Advances in Biology Laboratory Education, A Publication of the Association for Biology Laboratory Education
Volume 44, Article 11, 2024

An interactive online game that uses simulations of hunting vampire bats to introduce students to coding

Elizabeth A. Hobson[1], Sanjay Prasher[1], Gerald Carter[2], and Joshua Garland[3]

[1]University of Cincinnati, Department of Biological Sciences; 318 College Dr, Cincinnati, OH 45219 USA
[2]Ohio State University, Department of Evolution, Ecology, and Organismal Biology; 318 W 12th Ave, Columbus, OH 43210 USA
[3]Center on Narrative, Disinformation and Strategic Influence, Arizona State University; 21 E 6th Street, Tempe, AZ 85281 USA

Abstract

There is increasing awareness that data science and computational thinking are critical skills for undergraduates to develop but these can be difficult to integrate into undergraduate Biology classes. In this module, we describe how we have used a system for learning the programming language R that focuses on building students' skills and confidence in data exploration, management, and visualization. This activity pairs a hands-on virtual experiment where students simulate animal movements and social interactions to provide a friendly introduction to basic data science for biologists. During the activity, students play the "Bat Game", an online game which students access via an internet browser. Each student controls the movement decisions of one bat within a social group. The bats must search for cows they can bite to get a meal of blood. Students take the roles of bats in a series of foraging tasks. Students must follow "rules" and attempt to match their overall actions to those of their group members under different scenarios. The game platform collects all the locations of all bats in the game. After playing the game, students export the data they just created and analyze it to learn how to detect known patterns through basic summaries and plotting in R. All analyses and programming skills are presented in one cohesive R Markdown file, where students can read about the goals of each coding chunk, can run each chunk, and then answer questions about the biology of the social system as well as basic questions about the code used in the analyses. This approach decouples coding from statistics, assumes no prior knowledge, and uses a charismatic species to incentivize student participation. This module can be used in many courses including lab sections of large-enrollment introductory biology courses as well as smaller upper-level courses.

Keywords: inquiry-based learning, social networks, animal behavior, animal sociality, computational literacy, data science skills

Citation: Hobson EA, Prasher S, Carter G and Garland J. 2024. An interative online game that uses simulations of hunting vampire bats to introduce students to coding. Article 11 In: Boone E and Thuecks S, eds. *Advances in Biology Laboratory*

Education. Volume 44. Publication of the 44th Conference of the Association for Biology Laboratory Education (ABLE). DOI: https://doi.org/10.37590/able.v44.art11

Correspondence to: Elizabeth Hobson, Elizabeth.Hobson@uc.edu

INTRODUCTION

Data science skills are increasingly important to modern undergraduate training in biology (Carey & Papin, 2018; Markowetz, 2017), but data science, computational analysis, and statistics can be intimidating for students (Metz 2008; Chew and Dillon 2014; González et al. 2016). Because these skills are increasingly important, but courses built solely around introductory programming can be unattractive to biology students, it is crucial to find ways to make these topics engaging for students and better integrate this skills training more broadly into undergraduate biology curriculum.

This module demonstrates one way to introduce students who are complete coding novices to the power and potential of programming. We use R because it is an increasingly common language for statistical programming in biology (R Core Team 2021). We base our module around animal movements and vampire bats to tie the coding skill development to interesting and foundational biological concepts. Animal movements are key to many fundamental biological and conservation-relevant processes, like habitat choice, dispersal, territoriality, and social foraging. Vampire bats are charismatic species that are a textbook example of complex social relationships and cooperation in nature.

We developed an interactive online game students play to learn about animal movements, vampire bat biology, and basic R coding skills. Students join the virtual habitat and "role play" the movement decisions of individual animals, according to a set of social rules. Students then analyze the data they create from following different rule sets and compare how movement patterns differ when the underlying rules are changed. Students play four versions of the game: (1) wander, (2) random movement, (3) social foraging, and (4) bet-hedging social foraging. In "wander" students explore the game habitat and then compare their movements to the movements of 12 other vampire bats, whose movements are pre-set and who function as "NPCs" (Non-Player Characters). In "random movement", students try to match the random walk movements of the NPC bats. In "social foraging", each student picks a bat as a "buddy" with the goal of finding cows to feed on and staying close to their chosen buddy. In "bet-hedging" students play the social foraging game again, but this time with three "buddies" that they try to stay near. These movement patterns mimic parts of the social strategies that real vampire bats use to form relationships and forage in the wild.

After students play each version of the game, they analyze their data using an R Markdown file on Posit Cloud. These analyses allow students to compare their movement patterns with the NPC bats' movement to see whether they have followed the movement patterns of the NPC bats closely enough, or if they are still outliers in the social systems they are attempting to join. Student questions interspersed through the Markdown file direct students to critically consider the biology underlying all these patterns as well as having them read, examine, and make sense of the pre-written code to build a basic familiarity with analyses in R.

The interactive and flexible nature of our game's design also allows students to make a hypothesis about how movement patterns may be affected by certain factors and then test that hypothesis in a role-playing scenario. The flexibility of both the game and the module make them adaptable to many courses and institutions.

STUDENT OUTLINE

Objectives

1. Convince you that data science and coding skills are valuable for biologists and worth learning.
2. Provide you with basic natural history information for Vampire Bats, a fascinating species where individuals form relationships that lead to cooperative food sharing and social foraging.
3. Explore animal social interactions in online simulations. These "games" examine the outcomes of several "social rules" to see how changing your goals affects how individual animals interact with each other.
4. Introduce you to the R code underlying the analyses and provide a sampling of the power of custom analyses that is possible when coding.

