Perspectives on Behavior Science (2019) 42:565–581
https://doi.org/10.1007/s40614-019-00202-5

ORIGINAL RESEARCH

Furthering Open Science in Behavior Analysis: An Introduction and Tutorial for Using GitHub in Research

Shawn P. Gilroy[1] · Brent A. Kaplan[2]

Published online: 24 May 2019

Abstract

Open and transparent practices in scholarly research are increasingly encouraged by academic journals and funding agencies. Various elements of behavior analytic research are communicated transparently, though it is not common practice to archive study materials to support future replications. This tutorial presents a review of the Transparent and Open Practices guidelines provided by the Open Science Foundation and provides instructions on how behavior analysts can use GitHub transparency in research across multiple levels. GitHub is presented as a service that can be used to publicly archive various elements of research and is uniquely suited to research that is technical, data driven, and collaborative. The GitHub platform is reviewed, and the steps necessary to create an account, initialize repositories, archive study files, and synchronize changes to remote repositories are described in several examples. Implications of increased calls for transparency and modern statistical methods are discussed with regard to behavior analysis, and archiving platforms such as GitHub are reviewed as one means of supporting transparent research.

Introduction

Calls for greater research transparency in the social sciences have increased over the years (Grahe, 2018; Open Science Collaboration, 2015). Among the forces that have driven this trend towards more open research, the "replication crisis" has prompted a reevaluation of various practices in scholarly research (Open Science Collaboration, 2015). The "replication crisis" refers to recent, unsuccessful attempts to replicate many of the statistically significant effects reported in hallmark psychological studies (Open

✉ Shawn P. Gilroy
sgilroy1@lsu.edu

[1] Louisiana State University, 226 Audubon Hall, Baton Rouge, LA 70802, USA

[2] Fralin Biomedical Research Institute at VTC, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

Science Collaboration, 2012, 2015). Among the factors believed to contribute to these inconsistencies, various forms of questionable research and publication practices have been implicated (Branch, 2014; Gelman & Loken, 2013; John, Loewenstein, & Prelec, 2012; Nosek, Spies, & Motyl, 2012).

Concerning the field of behavior analysis, some have suggested that single-case research designs (SCRDs) are less prone to certain issues surrounding failed replications (Branch, 2018). For example, SCRDs avoid issues surrounding statistical inferences, such as misinterpretations of p values (Branch, 2014; Nuijten, Hartgerink, van Assen, Epskamp, & Wicherts, 2016) and varying levels of statistical power (Kyonka, 2018). In addition, SCRDs traditionally include some form of within- or between-subject replication (Horner et al., 2005; Smith, 2012) and the effects described in SCRDs are reported alongside some evidence that an effect was replicable (Kazdin, 2011; Kratochwill et al., 2012). Moreover, SCRDs are traditionally accompanied by thoroughly documented parameters, such as response definitions, designs, and procedures (Smith, 2012; Wolery & Ezell, 1993) and specificity in these areas is thought to support later replications (Kazdin, 2011; Kratochwill et al., 2012). Furthermore, study data are embedded into manuscripts to support the visual determination of effects at a later time (Wolery & Dunlap, 2001).

Threats to Replicability in Current Behavior Analytic Research

Although many qualities of SCRDs lend themselves to replicability, this is not to say that single-case research practices are immune to issues related to replicability (Hantula, 2019; Tincani & Travers, 2019). Within the present academic culture, both mainstream and single-case researchers are exposed to contingencies that can lead to questionable research practices (QRPs), e.g., “grant culture” (Hantula, 2019; Lilienfeld, 2017). That is, pressures to publish in prestigious journals and obtain/maintain funding contributes to the presence of QRPs across disciplines and methodologies (John et al., 2012; Tincani & Travers, 2019).

Regarding publication practices, a recent study by Shadish, Zelinsky, Vevea, and Kratochwill (2016) found that participants reviewing data from SCRD studies in behavior analysis were likely to recommend against the publication of studies with small or null effects. Tincani and Travers (2017) discussed this issue further and suggested that expectations regarding strong demonstrations of experimental control (i.e., moderate–large effects) towards publication may contribute to a “file drawer” effect for SCRD studies with negative, null, or small effects. Pressures to publish in prestigious behavior analytic journals and demonstrate large effects can set the occasion for QRPs in SCRDs as well. For example, Shadish et al. (2016) found that a subset of single-case researchers would omit participant data from their studies if that data were indicative of small effects. As such, phenomena such as publication bias can and do exist even in the absence of group design methodologies and statistical inference (Sham & Smith, 2014).

Threats to Replicability in Emerging Behavior Analytic Research

Although SCRDs will undoubtedly remain a defining feature of behavior analytic research and practice, statistical methods and practices have established a growing presence in behavior analytic research (Fisher & Lerman, 2014; Young, 2018; Zimmermann, Watkins, & Poling, 2015). The exploration of novel methods and approaches offers new possibilities to behavior analysts but also presents new challenges (Shull, 1999). For example, whereas many practices regarding SCRDs have traditionally been well-defined (Kratochwill et al., 2012), practices regarding the reporting of statistical methods historically have not. For example, study elements such as analyses (e.g., syntax), simulations (e.g., source code), and data may not be available to support reanalysis or replication. As behavior analysis ventures towards increased adoption of statistical practices and mainstream research designs, behavior analytic researchers must become aware of, and adapt to, the challenges associated with transparently conducting this manner of research (Nosek et al., 2015; Open Science Collaboration, 2015).

