' objective, species presence-absence data, and additional constraints all
#' are used for the integer programming analysis. Users may specify a minimum
#' survey length at each site ('min_survey_length', for which the default is
#' 14 days of media at each site), and a 'modeloutput_threshold' for which to
#' count each modeloutput as a detection or not. The argument
#' 'locations_per_species' may be used to indicate how many locations each
#' of the 'included_species' (if given) must be present at within the
#' output portfolio.
#'
#' If not enough data is
#' available to meet all constraints, no sites will be selected.
#'
#'
#' @return A list of results objects, including a portfolio of selected
#' locations
#' @export
#' @importFrom DBI dbIsValid dbReadTable dbGetQuery
#' @importFrom lpSolve lp
#' @importFrom AMMonitor qryEffort
#' @importFrom maps map
```

```
#' @import ggplot2
#'
#' @examples
#'
#' # load AMMonitor
#' library(AMMonitor)
#'
#' # create a mini demo database
#' fp <- ammCreateMiniDemo(tempdir())
#'
#' # connect to the database
#' conx <- dbSetCon(paste0(fp, '/database/demo.sqlite'))
#'
#' # add 'cost' and 'accessibility' to the locations table
#' new_column <-  data.frame(
#' pk_tablename = "locations",
#' pk_fieldname = "cost",
#' core = 0,
#' var_type = "REAL",
#' not_null_clause = NA_character_,
#' default_value = NA_character_,
#' max = NA,
#' min = NA,
#' fk_listid = NA_character_,
#' description = "Cost of maintaining each location")

#' # add the new column to the locations table
#' dbAddUserCol(
#'  con = conx,
#'  col_info = new_column,
#'  disconnect = FALSE)
#'
#' new_column <-  data.frame(
#' pk_tablename = "locations",
#' pk_fieldname = "accessibility",
#' core = 0,
#' var_type = "REAL",
#' not_null_clause = NA_character_,
#' default_value = NA_character_,
#' max = NA,
#' min = NA,
#' fk_listid = NA_character_,
#' description = "Ease of access per location")
```

```
#' # add the new column to the locations table
#' dbAddUserCol(
#'  con = conx,
#'  col_info = new_column,
#'  disconnect = FALSE)
#'
#'  # add example values for cost and accessbility
#'
#'  stmnt <- "UPDATE locations SET cost = 1;"
#'  DBI::dbExecute(conx, stmnt)
#'
#'  stmnt <- "UPDATE locations SET accessibility = 2;"
#'  DBI::dbExecute(conx, stmnt)
#'
#' # EXAMPLE 1: Minimize cost while keeping accessibility over 4, all species
#'
#' # set objective
#' objective <- 'min'
#' names(objective) <- 'cost'
#'
#' # set additional constraint, accessibility should have minimum value of 4
#' constraint <- 4
#' names(constraint) <- 'accessibility'
#'
#' # set direction of constraint
#' const_dir <- '>'
#'
#' # run siteSelection
#' results <- siteSelection(con = conx,
#' objective = objective,
#' constraints = constraint,
#' constraints_dir = const_dir,
#' data_source = 'annotations',
#' verified = FALSE,
#' min_survey_length = 1,
#' map_database = 'state',
#' regions = 'vermont')
#'
#' # View selected sites
#' results$selected_sites
#'
#' # View the presence-absence table for the whole analysis
```

```
#' View(results$pa_table_full)
#'
#' # View the presence-absence table for the selected sites (here all three)
#' View(results$pa_table_selected)
#'
#' # View the total cost (objective value)
#'  results$total_objective_value
#'
#' # View the cost per site
#' View(results$objective_per_site)
#'
#' # View the plot of the selected locations
#' results$map_selected_locations
#'
#'
#' # EXAMPLE 2: Maximize accessibility with a budget of 2, specifically for
#' conserving deer, bear, and raccoons
#'
#' # set objective
#' objective <- 'max'
#' names(objective) <- 'accessibility'
#'
#' # set additional constraint, cost must not be higher than 2
#' constraint <- 2
#' names(constraint) <- 'cost'
#'
#' # set direction of constraint
#' const_dir <- '<='
#'
#' # run siteSelection
#' results <- siteSelection(con = conx,
#' objective = objective,
#' constraints = constraint,
#' constraints_dir = const_dir,
#' data_source = 'annotations',
#' included_taxa = c('deer', 'bear', 'raccoon'),
#' verified = FALSE,
#' min_survey_length = 1,
#' map_database = 'state',
#' regions = 'vermont')
#'
#' # View the results objects and notice only 2 of the 3 sites were selected.
#' # All constraints were met by including only 2 sites.
```

