Study

Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution

Supplementary Information 1

Representing technical difficulty and degree of collaboration in use cases for Github in ecology and evolution

Appendix S1.1. Computing indices of technical difficulty and degree of collaboration

We aimed at distributing the use cases discussed in our study across gradients of technical difficulty and collaboration. For this, we estimated the perceived degree of collaboration and the perceived level of technical difficulty for each use case.

Degree of collaboration

Each author attributed a value ranging from 1 to 10 to their perceived degree of collaboration for a use case, with one representing a use case that offer very low advantages in terms of collaboration, while ten classified that use case as highly advantageous in terms of collaboration.

We determined the degree of collaboration for each GitHub use case as the range of collaborative contexts that can be achieved with each tool, as perceived by the authors. We identified the five levels of possible collaboration in the context of Ecology and Evolutionary Biology research, as follows:

Level	Degree of collaboration
1	Individual
2	Lab or group of collaborators
3	An organisation, or multiple labs
4	The research community
5	The general public

We determined the minimum and maximum degree of collaboration from these levels for each GitHub use

case.

Technical difficulty

We then assessed the technical difficulty of GitHub use cases in terms of their cognitive load.

As a note, we measured the perceived technical difficulty of each GitHub use case as the degree of difficulty as perceived by the authors, based on personal experience. This metric represents a consensus value among the authors, though the perceived degree of difficulty will ultimately vary among GitHub users depending on each person's previous experience with these use cases.

The table we used to quantify cognitive load was assembled collaboratively using Google Sheets, then manually saved as a `CSV` file called `cognitive_load_table.csv` in the `data` folder.

1. We first created a list of skills that are needed to implement different GitHub uses. These occupied a column each.
2. For each GitHub use listed on the first column, we went through each column, and marked a 1 if a skill was needed for that particular GitHub use, and 0 if the skill was not needed.
3. The sum of values across columns provides a measure of cognitive load required for a user that is a "GitHub beginner" to implement any GitHub use. The column *Cognitive load - real* reflects this number.
4. As many GitHub uses tied with the same cognitive load value, we resourced to perceived difficulty to break ties and avoid overlap of uses in the figure, mainly to improve visualization of the data. The column *Cognitive load - real - non-overlapping* shows this adjustment.
5. Finally, we ordered the GitHub uses in increasing order following *Cognitive load - real - non-overlapping*, and assigned a sequential value of cognitive load, from 1 to *maximum number of GitHub uses*. This is column *Cognitive load - sequential*

The collaborative Google Sheet used to quantify cognitive load is also available at the following link: https://docs.google.com/spreadsheets/d/1YipCCdg5Z8w-IQO67rFfMBSTpLjO-_7HiW44gqzTXPM/edit#gid=0.

Appendix S1.2. Annotated code and prose allowing the representation of indices of technical difficulty and degree of collaboration

Here, we provide the annotated `R` code and instructions that are required to load and present the indices of technical difficulty and degree of collaboration of the eleven use cases of GitHub we present in our piece.

We used Google Docs and Google Sheets to collect and manipulate data, and plotted results using the basic visualisation functions from `R` version 4.02.

Data collection

These data were assembled collaboratively using Google Docs following the methodology described in Appendix S1.1. then manually saved as a `CSV` file called `scatterblob_data_raw.csv` in the `data` folder. The collaborative Google Spreadsheet is also accessible at the following link: https://docs.google.com/spreadsheets/d/1zCRgctjfjE2xZN9MkqfFh222K6bvVvA3gT0UdkL_CS8/edit#gid=0.

The data table features a first column listing the ways to use GitHub that were addressed in the manuscript (Archiving citeable code and data; Storing a research compendium; Project management; Project continuity; Open scientific discussions; Virtual laboratory notebook; Educational materials; Academic websites; Writing a manuscript; Organizing and managing teams; Peer-reviews; Asynchronous collaborative coding and writing; Automation).

Creating Figure 2: A summary of ways GitHub can be used showing technical difficulty and degree of collaboration for each use case.

1. Loading the data `CSV` file:

```
scatterblob_data <- read_csv(here("data", "scatterblob_data_raw.csv"))
```

2. Cleaning the raw data and saving it as a `my_data` object:

```
# names(scatterblob_data)
scatterblob_data$x1 <- scatterblob_data$`Min degree of collaboration X1`
scatterblob_data$x2 <- scatterblob_data$`Max degree of collaboration X2`
scatterblob_data$y_real <- scatterblob_data$`Cognitive load - real - non-overlapping`
```

```
scatterblob_data$y_seq <- scatterblob_data$`Cognitive load - sequential`
my_data <- scatterblob_data[1:13,c("Ways to use GitHub","y_seq", "y_real", "x1", "x2")]
```

```
outpath <- here("content", "images")
```