Transparency and Openness Standards for Behavior Analytic Research

Recent trends towards "open science" have focused on increasing the transparency of research (Open Science Collaboration, 2012, 2015). High levels of openness and transparency have been suggested as one of several ways that the academic community might curb various forms of QRPs (Branch, 2014; John et al., 2012; Nosek et al., 2015; Open Science Collaboration, 2012). Although many have made similar calls for transparency, Nosek et al. (2015) provided a set of suggested guidelines for Transparency and Openness Promotion (TOP) and are listed in Table 1. These TOP guidelines establish standards for maintaining transparency in citation, data, analyses, research materials, designs, preregistrations of study designs and methods, and support for replications.

Regarding practices in behavior analytic research, certain standards are currently well-represented whereas others are less so. For instance, standards for citation are well-established in behavior analytic research and procedures derived from earlier works (e.g., functional analysis, stimulus choice assessments) are typically well-

Table 1 Transparency and openness guidelines

Guideline	Standards
1. Citation Standards	Data and methods used in a study are sufficiently cited.
2. Data transparency	Data are publicly archived and available for inspection.
3. Analytic methods transparency	Code/scripts are publicly archived and available for inspection.
4. Research methods transparency	Study protocols and relevant materials are publicly archived and available for inspection.
5. Design an analysis transparency	Standards are established and adhered to regarding study designs and expectations for publication.
6. Preregistration of studies	Studies are preregistered with time-stamped protocols available for public inspection.
7. Preregistration of analysis plans	Study analyses are preregistered with timestamped analytical plans available for public inspection.
8. Replication	The publication of replication studies is encouraged.

documented. Likewise, standards for acceptable research designs are also well-documented (e.g., multiple baseline, reversal). However, practices related to the public archiving of study data and materials are less represented in behavior analytic research. That is, the public archiving of study data, analyses, and materials are infrequently observed (e.g., time-stamped behavior data, interobserver agreement). From a replicability standpoint, the analytical reproducibility of a published work is limited without access to these elements (Klein et al., 2018; Wasserstein & Lazar, 2016). Further, few behavior analytic studies have used preregistration of the study methods and analytical plans. Although more often observed in mainstream studies and clinical trials, preregistration would offer similar benefits to behavior analytic research as well, as preregistration serves to better distinguish research that is confirmatory or hypothesis testing from research that is exploratory or hypothesis generating (Nosek et al., 2015). In addition, preregistering study parameters and methods may provide a further safeguard against QRPs more specific to SCRDs (e.g., criteria for dropping/excluding participants, criteria for advancing in treatment). Lastly, the remaining guideline speaks to the publication of direct replications. Although relatively new to behavior analytic publication practices, recent policies have been introduced to encourage the direct replication of behavior analytic studies (Hanley, 2017).

Archiving Behavior Analytic Research in Public Repositories

Regarding the TOP guidelines, most standards recommend archiving elements of research to a public repository (e.g., data, materials, analyses). Publicly archiving such resources has historically required some level of technical proficiency on the part of the research team (i.e., hosting a personal web service), but a variety of user-friendly options are now available for free or at a low cost (Soderberg, 2018). For example, the open science framework (OSF; https://www.osf.io) offers a range of research-related features such as preregistration and public archiving. Further, a range of newer options has also been developed to provide similar functionality with extended features (e.g., Code Ocean, Harvard Dataverse, Mendeley Data, DataNET, GitHub).

Each system for archiving functions as a form of *version control*. Put simply, version control is a mechanism whereby iterations of information (e.g., study data) are stored and retained for future inspection. Each iteration represents a change made by a researcher at some point in time. In terms of research, study materials or data archived in a repository might undergo various changes (e.g., removing a participant, editing materials) and these changes would be accompanied by documentation, such as the person who made the change, when, and for what reasons. For elements of a study, the full history of changes can be retained, and these can be inspected by reviewers or readers at any future time. Even further, these materials may also be refined following publication (e.g., corrections). The availability to document these changes along with time stamps is desirable because it allows the reader to view the historical record of the research over time (e.g., preregistration, recruitment, results). Readily accessible examples of version control can be found on the Wikipedia platform. For example, a reader can view the history of the content including what areas were changed.[1]

[1] The history of the Wikipedia entry related to "Perspectives on Behavior Science" is available at https://en.wikipedia.org/w/index.php?title=Perspectives_on_Behavior_Science&action=history

Version control systems such as those provided by the OSF (www.osf.io) are robust and suited to archiving many types of research files (e.g., word processing, spreadsheets), though more specialized systems exist for larger and more complicated projects. For example, projects including complex data, analyses, and source code may require a system that documents contributions from many researchers over time. In a Git-based approach to version control (e.g., GitHub, GitLab), collaborators "commit" changes to repositories that are reflected in the project's historical record. That is, Git-based version control tracks who modifies files, when those changes occur, and document the reasons why changes were made. Aside from high levels of transparency, Git-based approaches to version control are particularly useful for projects that involve many contributors. For example, projects such as the Linux kernel (Torvalds, 2018) have involved over 15,000 contributors and over 200 companies (Corbet & Kroah-Hartman, 2017).

In addition to systems that focus on publicly archiving binary files (e.g., spreadsheets, word documents), such as those OSF, Git-based archiving is especially suited to documenting individual changes in text files (e.g., statistical syntax, source code). That is, these systems mark the specific areas and content that was changed. This is desirable in the context of source code and statistical syntax, as very minor changes in text can dramatically affect results and their interpretation. As such, Git-based approaches to version control offer a robust archiving tool that is equally applicable to both current and future behavior analytic research. In particular, a Git-based approach to archiving is amenable to single-case research materials, statistical syntax associated with the adoption of more advanced statistics (Young, 2018), and the development and evaluation of behavior analytic software (Bullock, Fisher, & Hagopian, 2017; Gilroy, Franck, & Hantula, 2017; Gilroy, Kaplan, Reed, Koffarnus, & Hantula, 2018a; Kaplan, Gilroy, Reed, Koffarnus, & Hursh, 2018).