```
#'
#'
#'
#'
siteSelection <- function(con,
                          objective,
                          constraints = NULL,
                          constraints_dir = NULL,
                          data_source = 'annotations',
                          verified = FALSE,
                          includeNA = FALSE,
                          modeloutput_threshold = 0.1,
                          included_locations = NULL,
                          min_survey_length = 14,
                          included_taxa = NULL,
                          survey_date_range = NULL,
                          locations_per_species = NULL,
                          map_database = NA,
                          regions = NA) {

  # Argument checks and setup ----------

  # database connection

  if (!DBI::dbIsValid(con)) {
    stop('Please enter a valid database connection.')
  }


  # check objective

  locations <- DBI::dbReadTable(con, 'locations')

  if (length(objective) != 1) {
    stop('The objective argument must be a named vector of length 1.')
  }

  if (!names(objective) %in% colnames(locations)) {
    stop('The objective name must be a column in the locations table.')
  }

  if (objective != 'min' & objective != 'max') {
    stop("The value of the objective must be either 'min' or 'max'.")
```

```
}

# check constraints

if (!is.null(constraints)) {

  if (any(is.null(names(constraints)))) {
    stop("The constraints argument should be a named vector. Names should
    pertain to columns in the 'locations' table in the database, and values
    should specify a minimum or maximum value (direction should be assigned
    with the constraints_dir argument.")
  }

  if (any(!names(constraints) %in% colnames(locations))) {
    stop('All constraint names must be columns in the locations table.')
  }

}
# check data_source

if (data_source != 'annotations' & data_source != 'modeloutputs') {
  stop("The data_source argument must be either 'annotations' or 'modeloutputs'.")
}

# check locations are valid and assign locations to include

if (is.null(included_locations)) {

  included_locations <- unique(locations$pk_locationid)
  included_locations <- included_locations[
    which(included_locations != 'unknownLocation')
  ]

}

if (any(!included_locations %in% locations$pk_locationid)) {

  stop(paste0('The following locations are not in the locations table: ',
              paste(included_locations[
                which(!included_locations %in% locations$pk_locationid)
              ], collapse = ', ')))

}
```

```
included_locations <- included_locations[order(included_locations)]

# check date range argument
if (!is.null(survey_date_range)) {

  # check format
  if (any(
    is.na(
      as.Date(
        survey_date_range,
        format = "%Y-%m-%d")))) {
    stop('The survey_date_range must be in the format YYYY-MM-DD.')
  }

  # check start and end date only
  if (length(survey_date_range) != 2) {
    stop('Please enter a vector of length 2 for the survey_date_range
    argument, c(start_date, end_date). ')
  }

}

# make sure no locations_per_species if no included_taxa
if (!is.null(locations_per_species)) {
  if (is.null(included_taxa)) {
    stop("To use the 'locations_per_species' argument, an 'included_taxa'
    vector of the same order must be entered as well. ")
  }
}

# if only one species of interest, warning to include number of locations
if (!is.null(included_taxa)) {
  if (length(included_taxa) %in% seq(1:4) & is.null(locations_per_species)) {
    warning("This analysis includes a small number of 'included_taxa', and no
    'locations_per_species' argument is specified. The analysis will only
    solve for 1 location per species by default.")
  }
}

if (!is.null(included_taxa) & !is.null(locations_per_species)) {

  if (length(included_taxa) != length(locations_per_species)) {
```

```
    stop("The arguments 'included_taxa' and 'locations_per_species' must be of
    the same length. The order of each should match.")
  }
}

