Research Education Program Plan

Significance

Importance of the problem to be addressed

Insuring that biomedical research is performed in a rigorous and reproducible manner is critical to the advancement of science and improvement of human health. Significant emphasis has been placed on improving the rigor and reproducibility of laboratory science by improving the description of protocols, confirming the authenticity of strains and reagents, and improving experimental design (1–4). Implementing rigorous and reproducible practices in the analysis of the resulting experimental data has not received the same level of attention. This is perhaps because bench scientists receive extensive training in how to do laboratory techniques and the discussion of improving rigor and reproducibility fits nicely into traditional laboratory training. Laboratory training has not incorporated data analysis skills into existing courses and training programs have been slow to develop or adopt stand alone data science courses (5). Although many programs may require a course in statistics, these typically focus on experimental design and choosing the appropriate statistical tests. They rarely discuss data management, data curation, data visualization, or data dissemination (6, 7). Furthermore, faculty who appreciate that they need to develop these skills themselves have limited availability to do so. Given the heightened emphasis on rigor and reproducibility and the broad adoption of technologies that generate massive datasets, there is a great need for these skills. Because trainees' time is limited for activities outside of developing laboratory skills or for what is relevant to learning the background literature of their sub-discipline, training in data science skills has been neglected. To overcome this problem, workshops (also referred to as boot camps, short courses, or short form training) have been grown in popularity because they can provide a lot of information in a short period of time. For 2016, it was estimated that NIH and NSF provided such programs $27.8 million (8). Learners have positive experiences in these workshops and rate the value of the material highly (9–12). Yet, it is necessary to ask whether these intensive training activities are effective. A 2017 analysis measured learning outcomes and found that such activities did not have a statistically significant effect on learning (8). **Given the considerable time and financial investments made in these activities, it is important that we find effective means of training scientists the best practices in performing reproducible data science.** Furthermore, if poor data analysis practices persist, they will continue to undercut the rigor and reproducibility of biomedical research. This problem is central to the RFA that this proposal is in response to, which calls for the development of "exportable training modules with the potential to enhance the scientific rigor, reproducibility, and responsible conduct of biomedical data science research, and to provide for communication and coordination of the development and deployment of such modules."

Rigor of the Prior Research Supporting the Proposed Research Education Program

The reproducibility crisis. At the height of the "reproducible research crisis" there was concern that most biomedical research was not reproducible (13, 14). Ironically, the reports that heralded this crisis did not provide the level of rigor and transparency that they decried in the studies they claimed could not be reproduced. Less thoughtful commentators might see the lack of reproducibility as a sign of academic misconduct or that a result was incorrect. Others called such work "sloppy" (15). The reality is that academic misconduct is rare, reproducible research can be wrong, and *everyone* struggles to ensure that their work is reproducible. Performing reproducible research practices is hard. The descriptions of a reproducibility crisis put a much needed spotlight on well known difficulties within biomedical research (4, 15–18). These led to a renewed effort to improve the rigor and reproducibility across the lifecycle of a research project (1). As highlighted at the NIGMS Clearinghouse for Training Modules to Enhance Data Reproducibility, NIH has supported the development of instructional modules. This includes one that we generated for the field of microbiome research (19). **To improve reproducibility in science, the training must improve.**

Defining reproducibility. There is a general understanding that a result is reproducible if others can obtain the same result as the original researchers. Yet reproducibility and replicability are often used interchangeably or with different definitions (1, 15, 20–23). It is impossible to discuss improving "reproducibility" if the term is poorly defined. We previously described a framework for outlining how to think about reproducibility within the field of microbiome research

Table 1. System for defining concepts related to reproducibility

Methods	Same dataset	Different datasets
Same methods	Reproducibility	Replicability
Different methods	Robustness	Generalizability

that is easily generalized to other disciplines (Table 1) (24). We can also think of this framework in terms of the data analysis step of a research project (23).

Briefly, if someone were to take data and methods and generated the same results as the original researchers, then the results would be **reproducible**. While reproducibility should always be achieved, failure to generate a replicable, robust, or generalizable result is not necessarily a failure. If they used the same methods to test a hypothesis using data collected from different populations and got the same results, as one would do in a meta analysis, it would be **replicable**. Failure to replicate a result could indicate that there is some underlying variable that distinguishes the different datasets that needs to be better understood and could point to important biological phenomena. If different methods were applied to the same experimental system and the same results were produced, the result would be **robust**. Since not all methods are equally valid and may make different assumptions, failure to achieve a robust result is not necessarily a failure. In fact, as will be discussed next, subjective decisions that are made in an analysis can have a large impact on the conclusions of an analysis (25–28). Finally, if different methods were applied to multiple datasets, the results would be considered **generalizable**. Such results are relatively rare and point to overarching theories that drive science (e.g. climate change, evolution). Failure to generalize a result usually indicates that the underlying hypothesis is incorrect.

