Research Education Program Plan

A Significance

The recent NIH-Wide Strategic Plan [1] describes an integrative view of biology and human health that includes translational medicine, team science, and the importance of capitalizing on an exponentially growing and increasingly complex data ecosystem [2]. Underlying this view is the need to use, share, and re-use biomedical data generated from widely varying experimental systems and researchers. Basic sources of biomedical data range from relatively small sets of measurements, such as animal body weights and bacterial cell counts that may be recorded by hand, to thousands or millions of instrument-generated data points from various imaging, -omic, and flow cytometry experiments. In either case, there is a generally common workflow that proceeds from measurement to data recording, pre-processing, analysis, and interpretation. However, in practice the distinct actions of data recording, data pre-processing, and data analysis are often merged or combined as a single entity by the researcher using commercial or open source spreadsheets, or as part of an often proprietary experimental measurement system / software combination (Figure 1), **resulting in key failure points for reproducibility at the stages of data recording and pre-processing.**

It is widely known and discussed among data scientists, mathematical modelers, and statisticians [3, 4] that there is frequently a need to discard, transform, and reformat various elements of the data shared with them by laboratory-based researchers, and that data is often shared in an unstructured format, increasing the risks of introducing errors through reformatting before applying more advanced computational methods. Instead, a critical need for reproducibility is for the transparent and clear sharing across research teams of: (1) raw data, directly from hand-recording or directly output from experimental equipment; (2) data that has been pre-processed as necessary (e.g., gating for flow cytometry data, feature identification for metabolomics data), saved in a consistent, structured format, and (3) a clear and repeatable description of how the pre-processed data was generated from the raw data [3, 5].

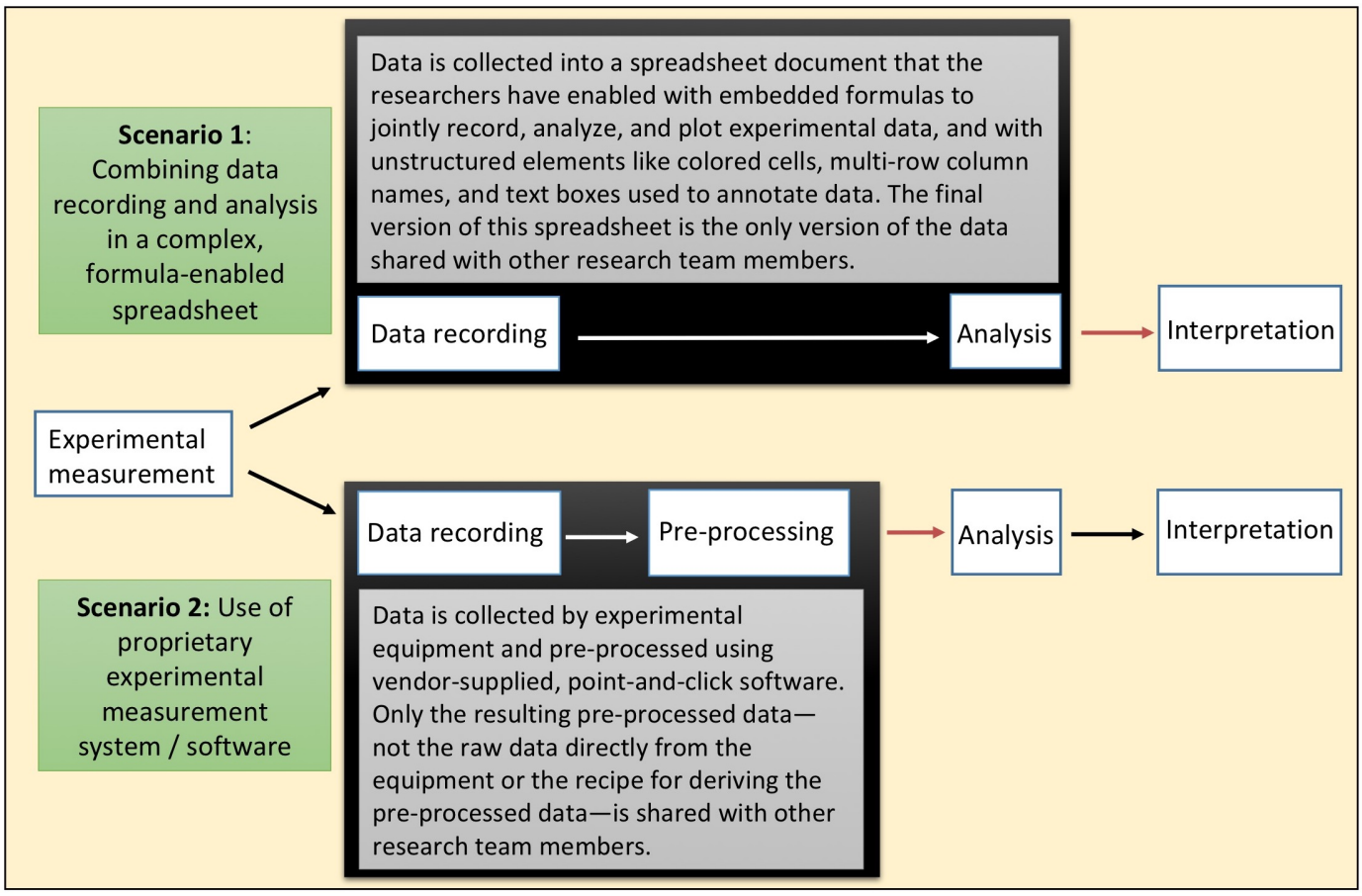

Figure 1: Two common scenarios where "black boxes" of non-transparent, non-reproducible data handling exist in research data workflows at the stages of data recording and pre-processing. These create potential points of failure for reproducible research. Red arrows indicate where data is passed to other research team members, including statisticians / data analysts, often within complex or unstructured spreadsheet files.

Here, we identify a clear separation among data recording, data pre-processing, and data analysis—breaking up commonly existing "black boxes" in data handling across the research process—as a critical point for enhancement of data reproducibility. Such a rigorous demarcation requires some change in the conventional understanding and use of spreadsheets and a recognition by biomedical researchers that recent advances in computer programming languages, especially the R programming language, provide user-friendly and accessible tools and concepts that can be used to extend a transparent and reproducible data workflow to the steps of data recording and pre-processing. Among our team, we have found that there are many common existing practices—including use of spreadsheets with embedded formulas that concurrently record and analyze experimental data, problematic management of project files, and reliance on proprietary, vendor-supplied point-and-click software for data pre-processing—that can interfere with the transparency, reproducibility, and efficiency of laboratory-based biomedical research projects, problems that have also been identified by others as key barriers to research reproducibility [3, 6, 5, 7]. Our team has worked together to craft a list of specific topics (Tables 1 and 2) we will address in training modules, choosing topics that tackle barriers to reproducibility that have straightforward, easy-to-teach solutions, but which are still very common in biomedical laboratory-based research programs.

As a wholesale adoption of these concepts and software tools may be disruptive to established laboratory procedures, our approach is to provide training modules that can be adopted as appropriate for a wide range of individuals with differing roles within small and large research groups. General principles and guidelines are presented separately from implementation and examples. Additionally, the training modules will be developed on-site at Colorado State University with NIH-funded microbiology and immunology laboratories devoted to anti-tuberculosis drug and vaccine development, as **our primary educational goal is to introduce the language, concepts, and tools of an R-based ecosystem for reproducible research to laboratory-based scientists whose attention is rather to their experimental technique and collection of accurate data, and who may have little or no background in the use of general purpose software tools.** While this proposal uses examples from microbiology and immunology experiments, the collection of primary data in a flexible, open source, transparent, and reproducible format, which can be kept in its primary state without the need for additional modification, is a solution that extends throughout large- and small-scale biomedical science projects.

B Innovation

We will use the innovative *bookdown* framework [8] to structure and publish all of our training materials as a free and open online book, with embedded video lectures and additional educational materials to help these training modules reach a global audience of biomedical researchers. The *bookdown* framework has been available for a little over two years and extends the principles of literate programming from *Rmarkdown* [9] to allow authors to create attractive online books that integrate programming code and text. With this format, code examples and related output do not need to be copied and pasted into a document, but instead are automatically generated. Through this innovative framework, we will create a searchable online book that weaves R code into the text and includes embedded video lectures, active web links to online references, and computationally reproducible examples and exercises (Figure 2). Further, trainees will be able to download the book as either a PDF or EPUB file to use as a reference as they continue learning to implement reproducibility tools in their research projects. Though this framework is new, it has proven itself reliable and effective—it serves as the framework for several popular and highly-accessed books on R programming, including the extremely popular *R for Data Science* [10].

The use of this framework will help us effectively reach a large audience of laboratory-based biomedical researchers in need of training materials to improve the reproducibility of data recording and pre-processing. We will post this book online using *GitHub Pages* [11], as we have done with a previous book created with the same framework [12], allowing anyone to freely access the content online. We will publish both the current version of the developing book and its underlying code openly online from the beginning of our development of the training materials. **By making the training materials available from day one, as they are developed, we will be able to attract early users to help disseminate the material as it evolves and to get early feedback.** Once the online book is developed, we will submit a static version of it to be posted on the homepage of the *bookdown.org* website [13], providing another way for our key audience of laboratory-based biomedical researchers to find and access the materials.

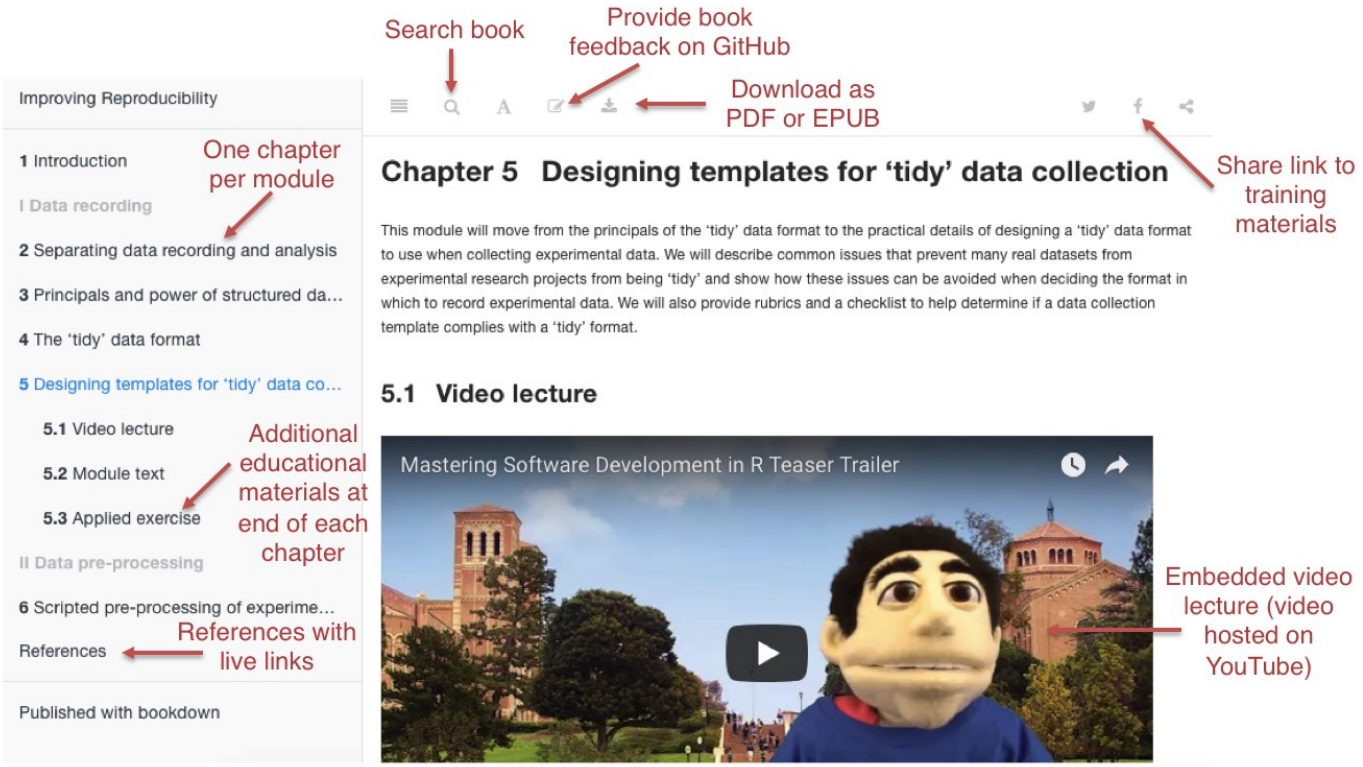

Figure 2: Prototype of online course book, with features highlighted.