GitHub and Git-Based Version Control

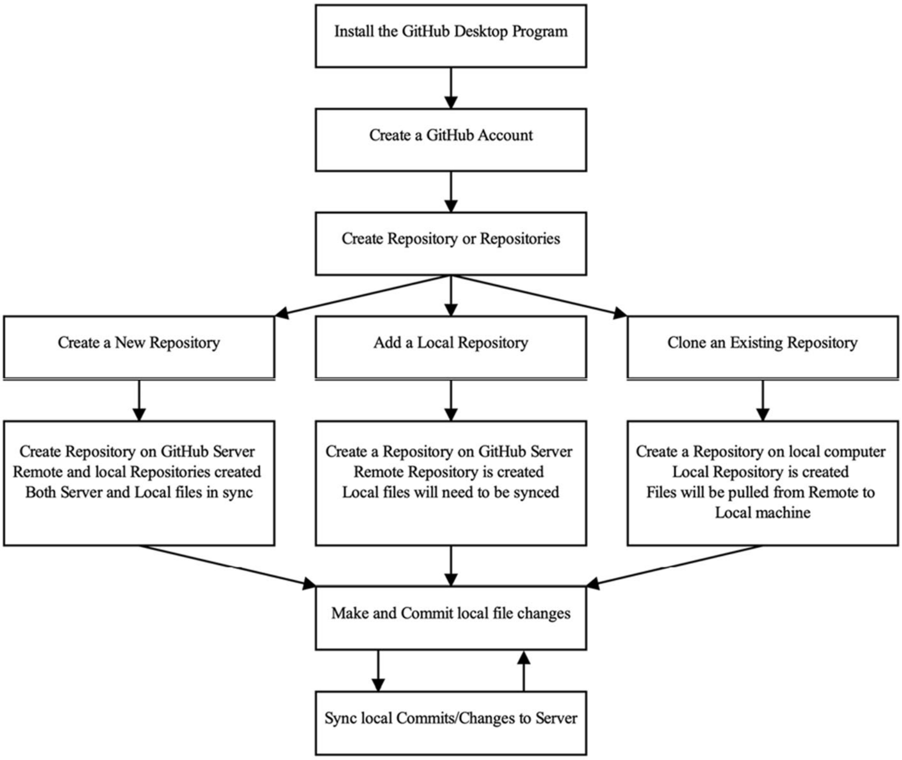
Notwithstanding the various services that use the Git protocol for version control, this tutorial focuses exclusively on GitHub (https://www.github.com). GitHub is a service that has been used extensively to archive and manage a variety of projects. At the time of this writing, GitHub maintains over 33 million public repositories and services 28 million users.[2] Regarding applicability to behavior analysts, GitHub is particularly useful because it provides options for those with varying levels of proficiency with computer programming. That is, users capable of contributing to the Linux kernel (i.e., central core of an open source operating system) would simply open a terminal (i.e., a command-line interface with the computer) and interact with the Git protocol directly (i.e., "git clone https://github.com/torvalds/linux"). As an alternative, GitHub offers interfaces for those who may not have the time to (or even wish to) learn how to work via command-line interfaces.

Among the interfaces available, GitHub provides a web-based portal wherein users may directly edit the repository (e.g., change text, drag-and-drop files). Likewise, GitHub also provides an installable program that provides the same

[2] Numbers provided from https://github.com/search?q=is:public as of September 1, 2018.

functionality, and a bit more. Both options offer a Graphical User Interface (GUI) for users who prefer a visual interface over the command line. For these reasons and others, GitHub is well-suited to behavior analysts given varying levels of technological proficiency. Regardless of the preferred interface, a general flow chart of the Git-based workflow is illustrated in Figure 1.

To illustrate the robust features and general applicability of GitHub to behavior analysts, several Use Cases are presented. Three Use Cases are presented to demonstrate broad applicability to behavior analytic research. Use Case #1 presents the most basic use of GitHub, focusing on publicly archiving files with the desktop program. This is likely the most generally applicable usage for behavior analysts. Use Case #2 builds upon Use Case #1 but focuses on pulling files from public archives. Although less generally applicable, this usage is more valuable to those who aim to replicate some aspect of a previously archived study. Finally, Use Case #3 builds upon the earlier cases to demonstrate how GitHub can be used to pull publicly archived files and, with the permission of the original repository owner, contribute changes to files. Each of these Use Cases is described in detail below.


Install the GitHub Desktop Program
Create a GitHub Account
Create Repository or Repositories
Create a New Repository
Add a Local Repository
Clone an Existing Repository
Create Repository on GitHub Server
Remote and local Repositories created
Both Server and Local files in sync
Create a Repository on GitHub Server
Remote Repository is created
Local files will need to be synced
Create a Repository on local computer
Local Repository is created
Files will be pulled from Remote to
Local machine
Make and Commit local file changes
Sync local Commits/Changes to Server

This is a visual depiction of the general workflow for version control using the Git protocol and GitHub.

Figure 1 Flow chart for git

Use Case #1: Using GitHub to Archive Study Files

This demonstration is focused on using GitHub to archive study files with the GitHub Desktop program. The process of using this program begins by downloading the software from the GitHub website.[3] Installation binaries are provided for both the Windows™ and macOS™ operating systems and users should download the necessary file for their system. Following installation, users input their GitHub credentials into the computer program once it starts. If the user does not have credentials, the program will assist the user in creating an account. For users with a fresh installation, the GitHub Desktop will have no *local* repositories (i.e., repositories on the current machine). Following the workflow identified in Figure 1, the next step for the user is to either *create* a new repository, *add* a local repository, or *clone* a repository existing somewhere on the internet. Our focus here is on creating a new repository—a folder that contains all the files that will be publicly archived. The global workflow for this Use Case is depicted in Figure 2.