As an example, Silberzhan et al. (25) performed a study that tested the reproducibility and robustness of the hypothesis that soccer referees are biased against players with darker skin tones. They recruited 29 teams of data scientists to analyze the same dataset to measure the bias and determine whether it was statistically significant. Although the methods varied by team, all of the methods were peer reviewed at multiple stages of the study. This level of oversight and transparency is not typical or practical for most studies, but points to the importance the teams placed on reproducibility. In spite of their individual reproducibility, the variation in study design reflected the teams' subjectivity and the peer review. The teams produced odds ratios ranging from 0.89 to 2.93. That 69% of the teams found a significant odds ratio suggests that the relationship between bias and skin tone was robust to differences in methodology. As the authors pointed out, the decisions made at each step are subjective and may have oversized impact on the conclusions of the studies. The key factor in this analysis is that because the individual analyses were reproducible and the methods were transparent, it was possible to understand how subjective decisions affected the robustness of the results. **Insuring that a result is reproducible is challenging; however, by following best practices we can understand the technical and biological reasons why a result fails to be reproducible, replicable, robust, or generalizable.**

Reproducibility. As we move forward through this proposal, ***reproducibility*** will be used in the sense of Table 1. Just as lacking reproducibility should not invalidate a result, a reproducible analysis is not necessarily correct because there may be limitations in the data and methods (22). However, if an incorrect result is obtained from a reproducible analysis, it is much easier to find and correct the problem and test the effect of the error on the final result. The past 10 years has seen a significant growth in the availability of tools and resources to help improve the reproducibility of data analyses (6, 7, 9, 11, 19, 29–33). We have found that assuming the analyst will need to reproduce their work in six months is a strong motivator to think more diligently about reproducibility. A six month gap is a good motivator because it is likely that any scientist will have one such gap in their analysis such as between when they finalized their analysis plans and when reviews come back from referees who ask for additional or different analyses to be done. There may be another such gap between when the author is done with the study and when readers start to ask questions about the analysis. If an analyst considers themselves six months from now as a muse, they will be more likely to ensure that the analysis is reproducible by their collaborators and third parties.

To highlight the challenges of reproducibility, Philip Bourne challenged researchers to reproduce his co-authors' 2010 study "The *Mycobacterium tuberculosis* drugome and its polypharmacological implications" (34). It is important to note that this study was performed before many of the tools used today to ensure reproducibility were popularized. The team of researchers attempting to reproduce the original work interacted with Bourne and his team to resolve questions. Bourne and his team have an excellent reputation as being concerned with reproducibility and conducting rigorous research. By the time the re-analysis was completed, the team estimated that it would take someone with basic bioinformatics skills 160 hrs to decipher the analysis and another 120 hrs to implement and execute the re-analysis (35). This would represent 13% of someone's annual effort to reproduce an analysis. This is a considerable cost and does not even consider the cost of data storage and processing, how much longer it would have taken if Bourne and colleagues were not as helpful, or

the cost of independent researchers repeating the same process. Although the re-analysis of the original study was ultimately reproducible, it was only with great expense. In hindsight, the re-analysis effort may have been easier had the original authors had started with the assumption that their analysis would be reproduced by someone in the future. **Through the development and use of improved tools, data scientists are now in a better position to insure the reproducibility of their data analyses than 10 years ago.**

Why reproducibility? There are three reasons that scientists should be concerned about the reproducibility of their research. The first is a "negative" reason. If a result is not reproducible, then other researchers will doubt the rigor of the original study and there will be cynical concerns of misconduct. Furthermore, failure to reproduce an earlier finding is expensive and consumes scarce resources. The second is a neutral reason. One of the more intriguing perspectives on the reproducibility crisis in data science was a call to see efforts to improve reproducibility as "preventative medicine" (22). The analogy suggests that using reproducible data analysis practices will help researchers better identify any problems that occur in their analysis. The third is a positive reason and what we consider the most salient. Reproducibility is important because scientists should want others to build upon their work. Another researcher cannot extend a result if they cannot reproduce the initial result. They also cannot apply innovative methods if they cannot reproduce the work. If scientists want their research to have the maximum impact, it must be reproducible. **This philosophy is central to the proposed Research Education Program: analysts need to ensure openness and reproducibility at every stage of their analysis so that they or anyone else can reproduce and then build upon the work.**

Necessary data analysis skills. As described above, preparing a data analysis to ensure that it is reproducible by yourself or others six months from now is not trivial. Software packages, databases, and operating systems change over time and may cause results to change or cause code from the original analysis to break. For researchers analyzing their data using a graphical user interface (GUI; e.g. Microsoft Excel, GraphPad Prism), ensuring reproducibility with such tools requires painstakingly documenting every operation. For those who use a scripting language (e.g. R or Python), the code can become part of the documentation, but it can also be written in a manner that is impossible to read and reproduce. Of course, accessibility to the raw data, code, and documentation is critical. Unfortunately, in some subfields this level of openness is not widely practiced. It may also be impractical to make protected, proprietary, or large datasets available. **The challenges of reproducibility are both cultural and technical.**