Dr. Anderson (PI) is an early adopter of the *bookdown* framework and served as an expert reviewer for *bookdown: Authoring Books and Technical Documents with R Markdown* [8]. Since the *bookdown* framework became available, she has used it to create two openly-available online books: the *R Programming for Research* online coursebook (https://geanders.github.io/RProgrammingForResearch/), which she uses as a joint textbook and website for the *R Programming for Research* course she teaches at Colorado State University, and *Mastering Software Development in R* (https://bookdown.org/rdpeng/RProgDA/), which she co-wrote with Dr. Roger Peng as a manual for developing advanced R programming skills and which has been downloaded by over 14,000 people from LeanPub (see letter from Dr. Peng).

C Approach—Proposed Research Education Program

Through this project, we aim to create training modules that will teach laboratory-based biomedical researchers how simple computational reproducibility principles and tools can improve reproducibility at the stages of data recording and pre-processing. The importance of *computational reproducibility*—in which all data and code for a research project are openly available and can be used to regenerate study findings either by the original researcher or by other researchers—is increasingly recognized by scientists, journals, and funding agencies [5, 14]. Our core project team combines two experts in R programming (Anderson, Lyons) with three biomedical researchers (Gonzalez-Juarrero, Henao-Tamayo, and Robertson) who have, collectively, spent decades in laboratory-based research to improve understanding of tuberculosis and other diseases. We met as fellow faculty members of the College of Veterinary Medicine & Biomedical Sciences at Colorado State University, and since have discovered that many of the tools that Drs. Anderson and Lyons teach and use to improve the reproducibility of *data analysis* for biomedical research can substantially improve reproducibility and transparency in the laboratory-based biomedical research projects of Drs. Gonzalez-Juarrero, Henao-Tamayo, and Robertson at the stages of *data recording* and *data pre-processing*. **Improving the computational reproducibility of research at these stages is critical, as these steps form the foundation, and provide input, for the later stages of data analysis, visualization, and interpretation. If data recording and pre-processing are not computationally reproducible, there is no chance to make the full research project reproducible.** Further, improving the computational reproducibility of the steps of data recording and pre-processing makes collaborations with statisticians and data analysts more efficient and less prone to errors [5], encouraging productive and high-quality "Team Science" to tackle large biomedical research projects [15].

The reproducibility of the steps of data recording and pre-processing is typically in the hands of laboratory-based, rather than computationally-based, researchers. While many excellent free training resources exist to improve the computational reproducibility of biomedical research, most of these materials—including some developed by Dr. Anderson for her online and Colorado State University-based courses in R programming [12, 16]—target researchers at the stage of *data analysis* for the audience of *computationally-based researchers*, and provide much less guidance on the principles and techniques to improve reproducibility of the earlier steps of *experimental data recording* and *experimental data pre-processing* for *laboratory-based researchers*. Over the past year, our team of co-investigators has begun to work together to improve the computational reproducibility of experimental data recording and pre-processing within our own research projects. For example, in Fall 2017 Dr. Gonzalez-Juarrero attended Dr. Anderson's (PI) course in *R Programming for Research* and has brought the ideas and techniques back to her research laboratory, and in Spring 2018 Dr. Henao-Tamayo and Dr. Anderson began co-advising a graduate student to implement open source tools for pre-processing flow cytometry in Dr. Henao-Tamayo's laboratory. Collectively, we are passionate about the idea that open source tools can be used to substantially improve the reproducibility of data recording and pre-processing in laboratory-based biomedical research, and yet we are also able to recognize that there are key barriers in implementing these tools in this setting, as well as in training laboratory-based researchers how to use these tools.

Our project's **primary goal** is to develop training modules that address the needs of laboratory-based biomedical researchers seeking to improve reproducibility, especially of experimental data recording and pre-processing, in their research projects. The **expected result** of this project is an online book that contains twenty short training modules as separate chapters, with video lectures, written text, and additional educational materials collected within each module's chapter. We consider it critical that these training materials be clear, relevant, and useful to a key audience of biomedical scientists whose primary research activities focus on laboratory research, rather than data analysts or statisticians. To meet this need, we propose to develop two sequences of modules, **Improving the Reproducibility of Experimental Data Recording** and **Improving the Reproducibility of Experimental Data Pre-Processing**. Each module will fall into one of three categories for teaching reproducibility: (1) *Principles*; (2) *Implementation*; and (3) *Examples*. *Principles* modules will be programming-language agnostic, while *Implementation* modules will focus on tools available through the popular open source R software and its *RStudio* interface. *Examples* modules will provide materials that can be used as a template for implementing R-based tools, but can also provide a top-level overview for non-programmers of how these tools can improve real biomedical research projects, and will be based on existing, NIH-funded research groups within our research team, for which we already have the data in hand, and in which we have been working to improve reproducibility. By co-authoring the modules with the biomedical laboratory-based co-investigators on our team, we will ensure that these modules and the examples used in them are approachable and useful to researchers with limited computational training. We have divided the content into modules in a way that will allow **trainees and investigators at any level** to create their own "tracks" by selecting relevant subsets of the modules to complete, potentially combining this content with other training modules available through the National Institute of General Medical Science's *Clearinghouse for Training Modules to Enhance Data Reproducibility* [17] to create a training experience aligned with their individual training needs. Table 3 gives a few examples of how different trainees could create and follow their own "track" through the training materials we propose to develop.

Sequence 1: Improving the Reproducibility of Experimental Data Recording

The first sequence will provide principles and tools for improving computational reproducibility at the stage of experimental data recording. This sequence will include eleven modules covering four main topics:

1. Separating data recording and analysis. Many biomedical laboratories currently use spreadsheets—with formulas creating underlying connections between spreadsheet cells—to jointly record, visualize, and analyze experimental data (Sequence 1 in Figure 1) [3]. This practice impedes the transparency and reproducibility of both data recording and data analysis. When a research group develops and uses an evolving spreadsheet template with embedded formulas, it leads to a data recording / analysis process that can become extraordinarily opaque and complex. To improve the computational reproducibility of a research project, it is critical for biomedical researchers to learn the importance of maintaining recorded experimental data as "read-only" files, separating data recording from any data pre-processing or data analysis steps [3, 7]. Statisticians have outlined specific methods that a laboratory-based scientist can take to ensure that data shared in an Excel spreadsheet are shared in a reliable and reproducible way, including avoiding

macros or embedded formulas, using a separate Excel file for each dataset, recording descriptions of variables in a separate code book rather than in the Excel file, avoiding the use of color of the cells to encode information, using "NA" to code missing values, avoiding spaces in column headers, and avoiding splitting or merging cells [5, 3]. These topics will covered in a *Principles* module on "Separating data recording and analysis" (Table 1).

2. Using a structured data format to record data. Every extra step of data cleaning is another chance to introduce errors in experimental biomedical data, and yet laboratory-based researchers often share experimental data with collaborators in a format that requires extensive additional cleaning before it can be input into data analysis [3]. Recording data in a "structured" format brings many benefits for later stages of the research process, especially in terms of improving reproducibility and reducing the probability of errors in analysis [5]. Data that is in a structured, tabular, two-dimensional format is substantially easier for collaborators to understand and work with, without additional data formatting [3]. Further, by using a consistent structured format across many or all data in a research project, it becomes much easier to create solid, well-tested code scripts for data pre-processing and analysis and to apply those scripts consistently and reproducibly across datasets from multiple experiments [3]. However, many biomedical researchers are unaware of this simple yet powerful strategy in data recording and how it can improve the efficiency and effectiveness of collaborations [5].

The "tidy" data format is one implementation of a tabular, two-dimensional structured data format that has quickly gained popularity among statisticians and data scientists since it was defined in a 2014 paper [18]. The "tidy" data format plugs into R's *tidyverse* framework, which enables powerful and user-friendly data management, processing, and analysis by combining simple tools to solve complex, multi-step problems [19, 20, 21, 10]. Since the *tidyverse* tools are simple and share a common interface, they are easier to learn, use, and combine than tools created in the traditional base R framework [19, 22, 23, 24]. This *tidyverse* framework is quickly becoming the standard taught in introductory R courses and books [25, 26, 27, 28, 23, 24], ensuring ample training resources for researchers new to programming, including books (e.g., [29, 30, 10]), massive open online courses (MOOCs), on-site university courses [26, 27, 28], and Software Carpentry workshops [31, 32]. Further, tools that extend the *tidyverse* have been created to enable high-quality data analysis and visualization in several domains, including text mining [33], microbiome studies [34], natural language processing [35], network analysis [36], ecology [37], and genomics [38].

These topics will be covered in a *Principles* module on "Principles and power of structured data formats", two *Implementation* modules on "The "tidy" data format: an implementation of a structured data format" and "Designing templates for tidy data collection", and one *Example* module called "Example: Creating a template for "tidy" data collection" (Table 1).

3. Managing all research project files in a single, structured directory. Reproducibility can also be improved, starting at the data recording stage, by using a single, structured directory to store all files related to the project. This "project" framework of structured and thoughtful file management has recently been encouraged by a number of researchers as a way to enable computationally reproducible research, especially for research conducted by teams [7, 39, 22]. If a consistent structure is used for these directories across different research projects, it can substantially increase the efficiency of, and reduce errors in, data pre-processing and analysis, as code scripts can be created that can be re-used across different project directories with few required changes [7]. This practice also improves the efficiency and effectiveness of collaborations with data scientists, mathematical modelers, and statisticians, as it allows the researchers to share critical research project files—raw data, processed data, and code scripts for extracting the processed data from the raw data [5, 40]—in a single directory. Further, some have suggested that if researchers learn better practices for managing "clean" project files, it will help increase their willingness to share those files and so meet standards of reproducibility [7].