Creating a GitHub Repository

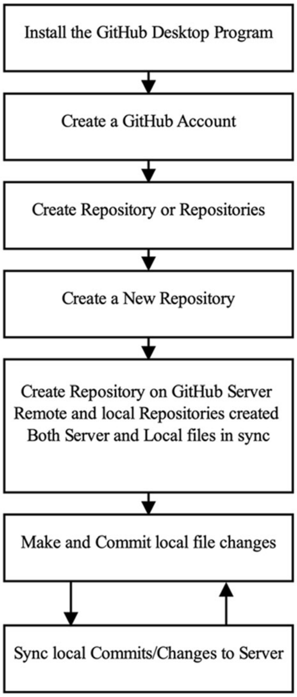
The process of creating a new local and remote repository with GitHub Desktop begins by selecting a directory (i.e., folder) on the user's local machine and instructing the GitHub Desktop program to use Git to place this location under version control.[4] With the assistance of the program, the user specifies a directory (e.g., the desktop) and a name for the repository (e.g., "use-case-1"). Although not mandatory, but highly recommended, most users initialize their repository with a README.md file and this file functions like a homepage for those viewing the repository online. Although not necessary at this point, a complete README.md file serves to orient others to the repository and to the contents contained within. For example, this file could contain contact information for the study authors, links to the published manuscript, and other information that might orient the reader (e.g., abstract text, files in the repository). Likewise, researchers are advised to include some documentation regarding how they wish to license their work. Discussions of specific licenses, and which to use, is beyond the scope of this tutorial and the reader is directed to the https://choosealicense.com website. This resource is hosted and curated by the GitHub development team. Regardless of the license type selected, the GitHub Desktop program assists the reader by including the LICENSE file that corresponds with their selection. Once selecting from these options, the users clicks "Create Repository" to create the repository and its contents (e.g., LICENSE, README.md).

Committing Research Files to Repository

Once repositories are created, files placed within the repository are placed under version control and *tracked*. The tracking of files functions like the "Track Changes"

[3] The current location for downloading this program is https://desktop.Github.com

[4] We note here that a ".gitignore" file can be used to determine which files are tracked and which are ignored. Intermediate build objects and other binaries are traditionally not tracked because such objects are unlikely to be useful on other machines and are not tracked for this reason.


Install the GitHub Desktop Program
Create a GitHub Account
Create Repository or Repositories
Create a New Repository
Create Repository on GitHub Server
Remote and local Repositories created
Both Server and Local files in sync
Make and Commit local file changes
Sync local Commits/Changes to Server

This illustrates how the GitHub Desktop program can be used to commit and publicly archive files to a remote repository on the internet.

Figure 2 Workflow for use case #1

functionality offered in word processing software. That is, changes within the repository are recorded and these changes can be *committed* (i.e., written into the history of the repository). Each of the changes committed to the repository is accompanied by documentation summarizing the changes, when those changes were committed, and who committed those changes. For example, if the reader opened the README.md file with a text editor of choice (e.g., Notepad on Windows, TextEdit on macOS) and appended the text “This is new text I've added” to the file, the GitHub Desktop program would indicate that there were changes made to the file.

The GitHub Desktop program displays information regarding content that has been added to files (illustrated in green with a “+”) as well as content that has been deleted (depicted in red with a “-”). Although appending text to a file is a trivial example, researchers could instead use this function commit various study elements to the repository. For example, users could place their study data, materials, statistical syntax, and any other information relevant to the study that might support future replication. Regardless of the specific changes made in the repository, the user would then commit these changes to the local repository. To do so using the GitHub Desktop program, users would select the change(s) they desire to commit and then provide a short summary of the change (i.e., “this is our first commit”). Users may, at their preference, provide a more thorough explanation of the commit in the “Description” area but this is not required. Once the necessary fields are populated, users commit these changes to the repository by pressing the “Commit to master” button.

Pushing Local Changes to Remote

Once changes are committed to a user's local repository, the sequence of changes are depicted in the "History" panel of the GitHub Desktop program. For example, data and study materials would be committed to the repository along with a permanent timestamp. This permits a public inspection of the full breadth of changes that have occurred within a repository, including individual files, since the creation of the repository. Each commit is accompanied by information regarding what changes were introduced, who committed those changes, when the committed took place, and a description of why files were changed.

At present, the changes exist only the local machine and this information is not yet synced with the *remote* repository (i.e., GitHub servers). The GitHub Desktop program makes this process painless—the user simply presses the "Push origin" button to *push* commits to the remote repository. Once this process begins, changes that have not yet been pushed to the remote repository on the internet will be pushed and the local and remote repositories will be synced. Once completed, all files will be publicly archived within the user's remote repository on the GitHub website (i.e., https://www.github.com/[username]/use-case-1).[5]

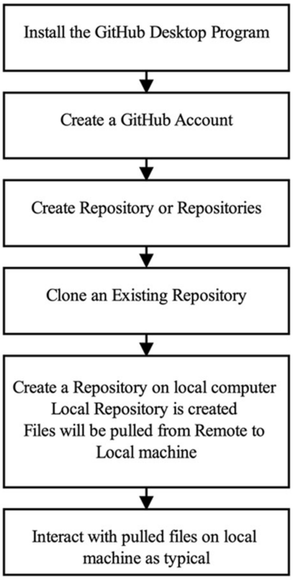
Use Case #2: Using GitHub to Retrieve Published Study Materials

In Use Case #2, this example illustrates how users would use GitHub to obtain information publicly archived in a GitHub repository. Use Case #1 consists of pushing files to a remote repository whereas Use Case #2 entails *pulling* files from a remote repository into a local one. Use Case #1 is most applicable to researchers publicly archiving their study materials, whereas Use Case #2 is most relevant to researchers who desire to replicate or examine archived study materials. This act of pulling a remote repository and creating a local, personal copy is referred to as *cloning*. Our focus in this example is on obtaining publicly archived files and the global workflow for this Use Case is depicted in Figure 3.