# Read in table data ---------------------

if (data_source == 'annotations') {

  if (verified) {

    if (!is.null(survey_date_range)) {

    stmnt <- paste0("SELECT annotationverifications.*,
    annotations.fk_taxonid, visits.fk_locationid FROM visits
    INNER JOIN (media INNER JOIN (annotations
    INNER JOIN annotationverifications
    ON annotations.pk_annotationid = annotationverifications.fk_annotationid)
    ON media.pk_mediaid = annotations.fk_mediaid)
    ON visits.pk_visitid = media.fk_visitid WHERE media.start_date > '",
                    survey_date_range[1],
                    "' AND media.start_date < '",
                    survey_date_range[2], "';")

      table_data <- DBI::dbGetQuery(con, stmnt)

    } else {

    stmnt <- "SELECT annotationverifications.*,
    annotations.fk_taxonid, visits.fk_locationid
    FROM visits INNER JOIN (media
    INNER JOIN (annotations INNER JOIN annotationverifications
    ON annotations.pk_annotationid = annotationverifications.fk_annotationid)
    ON media.pk_mediaid = annotations.fk_mediaid)
    ON visits.pk_visitid = media.fk_visitid ;"

      table_data <- DBI::dbGetQuery(con, stmnt)
    }

    table_data <- table_data[
      which(table_data$is_valid == 1),
    ]
```

```
    } else {

      if (!is.null(survey_date_range)) {

        stmnt <- paste0("SELECT * FROM (annotations
        INNER JOIN media ON annotations.fk_mediaid = media.pk_mediaid)
        INNER JOIN visits ON visits.pk_visitid = media.fk_visitid
        WHERE media.start_date > '",
                        survey_date_range[1],
                        "' AND media.start_date < '",
                        survey_date_range[2], "';")

        table_data <- DBI::dbGetQuery(con, stmnt)

      } else {

        table_data <- DBI::dbGetQuery(con,
                           statement = "SELECT * FROM
                           (annotations INNER JOIN media
                           ON annotations.fk_mediaid = media.pk_mediaid)
                           INNER JOIN visits
                           ON visits.pk_visitid = media.fk_visitid;")
      }
    }

  } else { # modeloutputs

    if (verified) { # modelverifications

      if (!is.null(survey_date_range)) {

        stmnt <- paste0("SELECT modelverifications.*, modeloutputs.*, visits.*
        FROM visits INNER JOIN (media INNER JOIN (modeloutputs
        INNER JOIN modelverifications
        ON modeloutputs.pk_modeloutputid = modelverifications.fk_modeloutputid)
        ON media.pk_mediaid = modeloutputs.fk_mediaid)
        ON visits.pk_visitid = media.fk_visitid WHERE media.start_date > '",
                        survey_date_range[1],
                        "' AND media.start_date < '",
                        survey_date_range[2], "';")

        table_data <- DBI::dbGetQuery(con, stmnt)
```

```
        table_data <- table_data[
          which(table_data$is_valid == 1),
        ]

      } else {

        stmnt <- "SELECT modelverifications.*, modeloutputs.*, visits.*
        FROM visits INNER JOIN (media INNER JOIN (modeloutputs
        INNER JOIN modelverifications
        ON modeloutputs.pk_modeloutputid = modelverifications.fk_modeloutputid)
        ON media.pk_mediaid = modeloutputs.fk_mediaid)
        ON visits.pk_visitid = media.fk_visitid ;"

        table_data <- DBI::dbGetQuery(con, stmnt)

      }

    } else { #modeloutputs

      if (!is.null(survey_date_range)) {

        stmnt <- paste0("SELECT modeloutputs.*, visits.*, media.*
FROM visits INNER JOIN (media INNER JOIN modeloutputs
ON media.pk_mediaid = modeloutputs.fk_mediaid)
ON visits.pk_visitid = media.fk_visitid WHERE media.start_date > '",
                        survey_date_range[1],
                        "' AND media.start_date < '",
                        survey_date_range[2], "';")

        table_data <- DBI::dbGetQuery(con, stmnt)

      } else {

        stmnt <- "SELECT modeloutputs.*, visits.*, media.*
        FROM visits INNER JOIN (media INNER JOIN modeloutputs
        ON media.pk_mediaid = modeloutputs.fk_mediaid)
        ON visits.pk_visitid = media.fk_visitid;"

        table_data <- DBI::dbGetQuery(con, stmnt)