One implementation of this practice is as an *RStudio* "Project". In *RStudio*, a researcher can collect all project files in a single, structured directory and save this directory as a "Project" [41]. *RStudio* "Projects" allow easy integration with version control tools (*git*) and online platforms for sharing a directory under version control (*GitHub*, *GitLab*). While a "Project" can have any internal structure, a common structure can be enforced across multiple research projects through the creation of a new "Project" template, which defines the required subdirectories, structure, and file names of common elements for each project's structured directory [42]. This template, when selected from a menu bar in *RStudio* by a future user, will create a new directory

with a "skeleton" structure, potentially including templated files (e.g., for metadata that a researcher wants to remember to record for each project) [42]. If there is a standard for organizing files for a researcher's scientific area, this format can be encoded as a reusable template; use of the structure imposed by this template will make it easier for other researchers in the field to easily navigate code and data that is made public in efforts to increase reproducibility [7].

These topics will be covered in a *Principles* module on the "Power of using a single structured 'Project' directory for storing and tracking research project files", an *Implementation* module on "Creating 'Project' templates", and an *Example* module called "Example: Creating a 'Project' template" (Table 1).

4. Implementing version control. As a research project progresses, a typical practice in many laboratory-based research groups is to save new, renamed versions of each file (e.g., "draft1.doc", "draft2.doc") [6], so that the researchers can revert to earlier versions of a file. However, this practice leads to an explosion of files, and it becomes hard to track which files represent the "current" state of a project. Version control—which tracks and documents changes to any files that the user chooses to track in a directory—allows researchers to edit and change research project files more cleanly, keeping a single copy of each file in the directory rather than multiple versions, while maintaining the power to backtrack to previous versions. Further, with version control, "commit messages" are required to explain changes, making both the changes and the reasoning behind them transparent. The use of the version control software *git* has been encouraged as a tool to enable reproducible research [43, 14, 6, 22, 44]. Once a researcher has learned to use *git* on their own computer for local version control, they can begin using version control platforms (e.g., *GitLab*, *GitHub*) to collaborate with others in their research group while keeping the project under version control [6, 40]. Platforms like *GitHub* and *GitLab* allow all collaborators to share a current version of a project directory (similar to Dropbox), but in a way that allows easy use of version control and that is more efficient for exploring (and, when necessary, undoing) the changes each team member has made to project files [6]. For many years, using *git* version control required use of the command line, limiting its accessibility to researchers with limited programming experience. Graphical interfaces, however, have removed this barrier, and *RStudio* has particularly user-friendly tools for implementing version control. At Colorado State University, we have found that *git* and *GitHub* can be quickly taught to researchers in their first programming course, when using the *RStudio* / *GitHub* interfaces, so that they can successfully use *GitHub* to submit class assignments; others have reported similar success at other institutions [6]. These topics will be covered in two *Principles* modules called "Harnessing version control for transparent data recording" and "Enhance the reproducibility of collaborative research with version control platforms" as well as an *Implementation* module on "Using *git* and *GitLab* to implement version control" (Table 1).

Sequence 2: Improving the Reproducibility of Experimental Data Pre-Processing

The second sequence will provide principles and tools for improving computational reproducibility as experimental data is pre-processed. **By "pre-processing", we mean the steps taken to convert data from the raw data—either collected by hand or output by laboratory equipment—into a format ready for data analysis.** We will focus particularly on improving reproducibility of pre-processing the complex raw data output by laboratory equipment—for example feature identification and quantification in mass spectrometry data and gating in flow cytometry data—as well as general pre-processing steps like normalization and scaling. This sequence will include nine modules covering three main topics:

1. Using code scripts to pre-process experimental data. The experimental data collected for biomedical research often requires pre-processing before it can be analyzed. While point-and-click software is often available for this pre-processing (Scenario 2 in Figure 1), it is used interactively and often does not create a history of steps and choices [45], at least not in a format that is easy for a statistician to navigate, understand, and check [46, 45]. Statisticians have clearly stated that, to ensure research is reproducible and that a collaboration is efficient, they would like to receive raw experimental data (i.e., the direct output from experimental equipment), the processed data (preferably in a "tidy" format), and an "explicit and exact recipe" for how the processed data was derived from the raw data [5]. A code script, like an R script, provides this "explicit and exact recipe". Scripted pre-processing can also help reduce the temptation to manually edit data during pre-processing, including manually changing file formats, which prevents reproducibility and impedes transparency [45]. While many of the pre-processing tasks required for biomedical experimental data are complex (e.g., feature identification, gating), R has package extensions that can be used for these tasks, many hosted on Bioconductor [47]. Open-source scriptable software tools bring other key advantages

compared to proprietary software in terms of data pre-processing, including that open source choices are transparent and often more robust and easier to extend [44, 47, 48, 43, 49]. These topics will be covered in one *Principles* module on "Principles and benefits of scripted pre-processing of experimental data" and two *Implementation* modules called "Introduction to scripted data pre-processing in R" and "Simplify scripted pre-processing through R's *tidyverse* tools" (Table 2).

2. Working with complex data types during pre-processing. Many R functions output data in a format that is "untidy" [50], in the sense that it does not comply with the structured "tidy" format required by R's *tidyverse* tools [18]. This is particularly true for raw data from many biomedical experiments, especially machine-generated data (e.g., output from a flow cytometer or mass spectrometer). Further, Bioconductor, which hosts many R packages useful for pre-processing and analyzing experimental biomedical data, relies heavily on an object-oriented framework [51]. This aids interoperability among Bioconductor packages and helps contain different types of data from an experiment (expression measurements, phenotypes, and administrative data) [51]. While these formats are well-justified within open source software for pre-processing complex biomedical data, they add a critical barrier for researchers wishing to implement reproducibility tools [50]. This hurdle can be surmounted by skilled R programmers, less so for researchers new to scripting tools [50], and it can reduce transparency of analysis by requiring obscure, lengthy code to extract and tidy data from the complex data object [50]. Very recently, the *broom* and *biobroom* R packages have been developed to extract a "tidy" dataset from many common complex data formats that are output by R functions [50, 52]. The *biobroom* package, in particular, can "tidy" data within many popular Bioconductor data formats, including *ExpressionSet* objects [52]. These topics will be covered in a *Principles* module on "Complex data types in experimental data pre-processing", a *Implementation* module on "Complex data types in R and Bioconductor", and an *Example* module called "Example: Converting from complex to "tidy" data formats" (Table 2).

3. Reproducible data pre-processing protocols. Pre-processing software requires many parameter choices, which can lead to an explosion of possible combinations of ways to pre-process experimental biomedical data [15, 40, 45]. To ensure transparent and high-quality biomedical research, it is important to be thoughtful in selecting these parameter values and to keep a detailed record of which parameter values were selected and why. Ideally, a research group should maintain "protocols" describing this pre-processing that are as detailed and reproducible as their experimental protocols [51]. Creating and maintaining such *data pre-processing protocols* helps make it easier and clearer for laboratory-based researchers to share their pre-processing "recipe" with collaborators and to publish journal articles that meet reproducibility guidelines. It also helps ensure continuity in methods for a research group, ensuring that the same pre-processing is maintained as researchers join and leave the group [40]. Reproducibility tools can be used to create *reproducible data pre-processing protocols*—documents that combine code and text in a "knitted" document, which can be re-used across experiments and research projects. *RMarkdown* can be used to combine code and text to create these reproducible data pre-processing protocols. These topics will be covered in a *Principles* module called "Introduction to reproducible data pre-processing protocols", an *Implementation* module on "RMarkdown for creating reproducible data pre-processing protocols", and an *Example* module called "Example: Creating a reproducible data pre-processing protocol" (Table 2).

Choice of tools to include in "Implementation" modules.

To improve reproducibility practices among our target audience of laboratory-based biomedical researchers, it is important that we provide them with some instruction on how to *implement* the reproducibility principles presented. For some of these principles, there are several reasonable and well-developed tools that could be used for implementations. However, few researchers are interested in learning every tool, and instead would prefer to learn one set of tools that "just work", and presenting a single set of tools improves the chance of trainees mastering the tools [53].

We have therefore chosen for the *Implementation* modules to focus on tools from the open source R programming language ecosystem. R can be freely, quickly, and easily downloaded and installed to a user's computer, allowing new users to get started quickly, a critical consideration for usable scientific software [54]. R has been maintained for over a decade by the R Development Core Team and works with all major computing platforms, ensuring widespread access, stability, and compatibility, which also helps ensure ease-of-use [49, 55]. R offers a well-developed environment for creating new tools that extend the core

language [56, 51] and includes ample tools for documenting research workflows [9, 8]. R's status as a common tool among statisticians and biostatisticians means that its use in early stages of experimental data recording and pre-processing can help foster closer collaborations between laboratory-based scientists and statisticians throughout the research process. *RStudio* is a free, open source Integrated Development Environment (IDE) for the R programming language. It is actively developed by a team of some of the best R programmers worldwide, including the creators of the *tidyverse* (Hadley Wickham) and *RMarkdown* (Yihui Xie). Some of the implementation tools—*git* and *GitLab*, for example—are separate from R but can be mastered much more easily if trainees are taught to use them through *RStudio*'s user-friendly interfaces rather than using the command line or other alternative interfaces.

We appreciate that many laboratory-based biomedical researchers do not know the R language, and some of them may want to learn more about improving reproducibility without needing to learn a new programming language. Although literacy in programming is increasing in the sciences [14], and many now recommend programming as a critical skill for all biology Ph.D. students [54], expertise with a scripting language is not universal across biomedical researchers. For this reason, we have deliberately separated the *Principles* and *Examples* content in our modules from the *Implementation* modules, so that a researcher can select a track of our modules that does not require programming knowledge. The code examples themselves will follow literate programming principles with small, modular, well-described code that trainees can use as templates in their own research projects. Further, while all of the *Implementation* modules are conceptually focused on tools that an R programmer would use, several of the modules could be appreciated and used to improve reproducibility without a mastery of R. For example, *RStudio* and its "Projects" functionality can be used to help manage research project files, keep them under version control, and interface with *GitLab* without using any R code within the project. Similarly, while the "tidy" data format is currently an important implementation common within R for structuring data, understanding its principles and characteristics does not require any knowledge of R. In our table of example "tracks" (Table 3), the track we give for an example Principal Investigator is one example of a track that requires no prior knowledge of R or other programming languages.

Table 1: Modules for the first sequence, **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages.