Cloning Existing GitHub Repositories

Like Use Case #1, the GitHub Desktop program can be used to clone remote repositories on GitHub. To create a repository, the user begins by pressing the "File" dropdown and selecting "Clone repository." Once selected, the GitHub Desktop program offers several options for identifying which repository to clone (e.g., personal remote repositories, repositories from an organization, URLs). Once a repository of interest is selected, the program asks the user to specify where to create their local clone of the repository. Once these operations are performed, the user has a local clone of the remote repository. At this point, users may freely interact with local files and commit changes to their local clone of the repository.

[5] We note that GitHub users may push changes to private repositories, though the primary focus of this tutorial is on the public archiving of files. Repositories that are kept private can be made public later (e.g., after submission for publication).


Install the GitHub Desktop Program
Create a GitHub Account
Create Repository or Repositories
Clone an Existing Repository
Create a Repository on local computer
Local Repository is created
Files will be pulled from Remote to
Local machine
Interact with pulled files on local
machine as typical

This illustrates how the GitHub Desktop program can be used to retrieve and access previously published files from a remote repository on the internet.

Figure 3 Workflow for use case #2

Retrieving Study Materials: Gilroy, McCleery, and Leader (2018b)

In situations where researchers are attempting to replicate findings from earlier studies, it may become necessary to use the exact materials from an original publication. GitHub makes these materials easy to maintain and supports replication of applied studies. As an example of how GitHub repositories can be used to support the replication of applied studies, published materials were cloned from Gilroy, McCleery et al. (2018). Briefly, Gilroy, McCleery et al. (2018) conducted a randomized-control comparison of high- and low-tech forms of Augmentative and Alternative Communication (AAC) in children diagnosed with Autism Spectrum Disorder (ASD). That is, the intervention consisted of a specially designed mobile application that children operated using handheld devices. The authors publicly archived their software in a GitHub repository to allow future researchers the opportunity to inspect the software as well as replicate study effects.[6]

Using the GitHub Desktop program, users can clone this repository by using the "File" dropdown, selecting "Clone a repository," and inserting the relevant text into the "URL" dialogue (i.e., "miyamot0/FastTalkerSkiaSharp"). Once a local path is selected for the local repository, the user would press the "Clone" button to retrieve a copy of all relevant source code to build this mobile application. Once completed,

[6] The GitHub repository for the AAC application is available at https://www.github.com/miyamot0/FastTalkerSkiaSharp

files in the local repository include all the relevant assets (i.e., images) and source code to create the software that was used with children with ASD in this study. However, although it is easy to simply clone the files in this repository, the user's local machine must be configured to build and deploy the application before it could be used as normal.

Retrieving Study Data: Gilroy, Kaplan, Reed, Hantula, and Hursh (2019)

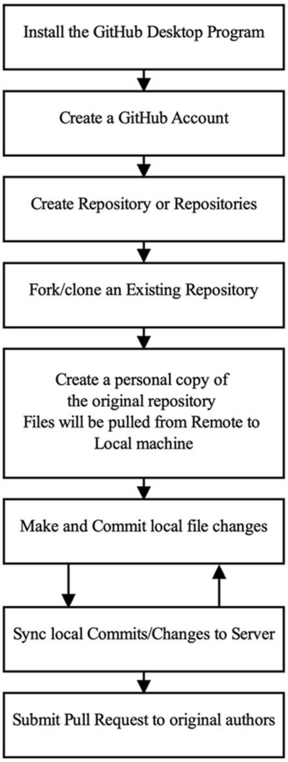
As a step beyond retrieving study materials, researchers may wish to revisit specific analyses and data from peer-reviewed studies. The capability of revisiting these findings speaks to the analytical replicability of published studies (Klein et al., 2018; Wasserstein & Lazar, 2016). As an example of this, study data and analysis from a published study were retrieved from Gilroy et al. (2019). Briefly, Gilroy et al. evaluated a novel method for evaluating unit elasticity in models of operant demand using computer simulation. Within the published manuscript, a link to the corresponding GitHub repository was provided to allow reviewers and other researchers to inspect the source code, view the nature of the simulated data, and how the novel method was performed.[7]

Using the GitHub Desktop program, researchers would clone this repository by using the "File" dropdown, selecting "Clone a repository," and highlighting the "URL" tab. In the URL entry field, the location of the GitHub repository takes the form of the username and the repository separated by a forward slash (e.g., "miyamot0/PmaxEvaluation"). Once a local path is selected for the local repository, the user would press the "Clone" button to retrieve a copy of all files necessary to generate simulated data and repeat study analyses using the open source R Statistical Program (R Core Team, 2017). As noted above, the user's local machine must be configured to run the R Statistical Program with all necessary packages before the scripts can be used to simulate study data, perform study analyses, and construct study figures and tables.

Use Case #3: Using GitHub to Collaborate with Researchers

Whereas Use Cases #1 and #2 provide examples of pushing and pulling files, the capabilities of GitHub extend much further. As noted earlier, many GitHub projects are complex, grow over time, and incorporate changes from many collaborators. Use Case #3 provides an example relevant to researchers seeking to replicate and expand upon earlier research. That is, researchers may use GitHub as an intermediary to collaborate with other researchers. This act of pulling a remote repository and creating a personal clone that can share changes back to the original is referred to as *forking*. The purpose of this example is to obtain publicly archived files, commit changes, and share those changes with others. The global workflow for this Use Case is depicted in Figure 4.