      }

    } # end non-verified modeloutputs
```

```
  # remove any below threshold for modeloutputs
  if (nrow(table_data) == 0) {
    stop('There is no data in the selected table for this date range.')
  }

  # include NAs or not, and subset by model threshold

  if (includeNA == FALSE) {

    table_data <- table_data[
      which(table_data$value_num > modeloutput_threshold),
    ]

  } else {

    table_data <- table_data[
      which(table_data$value_num > modeloutput_threshold |
              is.na(table_data$value_num)),
    ]

  }

} # end getting table_data

if (nrow(table_data) == 0) {
  stop('There is no data in the selected table for this date range.')
}


# check taxa and assign pk_taxonid if given as argument

 # read in taxa table
  taxa <- DBI::dbReadTable(con, 'taxa')

if (is.null(included_taxa)) {

  # include all taxa from table in analysis
  taxa_ids <- unique(table_data$fk_taxonid)

  taxa_ids <- taxa_ids[which(!taxa_ids %in% c('Human',
                                              'animal sp.',
                                              'no-species') )]
```

```
} else {

  taxa_ids <- included_taxa

  # exchange common and scientific names for pk_taxonid

  for (i in 1:length(taxa_ids)) {

    if (tolower(taxa_ids[i]) %in% tolower(taxa$pk_taxonid)) {

      taxa_ids[i] <- taxa$pk_taxonid[
        which(tolower(taxa_ids[i]) == tolower(taxa$pk_taxonid))
      ]

      next

    } else if (tolower(taxa_ids[i]) %in% tolower(taxa$common_name)) {

      # if common name given, match to pk_taxonid
      taxa_ids[i] <- taxa$pk_taxonid[
        which(taxa$common_name == taxa_ids[i])
      ]

    } else if (tolower(taxa_ids[i]) %in% tolower(taxa$rank_species)) {

      # if scientific name given, match to pk_taxonid
      taxa_ids[i] <- taxa$pk_taxonid[
        which(taxa$rank_species == taxa_ids[i])
      ]

    } else {

      stop(paste0("The following taxa are not in the database: ",
                  paste(
                    taxa_ids[
                      which(
                        !taxa_ids %in% taxa$pk_taxonid &
                          !taxa_ids %in% taxa$common_name &
                          !taxa_ids %in% taxa$rank_species)
                    ],
                    collapse = ', '
                  )))
    } # end else
```

```
    # stop if included_taxa not in table_data

    if (!taxa_ids[i] %in% table_data$fk_taxonid) {
      stop(
        paste0(
          taxa_ids[i], ' is not in the specified table of the database for
          this date range.'
        )
      )
    }

  } # end looping through taxa_ids to match to pk_taxonid

} # end if/else included_taxa


# Creating the presence-absence table --------------

# create an empty dataframe with taxa_ids as column names
pres_abs <- data.frame(matrix(nrow = 0,
                              ncol = length(taxa_ids)))
names(pres_abs) <- taxa_ids

# create list of excluded locations
excluded_locs <- c()

# get dates of survey effort (LB, media dates only)
effort_dates <- AMMonitor::qryEffort(con)
media_effort_dates <- effort_dates[!is.na(effort_dates$activeStartLB),]
media_effort_dates[, 3] <- as.Date(media_effort_dates[, 3])
media_effort_dates[, 4] <- as.Date(media_effort_dates[, 4])
media_effort_dates[, 5] <- as.Date(media_effort_dates[, 5])
media_effort_dates[, 6] <- as.Date(media_effort_dates[, 6])


# loop through included locations
for (loc in included_locations) {

  # get indices of visits at location from effort
  indices <- which(media_effort_dates$fk_locationid == loc)

  # get lower bound effort from media dates
```

```
lb_effort_days <- difftime(
  time1 = media_effort_dates$activeEndLB[indices],
  time2 = media_effort_dates$activeStartLB[indices],
  units = 'days'
)

# skip location if sum of effort across visits does not equal min survey
# length
if (sum(lb_effort_days) < min_survey_length) {

  excluded_locs <- c(excluded_locs, loc)
  next
}

# initialize a row to add to the pres_abs dataframe
pres_abs_loc <- data.frame(matrix(nrow = 1,
                              ncol = length(taxa_ids)))
names(pres_abs_loc) <- taxa_ids