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
Separating data recording and analysis	Many biomedical laboratories use spreadsheets, with embedded formulas, to both record and analyze experimental data. This practice impedes transparency and reproducibility of both data recording and data analysis. In this module, we will describe this common practice and will outline alternative approaches that separate the steps of data recording and data analysis.	• Explain the difference between data recording and data analysis • Understand why collecting data on spreadsheets with embedded formulas impedes reproducibility • List alternative approaches to improve reproducibility	15	• Discussion questions about data recording approaches the trainee has previously used in research projects and the benefits and limitations for data transparency and reproducibility • Short audio recording of two Co-Is giving their answers
Principles and power of structured data formats	The format in which experimental data is recorded can have a large influence on how easy and likely it is to implement reproducibility tools in later stages of the research workflow. Recording data in a 'structured' format brings many benefits. In this module, we will explain what makes a dataset 'structured' and why this format is a powerful tool for reproducible research.	• List the characteristics of a structured data format • Describe benefits for research transparency and reproducibility • Outline other benefits of using a structured format when recording data	10	• Applied exercise: For example datasets, specify whether each is in a structured data format and, if not, draft a structured version • Video walking trainees through solutions to the applied exercise
The 'tidy' data format: an implementation of a structured data format	The 'tidy' data format is an implementation of a structured data format popular among statisticians and data scientists. By consistently using this data format, researchers can combine simple, generalizable tools to perform complex tasks in data processing, analysis, and visualization. We will explain what characteristics determine if a dataset is 'tidy' and how use of the 'tidy' implementation of a structure data format can improve the ease and efficiency of 'Team Science'.	• List characteristics defining the the 'tidy' structured data format • Explain the difference between the a structured data format (general concept) and the 'tidy' data format (one popular implementation)	15	• Quiz questions: For example datasets, correctly identify which of the 'tidy' data principles the dataset has or lacks • Video explaining quiz solutions
Designing templates for tidy data collection	This module will move from the principles of the 'tidy' data format to the practical details of designing a 'tidy' data format to use when collecting experimental data. We will describe common issues that prevent biomedical research datasets from being 'tidy' and show how these issues can be avoided. We will also provide rubrics and a checklist to help determine if a data collection template complies with a 'tidy' format.	• Identify characteristics that keep a dataset from being 'tidy' • Convert data from an 'untidy' to a 'tidy' format	20	• Applied exercise: For an 'untidy' dataset, explain why it is not 'tidy' and convert to a 'tidy' format • Video providing a detailed solution to the applied exercise

Table 1: Modules for the first sequence, **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
Example: Creating a template for 'tidy' data collection	We will walk through an example of creating a template to collect data in a 'tidy' format for a laboratory-based research project, based on a research project on drug efficacy in murine tuberculosis models. We will show the initial 'untidy' format for data recording and show how we converted it to a 'tidy' format. Finally, we will show how the data can then easily be analyzed and visualized using reproducible tools.	• Understand how the principles of 'tidy' data can be applied for a real, complex research project; • List advantages of the 'tidy' data format for the example project	15	• Discussion questions, including listing examples of experiences collecting data in an 'untidy' format • Short audio recording of two Co-Is giving their answers
Power of using a single structured 'Project' directory for storing and tracking research project files	To improve the computational reproducibility of a research project, researchers can use a single 'Project' directory to collectively store all research data, meta-data, pre-processing code, and research products (e.g., paper drafts, figures). We will explain how this practice improves the reproducibility and list some of the common components and subdirectories to include in the structure of a 'Project' directory, including subdirectories for raw and pre-processed experimental data.	• Describe a 'Project' directory, including common components and subdirectories • List how a single 'Project' directory improves reproducibility	20	• Quiz questions: What is a structured 'Project' directory and what are its benefits to reproducibility • Video with detailed discussion of quiz solutions
Creating 'Project' templates	Researchers can use RStudio's 'Projects' can facilitate collecting research files in a single, structured directory, with the added benefit of easy use of version control. Researchers can gain even more benefits by consistently structuring all their 'Project' directories. We will demonstrate how to implement structured project directories through RStudio, as well as how RStudio enables the creation of a 'Project' for initializing consistently-structured directories for all of a research group's projects.	• Be able to create a structured 'Project' directory within RStudio • Understand how RStudio can be used to create 'Project' templates	25	• Discussion questions on how the trainee has saved and tracked research project files for previous research projects and related barriers to reproducibility • Short audio recording of two Co-Is discussing their answers
Example: Creating a 'Project' template	We will walk through a real example, based on the experiences of one of our Co-Is, of establishing the format for a research group's 'Project' template, creating that template using RStudio, and initializing a new research project directory using the created template. This example will be from a laboratory-based research group that studies the efficacy of tuberculosis drugs in a murine model.	• Create a 'Project' template in RStudio to initialize consistently-formatted 'Project' directories • Initialize a new 'Project' directory using this template	15	• Applied exercise: Create and save a 'Project' template that meets specifications provided for an example research group • Video demonstrating a detailed solution

Table 1: Modules for the first sequence, **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
Harnessing version control for transparent data recording	As a research project progresses, a typical practice in many experimental research groups is to save new versions of files (e.g., 'draft1.doc', 'draft2.doc'), so that changes can be reverted. However, this practice leads to an explosion of files, and it becomes hard to track which files represent the 'current' state of a project. Version control allows researchers to edit and change research project files more cleanly, while maintaining the power to 'backtrack' to previous versions, messages included to explain changes. We will explain what version control is and how it can be used in research projects to improve the transparency and reproducibility of research, particularly for data recording.	• Describe version control • Explain how version control can be used to improve reproducibility for data recording	10	• Discussion questions, including discussion of how the trainee has managed evolving research project files in previous projects and related barriers to reproducibility • Short audio recording of two Co-Is giving their own answers
Enhance the reproducibility of collaborative research with version control platforms	Once a researcher has learned to use *git* on their own computer for local version control, they can begin using version control platforms (e.g., *GitLab*, *GitHub*) to collaborate with others under version control. We will describe how a research team can benefit from using a version control platform to work collaboratively.	• List benefits of using a version control platform to collaborate on research projects, particularly for reproducibility • Describe the difference between version control (e.g., *git*) and a version control platform (e.g., *GitLab*)	10	• Discussion questions: Describe how past research projects shared files without using version control • Short audio file with two Co-Is discussing their answers
Using git and GitLab to implement version control	For many years, use of version control required use of the command line, limiting its accessibility to researchers with limited programming experience. However, graphical interfaces have removed this barrier, and RStudio has particularly user-friendly tools for implementing version control. In this module, we will show how to use *git* through RStudio's user-friendly interface and how to connect from a local computer to *GitLab* through RStudio.	• Understand how to set up and use *git* through RStudio's interface • Understand how to connect with *GitLab* through RStudio to collaborate on research projects while maintaining version control	20	• Applied exercise: Use RStudio to initialize *git* version control for a directory and to make several tracked changes. Create a matching *GitLab* repository and use RStudio to push local changes to this *GitLab* version of the directory • Video walking trainees through a detailed solution

Table 2: Modules for the second sequence, **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages.

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
Principles and benefits of scripted pre-processing of experimental data	The experimental data collected for biomedical research often requires pre-processing before it can be analyzed (e.g., gating of flow cytometry data, feature finding / quantification for mass spectrometry data). Use of point-and-click software can limit the transparency and reproducibility of this analysis stage and is time-consuming for repeated tasks. We will explain how scripted pre-processing, especially using open source software, can improve transparency and reproducibility.	• Define 'pre-processing' of experimental data • Describe an open source code script and explain how it can increase reproducibility of data pre-processing	15	• Discussion questions, including common pre-processing needs and practices • Short audio recording of two Co-Is giving their answers
Introduction to scripted data pre-processing in R	We will show how to implement scripted pre-processing of experimental data through R scripts. We will demonstrate the difference between interactive coding and code scripts, using R for examples. We will then demonstrate how to create, save, and run an R code script for a simple data cleaning task.	• Describe what an R code script is and how it differs from interactive coding in R • Create and save an R script to perform a simple data pre-processing task • Run an R script • List some popular packages in R for pre-processing biomedical data	10	• Applied exercise: Given a simple example dataset and a data cleaning task, write and run an R script to perform the task. Then adapt that script to re-use it on a second dataset. Hints will be provided for those new to R • Video providing a detailed walk-through of a solution to the applied exercise
Simplify scripted pre-processing through R's 'tidyverse' tools	The R programming language now includes a collection of 'tidyverse' extension packages that enable user-friendly yet powerful work with experimental data, including pre-processing and exploratory visualizations. The principle behind the 'tidyverse' is that a collection of simple, general tools can be joined together to solve complex problems, as long as a consistent format is used for the input and output of each tool (the 'tidy' data format taught in other modules). In this module, we will explain why this 'tidyverse' system is so powerful and how it can be leveraged within biomedical research, especially for reproducibly pre-processing experimental data.	• Define R's 'tidyverse' system • Explain how the 'tidyverse' collection of packages can be both user-friendly and powerful in solving many complex tasks with data • Describe the difference between base R and R's 'tidyverse'.	15	• Quiz questions: What is R's 'tidyverse' and why is it a powerful yet user-friendly tool for improving the reproducibility of research projects • Video with detailed answers and explanations for the quiz questions • Links to free sources for developing more 'tidyverse' coding skills

Table 2: Modules for the second sequence, **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
Complex data types in experimental data pre-processing	Raw data from many biomedical experiments, especially those that use high-throughput techniques, can be very large and complex. Because of the scale and complexity of these data, software for pre-processing the data in R often uses complex, 'untidy' data formats. While these formats are necessary for computational efficiency, they add a critical barrier for researchers wishing to implement reproducibility tools. In this module, we will explain why use of complex data formats is often necessary within open source pre-processing software and outline the hurdles created in reproducibility tool use among laboratory-based scientists.	• Explain why R software for pre-processing biomedical data often stores data in complex, 'untidy' formats • Describe how these complex data formats can create barriers to laboratory-based researchers seeking to use reproducibility tools for data pre-processing	15	• Quiz questions: Why are complex data formats often used within steps of experimental data pre-processing in open-source software and how does their use complicate the use of reproducibility tools • Video providing detailed answers
Complex data types in R and Bioconductor	Many R extension packages for pre-processing experimental data use complex (rather than 'tidy') data formats within their code, and many output data in complex formats. Very recently, the *broom* and *biobroom* R packages have been developed to extract a 'tidy' dataset from a complex data format. These tools create a clean, simple connection between the complex data formats often used in pre-processing experimental data and the 'tidy' format required to use the 'tidyverse' tools now taught in many introductory R courses. In this module, we will describe the 'list' data structure, the common backbone for complex data structures in R and provide tips on how to explore and extract data stored in R in this format, including through the *broom* and *biobroom* packages.	• Describe the structure of R's 'list' data format • Take basic steps to explore and extract data stored in R's complex, list-based structures • Describe what the *broom* and *biobroom* R packages can do • Explain how converting data to a 'tidy' format can improve reproducibility	15	• Applied exercise: Starting with example data in a complex, list-based format, explore the data and extract specified elements, including with the *broom* and *biobroom* packages; • Video providing a detailed walk-through of the solution to this exercise
Example: Converting from complex to 'tidy' data formats	We will provide a detailed example of a case where data pre-processing in R results in a complex, 'untidy' data format. We will walk through an example of applying automated gating to flow cytometry data. We will demonstrate the complex initial format of this pre-processed data and then show trainees how a 'tidy' dataset can be extracted and used for further data analysis and visualization using the popular R 'tidyverse' tools. This example will use real experimental data from one of our Co-Is research on the immunology of tuberculosis.	• Describe how tools like *biobroom* were used in this real research example to convert from the complex data format from pre-processing to a format better for further data analysis and visualization • Understand how these tools would fit in their own research pipelines	20	• Applied exercise: With an example dataset in a complex, 'untidy' data format in R, convert it to a 'tidy' format and create simple plots with this 'tidy' dataset • Video demonstrating a detailed solution to the applied exercise