Introduction

Vampire bats are fascinating social creatures. To survive, they must drink blood from other larger animals, like cows, but they sometimes fail to get a meal. To buffer against the risk of starvation, vampire bats can share portions of their own blood meals with each other. This cooperative food sharing affects how the bats' relationships form and change over time. Their social relationships also influence their movement decisions while hunting (foraging) and choosing where to roost. These movements shape how the group's social structure forms and changes over time. In this one-day lab module, you will first get an introduction to vampire bat biology by watching a series of short videos. You will then play a series of online games where you will attempt to join a vampire bat society as the bats move around a habitat looking for cows to feed on. After you play each version of the game, you'll use the programming language R to analyze your data. You'll also be asked to read over the R code and answer questions about both the code and the biological implications of each different game scenario for the bats. At the end of the lab you will "knit" your code to compile all of your R code, analyses, data summaries, plots, and answers to all the questions within a single HTML file to make a final report.

Methods and Data Collection

Part A: Learn the basics of vampire bat biology

Vampire bats are incredibly fascinating animals. Watch the two videos to hear researcher Gerry Carter give an overview of vampire bat biology. Gerry has spent his career specializing on vampire bat cooperation and sociality. For more information on his research, see https://socialbat.org/research/.

For the lab today, the main points you need to know about vampire bats are:

1. Vampire bats feed on blood to survive
2. They roost in caves or tree cavities during the day and fly around to forage on cattle at night
3. Without feeding, vampire bats can starve quickly
4. If a vampire bat failed to feed on blood, its 'friends' can share food with it by regurgitation. This helping behavior appears to be reciprocal: bats feed partners that later feed them

Today, you'll also use the coding language R. R is a coding language that is widely used in biological research. It's particularly good at both statistical analyses and plotting data in customized ways. Knowing how to code can give students important advantages and make them more competitive on the job market both in academia as well as in a wide range of non-academic careers. Beyond being an important skill, it will make your science more reproducible because you have a record of everything that you did to wrangle your data, plot your plots, and analyze your statistics. We are not doing any stats today -- we're focusing just on data wrangling, summarizing, and plotting. The lab assumes no prior experience with coding and will walk you through all the steps. If you have never coded before, don't worry! This lab will allow you to use custom analyses in a way that walks you through the code but doesn't require you to write any new code. This lab is specifically targeted to complete beginners in R.

The main points you need to know about coding in R to complete the lab are:

1. R is sensitive to capitalization and spelling/typos. If you are told to save data as "random.csv" and instead save it as "Random.csv", the code will fail! If you get an error, check for typos or differences between your file names in your project and the file names in the code - these must be exactly the same.

2. We are working in an *R Markdown* file. This file format is a nice way to combine code chunks and text blocks. At the end of the lab, you will "knit" the file which means compile all the code, plots, tables, results, and your answers to all the questions into a single HTML file which you will submit to your instructor. These HTML files are viewable in your browser.
3. In the markdown file, all code appears in grey boxes enclosed by backquote or backtick symbols (see more information about this symbol here: https://www.computerhope.com/jargon/b/backquot.htm). There are three backquote symbols at the start of the code block and three backquote symbols at the end of the code block. Don't delete these! Your answer text blocks are also enclosed with these backquote symbols (just replace the "YOUR ANSWER HERE" text with your own answer)
4. To run each code block, refer to the provided powerpoint slides for pictures and descriptions (https://doi.org/10.6084/m9.figshare.25809901).. To run code, first read the coding explanation text. Then click the "play button" on the right-hand side of the screen to run that chunk. You will see a stop sign appear while the code is running and a green line will display on the left side of the screen to show progress on running the code. Once a code chunk has run, the results will display under the code chunk (the table or plot will show). You do not need to click the "play button" to run your answers to questions (that will be run at the end when you Knit the whole file).
5. Notes appear within the grey code blocks as green text with a "#" sign in front of it. This is a way to add annotations and notes into R code - R will not run whatever is on the same line behind the # symbol
6. Make sure you read each section! These descriptions provide a walk-through of the code and important information for each part of the lab
7. Watch for questions -- These are all labeled "**Q..." and will appear as bright blue text within the R Markdown file. Make sure you respond to each of the questions and answer the question in the space provided (just replace the "YOUR ANSWER HERE" text with your own answer -- DO NOT DELETE the backquote symbols and make sure they stay on their own lines without any extra text on them!)
8. MAKE SURE YOU SAVE YOUR WORK!! Click the save icon in the menu directly above the R Markdown code you are working on.

Here are links to the first two videos we watched about the bats today:

- Gerry gives an overview of vampire bats: https://www.youtube.com/watch?v=pk2brNQxQxY
- Video of two vampire bats feeding on a cow at night: https://www.youtube.com/watch?v=lJDq9yjWXSk

Part B: Play the four game versions and analyze your data

Follow the student guide slides to (1) Play the first version of the game, "Explore", (2) sign up for a free Posit Cloud account, (3) create a new Posit Cloud project to pull code and data from GitHub, and (4) upload your "Explore" game data. Use this link (http://hobsonresearch.com/index.php/sociality-game/the-sociality-game-vampire-bat-version/) to navigate to the Bat Game online. Refer to the provided powerpoint slides for pictures and descriptions of steps (https://doi.org/10.6084/m9.figshare.25809901).