# get annotations for this location
data_loc <- table_data[
  which(table_data$fk_locationid == loc),
]

# if no data in table, skip loc
if (nrow(data_loc) == 0) {

  excluded_locs <- c(excluded_locs, loc)

  next
}

# loop through taxa_ids
for (t in 1:length(taxa_ids)) {

  taxon <- taxa_ids[t]

  # assign presence/absence 1/0
  if (taxon %in% data_loc$fk_taxonid) {

    pres_abs_loc[1, t] <- 1

  } else {
```

```
      pres_abs_loc[1, t] <- 0

    }

  } # end loop through taxa_ids per location

  # add location's presence/absence data to the pres_abs dataframe
  pres_abs <- rbind(pres_abs, pres_abs_loc)

} # end loop through locations for presence absence table

# update cost vector to remove excluded_locs
included_locations <- included_locations[
  which(!included_locations %in% excluded_locs)]

# assign location names to presence/absence table
row.names(pres_abs) <- included_locations

# remove taxa without any presence data (1s)
pres_abs <- pres_abs[which(colSums(pres_abs) != 0)]


# Setup lpSolve function ----------------

# cost object
locations <- locations[
  which(locations$pk_locationid %in% included_locations),
]

locations <- locations[order(locations$pk_locationid),]

f.obj <- locations[,
  which(colnames(locations) == names(objective))
]

# transpose pres-abs data
f.con <- t(pres_abs)

# Set inequality signs for times = num species
f.dir <- rep(">=", times = nrow(f.con))

# Set right hand side coefficients
```

```
if (!is.null(locations_per_species)) {
  f.rhs <- locations_per_species
} else {
  f.rhs <- rep(1, nrow(f.con))
}

# Add other constraints to this f.con matrix and f.dir -------------

if (!is.null(constraints)) {

  for (i in 1:length(constraints)) {

    constraint_data <- locations[,
      which(colnames(locations) == names(constraints)[i])
    ]

    f.con <- rbind(f.con, constraint_data)

    row.names(f.con)[
      which(row.names(f.con) == 'constraint_data')
    ] <- names(constraints)[i]

    f.dir <- c(f.dir, constraints_dir[i])

    f.rhs <- c(f.rhs, constraints[i])

  } # end loop through constraints

} # end adding additional constraints

# Run lp function ---------------
results <- lpSolve::lp(direction = objective,
                       objective.in = f.obj,
                       const.mat = f.con,
                       const.dir = f.dir,
                       const.rhs = f.rhs,
                       all.bin = TRUE
           )

# get results
total_objective_value <- results$objval

# solution = selected locations
```

```
selected_locations <- results$solution
names(selected_locations) <- included_locations

selected_locations <- selected_locations[
  which(selected_locations == 1)
]

selected_locations <- names(selected_locations)

# return the locations table for selected sites
locations_table_selected <- locations[
  which(locations$pk_locationid %in% selected_locations),
]

num_locations_included <- results$x.count

num_species_included <- results$const.count

pa_table <- pres_abs

pa_table_selected_locs <- pres_abs[
  which(rownames(pres_abs) %in% selected_locations), ]

# Output -------------------

output <- list()

# locations table for selected sites
output$selected_sites <- locations_table_selected

# objective value total
output$total_objective_value <- total_objective_value

# presence-absence table
output$pa_table_full <- pa_table

# only include pa_table_selected if more than one taxon
if (length(taxa_ids) > 1) {
  output$pa_table_selected <- pa_table_selected_locs
}

# Map output -----------------------
```

```
locations <- locations[
  which(locations$pk_locationid %in% selected_locations),
]

locations_xy <- locations[stats::complete.cases(
  locations$lat, locations$long), ]

if (nrow(locations_xy) > 0) {

 if (!is.na(map_database)) {
  g <- ggplot2::ggplot(
    locations_xy,
    ggplot2::aes(
      x = long,
      y = lat))  +
    ggplot2::geom_point(
      data = locations,
      ggplot2::aes(
        x = long,
        y = lat),
      size = 3) +
    ggplot2::coord_fixed() +
    ggplot2::labs(
      title = "Selected Locations",
      x = "Longitude",
      y = "Latitude") +
    ggplot2::theme_minimal()