Table 2: Modules for the second sequence, **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principles** (blue), **Implementation** (red), or **Case study examples** (black). This table is continued over several pages. *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video length (min.)	Extra educational materials
Introduction to reproducible data pre-processing protocols	Reproducibility tools can be used to create reproducible data pre-processing protocols—documents that combine code and text in a 'knitted' document, which can be re-used to ensure data pre-processing is consistent and reproducible across research projects. In this module, we will describe how reproducible data pre-processing protocols can improve reproducibility of pre-processing experimental data, as well as to ensure transparency, consistency, and reproducibility across the research projects conducted by a research team.	• Define a 'reproducible data pre-processing protocol' • Explain how such protocols improve reproducibility at the data pre-processing phase • List other benefits, including improving efficiency and consistency of data pre-processing	15	• Discussion questions: How reproducible data pre-processing protocols can make biomedical research more reproducible at the data pre-processing stage in the trainee's research area • Short audio recording of two Co-Is giving their own answers to these discussion questions
RMarkdown for creating reproducible data pre-processing protocols	The R extension package RMarkdown can be used to create documents that combine code and text in a 'knitted' document, and it has become a popular tool for improving the computational reproducibility and efficiency of the data analysis stage of research. This tool can also be used earlier in the research process, however, to improve reproducibility of pre-processing steps. In this module, we will provide detailed instructions on how to use RMarkdown in RStudio to create documents that combine code and text. We will show how an RMarkdown document describing a data pre-processing protocol can be used to efficiently apply the same data pre-processing steps to different sets of raw data.	• Define RMarkdown and the documents it can create • Explain how RMarkdown can be used to improve the reproducibility of research projects at the data pre-processing phase • Create a document in RStudio using RMarkdown • Apply it to several different datasets with the same format	15	• Applied exercise: Create, save, and render their own RMarkdown document through RStudio • Video providing a detailed walk-through of a solution to the applied exercise
Example: Creating a reproducible data pre-processing protocol	We will walk through an example of creating a reproducible protocol for the automated gating of flow cytometry data for a project on the immunology of tuberculosis lead by one of our Co-Is. This data pre-processing protocol was created using RMarkdown and allows the efficient, transparent, and reproducible gating of flow cytometry data for all experiments in the research group. We will walk the trainees through how we developed the protocol initially, the final pre-processing protocol, how we apply this protocol to new experimental data.	• Explain how a reproducible data pre-processing protocol can be integrated into a real research project • Understand how to design and implement a data pre-processing protocol to replace manual or point-and-click data pre-processing tools	20	• Quiz questions: Test understanding of how and why we created a reproducible data pre-processing protocol for this pre-processing step, and how this improves reproducibility for the research group; • Short video with a detailed discussion of quiz questions

Table 3: Examples of how different types of trainees might use subsets of the training modules to meet their specific training needs.

	Graduate student who would like to learn in detail how to use reproducibility tools for data recording and pre-processing and is willing to learn R programming tools	**Principal investigator** who does not program but would like to learn how his/her research team could improve reproducibility of data recording and pre-processing	**Biostatistician** who would like to understand barriers faced by collaborators in implementing reproducibility principles early in research projects	**Technician** in charge of running and pre-processing mass spectrometry data	**Undergraduate student** who wants an introduction to improving reproducibility of data recording
Improving the Reproducibility of Experimental Data Recording					
• Separating data recording and analysis	Yes	Yes	Yes	No	Yes
• Principles and power of structured data formats	Yes	Yes	No	No	Yes
• The 'tidy' data format: an implementation of a structured data format	Yes	Yes	No	No	No
• Designing templates for 'tidy' data collection	Yes	Yes	No	No	No
• Example: Creating a template for 'tidy' data collection	Yes	Yes	Yes	No	No
• Power of using a single structured 'Project' directory for storing and tracking research project files	Yes	Yes	No	No	Yes
• Creating 'Project' templates	Yes	No	No	No	No
• Example: Creating a 'Project' template	Yes	Yes	Yes	No	No
• Harnessing version control for transparent data recording	Yes	Yes	No	No	Yes
• Enhance the reproducibility of collaborative research with version control platforms	Yes	Yes	No	No	Yes
• Using git and GitLab to implement version control	Yes	No	No	No	No
Improving the Reproducibility of Experimental Data Pre-Processing					
• Principles and benefits of scripted pre-processing of experimental data	Yes	Yes	No	Yes	No
• Introduction to scripted data pre-processing in R	Yes	No	No	Yes	No
• Simplify scripted pre-processing through R's 'tidyverse' tools	Yes	No	No	Yes	No
• Complex data types in experimental data pre-processing	Yes	Yes	Yes	Yes	No
• Complex data types in R and Bioconductor	Yes	No	Yes	Yes	No
• Example: Converting from complex to 'tidy' data formats	Yes	Yes	Yes	Yes	No
• Introduction to reproducible data pre-processing protocols	Yes	Yes	No	Yes	No
• RMarkdown for creating reproducible data pre-processing protocols	Yes	No	No	Yes	No
• Example: Creating a reproducible data pre-processing protocol	Yes	Yes	Yes	Yes	No

Format for the training modules

Online book. We will use an online book to collect all the training materials we develop in a single online document. Each chapter of the book will contain the materials for one of the modules listed in Tables 1 and 2, for twenty chapters total. We have created a prototype (Figure 2) to demonstrate some features of the final book. Users will be able to quickly navigate through chapters with a navigation bar on the left of the webpage, with chapter subsection links opening when a chapter is selected. We will embed a lecture video at the start of the chapter, allowing a user to watch the video content without leaving the book website. This type of format, in which video content is woven together with written text and additional educational materials, has been praised as an effective format for presenting online training materials [57]. The book will include a link for the trainee to download a copy as a PDF or EPUB file to use as a future reference offline if desired. The format also includes buttons that can be used to share the link to the online book with others through Twitter and other platforms, as well as a link to the book's *GitHub* repository, to allow early users of the in-development materials to provide feedback on typos, broken links, unclear materials, and other issues as we develop the materials. We will use *GitHub Pages* [11] to freely post this book online. Dr. Anderson (PI) has previously created two *bookdown*-based books, *R Programming for Research* [12] and *Mastering Software Development in R* [16], and has posted and maintained *R Programming for Research* online through *GitHub Pages* continuously since Fall 2016, providing evidence of the robustness of this method of dissemination.

Video lectures. Since videos can engage online learners better than some other types of learning materials [58], each chapter will include a video lecture that covers the module's material, with the approximate length of each lecture listed in Tables 1 and 2. We will record these video lectures in Colorado State University's Computer Assisted Teaching Support laboratory (see letter from Dr. Andrew West), which includes equipment and staff for creating professional-quality video lectures. We will use YouTube [59] to freely host these videos, which will allow us to embed the video within the text of the module's chapter in the online book (see Figure 2 for an example of how this will look to trainees). Hosting the video lectures through YouTube will allow us to take advantage of YouTube's free, and well-tested platform for sharing videos, as well as allow us to collect detailed analytics on how often each video is watched and for how long, to help us assess the use of this component of the training material.

Additional educational materials. Each module will contain additional educational material, to help the trainee absorb the material and assess his or her mastery of the topics. Depending on the module, this additional content will either be a quiz, questions for discussion, or an applied exercise (see Tables 1 and 2 for the specific material planned for each module). For many of the *Implementation* modules, these extra educational materials will be applied exercises, since providing tutorials, example code, and example datasets can substantially improve the ability of new users to learn software tools [54, 58, 60]. Including quizzes as the additional educational material within some of the modules will help trainees self-evaluate their mastery of the material [58, 60]. We will also include audio and video content walking the trainee through answers and solutions for these additional materials. While not a substitute for in-person interactions, these video and audio discussions help mimic the detailed walk-throughs and discussions that we would do after a student attempted these materials if we were teaching these materials in person. As with the video lectures, we will tape this video and audio content in Colorado State University's Computer Assisted Teaching Support laboratory. We will host the video content through YouTube [59] and the audio content through SoundCloud [61]. We will use Google Forms [62] as a free and unlimited way to create the quizzes and embed them in the online book.

Insuring compliance with Rehabilitation Act. We have plans for making our proposed training module compliant with the Rehabilitation Act, as amended by the Workforce Investment Act of 1998. Much of the online content will be text based. For figures and other images in the book, we will use *alt* and *longdesc* attributes within the image tag to provide a text alternative. Using YouTube to host the lecture videos will allow us to draw on that platform's functionality for accessibility, including "enough time" functionality, in terms of being able to pause and turn off the content. YouTube allows users to add and edit closed caption content on their videos, which we will use to add optional closed captions to this content—in addition to improving accessibility of the content, it may also help trainees for whom English is a second language to follow the video lectures. In the first two years of the grant, our team will include a student hourly who will assist Dr. Anderson in the technical implementation of the online book, and helping to ensure the content is compliant with the Rehabilitation Act will be one of his or her key tasks. If this task requires the use of interesting

techniques or technologies, Dr. Anderson and the student may prepare a journal article describing these techniques and submit it to *The R Journal* during the project period. The R community is very interested in improving accessibility, as evidenced by previous publications, presentations, and software on these topics (e.g., [63, 64]).

C.1 Program Director / Principal Investigator and Project Team

Our core team (Table 4) combines experts in R programming (Anderson, Lyons) with laboratory-based biomedical researchers in microbiology and immunology (Henao-Tamayo, Gonzalez-Juarrero, Robertson) who are **attuned to the needs of and barriers to improving the reproducibility of experimental data collection and pre-processing among laboratory-based biomedical researchers**. Our team will allow us to develop training modules that present state-of-the-art approaches and tools for reproducibility, but do so in a way that is prioritized to be most useful and accessible to biomedical researchers whose training has focused on laboratory-related, rather than computational, research. Our team also includes a to-be-named undergraduate student hourly and Dr. Julie Maertens (Research Associate), a senior evaluator at the Colorado State University Science, Technology, Engineering, and Math (STEM) Center. The student hourly will assist Dr. Anderson in the technical work of publishing the content our team develops in an online book. Dr. Maertens will assist in the design and implementation of project evaluation throughout this project. Dr. Maertens will only be involved in the project to assist in planning and implementing evaluation, and her percent effort is capped at 0.85% to reflect the RFA's budget restriction of $3,000 on program evaluation, including salary support.

Coordination and management of the team

The Principal Investigator and all four Co-Investigators will collaborate to develop the materials in the training modules. We have designed a plan (Table 5) in which, for each module, Dr. Anderson and one of the Co-Investigators will collaborate as the **primary authors** of the written text, lecture slides, and additional education materials (quiz, discussion questions, or applied exercise). A second co-investigator will serve as the **first tester** of the module and will evaluate and test all module material, providing detailed feedback to the two authors of the module to allow them to refine the material. The training materials for the module will then be tested in an in-person Colorado State University pilot testing session (sessions will occur twice per year over the project period, with each session testing the content for approximately five modules), and further refined based on that feedback, as well as on feedback from the two Co-Investigators who did not have roles of author or first tester for that module, and any feedback from early online users. After this refinement based on early feedback, Dr. Anderson will film the video lecture for the module. Throughout the module development period (first two years of the project), Dr. Anderson and the undergraduate student will transfer the developed training content into the online book format and publish it online, and Dr. Anderson will be in charge of maintaining and updating the online material during the third project year. Dr. Anderson will lead the planning and implementation of the Colorado State University pilot testing sessions and other activities related to pilot testing and refinement of the training modules. Dr. Maertens will assist Dr. Anderson at points throughout the project period to develop materials for evaluation, including feedback surveys to use with Colorado State University pilot testers, survey questions to include in the online book for evaluation purposes, and brief training on how to best elicit qualitative feedback from pilot testers during the bi-annual Colorado State University pilot testing sessions. The four co-investigators will play key roles in disseminating the training materials to our key audience, as they are all well-connected within the microbiology and immunology research communities. Throughout the project period, Dr. Anderson, the four co-investigators, Dr. Maertens, and the student hourly will meet as a group at least four times per project year to check in on progress on the project.