Game overview:

- The Bat Game is a single-player game played within a browser (must have internet connection).
- The game shows a habitat with: 3 roost trees, 6 cows, 12 pre-programed bats, and 1 "Player" bat. Bats labeled 0-11 are "non-player characters" (NPCs).
- Each student controls the bat named "Player" by using their arrow keys to move the bat around the habitat. The player follows simple "rules" to join an existing social system. These rules change based on which version of the game is played.
- In the background, the game collects all locations of all the bats. Each game lasts 2-3 minutes. At the end of the game, each student can download these locations and then analyze them in R.
- The analyses show how well each player conformed to the social rules of the system.

To help you complete your project, you will be guided through the steps in today's lab by the R Markdown file "analysis_code.RMD". Read through each section carefully as it contains critical information for completing the lab, descriptions of the goals of each coding block, and questions you will need to answer for your report.

Part B1: Play the "Explore" version of the game & save your data.

In this version of the game, your goal is to wander around the habitat exploring. Choose the "Random Walk" movement style and start the game. During the game, you can choose where you move and which roosts, cows, or NPC bats to visit. When the game ends (after about 2.5 minutes), you will see a prompt to either download the data or reset the game. Choose "Download CSV". CSV means comma separated values; it is a file format for saving data. Change the name of the file to "explore" (without the quotation marks) and save the file as a CSV somewhere easy to find on your computer (for example, in your "Downloads" folder). As a reminder, R will cause an error if you have any differences in spelling/capitalization/etc. in your file names – make sure you have spelling and capitalization exactly the same as directed.

Part B2: Register for a free Posit Cloud account.

Next you will register for a free Posit Cloud account. Visit the Posit Cloud site (https://posit.cloud/), click the "Sign Up" button in the upper right of the screen, then select the "Free" account version. You may need to confirm your email. Refer to the provided powerpoint slides for pictures and descriptions.

Part B3: Set up a new project in Posit Cloud & import code and data from GitHub repository.

Once you have your account opened and you are signed in, make sure you are in "Your Workspace" (click the button with this name in the left-hand menu if you need to navigate there; refer to the provided powerpoint slides for pictures and descriptions https://doi.org/10.6084/m9.figshare.25809901). Now we'll make a new project for the Bat Game. Click the "New Project" button in the upper right of your workspace screen. You'll see a drop-down menu: choose the option "New Project from Git Repository". When prompted, use this link to access the repository https://github.com/EAHobson/BATGAME. This will link your project with the Bat Game centralized code and will provide you with the analysis code, the locations of all the landmarks, some example data, and other aspects needed for the lab today. Call your new project "BATGAME".

Part B4: Upload the "explore" data you created playing the first version of the game.

To upload your data to your Posit Cloud account, first find the panel on the right with the tabs "Files", "Plots", "Packages" etc. (refer to the provided powerpoint slides for pictures and descriptions). Right below this you will see an "Upload" button. Click Upload, then click "Choose File" and navigate to where you saved the explore.csv file on your computer. Select it, then click "OK" (you do not need to change anything about the "Target Directory". Once you click OK you should see the explore.csv file displayed in your file list.

Play four versions of the game

During lab today, you'll play four versions of the Bat Game. Each version has slightly different rules you will follow to structure your movement decisions. This section will give you an overview of each game version and the Data Analysis section will summarize all the major analyses you'll do in lab today. Follow the instructions in the Markdown file which will walk you through when to play each game and which analyses to run. You will alternate between playing a game version and analyzing the data you produce in Posit Cloud, then going back to the game to play another version and returning to Posit Cloud to analyze the new data.

Here is a summary of the four versions of the game you will play and analyze today:

Game #1: Explore.

In the first play of the game, your goal is to explore the habitat. We'll then compare your movements with the rest of the bats. Choose the "Random Walk" option from the initial game menu at https://eahobson.github.io/. You have no rules in this game other than the goal of exploring the space. Play the game, which takes 2-3 minutes, then save your data as "explore.csv".

Game #2: Random.

The second time you play the game, your goal is to try to mimic the random movements of the other bats (NPC bats). Choose the "Random Walk" option from the initial game menu at https://eahobson.github.io/. Play the game, which takes 2-3 minutes, then save your data as "random.csv".

Game #3: Social preferences.

When you played the first two versions using the Random Walk option, all the other (NPC) bats were programmed to just move randomly around the habitat, without responding to anything in the habitat. Real bats are unlikely to move that way--as you noticed in the explore and random games, moving randomly makes it so almost none of the bats actually find a cow. This would be a terrible strategy for a real bat. Real bats use a suite of rules to find food, but also seek out other bats that they have strong relationships with (for an overview, watch this video of Gerry explaining how they study bats in the wild: https://www.youtube.com/watch?v=BeAiUBM18Cs&t=248s). The third time you play the game, you'll choose "Social Preferences" as the movement type (https://eahobson.github.io). In the "Social Preferences" version of the game, the NPC bats have new rules that they use to structure their movements. Each NPC bat is assigned a "buddy". The NPC bats now have two goals: (1) they need to find cows to forage on, but (2) they also prefer to be near their assigned buddy. In this version of the game, activate the game and wait 10 seconds. Watch how the bats start to move around the habitat. Use this time to pick one bat to be your "buddy". Once the 10 seconds have passed, you will use the same rules that the NPC bats use: your goal is to move around and find cows to feed on, but you also need to stay close to the bat you've chosen as your buddy. Play the game, which takes 2-3 minutes, then save your data as "socforag.csv".