3. Plotting cognitive load

We used the color palette "turbo" from the R package viridis:

```
my_colors <- viridis::turbo(n = 100, alpha = 0.9)[c(2, 5, 15, 26, 28, 30,
                            68, 72, 76, 80, 84, 94, 100)]
plot_name <- "scatterblob_1-viridis-turbo"

#################################################################
# start the png connection and set graphical parameters
#################################################################

plot_format <- "png"
if (plot_format == "png") {
  png(file = here(outpath, paste0(plot_name, ".png")),
      width = 8.25,
      height = 6,
      units = "in",
      res = 300,
      bg = "white")
}
par(xpd = NA,
    mai = c(1.02, 1, 0.82, 4),
    bty="n")

#################################################################
```

```r
# create the plot background
#####################################################################
plot(x = c(0.5, 5.5),
     y = c(0.5, 13.5),
     xlab = "",
     ylab = "",
     col = "white",
     # "hide" the tick labels so we can put some words instead of numbers:
     col.axis = "white",
     tck = 0.02)
title(ylab = "Technical Difficulty",
      line = 4,
      cex.lab = 1)
title(xlab = "Degree of Collaboration",
      line = 3.5,
      cex.lab = 1)

#####################################################################
# add tick labels
#####################################################################

text(x = 1:5,
     y = c(-0.65, rep(-0.7, 4)),
     cex = 0.7,
     labels = c("Personal",
                "Lab \nMembers",
                "Other \nLabs",
                "EEB \nCommunity",
                "All \nUsers"))
```

```
text(x = c(1, 3, 5),
     y = -1.7,
     labels = c("Low", "Medium", "High"))

# Technical difficulty levels from https://www.londonschool.com/level-scale/
text(x = 0.2,
     y = seq(1, 13, length.out = 5),
     cex = 0.7,
     adj = 1,
     labels = c("Beginner",
                "Low\nIntermediate",
                "Intermediate",
                "Pre \nAdvanced",
                "Advanced"))

#####################################################################
# plot scatter blobs on sequential cognitive load
#####################################################################
for (i in seq(nrow(my_data))) {
  X0 <- my_data$x1[i]
  X1 <- my_data$x2[i]
  Y <- my_data$y_seq[i]
  if (X0 == X1) {
    X0 <- X0 - 0.01
    X1 <- X1 + 0.01
   }
  segments(x0 = X0,
           x1 = X1,
           y0 = Y,
```

```
            col = my_colors[i],
            lwd = 23)
}


#####################################################################
# add github uses text
#####################################################################
text(x = 5.5,
    y = my_data$y_seq,
    labels = my_data$`Ways to use GitHub`,
    adj = 0,
    col = my_colors)
dev.off()
```

```
## pdf
##   2
```

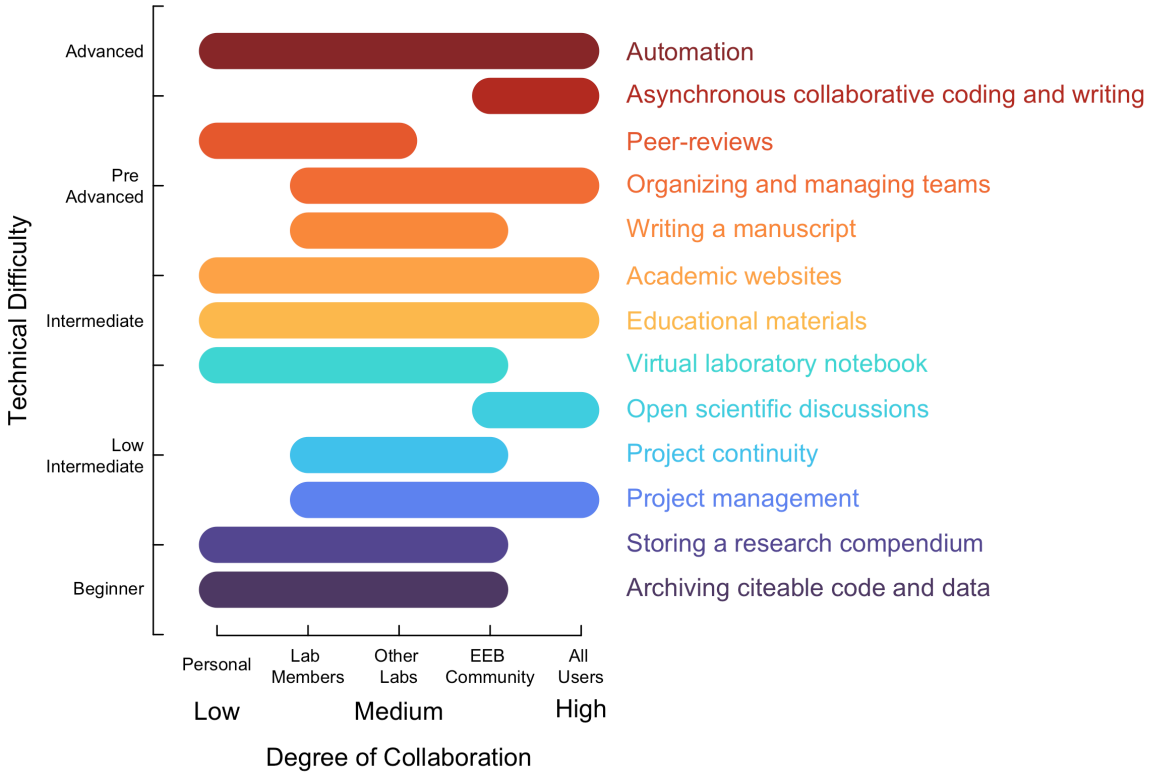

Figure 1: A summary of ways GitHub can be used showing technical difficulty and degree of collaboration for each use case.