Our plan of content development and publishing is ambitious, but we are confident we can meet it. Dr. Anderson previously developed the online content, in collaboration with a co-instructor, for a five-course specialization (*Mastering Software Development with R*, see letter from Dr. Roger Peng) with approximately twice as much content in under nine months. She has experience developing and publishing online training materials for learning R-based tools, including through the *bookdown* online book framework we plan to use here. In this previous development of online training materials, Dr. Anderson learned to collaborate closely with co-authors in developing and refining online educational content and in developing quizzes and applied exercises to allow trainees to test their understanding of that content. She will bring that experience in the

Table 4: Principal and co-investigators on our project team.

Person / role	Description
Brooke Anderson Principal Investigator *Assistant Professor, Dept of Environmental & Radiological Health Sciences*	Dr. Anderson is an expert in R programming and has created and published several open-source R packages, in particular to facilitate environmental epidemiological research. She has experience creating R programs to work with large data, including climate model output and large weather datasets, as well as programs that interface with open web-based datasets. She is the co-instructor of a series of Massive Open Online Courses on *Mastering Software Development in R* through Coursera and an associated open online book.
Michael Lyons Co-Investigator *Assistant Professor, Dept. of Microbiology, Immunology & Pathology*	Dr. Lyons works on the computational biology and pharmacology of tuberculosis infection and treatment in experimental animal models and tuberculosis patients. Prior to joining CSU full-time in 2011, he was a software engineer in the computer industry for 12 years, and prior to that, a theoretical physicist. Through a K25 award, he obtained significant classroom and hands-on training and exposure to laboratory methods related to drug and vaccine development for tuberculosis, providing him with a solid understanding of how preclinical and clinical data are used for evidence-based decision making in the biomedical sciences. He is highly attuned to the problems that this project aims to address, and he has a clear understanding of the practical limitations and challenges for both the laboratory scientist and data analyst. He uses R daily in his academic research
Mercedes Gonzalez-Juarrero Co-Investigator *Associate Professor, Dept. of Microbiology, Immunology & Pathology*	Dr. Gonzalez-Juarrero studies the basic nature of the cell mediated immune response to mycobacteria infections. During the last ten years, her research group has undertaken studies to investigate the emergence of immunosuppression during pulmonary tuberculosis, with the primary goal of learning how and where to target the latently infected host to fully recover the antimicrobial activity of the infected cell, and how to use this information in the context of current chemotherapeutic and multidrug resistant tuberculosis infections. Dr. Gonzalez-Juarrero became particularly interested in how to improve the reproducibility, transparency, and efficiency of experimental data recording within her research projects when she attended Dr. Anderson's CSU course on *R Programming for Research* in Fall 2017 and learned about the principles of structured data formats, including the "tidy" data format now popular with statisticians, and she has begun implementing these principles in her research laboratory.
Marcela Henao-Tamayo Co-Investigator *Assistant Professor, Dept. of Microbiology, Immunology & Pathology, Co-Director of CSU-Flow Cytometry Facility*	Dr. Henao-Tamayo studies the immunopathogenesis of tuberculosis using animal models to evaluate the role of different types of T cells and myeloid-derived cells in tuberculosis and Bacille Calmette Guerin vaccination. She has tested numerous vaccine candidates evaluating the immune response they elicit in association with protection against tuberculosis disease. She is interested in how existing tools for computational reproducibility can be applied to data recording and pre-processing in her own research laboratory, and she and Dr. Anderson (PI) co-advise a graduate student who is integrating open-source R software into the regular practice of Dr. Henao-Tamayo's research work, including through implementation of reproducible automated gating of flow cytometry data.
Gregory Robertson Co-Investigator *Assistant Professor, Dept. of Microbiology, Immunology & Pathology*	Dr. Robertson has more than 20 years of classical and clinical microbiology experience, with an emphasis in antibacterial discovery and mode-of-action studies for novel and existing classes of antimicrobials. This includes efforts in academia, and also with larger pharmaceutical corporations (Eli Lilly and Co) and smaller bio-pharmaceutical groups (Cumbre Pharmaceuticals). His current research is focused on *Mycobacterium tuberculosis* host-pathogen interactions and the development and application of novel preclinical animal models to further anti-tuberculosis drug development and evaluate drug resistance. In the context of improving reproducibility in biomedical research, Dr. Robertson is particularly passionate about the perils of using spreadsheets with embedded formulas as a tool for recording and analyzing experimental data.

Table 5: Roles of our team of investigators in developing the proposed training materials.

	Brooke Anderson	Michael Lyons	Mercedes Gonzalez-Juarrero	Marcela Henao-Tamayo	Gregory Robertson
Improving the Reproducibility of Experimental Data Recording					
• Separating data recording and analysis	Author	Tester			Author
• Principles and power of structured data formats	Author	Author			Tester
• The 'tidy' data format: an implementation of a structured data format	Author		Tester	Author	
• Designing templates for 'tidy' data collection	Author	Tester	Author		
• Example: Creating a template for 'tidy' data collection	Author		Tester		Author
• Power of using a single structured 'Project' directory for storing and tracking research project files	Author		Author	Tester	
• Creating 'Project' templates	Author			Author	Tester
• Example: Creating a 'Project' template	Author		Author	Tester	
• Harnessing version control for transparent data recording	Author	Author			Tester
• Enhance the reproducibility of collaborative research with version control platforms	Author	Tester			Author
• Using git and GitLab to implement version control	Author	Author	Tester		
Improving the Reproducibility of Experimental Data Pre-Processing					
• Principles and benefits of scripted pre-processing of experimental data	Author		Author	Tester	
• Introduction to scripted data pre-processing in R	Author	Author	Tester		
• Simplify scripted pre-processing through R's 'tidyverse' tools	Author		Author		Tester
• Complex data types in experimental data pre-processing	Author		Tester	Author	
• Complex data types in R and Bioconductor	Author	Tester		Author	
• Example: Converting from complex to 'tidy' data formats	Author	Tester		Author	
• Introduction to reproducible data pre-processing protocols	Author	Author			Tester
• RMarkdown for creating reproducible data pre-processing protocols	Author			Tester	Author
• Example: Creating a reproducible data pre-processing protocol	Author			Tester	Author

proposed project, to help organize and integrate the contributions of our team of co-investigators. For the technical development of the book, she will be assisted by and supervise an undergraduate hourly during the first two years of the project, to help in turning the intellectual content that she and the co-investigators develop into the formatted online book.

C.2 Institutional Environment and Commitment

Colorado State University is a Research I institution, with vibrant research programs in a variety of scientific, engineering, and health-related fields. As Colorado's land-grant university, Colorado State University has a 100-year-old extension program to help researchers disseminate their research results to members of the wider community and strongly values its history and continuing commitment to sharing cutting-edge scientific knowledge through outreach education activities. Colorado State University is very supportive of improving the reproducibility of research, and it offers ample resources that will help us ensure the success of the proposed project. **Our institution has clearly expressed its support for our proposed effort to create, refine, evaluate, and disseminate these training materials** in institutional letters of support from Dr. Jac Nickoloff (Chair of the Department of Environmental & Radiological Health Sciences, Colorado State University), Dr. Gregg Dean (Chair of the Department of Microbiology, Immunology, & Pathology, Colorado State University), and Dr. Alan Rudolph (Vice President for Research, Colorado State University). As expressed in the letter of support from Dr. Nickoloff, Dr. Anderson's teaching expectations during the

project period are capped at three courses every two years, allowing adequate time for her to complete the tasks required by this proposal. As also expressed in that letter, Dr. Anderson's department supports that the training materials developed under this grant will be made freely available through online publication under a Creative Commons license. Dr. Anderson's department has supported her previous development and publication of similarly free and open training materials following a similar format [12, 16].

Colorado State University is host to a large number of laboratory-based biomedical researchers, especially in fields related to drug and vaccine development for infectious diseases. We will **take advantage of this environment** to help us pilot test and refine the training materials, to ensure they are clear, relevant, and useful to our target population of laboratory-based biomedical researchers. We have received statements of support from relevant programs—including the director of the Colorado State University Interdisciplinary Graduate Program in Cell & Molecular Biology and the Associate Director of the Interdisciplinary NSF-NRT GAUSSI training program in Computational Biology—to help disseminate our materials to pilot testers and early users of our training materials on-campus at Colorado State University (see letters from Drs. Gregg Dean, Carol Wilusz, and Jeff Wilusz).

Colorado State University provides meeting rooms that we can reserve free-of-charge to use for the Colorado State University-based user testing sessions. The Computer Assisted Teaching Support (CATS) laboratory at Colorado State University's Academy for Teaching and Learning has professional-grade recording equipment that we will use to record the video lectures within each module (see letter from Dr. Andrew West), and Colorado State University's Science, Technology, Engineering, and Math (STEM) Center will provide help in evaluating the training modules, including through the budgeted involvement of Dr. Julie Maertens, a senior evaluator at the center.

C.3 Evaluation Plan

Pilot testing

We will pilot test the training materials using three methods: (1) bi-annual on-campus day-long pilot testing at Colorado State University (one session in year 1 and two each in years 2 and 3, five total over the project); (2) a workshop at the American Society for Microbiology Conference in year 2 of the project; and (3) early online users of the training materials. We expect that these different methods will allow us to collect different sets of feedback on the frequency of use and usefulness of the training materials (Table 6), providing a rich collection of ideas for refining the materials. For each of the modules, we have outlined learning objectives will help us evaluate if the training modules are useful in achieving their educational goals (Tables 1 and 2).

Biannual Colorado State University pilot testing sessions. We will conduct two day-long pilot testing sessions at Colorado State University in each year of the project. The Colorado State University pilot testing participants will include current and future laboratory-based biomedical researchers. We will recruit trainees with a variety of research roles, including undergraduate students, graduate students, postdoctoral fellows, research associates, and principal investigators. We have informed several Colorado State University biomedical researchers about these proposed testing sessions and received their support in encouraging researchers within their groups and departments to participate (see letters from Drs. Gregg Dean, Carol Wilusz, and Jeff Wilusz).

In the first two project years, each session will test the set of modules developed since the last user testing (approximately five modules will be tested in each of these session). The sessions will begin with our team giving live lectures of the same content we plan to film for the video lectures. This will allow us to improve and refine this content, based on detailed feedback from testers representative of our target audience, before filming the final video lectures. During the rest of the session, we will divide the trainees into small teams to work through the additional educational materials (applied exercises, quiz questions, and discussion questions) to iron out problems with the clarity or implementation of these materials. The trainees will have access to the in-development online book as they work through these materials. In the last year of the project, these sessions will revisit the material in modules that proved problematic in their first round of pilot testing, allowing us to re-test approximately ten of the modules (five tested per session in project year 3). To solicit specific feedback from these sessions, we will survey the participants during the sessions (for examples of the types of feedback we will collect through these surveys, see Table 6). Six months after each pilot testing session, we will send participants a follow-up survey, to help evaluate longer-term success

Table 6: Examples of types of feedback we anticipate to generate from pilot testing among different groups to help us refine the training materials.