Game #4: Bet-hedging.

When you played the first version of the social preferences game (Game #3), you picked one bat to be your "buddy" and followed two rules: (1) visit the cows to feed on them (forage), and (2) try to stick close to your buddy while foraging. In the real world, this strategy of having one very strong friend might be risky: if something happens to your one friend (it gets sick, is predated, wanders off, etc.) and you desperately need some food sharing, you wouldn't have any other strong friendships to rely on. Real vampire bats appear to develop more friendships than they seem to need as a way to "bet-hedge" against the risk of any one friend going missing. Gerry wrote a paper about this and you can read it here: https://doi.org/10.1098/rsbl.2017.0112 (Carter et al. 2017). In this version of the game, you will play the "Social Preferences" version again. However, this time you will use the first 10 seconds of the game to pick THREE BUDDIES. Once the first 10 seconds have passed, your goals are now to (1) visit the cows to forage, and (2) spend as much time as you can near all three of your buddies. Play the game, which takes 2-3 minutes, then save your data as "bethedge.csv".

Data Analysis

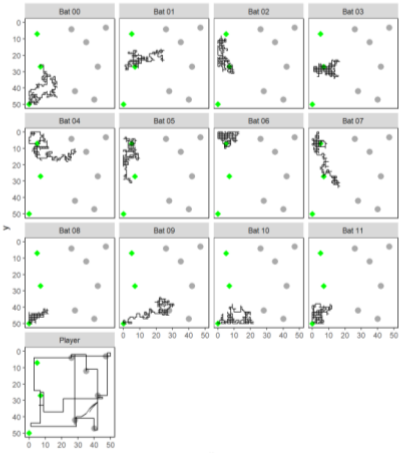
Once you have your first game data file uploaded to your BATGAME project in Posit Cloud, you'll follow the directions in the Markdown file to complete each stage of the analyses. To open the analysis code file, find it in the list of files in your project (see tab "Files" on the right-side panel). You should see the file "analysis_code.RMD". Click on the file name (not the check box next to it). This will open the code in the left-hand window. Follow the directions in the Markdown file to progress with your analyses. To run each code chunk, click the green "play" symbol in the upper right of each code block. You must run these blocks in the order they appear. Do not alter any of the code within the code blocks, but you will add text in the sections indicated in the file. Make sure to save your work as you go (click the "Save" icon above the coding window).

In each part of the analyses, you will compare your behavior to that of the NPC bats who are following certain sets of social and movement decision rules. You will compare your behavior using several summary measures and plot the comparison of your behavior versus that of the NPC bats. You will not use each comparison for each of the games – refer to the Markdown file code for instructions on which summaries to plot and compare for each of the four games. Below, you'll find a summary of each of the measures and plots you will generate for parts of the lab today.

1. Plot bat trajectories

We will plot all the movement trajectories for each of the bats in the game, one per panel. These panels each show the x and y coordinates for all locations and the black line traces where each bat went during the game. The colored points show the location of the cow and roost landmarks. Your data is labeled "Player" and appears last. For this plot, visually compare your trajectory to that of the NPC bats. Is it similar or different? In Figure 1, you can see an example where the player behaved very differently from the NPC bats.

Figure 1: Plot of bat trajectories



2. Summarize characteristics of trajectories and plot

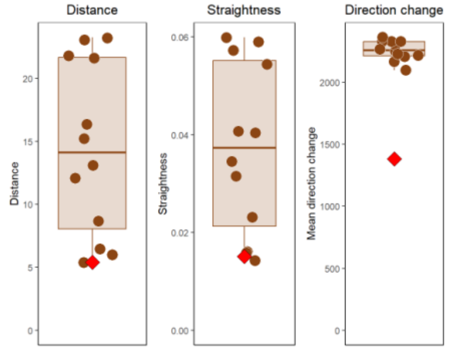
We will summarize several characteristics of these trajectories using the R package "traj" (see (McLean & Skowron Volponi, 2018) and https://cran.r-project.org/web/packages/trajr/trajr.pdf for more detail about the measurements).

We will measure:

a) **Distance**: how far each bat traveled. A higher number indicates that the bat traveled a longer distance than other bats.
b) **Path straightness**: a measure comparing the beeline distance between the first and last points in the trajectory and the path length travelled (Batschelet, 1981). A lower number indicates that a bat traveled a straighter path between its starting and ending destinations than other bats.
c) **Mean direction change**: a measure of how much the direction of movement changes between two steps in the trajectory, divided by the time difference between the two steps. A higher number indicates a bat that more drastically changes its direction from one step to the next, rather than continuing in the same direction for awhile.

We will then plot these measures for all the NPC bats as a boxplot and compare the player measures to the NPC measures. For an overview of boxplots, visit https://www.rforecology.com/post/2022-04-06-how-to-make-a-boxplot-in-r/ and read the section "Boxplot Components". We'll plot measure separately and then combine all of them into a multi-panel plot where we can compare all the measures side by side. These plots all show the NPC data with raw points shown in brown circles and the summary of the variability in the NPC data shown as a boxplot Generally, any points that are higher or lower than the highest and lowest ends of the "whiskers" of the box plot are considered "unusually" larger or smaller than most of the data points and are treated as outliers. The raw NPC data are "jittered" in x-axis space so that points are easier to see and are less likely to overlap.