    # map of locations by type
    borders <- maps::map(
      database = map_database,
      regions = regions,
      fill = TRUE,
      plot = FALSE
    )
    g <- g + ggplot2::geom_polygon(
      data = borders,
      ggplot2::aes(x = long, y = lat),
      fill = NA,
      color = "black")

    output$map_selected_locations <- g
  } # end map output
```

```
  } # end if locations_xy has rows

  message(paste0(length(excluded_locs),
             ' locations were excluded from analysis due in insufficient data
             or survey effort.\n'))

  # if no solution was found with lp function, message
  if (all(results$solution == 0)) {

    message('No solution was found with the given objective, constraints, and
    data.')

  } else {

    # give output message and output object
    if (objective == 'min') {
      ob_dir <- 'Minimize'
    } else {
      ob_dir <- 'Maximize'
    }

    message(paste0('Number species and other constraints included: ',
    num_species_included, '\n',
                   'Number locations included: ',
                   num_locations_included, '\n\n',
                   'Objective: ', ob_dir, ' ',
                   toupper(names(objective)), '\n',
                   'Total objective value: ',
                   total_objective_value, '\n',
                   'Selected sites: ',
                   paste(selected_locations, collapse = ', '), '\n\n',
                   'See the output object for the presence/absence table and
                   cost breakdown per site. If map_database was specified,
                   the output object will have a plot of the selected
                   locations.'))

    return(output)
  }


} # end of function
```

References

Altman, B. & Sallabanks, R. (2020). Olive-sided flycatcher (contopus cooperi) (A.F. Poole, Ed.). *Birds of the World*.

Berkelaar, M. & others. (2023). *lpSolve: Interface to 'lp_solve' v. 5.5 to solve linear/integer programs*.

Clarfeld, L., Tang, C., Huber, K., Balantic, C. & Donovan, T. (2024). *AMMonitor: Remote monitoring of biodiversity in an adaptive management framework*.

Confer, J.L., Hartman, P. & Roth, A. (2020). Golden-winged warbler (vermivora chrysoptera) (A.F. Poole, Ed.). *Birds of the World*.

Donovan, T., Hines, J.E. & MacKenzie, D. (2023). *occupancyTuts: Tutorials for occupancy modeling with RPresence*.

Evans, M., Gow, E., Roth, R.R., Johnson, M.S. & Underwood, T.J. (2020). Wood thrush (hylocichla mustelina) (A.F. Poole, Ed.). *Birds of the World*.

Gill, F.B., Canterbury, R.A. & Confer, J.L. (2020). Blue-winged warbler (vermivora cyanoptera) (A.F. Poole & F.B. Gill, Eds.). *Birds of the World*.

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

Lapp, S., Larkin, J.L., Parker, H.A., Larkin, J.T., Shaffer, D.R., Tett, C., McNeil, D.J., Fiss, C.J. & Kitzes, J. (2023). Automated recognition of ruffed grouse drumming in field recordings. *Wildlife Society Bulletin*, **47**, e1395.

Mack, D.E. & Yong, W. (2020). Swainson's thrush (catharus ustalatus) (A.F. Poole & F.B. Gill, Eds.). *Birds of the World*.

McAuley, D.G., Keppie, D.M. & Whiting Jr., R.M. (2020). American woodcock (scolopax minor) (A.F. Poole, Ed.). *Birds of the World*.

Pérez-Granados, C. (2023). BirdNET: Applications, performance, pitfalls and future opportunities. *Ibis*, **165**, 1068–1075.

Porter, W.F. & Jarzyna, M.A. (2013). Effects of landscape-scale forest change on the range contraction of ruffed grouse in new york state, USA. *Wildlife Society Bulletin (2011-)*, **37**, 198–208.

Renfrew, R.B. (Ed.). (2013). *The second atlas of breeding birds of vermont.* University Press of New Hampshire.

Rusch, D.H., Destefano, S., Reynolds, M.C. & Lauten, D. (2020). Ruffed grouse (bonasa umbellus) (A.F. Poole & F.B. Gill, Eds.). *Birds of the World.*

Squires, J.R. & Reynolds, R.T. (2023). American goshawk (accipiter atricapillus) (N.D. Sly, Ed.). *Birds of the World.*