	CSU pilot testers	ASM workshop participants	Early online users
Characteristics of the trainees?			
• Demographics	Yes	Yes	Yes
• Highest educational degree	Yes	Yes	Yes
• Research role (e.g., principal investigator, research associate, graduate student)	Yes	Yes	Yes
How often the training materials are used			
• How many trainees have accessed the online book?	No	No	Yes
• How are online book users distributed across the U.S.?	No	No	Yes
• How many international trainees have accessed the online book?	No	No	Yes
• How many trainees attended the ASM workshop?	No	Yes	No
• How many trainees attended on-campus CSU piloting?	Yes	No	No
Patterns in use of each module			
• How long do trainees stay on the webpage for the module?	No	No	Yes
• For each module video, how often has it been watched?	No	No	Yes
• When trainees watch a module's video, on average what percent do they watch?	No	No	Yes
• How often are additional educational materials (quizzes, applied exercise materials) used?	No	No	Yes
• How often is the entire book downloaded as a PDF or EPUB file?	No	No	Yes
• Which of the modules are used most frequently?	No	No	Yes
Usefulness of each module			
• What were the trainee's goals in using this training material?	Yes	Yes	No
• Did this module provide the trainee novel information?	Yes	Yes	Yes
• Does the trainee plan to change research practices based on having taken the module?	Yes	Yes	Yes
• Is so, how does the trainee plan to change research practices based on having taken the module?	Yes	Yes	No
• Was the module useful enough that the trainee would recommend it to other scientists?	Yes	Yes	Yes
• Which elements of the training modules (video lecture, written text, additional educational materials) did the trainee find most useful?	Yes	Yes	No
• For each module video, are there spots where it is common for trainees to stop watching?	No	No	Yes
• How did the trainee choose which modules to use?	Yes	Yes	No
• For the modules taken, what content did the trainee wish had been covered but was not?	Yes	Yes	No

of the training in improving the reproducibility of data recording and pre-processing. Dr. Anderson (PI) has experience in productively conducting these kinds of user testing sessions at Colorado State University. She has run several two-hour user testing sessions with students from various departments of Colorado State University prior to releasing R software packages [65, 66]. Further, in April 2016, she led a longer, two-day user testing session through a Weather Data Hackathon at Colorado State University (Figure 3). Some of the ideas and code developed during this Hackathon have since led to development and publication of open source software [67, 68].

American Society for Microbiology pre-conference workshop. The American Society for Microbiology presents pre-conference workshops at its annual conference, for which scientists can submit proposals. We will submit a proposal to lead a day-long workshop at the American Society for Microbiology conference in year 2 of the project. This workshop will cover the content of most of the training modules listed in Tables 1 and 2. It will include live lectures of the materials from the online video lectures, as well as directed

Figure 3: Some of the approximately 15 undergraduate students, graduate students, postdoctoral fellows, and professors who participated in a two-day Weather Data Hackathon at Colorado State University in April 2016 led by Dr. Anderson. Around 15 people participated, including from Colorado State University's Departments of Atmospheric Sciences, Civil & Environmental Engineering, Microbiology, and Statistics.

work through the additional educational materials for each module. As with the Colorado State University pilot testing sessions, we will use surveys to get immediate and follow-up (six months after, to evaluate longer-term outcomes) feedback from the participants of this workshop.

Early online users. We will develop the book openly online from the start of the project, including posting all underlying code online at *GitHub* and posting the current version of the online book through *GitHub Pages*, using *Travis Continuous Integration* to rebuild the book each time we commit changes to its underlying code. Dr. Anderson has used a similar development process with an online coursebook [12]; this open development process helps attract early users and also provides book users with a real-life view of how version control tools and platforms can enable "live" development of a project. In fact, a similar process was used to develop this proposal (https://github.com/geanders/reproducible_modules), following the lead of a previous project funded under this mechanism (https://github.com/riffomonas/RR_R25_2014).

By six months into year 1 of the project, we will have a first version of the online book's written text for at least five modules. At this point, we will actively recruit early online users through social media and through our networks R users (Anderson and Lyons) and biomedical researchers (Gonzalez-Juarrero, Henao-Tamayo, and Robertson) (see letters from Drs. Roger Peng and Jorge Henao-Mejia). We will enable Google Analytics [69] on the online book's webpage, which will allow us to track how many people access the online book, how often they stay on the book's page, as well as the distribution of these early users across the United States and the world. Similarly, analytics from the platforms we use to host embedded material (YouTube [59], SoundCloud [61], and Google Forms [62]) will allow us to track the use of those materials. We will use Google Forms [62] to create voluntary surveys within the online book, to generate additional feedback from these early users in terms of their characteristics and on the usefulness of the modules. While this voluntary survey will not provide data on *all* early online users, it will help characterize a collection of the users. We are selecting to make the survey voluntary (rather than required to access the book) in hopes of making the book as accessible as possible, and to avoid discouraging users who might be drawn in by browsing the materials before committing to provide their information or feedback.

External program evaluation

An external evaluation of the proposed training modules program will be conducted by Julie Maertens in the Colorado State University STEM Center, which facilitates collaborations among STEM-related education and outreach projects within and outside Colorado State University. Dr. Maertens has conducted several large-scale, multi-site youth program evaluations in conjunction with the State of Colorado, and also regularly works with researchers in- and outside of Colorado to implement STEM education and outreach

evaluations. She currently leads the evaluation of three federally funded projects designed to: (1) Create a pre-service secondary teacher program that integrates the requirements for traditional engineering undergraduate degree programs; (2) Improve recruitment and persistence of women in the geosciences using a deliberate mentorship approach, and (3) Improve recruitment and retention of underrepresented minorities in computational biology and genomics by creating and conducting week-long workshops in Todos Santos, Mexico.

The conceptual framework for the evaluation will be organized using the CIPP model, which is designed to guide project decision-making based on assessment of a programs context, input, process and product [70]. The operational framework will use observable data to conduct both formative [F] and summative [S] evaluation of the program, and will focus on 2 main areas of the conceptual evaluation model to determine: (1) How well the program is implemented (*process*), and (2) How well the proposed modules and training activities meet the program objectives (*product*).

Evaluation data will include baseline program metrics as well as measures to gauge the short- and long-term success of the proposed training modules in achieving program objectives. Examples of data to be collected include:

- Number, educational level, and demographics of online module users (where available; survey participation is optional)
- Brief post-course evaluation of module usefulness (e.g., accessibility, content, goals, structure, overall experience) among online module users (where available; survey participation is optional)
- Number, educational level, and demographics of in-person test users
- In depth, 15-point post-course evaluation of module usefulness among in-person test users
- Follow-up surveys among in-person test users to determine future usefulness of the modules (e.g., how and whether module content is applied to future research activities)
- Project team implementation surveys

In addition to the proposed data collection, Dr. Maertens will train the project team to collect qualitative data among test users during the day-long user testing sessions (on-campus at Colorado State University and at the American Society for Microbiology workshop), and how to use a qualitative content analysis process to extract themes related to feedback (e.g., strengths, weaknesses) about the modules. These themes may be used, along with the evaluation survey data, to make iterative improvements to the modules each year.

Survey results will be presented annually via written report, and will summarize all findings and iterative program changes and provide recommendations for future programming. See Table 7 for evaluation questions and methods.

Table 7: External evaluation questions, methods, and timeline.

	Evaluation Questions & Function	**Data Collection**	**Timing**
Process	To what extent is the program implemented as proposed? *(Formative [F] & Summative [S])*	• Project team surveys to determine the extent to which the program activities are implemented as planned, whether activities take place within the proposed time frame, what barriers to implementation are encountered, and whether and how data collected are used to make improvements or refinements to the modules	Years 1–3
Product	How well does implementation of the training modules meet the program objectives? *(Formative [F] & Summative [S])*	• Survey data collected to understand whether and how well the training modules impact short- and long-term participant learning and utility outcomes of interest	Years 1–3

C.4 Dissemination Plan

Using *GitHub Pages* [11], we will publish the book freely online, with all materials published under the Creative Commons Attribution-ShareAlike 4.0 International License, **making all materials freely accessible, both nationally and internationally**. From the beginning of the project, we will publish the book online as it develops, and we will promote this material through social media (e.g., Twitter) and through our network of colleagues in biomedical research and the R programming community (for example, see letters from Drs. Roger Peng and Jorge Henao-Mejia). Since this book will be hosted online, it will be easy to link to from the National Institute of General Medical Sciences' *Clearinghouse for Training Modules to Enhance Data Reproducibility* [17]. There will be no paywall or other restriction on accessing any of the training materials, and source code for the book and exercises will be published on *GitHub*. All video and audio content will be published online in free formats through YouTube [59] and SoundCloud [61], with the content embedded in the online book (see Figure 2). At the end of project years 2 and 3, we will also post a static version of the book to the website *bookdown.org*, where people can go to find free online books published using the *bookdown* format and which invites direct submissions from authors that have used these framework.

In addition to these methods of disseminating the training materials to a general audience, **we will also take specific steps to make sure that our target audience—laboratory-based biomedical scientists—are aware of these training materials.** We will apply to present posters or oral presentations in years 2 and 3 of the project at three national and international conferences (American Society for Microbiology Conference, American Association of Immunologists Meeting, and International Society for the Advancement of Cytometry) to help get out the news among our target audience that these materials are freely available. We will also invite colleagues (and their research group members) at and outside of Colorado State University to serve as early online users of the our materials (see letters from Drs. Gregg Dean, Carol Wilusz, Jeffrey Wilusz, and Jorge Henao-Mejia). In addition to providing us with feedback to help refine our materials, this will help us disseminate the materials. Finally, we will write and submit a paper describing these training materials and highlighting their content in a biomedical journal relevant to our target audience.

The PI has previously had substantial success in disseminating online training materials. She is the co-instructor of a five-course specialization on *Mastering Software Development in R* through the Massive Open Online Course platform Coursera. This series has had over 50,000 participants since it was opened in Fall 2016, and an accompanying online book on the LeanPub platform has been downloaded by over 14,000 people.

C.5 Program Participants

Here we propose to develop training modules that are freely available nationally and internationally. Unlike traditional, on-site training or educational programs, we do not have a constrained program participant group that we plan to enroll in the final program funded by this award, which we consider use of the online training materials. However, we anticipate that the key participant group in this program, in terms of accessing and using the developed online materials, will be laboratory-based biomedical researchers interested in improving reproducibility within their laboratories. Table 3 gives some examples of different types of researchers we hope will participate in this program through use of the final online materials. Given the open dissemination of these materials, other participants could include researchers from other fields or people from outside academic research. We anticipate that at least a few hundred people will participate in using the final training materials, although based on our previous work in disseminating training materials openly online, the number of participants could reach several thousand.

C.6 Potential Pitfalls and Alternative Plans

We have high confidence that we will be able to successfully develop, test, refine, and disseminate the training materials as outlined in this proposal, based on our previous success with similar projects [16, 12]. However, we have considered alternative plans in case there are complications. Specifically, if there are any problems in disseminating the full collection of training materials using the *bookdown* format and with free web hosting through *git Pages*, we will explore using the *DataCamp* platform [71] to host the training materials. This is a well-developed online learning platform, and it allows academics to develop and post their own training materials through "Course Editor". Content that is created and published through the "Community" section of DataCamp are freely accessible to anyone [72].

Figure 4: Timeline for proposed activities for this project. The vertical solid red lines show the start and end of the project period, while the vertical dotted red lines separate each project year. All training materials will be freely and publicly available online by the end of the second project year.