Figure 2: Summary of trajectory characteristics

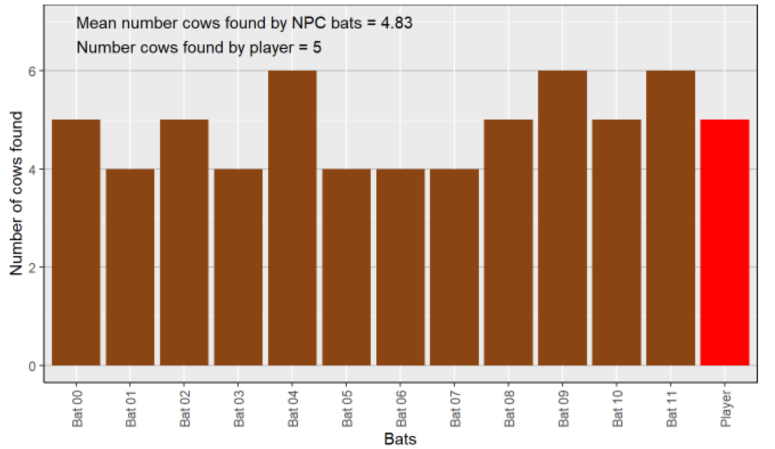


To answer the questions in the lab, compare where your data (Player, red diamond) falls compared to the NPC bats, who are all plotted in brown. If the red point is higher than all of the brown NPC bat data points, and is higher than the highest end of the whisker on the boxplot, then your behavior in that measure was unusually higher than that of the NPC bats. If your red point is lower than the lowest end of the boxplot whisker, your behavior in that measure is unusually lower than the behavior of the NPC bats. For example, in Figure 2 you can see that the Player points are on the low end of measures for Distance and Straightness compared to the NPC bats, but not outside of the whiskers of the boxplot, so are not unusually low compared to the NPC bat measures. However, the player's Direction change measure is very much less than the NPC bats' measures and falls completely outside of the range of the NPC data, indicating that the player here had an unusually low mean direction change compared to the NPC bats.

3. Quantify bat to landmark distances and determine how many cows were found by each bat

In this part of the analysis, we'll determine how far each bat is from each landmark. We'll quantify these distances every 10 seconds. We'll use a distance threshold to determine whether bats are "near" or "not near" each landmark at each time period. In the code chunk, we count a bat as "near" a landmark if the bat is 5 units or less distance away from the landmark. The code keeps data for any time any bat is recorded as "near" a landmark and gets rid of all the times bats were not near a particular landmark. Then we'll summarize this data to determine how many unique cows each bat visited, and plot the data as a bar plot with NPC bat data plotted in brown and the player's data plotted in red.

Figure 3: Quantify number of cows visited by each bat



To answer the questions in the lab, look at the data in the plot and the summary text showing the mean number of cows visited by NPC bats compared to the number of cows visited by your player bat. In Figure 3, the mean number of cows visited by NPC bats was 4.83 and all NPC bats visited at least 4 cows. The player bat visited 5 cows, which is very similar behavior to the rest of the NPC bats. In this case, the player bat is acting "normally" compared to the NPC bats.

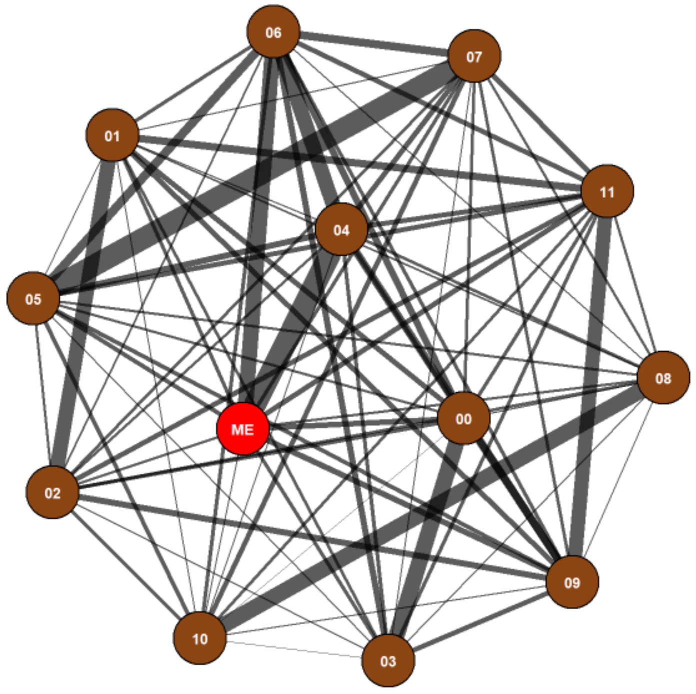
4. Use location data to find distances between all bats; use these to build proximity networks

Now we'll quantify and plot the social networks for the bats in the social preferences game. These networks will summarize the proximity network, where we'll count the number of times each bat was observed as "near" other bats. Similar to how we found the distances between each bat and each landmark, we'll now calculate the distance from one bat to every other bat once every 10 seconds. We'll count bats as "near" each other if they are within 5 or less distance units from each other, just like we used for the landmarks earlier. We'll then use these summaries to build social networks showing which bats were near each other and measure two aspects of those networks.