[7] The GitHub repository for the novel behavioral economic calculation is available at https://www.github.com/miyamot0/PmaxEvaluation


Install the GitHub Desktop Program
Create a GitHub Account
Create Repository or Repositories
Fork/clone an Existing Repository
Create a personal copy of the original repository Files will be pulled from Remote to Local machine
Make and Commit local file changes
Sync local Commits/Changes to Server
Submit Pull Request to original authors

This illustrates how the Git protocol can be used to fork an existing repository, commit changes, sync these changes to the remote repository, and submit these changes to the original authors for potential inclusion in the original repository.

Figure 4 Workflow for use case #3

Forking Existing Repositories

Whereas the examples provided in Use Case #2 focused on cloning personal repositories, Use Case #3 illustrates how to create and use a *fork* of an existing repository. As noted above, a forked repository is a cloned repository that can feed committed changes *upstream* to the original repository. That is, researchers in a related area may follow the work of a lab and use GitHub to contribute to its projects. This function allows researchers and other collaborators to interact with materials used by others, and, with permission of the original project owners, contribute those changes back to the original repositories. The collaborative nature of open source and open science has given rise to a number of projects that would not have been possible by a single team, such as the Linux kernel (Corbet & Kroah-Hartman, 2017; Torvalds, 2018).

The forking of a repository is most easily performed using the GitHub website. In general, this operation is performed within the web browser, because this is the most natural way to view the repositories of individual users. Users logged into the GitHub may create a fork of any public repository by pressing the “Fork” button located in the upper portion of the GitHub website. Once this operation is complete, users have a fork of the remote repository among their personal repositories. This personal copy serves as a working repository, with changes committed here remaining in this repository unless submitted to the original repository as a *pull request*.

Extending Published Studies: Kaplan et al. (2018)

In situations where researchers may wish to systematically replicate and extend methods from earlier studies, researchers may wish to incorporate their findings into existing open source projects. GitHub makes this type of operation easy to perform and supports the collaborative nature of research. As an example of how GitHub repositories can be used to support research collaboration, Use Case #3 focuses on creating a fork of open source software developed by Kaplan et al. (2018) and submitting a pull request to that repository. In brief, Kaplan et al. (2018) developed an open-source statistical package (*beezdemand*) to aid researchers in applying behavioral economic methods using the R Statistical Program (R Core Team, 2017). Although robust and possessing broad functionality, the features provided by *beezdemand* are not exhaustive and on-going development is necessary to ensure that features are kept consistent with current research needs. That is, recent contributions such as those from Gilroy et al. (2019; see Use Case #2) could enhance the utility of this tool.[8]

To contribute to *beezdemand* repository, the user would navigate to the *beezdemand* repository in their browser and press the "fork" button. Once the fork operation is completed, the personal repository may be cloned using the GitHub Desktop program[9] as performed in Use Case #2. This would create a clone of the files necessary to build the *beezdemand* package. At this point, the user would make changes to the necessary files (i.e., adding new methods) and commit these changes to our local repository. Changes would be committed and pushed to the personal remote repository, as performed in Use Case #1. Visiting the personal repository on the GitHub website, the user would then create a pull request to query the original authors and, at their discretion, incorporate those contributions in the original project. If accepted, the original repository retains a record who contributed to the project, what changes were made, and when those were made.

Discussion

The purpose of this tutorial was to review how GitHub can be used to support open and transparent research in behavior analysis. Although relatively less represented in behavior analytic research, many mainstream scholarly outlets (Nuijten et al., 2017) and funders (Houtkoop et al., 2018) are increasingly requiring more transparency in research. This tutorial provided a review of how behavior analysts can leverage the capabilities of the free GitHub service to archive their study materials (e.g., analyses, raw data, preprints), support replicability, and collaborate with other researchers.

[8] We note here that Gilroy et al. (2019) validated a novel calculation of unit elasticity for models of operant demand.

[9] Given nature of this particular example, it is likely that users of this level of proficiency would more likely perform this operation via the command line using "git clone https://github.com/[username]/beezdemand.git."

Although behavior analysts have long had the option to archive elements of their peer-reviewed studies as online supplemental materials, this approach is limited in comparison to the GitHub platform for several reasons. First, platforms such as GitHub can provide greater visibility to individual researchers and their work. For example, repositories can be assigned individualized digital object identifiers (DOIs) like other peer-reviewed articles. Outlets such as the *Journal of Open Source Software* regularly provide DOIs for peer-reviewed articles as well as archived software. The ability to mint DOIs for published works as well as supplemental content serves to increase the visibility of published works. This is likely to become increasingly relevant as behavior analysts publish data sets and other study elements. Second, the GitHub platform supports collaborative and iterative science. For example, novel methods developed by behavior analysts may be updated and refined over time from publication to publication (Gilroy et al., 2017; Gilroy & Hantula, 2018). Simply attaching files as supplemental materials in a publication would not provide information regarding changes to individual files, when these changes were performed, and who contributed to those changes. Third, publicly archiving study materials on GitHub provides a means for researchers to contact and interact with the repository owners through GitHub profiles. This feature supports communication with repository owners and allows the community to interact with researchers who gathered study data, wrote code, and performed the analyses. In short, the community would be able to consult directly with those most familiar with these resources (Lo & Demets, 2016). Lastly, GitHub provides opportunities for research collaborations. Open source collaborations typically occur regardless of geographical location, nationality, or affiliation, and this level of visibility opens new opportunities for behavior analysts. Indeed, the GitHub platform has already facilitated international behavior analytic collaborations in the areas of augmentative and alternative communication (Gilroy, 2016; Gilroy, McCleery et al., 2018) as well as in applied behavioral economics (Gilroy et al., 2017; Kaplan et al., 2018).