References

[1] U.S. Department of Health and Human Services, National Institutes of Health. NIH-Wide Strategic Plan, Fiscal Years 2016-2020: Turning Discovery Into Health. 2016. URL https://www.nih.gov/sites/default/files/about-nih/strategic-plan-fy2016-2020-508.pdf. Accessed: 2018-06-24.

[2] U.S. Department of Health and Human Services, National Institutes of Health. NIH Strategic Plan for Data Science. 2018. URL https://datascience.nih.gov/sites/default/files/NIH_Strategic_Plan_for_Data_Science_Final_508.pdf. Accessed: 2018-06-24.

[3] Karl W Broman and Kara H Woo. Data organization in spreadsheets. *The American Statistician*, 72(1): 2–10, 2018.

[4] Sanjay Krishnan, Daniel Haas, Michael J Franklin, and Eugene Wu. Towards reliable interactive data cleaning: A user survey and recommendations. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, page 9. ACM, 2016.

[5] Shannon E Ellis and Jeffrey T Leek. How to share data for collaboration. *The American Statistician*, 72 (1):53–57, 2018.

[6] Jennifer Bryan. Excuse me, do you have a moment to talk about version control? *The American Statistician*, 72(1):20–27, 2018.

[7] Ben Marwick, Carl Boettiger, and Lincoln Mullen. Packaging data analytical work reproducibly using R (and friends). *The American Statistician*, 72(1):80–88, 2018.

[8] Yihui Xie. *Bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall, 2016.

[9] Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall, Boca Raton, 2016.

[10] Hadley Wickham and Garrett Grolemund. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media, 2016.

[11] GitHub Pages. https://pages.github.com. Accessed: 2018-06-24.

[12] Brooke Anderson and Rachel Severson. R Programming for Research. https://geanders.github.io/RProgrammingForResearch/. Accessed: 2018-06-24.

[13] Bookdown. https://bookdown.org. Accessed: 2018-06-24.

[14] Karthik Ram. Git can facilitate greater reproducibility and increased transparency in science. *Source Code for Biology and Medicine*, 8(1):7, 2013.

[15] Marcus R Munafò, Brian A Nosek, Dorothy VM Bishop, Katherine S Button, Christopher D Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J Ware, and John PA Ioannidis. A manifesto for reproducible science. *Nature Human Behaviour*, 1:0021, 2017.

[16] Roger Peng, Sean Kross, and Brooke Anderson. Mastering Software Development in R. https://bookdown.org/rdpeng/RProgDA/. Accessed: 2018-06-24.

[17] National Institute of General Medical Sciences: Clearinghouse for Training Modules to Enhance Data Reproducibility. https://www.nigms.nih.gov/training/pages/clearinghouse-for-training-modules-to-enhance-data-reproducibility.aspx. Accessed: 2018-06-24.

[18] Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10):1–23, 2014.

[19] Zev Ross, Hadley Wickham, and David Robinson. Declutter your R workflow with tidy tools. *PeerJ Preprints*, 5:e3180v1, 2017.

[20] Julia Silge and David Robinson. tidytext: Text mining and analysis using tidy data principles in R. *The Journal of Open Source Software*, 1(3), 2016.

[21] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer, New York, 2nd edition, 2016.

[22] Julia S Stewart Lowndes, Benjamin D Best, Courtney Scarborough, Jamie C Afflerbach, Melanie R Frazier, Casey C O'Hara, Ning Jiang, and Benjamin S Halpern. Our path to better science in less time using open data science tools. *Nature Ecology & Evolution*, 1(6):160, 2017.

[23] Allan M Reviewer-Miller. Review of 'R for Data Science: Import, Tidy, Transform, Visualize, and Model Data' by Hadley Wickham and Garrett Grolemund. *ACM SIGACT News*, 48(3):14–19, 2017.

[24] Amelia McNamara. On the state of computing in statistics education: Tools for learning and for doing. *arXiv preprint arXiv:1610.00984*, 2016.

[25] Stephanie C Hicks and Rafael A Irizarry. A guide to teaching data science. *The American Statistician*, In Press, 2017. doi:10.1080/00031305.2017.1356747.

[26] Ben Baumer. A data science course for undergraduates: Thinking with data. *The American Statistician*, 69(4):334–342, 2015.

[27] Daniel Kaplan. Teaching stats for data science. *The American Statistician*, 72(1):89–96, 2018.

[28] Julian Stander and Luciana Dalla Valle. On enthusing students about big data and social media visualization and analysis using R, RStudio, and RMarkdown. *Journal of Statistics Education*, 25(2):60–67, 2017.

[29] Benjamin S Baumer, Daniel T Kaplan, and Nicholas J Horton. *Modern Data Science with R*. CRC Press, Boca Raton, 2017.

[30] Rafael A. Irizarry and Michael I. Love. *Data Analysis for the Life Sciences with R*. Chapman and Hall, 2016.

[31] G Wilson. Software Carpentry: lessons learned. *F1000Research*, 3:62–62, 2014.

[32] Aleksandra Pawlik, Celia WG van Gelder, Aleksandra Nenadic, Patricia M Palagi, Eija Korpelainen, Philip Lijnzaad, Diana Marek, Susanna-Assunta Sansone, John Hancock, and Carole Goble. Developing a strategy for computational lab skills training through Software and Data Carpentry: Experiences from the ELIXIR Pilot action. *F1000Research*, 6:ELIXIR–1040, 2017.

[33] Julia Silge and David Robinson. *Text Mining with R: A Tidy Approach*. O'Reilly Media, Sebastopol, 2017.

[34] Paul J McMurdie and Susan Holmes. phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PloS one*, 8(4):e61217, 2013.

[35] Taylor Arnold. A Tidy Data Model for Natural Language Processing using cleanNLP. *The R Journal*, 9 (2):248–267, 2017.

[36] Sam Tyner, François Briatte, and Heike Hofmann. Network Visualization with ggplot2. *The R Journal*, 9(1):27–59, 2017.

[37] TC Hsieh, KH Ma, and Anne Chao. iNEXT: an R package for rarefaction and extrapolation of species diversity (Hill numbers). *Methods in Ecology and Evolution*, 7(12):1451–1456, 2016.

[38] Tengfei Yin, Dianne Cook, and Michael Lawrence. ggbio: an R package for extending the grammar of graphics for genomic data. *Genome Biology*, 13(8):R77, 2012.

[39] Hilary Parker. Opinionated analysis development. *PeerJ Preprints*, 5:e3210v1, 2017.

[40] Ashley Shade and Tracy K Teal. Computing workflows for biologists: a roadmap. *PLoS Biology*, 13(11): e1002303, 2015.

[41] R Studio Projects. https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects, . Accessed: 2018-06-24.

[42] R Studio Project Templates. https://rstudio.github.io/rstudio-extensions/rstudio_project_templates.html, . Accessed: 2018-06-24.

[43] Stephen R Piccolo and Michael B Frampton. Tools and techniques for computational reproducibility. *GigaScience*, 5(1):30, 2016.

[44] Mine Çetinkaya-Rundel and Colin Rundel. Infrastructure and tools for teaching computing throughout the statistical curriculum. *The American Statistician*, (In Press), 2017.

[45] Cyril Pernet and Jean-Baptiste Poline. Improving functional magnetic resonance imaging reproducibility. *Gigascience*, 4(1):15, 2015.

[46] Roger D Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011.

[47] Wolfgang Huber, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, et al. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, 12(2):115–121, 2015.

[48] C Titus Brown, Alexis Black Pyrkosz, and Likit Preeyanon. Reproducible bioinformatics research for biologists. In *Implementing Reproducible Research*, pages 205–238. Chapman and Hall/CRC, 2014.

[49] Benjamin S Baumer. Lessons from between the white lines for isolated data scientists. *The American Statistician*, 72(1):66–71, 2018.

[50] David Robinson. broom: An R package for converting statistical analysis objects into tidy data frames. *arXiv preprint arXiv:1412.3565*, 2014.

[51] Robert C Gentleman, Vincent J Carey, Douglas M Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80, 2004.

[52] Andrew J. Bass, David G. Robinson, Steve Lianoglou, Emily Nelson, John D. Storey, and with contributions from Laurent Gatto. *biobroom: Turn Bioconductor objects into tidy data frames*, 2017. URL https://github.com/StoreyLab/biobroom. R package version 1.8.0.

[53] Neil CC Brown and Greg Wilson. Ten quick tips for teaching programming. *PLoS Computational Biology*, 14(4):e1006023, 2018.

[54] Markus List, Peter Ebert, and Felipe Albrecht. Ten simple rules for developing usable software in computational biology. *PLoS Computational Biology*, 13(1):e1005265, 2017.

[55] Stephen Altschul, Barry Demchak, Richard Durbin, Robert Gentleman, Martin Krzywinski, Heng Li, Anton Nekrutenko, James Robinson, Wayne Rasband, James Taylor, et al. The anatomy of successful computational biology software. *Nature Biotechnology*, 31(10):894–897, 2013.

[56] Hadley Wickham. *R packages: Organize, Test, Document, and Share Your Code*. O'Reilly Media, 2015.

[57] David B Searls. An online bioinformatics curriculum. *PLoS Computational Biology*, 8(9):e1002632, 2012.

[58] David B Searls. Ten simple rules for online learning. *PLoS Computational Biology*, 8(9):e1002631, 2012.

[59] Youtube. https://www.youtube.com. Accessed: 2018-06-24.

[60] Allegra Via, Javier De Las Rivas, Teresa K Attwood, David Landsman, Michelle D Brazas, Jack AM Leunissen, Anna Tramontano, and Maria Victoria Schneider. Ten simple rules for developing a short bioinformatics training course. *PLoS Computational Biology*, 7(10):e1002245, 2011.

[61] Soundcloud. https://soundcloud.com. Accessed: 2018-06-24.

[62] Google forms. https://www.google.com/forms/about, . Accessed: 2018-06-24.

[63] Josh London. *uswebr: RMarkdown template based on the U.S. Web Design Standards*, 2018. R package version 1.5.

[64] A Jonathan R Godfrey. Statistical software from a blind person's perspective. *R Journal*, 5(1):73–79, 2013.

[65] Brooke Anderson, Colin Eason, and Elizabeth Barnes. *futureheatwaves: Find, Characterize, and Explore Extreme Events in Climate Projections*, 2017. R package version 1.0.3.

[66] Rachel Severson and Brooke Anderson. *countyweather: Compiles Meterological Data for U.S. Counties*, 2016. URL https://CRAN.R-project.org/package=countyweather. R package version 0.1.0.

[67] Rod Lammers and Brooke Anderson. *countyfloods: Quantify United States County-Level Flood Measurements*, 2017. URL https://CRAN.R-project.org/package=countyfloods. R package version 0.0.2.

[68] Brooke Anderson and Ziyu Chen. *noaastormevents: Explore NOAA Storm Events Database*, 2017. URL https://github.com/zailchen/noaastormevents. R package version 0.1.0.

[69] Google Analytics. https://www.google.com/analytics/, . Accessed: 2018-06-24.

[70] Daniel L Stufflebeam. The CIPP model for evaluation. In Thomas Kellaghan and Daniel L Stufflebeam, editors, *International Handbook of Educational Evaluation*, pages 31–62. Kluwer Academic Publishers, Dordrecht, 2012.

[71] DataCamp. https://www.datacamp.com. Accessed: 2018-06-24.

[72] Authoring in DataCamp. https://authoring.datacamp.com. Accessed: 2018-06-24.