These proximity networks will show the number of times each bat was observed as "near" other bats. The bats will be the "nodes" in the network (plotted as circles) and the proximity observations will be the links or "edges" (plotted as lines connecting the nodes). The width of each edge will represent how often those two bats were seen in close proximity to each other. Bats that are often seen close together will have wider edges connecting them, bats that are rarely observed close together will have narrow edges connecting them, and bats that are never seen together will not have a connection.

In Figure 4, the “Player” bat is shown in red and labeled “ME” and the NPC bats are shown in brown and labeled with each of their identification numbers. You can see that the player bat is most often in proximity to Bat 04 (wide edge connecting them), somewhat often near Bat 06, and rarely near Bat 03 (narrow edge connecting them).

Figure 4: Social network showing bat-to-bat proximities



5. Quantify measures of proximity network structure

We'll use network metrics to quantify two social measures: "degree" and "strength".

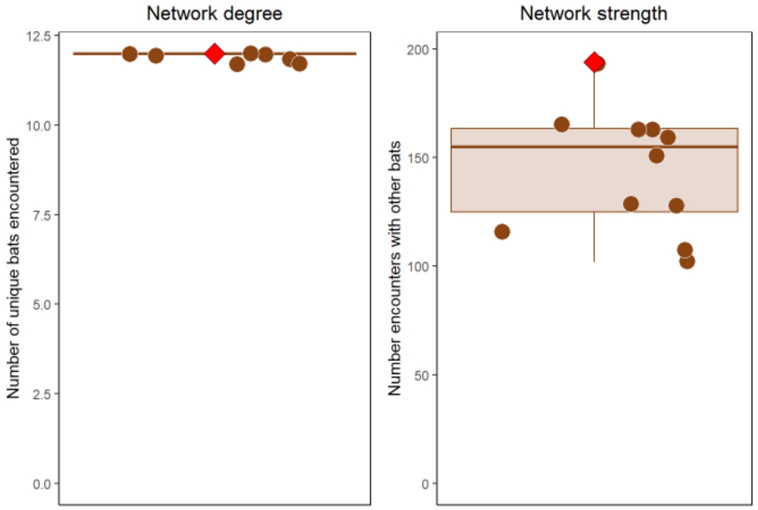
a) **Network degree**: Degree quantifies how many others each individual is connected to in the network, and represents a count of the nodes each individual is directly connected to. Here, each bat's degree is the total number of other bats that it was ever recorded as "near". A bat with a degree of 3 was observed "near" 3 other bats.

© 2024 by Elizabeth Hobson, Sanjay Prasher, Gerald Carter and Joshua Garland

b) **Network strength**: Strength quantifies how often individuals are connected to others in the network and represents the sum of all the weights of the edges in the network.

In Figure 5, you can see that all the NPC bats and the player bat all had a network degree measure of 12, which means that all the bats were in close proximity to each other bat in the group at least once during the game. The player degree overlaps with what the NPC bats had, so the player bat is acting "normally" compared to NPC bats with this measure. The player bat had a network strength measure that was on the high side compared to the NPC bats, but the measure wasn't higher than the highest point of the whiskers on the boxplot, so wouldn't be considered quite abnormally extreme in its strength measure.

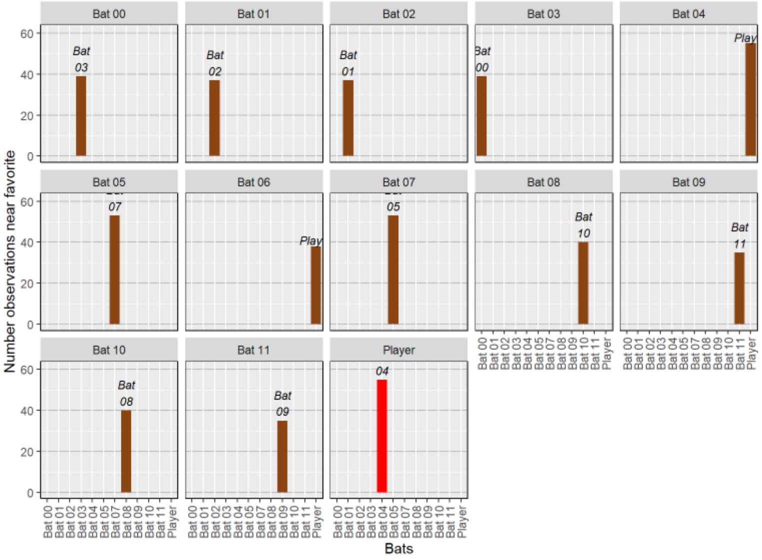
Figure 5: Social network measures of NPC and player bats who are categorized as "near" each other



6. Find each bat's closest associate in the proximity network

In this section you will use the proximities to find each bat's strongest partner or partners - this is whichever bat each bat is most often scored as "near" in location. For some analyses, you'll also find the top three associates.