Limitations

Whereas open science and transparency are noble and have merit in behavior analysis, these goals present several new challenges to the behavior analytic community. First and foremost, there are concerns that adopting open standards would place additional burdens on both researchers and editors alike (Tenopir et al., 2011). That is, introducing additional requirements would place even greater demands upon reviewers and editors already tasked with many responsibilities. This concern may be at least partially mitigated by accessible tutorials such as this and other online resources (e.g., https://guides.github.com, https://www.youtube.com/githubguides, https://stackoverflow.com/questions/tagged/github). In addition, individual researchers may have concerns that openly sharing their data may expose their careers to professional risks (Laine, 2017). That is, there is the possibility that other teams may "scoop" potential findings that otherwise may have been discovered or published by the original study authors. As noted earlier, GitHub repositories can be made private or by invitation only. During the peer-review process or once accepted for publication, users can choose to make their repositories public at that time. However, each of these limitations warrants careful consideration in light of the potential challenges related to replicability in behavior analytic research.

References

Branch, M. N. (2014). Malignant side effects of null-hypothesis significance testing. *Theory & Psychology, 24*(2), 256–277. https://doi.org/10.1177/0959354314525282.

Branch, M. N. (2018). The reproducibility crisis: Might the methods used frequently in behavior-analysis research help? *Perspectives on Behavior Science*. https://doi.org/10.1007/s40614-018-0158-5.

Bullock, C. E., Fisher, W. W., & Hagopian, L. P. (2017). Description and validation of a computerized behavioral data program: BDataPro. *The Behavior Analyst, 40*(1), 275–285. https://doi.org/10.1007/s40614-016-0079-0.

Corbet, J., & Kroah-Hartman, G. (2017). 2017 Linux Kernel development report. *Linux Foundation.* Retrieved from https://www.linuxfoundation.org/2017-linux-kernel-report-landing-page/.

Fisher, W. W., & Lerman, D. C. (2014). It has been said that, There are three degrees of falsehoods: Lies, damn lies, and statistics. *Journal of School Psychology, 52*(2), 243–248. https://doi.org/10.1016/j.jsp.2014.01.001.

Gelman, A., & Loken, E. (2013). The garden of forking paths: Why multiple comparisons can be a problem, even when there is no “fishing expedition” or “p-hacking” and the research hypothesis was posited ahead of time. Department of Statistics, Columbia University.

Gilroy, S. P. (2016). FastTalker: Cross-platform AAC application. *GitHub Repository*. Retrieved from https://github.com/miyamot0/FastTalkerSkiaSharp.

Gilroy, S. P., Franck, C. T., & Hantula, D. A. (2017). The discounting model selector: Statistical software for delay discounting applications. *Journal of the Experimental Analysis of Behavior, 107*(3), 388–401. https://doi.org/10.1002/jeab.257.

Gilroy, S. P., & Hantula, D. A. (2018). Discounting model selection with area-based measures: A case for numerical integration. *Journal of the Experimental Analysis of Behavior, 109*(2), 433–449. https://doi.org/10.1002/jeab.318.

Gilroy, S. P., Kaplan, B. A., Reed, D. D., Hantula, D. A., & Hursh, S. R. (2019). An exact solution for unit elasticity in the exponential model of operant demand. *Experimental & Clinical Psychopharmacology.* https://doi.org/10.1037/pha0000268.

Gilroy, S. P., Kaplan, B. A., Reed, D. D., Koffarnus, M. N., & Hantula, D. (2018a). The Demand Curve Analyzer: Behavioral economic software for applied researchers. *Journal of the Experimental Analysis of Behavior.*

Gilroy, S. P., McCleery, J. P., & Leader, G. (2018b). A community-based randomized-controlled trial of speech generating devices and the Picture Exchange Communication System for children diagnosed with autism spectrum disorder. *Autism Research.* https://doi.org/10.1002/aur.2025.

Grahe, J. (2018). Another step towards scientific transparency: Requiring research materials for publication. *Journal of Social Psychology, 158*(1), 1–6. https://doi.org/10.1080/00224545.2018.1416272.

Hanley, G. P. (2017). Editor's note. *Journal of Applied Behavior Analysis, 50*(1), 3–7. https://doi.org/10.1002/jaba.366.

Hantula, D. A. (2019). Editorial: Replication and reliability in behavior science and behavior analysis: A call for a conversation. *Perspectives on Behavior Science.* https://doi.org/10.1007/s40614-019-00194-2.

Horner, R. H., Carr, E. G., Halle, J., McGee, G., Odom, S., & Wolery, M. (2005). The use of single-subject research to identify evidence-based practice in special education. *Exceptional Children, 71*(2), 165–179. https://doi.org/10.1177/001440290507100203.

Houtkoop, B. L., Chambers, C., Macleod, M., Bishop, D. V. M., Nichols, T. E., & Wagenmakers, E.-J. (2018). Data sharing in psychology: A survey on barriers and preconditions. *Advances in Methods & Practices in Psychological Science, 1*(1), 70–85. https://doi.org/10.1177/2515245917751886.

John, L. K., Loewenstein, G., & Prelec, D. (2012). Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science, 23*(5), 524–532. https://doi.org/10.1177/0956797611430953.

Kaplan, B. A., Gilroy, S. P., Reed, D. D., Koffarnus, M. N., & Hursh, S. R. (2018). The R package beezdemand: Behavioral Economic Easy Demand. *Perspectives on Behavior Science.* https://doi.org/10.1007/s40614-018-00187-7.

Kazdin, A. E. (2011). *Single-case research designs: Methods for clinical and applied settings*. Oxford, UK: Oxford University Press.