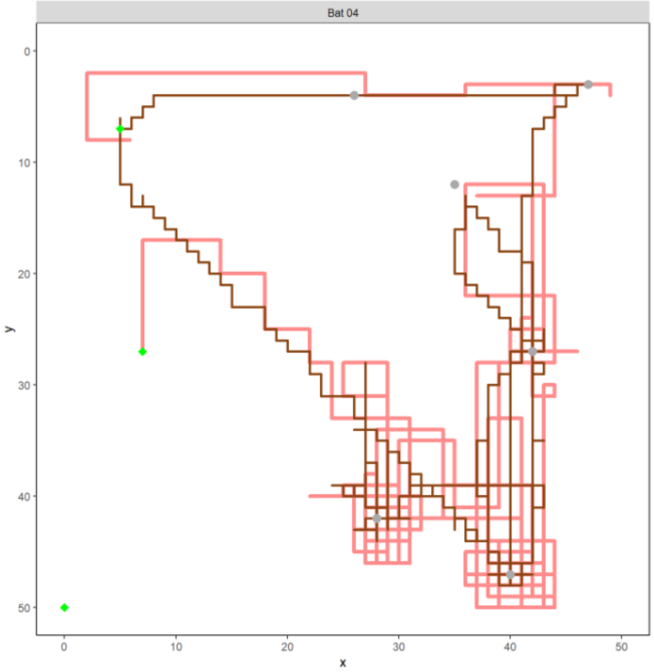
Figure 6: Closest associates



We'll plot this data in a bar plot, where each bat has a panel and the bar in that panel shows that bat's closest associate(s), shown in Figure 6. The NPC bats are shown in brown and the player bat is shown in red. On the y-axis is the number of times each bat was observed in close proximity ("near") each bat of interest, and on the x-axis are all the possible bats.

In Figure 6, you can see that the player's closest associate in this game was Bat 04, while Bat 11's closest associate was Bat 09.

Figure 7: Trajectory matching with close social associate(s)



7. Comparison of trajectories for strongest partners

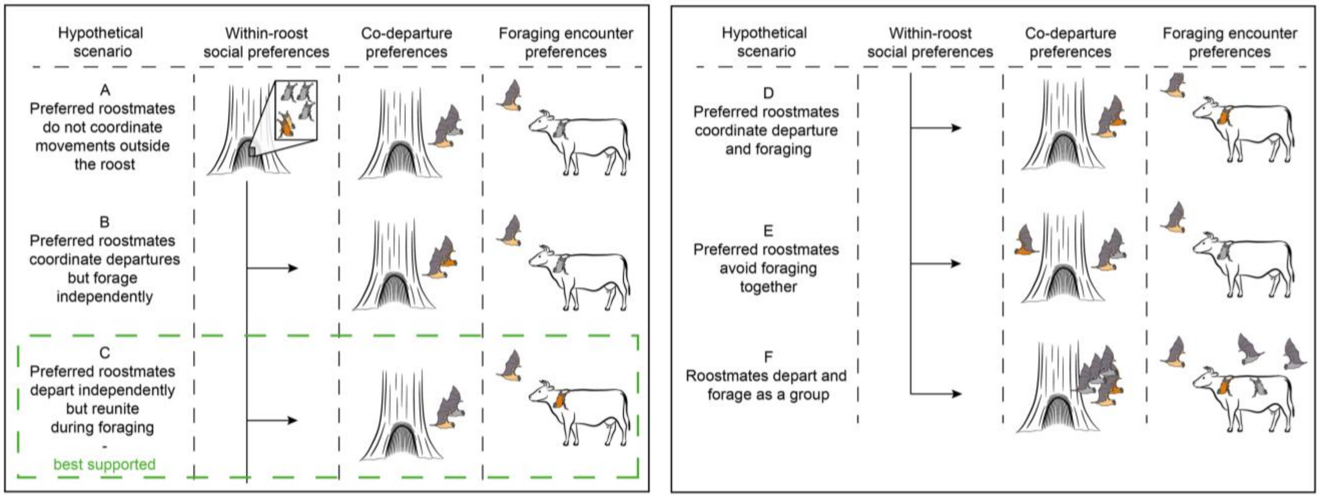
In this section, we'll compare the trajectory of the player with the trajectories of its top (or top three) closest partners. This is similar to what we plotted in Figure 1 but this version provides a way to directly compare how well the player matched the movements of their chosen buddy.

In Figure 7, the player's trajectory is plotted with the red line and the closest social associate's trajectory (for Bat 04 in this case) is shown with the brown line. You can see that the player overlapped the general path of Bat 04 pretty well, but did deviate from Bat 04's path in some small ways. Overall this is a good match between the player's movement and the strongest associate's movements.

Discussion

At the end of the lab today, once you have played four versions of the game and followed different rules each time, you will answer several discussion questions about the broader implications of the patterns you observed. These questions will ask you to carefully consider your experiences in the game and what you now know about vampire bat foraging and survival. In the social foraging game, you had two "rules" you used to make your movement decisions: you wanted to find cows to forage on and you wanted to stay relatively close to your chosen "buddy". Real bats can use similar rules to structure their movements or may do something different. Several potential strategies are summarized in Figure 8.

Figure 8: Several potential social foraging strategies that vampire bats could use. This figure is modified slightly from the figure that appears in Ripperger & Carter, 2021. The movements of real bats matched most closely to scenario C; preferred roostmates who also groomed and shared food left their roost individually but then reunited while foraging. It remains a mystery why preferred roostmates never left the roost together.



What rules would you need in the following additional scenarios?

- Scenario A: Preferred roostmates do not coordinate movements outside the roost while foraging.
    - What rules should the player and the NPC bats follow in a game like Scenario A?
    - If you were a researcher in the field studying these bats, how could you tell whether bats were following Scenario A? What would you look for? How would you tell which bats preferred each other?
- Scenario B: Preferred roostmates coordinate their roost departures but avoid foraging with each other
    - What rules should the player and the NPC bats follow in a game like Scenario B?
    - If you were a researcher in the field studying these bats, how could you tell whether bats were following Scenario B? What would you look for? How would you tell which bats preferred each other?
- Scenario F: All roostmates depart and forage as a group.
    - What rules should the player and the NPC bats follow in a game like Scenario F?
    - If you were a researcher in the field studying these bats, how could you tell whether bats were following Scenario F? What would you look for? How would you tell which bats preferred each other?

Long-term social strategies for survival

Vampire bats can share food with each other if one bat isn't successful at foraging, but generally only share blood when they have a strong relationship based on clustering and social grooming. Food sharing occurs after foraging once bats have returned to the roost. If you are a bat living in an uncertain environment, where your chances of being unsuccessful at foraging and needing food sharing are higher, what social strategies could you use to increase your chances of survival?

How would you change your foraging patterns to increase your chances of survival? What strategy would you use and what rules would you use to move around?

How would you change your roosting patterns to increase your chances of survival? What strategy would you use and what rules would you use to decide which roost to use and who to share a roost with?

Cited References

Carey MA, Papin JA. Ten simple rules for biologists learning to program. PLoS Computational Biology. 2018 Jan 4;14(1):e1005871.

Carter GG, Farine DR, Wilkinson GS. Social bet-hedging in vampire bats. Biology Letters. 2017 May 31;13(5):20170112.

Chew PK, Dillon DB. Statistics anxiety update: Refining the construct and recommendations for a new research agenda. Perspectives on Psychological Science. 2014 Mar;9(2):196-208.

González A, Rodríguez Y, Faílde JM, Carrera MV. Anxiety in the statistics class: Structural relations with self-concept, intrinsic value, and engagement in two samples of undergraduates. Learning and Individual Differences. 2016 Jan 1;45:214-21.

Markowetz F. All biology is computational biology. PLoS biology. 2017 Mar 9;15(3):e2002050.

McLean DJ, Skowron Volponi MA. trajr: an R package for characterisation of animal trajectories. Ethology. 2018 Jun;124(6):440-8.

Metz AM. Teaching statistics in biology: using inquiry-based learning to strengthen understanding of statistical analysis in biology laboratory courses. CBE—Life Sciences Education. 2008 Sep;7(3):317-26.

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ripperger SP and Carter GG. 2021. Social foraging in vampire bats is predicted by long-term cooperative relationships. PLOS Biology 19(9): e3001366.

MATERIALS

Each student needs access to a computer with an internet connection. Each student also needs to register for a free Posit Cloud account (https://posit.cloud/).

NOTES FOR THE INSTRUCTOR

Please refer to the included powerpoint slides linked below for visual aids for some steps, as well as introductory slides that can be shown to students at the start of the lab session.

Hobson, Elizabeth (2024). BATGAME class slides. figshare. Presentation. https://doi.org/10.6084/m9.figshare.25809901

ACKNOWLEDGEMENTS

We thank ABLE for the funding to build the Bat Game and our game developer Lukas Potters for coding the mechanics of the Bat Game. NSF-IOS 2015932 provided additional support for game development.

About the Authors

Liz Hobson is an Assistant Professor in the Department of Biological Sciences at the University of Cincinnati. She teaches a range of undergraduate and graduate courses with a focus on incorporating coding skills and analytical approaches. She developed the course “Analytical Tools for Behavior” which is a semester-long introduction to R coding for upper-level undergraduates as well as designing and implementing shorter data science modules for large introductory biology courses.

Sanjay Prasher is a PhD candidate in the Hobson Lab at the University of Cincinnati, where he studies animal social cognition using empirical and computational approaches. He has worked as a teaching assistant for a variety of biology lab courses (e.g., introductory biology, human and comparative anatomy).

Gerry Carter is an Associate Professor at The Ohio State University and a Research Associate at the Smithsonian Tropical Research Institute. He primarily teaches about evolution, ecology, and animal behavior.

Josh Garland is an Associate Professor in the Center on Narrative, Disinformation and Strategic Influence at Arizona State University. While Josh is primarily a mathematician working in complex systems research, he enjoys collaborating on fun biological research projects. He is a proponent for broader teaching of coding skills, but prefers Python to R

© 2024 by Elizabeth Hobson, Sanjay Prasher, Gerald Carter and Joshua Garland

Mission, Review Process & Disclaimer

The Association for Biology Laboratory Education (ABLE) was founded in 1979 to promote information exchange among university and college educators actively concerned with teaching biology in a laboratory setting. The focus of ABLE is to improve the undergraduate biology laboratory experience by promoting the development and dissemination of interesting, innovative, and reliable laboratory exercises. For more information about ABLE, please visit https://www.ableweb.org/.

Papers published in *Advances in Biology Laboratory Education: Peer-Reviewed Publication of the Conference of the Association for Biology Laboratory Education* are evaluated and selected by a committee prior to presentation at the conference, peer-reviewed by participants at the conference, and edited by members of the ABLE Editorial Board.

Compilation © 2024 by the Association for Biology Laboratory Education, ISSN 2769-1810. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the copyright owner. ABLE strongly encourages individuals to use the exercises in this volume in their teaching program. If this exercise is used solely at one's own institution with no intent for profit, it is excluded from the preceding copyright restriction, unless otherwise noted on the copyright notice of the individual chapter in this volume. Proper credit to this publication must be included in your laboratory outline for each use; a sample citation is given below the abstract.