Klein, O., Hardwicke, T. E., Aust, F., Breuer, J., Danielsson, H., Hofelich Mohr, A., . . . Frank, M. C. (2018). A practical guide for transparency in psychological science. *Collabra: Psychology, 4*(1). doi:https://doi.org/10.1525/collabra.158.

Kratochwill, T. R., Hitchcock, J. H., Horner, R. H., Levin, J. R., Odom, S. L., Rindskopf, D. M., & Shadish, W. R. (2012). Single-case intervention research design standards. *Remedial & Special Education, 34*(1), 26–38. https://doi.org/10.1177/0741932512452794.

Kyonka, E. G. E. (2018). Tutorial: Small-N power analysis. *Perspectives on Behavior Science.* https://doi.org/10.1007/s40614-018-0167-4.

Laine, H. (2017). Afraid of scooping: Case study on researcher strategies against fear of scooping in the context of open science. *Data Science Journal, 16*. doi:https://doi.org/10.5334/dsj-2017-029.

Lilienfeld, S. O. (2017). Psychology's replication crisis and the grant culture: Righting the ship. *Perspectives on Psychological Science, 12*(4), 660–664. https://doi.org/10.1177/1745691616687745.

Lo, B., & Demets, D. L. (2016). Incentives for clinical trialists to share data. *New England Journal of Medicine, 375*(12), 1112–1115. https://doi.org/10.1056/NEJMp1608351.

Nosek, B. A., Alter, G., Banks, G. C., Borsboom, D., Bowman, S. D., Breckler, S. J., et al. (2015). Promoting an open research culture. *Science, 348*(6242), 1422–1425. https://doi.org/10.1126/science.aab2374.

Nosek, B. A., Spies, J. R., & Motyl, M. (2012). Scientific utopia: II. Restructuring incentives and practices to promote truth over publishability. *Perspectives on Psychological Science, 7*(6), 615–631. https://doi.org/10.1177/1745691612459058.

Nuijten, M. B., Borghuis, J., Veldkamp, C. L. S., Dominguez-Alvarez, L., Van Assen, M. A. L. M., & Wicherts, J. M. (2017). Journal data sharing policies and statistical reporting inconsistencies in psychology. *Collabra: Psychology, 3*(1). https://doi.org/10.1525/collabra.102.

Nuijten, M. B., Hartgerink, C. H., van Assen, M. A., Epskamp, S., & Wicherts, J. M. (2016). The prevalence of statistical reporting errors in psychology (1985–2013). *Behavior Research Methods, 48*(4), 1205–1226.

Open Science Collaboration. (2012). An open, large-scale, collaborative effort to estimate the reproducibility of psychological science. *Perspectives on Psychological Science, 7*(6), 657–660. https://doi.org/10.1177/1745691612462588.

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science, 349*(6251), aac4716. https://doi.org/10.1126/science.aac4716.

R Core Team. (2017). R: A language and environment for statistical computing (Version 3.4.1). R Foundation for Statistical Computing.

Shadish, W. R., Zelinsky, N. A., Vevea, J. L., & Kratochwill, T. R. (2016). A survey of publication practices of single-case design researchers when treatments have small or large effects. *Journal of Applied Behavior Analysis, 49*(3), 656–673. https://doi.org/10.1002/jaba.308.

Sham, E., & Smith, T. (2014). Publication bias in studies of an applied behavior-analytic intervention: an initial analysis. *Journal of Applied Behavior Analysis, 47*(3), 663–678. https://doi.org/10.1002/jaba.146.

Shull, R. L. (1999). Statistical inference in behavior analysis: Discussant's remarks. *The Behavior Analyst, 22*(2), 117–121. https://doi.org/10.1007/bf03391989.

Smith, J. D. (2012). Single-case experimental designs: A systematic review of published research and current standards. *Psychological Methods, 17*(4). doi:https://doi.org/10.1037/a0029312

Soderberg, C. K. (2018). Using OSF to share data: A step-by-step guide. *Advances in Methods & Practices in Psychological Science, 1*(1), 115–120. https://doi.org/10.1177/2515245918757689.

Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., et al. (2011). Data sharing by scientists: practices and perceptions. *PLoS One, 6*(6), e21101. https://doi.org/10.1371/journal.pone.0021101.

Tincani, M., & Travers, J. (2017). Publishing single-case research design studies that do not demonstrate experimental control. *Remedial & Special Education, 39*(2), 118–128. https://doi.org/10.1177/0741932517697447.

Tincani, M., & Travers, J. (2019). Replication research, publication bias, and applied behavior analysis. *Perspectives on Behavior Science.* https://doi.org/10.1007/s40614-019-00191-5.

Torvalds, L. (2018). Linux kernel source tree. *GitHub Respository.* Retrieved from https://github.com/torvalds/linux.

Wasserstein, R. L., & Lazar, N. A. (2016). The ASA's statement on p-values: Context, process, and purpose. *The American Statistician, 70*(2), 129–133. https://doi.org/10.1080/00031305.2016.1154108.

Wolery, M., & Dunlap, G. (2001). Reporting on studies using single-subject experimental methods. *Journal of Early Intervention, 24*(2), 5.

Wolery, M., & Ezell, H. K. (1993). Subject descriptions and single-subject research. *Journal of Learning Disabilities, 26*(10), 642–647. https://doi.org/10.1177/002221949302601001.

Young, M. E. (2018). A place for statistics in behavior analysis. *Behavior Analysis: Research and Practice, 18*(2), 193–202. https://doi.org/10.1037/bar0000099.

Zimmermann, Z. J., Watkins, E. E., & Poling, A. (2015). JEAB research over time: Species used, experimental designs, statistical analyses, and sex of subjects. *Behav Anal, 38*(2), 203–218. https://doi.org/10.1007/s40614-015-0034-5.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.