Someone wanting to become a data scientist needs to know how to *program*, but they also need to understand *project organization*, *data visualization*, *statistics*, *version control*, and *automation*. This "stack" of skills is overwhelming to most people setting out to engage in reproducible data science practices. In earlier work, we developed the Riffomonas project to lead microbiome scientists through the development of these skills (19). The proposed project will build off of the earlier Riffomonas platform to generalize the concepts to other biomedical and general science fields. In a Commentary that came from that work, Schloss proposed an aspirational rubric for how researchers could grade the reproducibility of a study (24). The practices outlined in the rubric highlight the skills that a scientist needs to develop to perform reproducible analyses

- Handling of confounding variables
- Sex/gender as confounding variables
- Experimental design considerations
- Data analysis plan
- Clarity of software descriptions
- Availability of data products
- Availability of metadata
- Data analysis organization
- Availability of data analysis tools
- Documentation of data analysis workflow
- Use of random number generator seed
- Defensive data analysis
- Insuring short and longterm reproducibility
- Open science to foster reproducibility
- Transparency of data analysis

Each of these practices were associated with a grade of “good”, “better”, or “best” depending on how the scientist answered specific questions. For example, under the “Documentation of data analysis workflow” practice:

- **Good: Is our code well documented? Do we use a self-commenting coding practice?** To get this grade, one would need to be able to program and use its commenting system along with using descriptive function and variable names.
- **Better: Do each of our scripts have a header indicating the inputs, outputs, and dependencies? Is it documented how files relate to each other?** This grade requires a more extensive use of commenting and organization.
- **Best: Are automated workflow tools like GNU Make and CommonWL used to convert raw data into final tables, figures, and summary statistics?** A grade of best requires the use of an automation tool and a high level of organization across the project in addition to well documented code.

These questions demonstrate that the ability to answer “yes” is often dependent on using multiple tools. Considering it is an *aspirational* rubric, the expectation is not that every practice be rated “best”. Rather, the goal should always be at least “good” and the scientist should be striving to move to the “better” and “best” grades. **This rubric requires both the basic knowledge of how to use the tools, but also the skill to integrate tools to achieve a goal.**

Teaching data analysis skills. Regardless of the challenges, we have found that as traditionally-trained bench scientists are expected to do more of their own analysis using ever growing datasets, they are ill-equipped to employ modern approaches to maximize reproducibility.

Teaching the self-learner. The past 10 years has seen an explosion in the availability of materials to help people learn to analyze data. Much of these materials have been targeted to self-learners. This is evident in the myriad books teaching people programming and data analysis skills using languages such as R and Python, the popularity of websites such as Stack Overflow (https://stackoverflow.com) that answer programming questions, the accessibility of online tutorials through for profit companies including DataCamp (https://www.datacamp.com) and Codecademy (https://www.codecademy.com), and in the number of YouTube tutorial videos. We have created our own online content as text and videos covering topics related to reproducible research practices (https://www.riffomonas.org). Although these resources are generally excellent, there are multiple challenges for the self-learner (36). The first is deciphering what they need to learn and how sift through the various opinions to learn best practices. The second is how to piece together tools from different areas of data science to perform a complete analysis since most tutorials focus on teaching a single concept rather than on how the concept fits in with other concepts. The third is that by definition, an self-learner lacks a community in which to develop, correct, and strengthen their new skills. These challenges likely limit the progress of bench scientists trying to develop data analysis skills leading them to persist in their skills that limit reproducibility.

Teaching in workshops. Simultaneously, the traditional spread out format of a traditional semester-long course has been converted into a concentrated format giving rise to workshops and bootcamps. Organizations such as The Carpentries (also known as Software Carpentry and Data Carpentry) have popularized the use of workshops to introduce data analysis skills to learners in a concentrated format. These workshops are popular because they are free, require a minimal time commitment from the learner (i.e. 2 to 3 days), and offer a welcoming and inclusive environment (9, 12). Software and Data Carpentry workshops cover command line tools, programming, and version control. Other, for profit, bootcamps and university-based certificate programs, require a more extensive time commitment and often assume a foundation in math, statistics, and programming. For the past 12 years, we have taught our own 3-day data analysis workshops covering reproducible research topics for scientists studying the host-associated and other microbiomes. An important study from Feldon et al. (8) acknowledged the popularity of workshops, but wanted to know whether the workshop model was effective for long-term retention of the material. Using a cohort of 294 life sciences PhD students they assessed skill development, productivity, and socialization among students 1 and 2 years after joining their graduate program. Among those students who participated in a boot camp or bridge program designed to enhance data analysis and writing skills and acclimate students to academia prior to starting

graduate school, there were no significant benefits of the training relative to those that did not participate in a program. This result was jarring, but aligns with anecdotal evidence of past workshop learners who comment that they are taking a workshop covering the same content for the second or third time. It also aligns with the education literature, as discussed below.

Teaching via live coding online. Data science workshops can be taught in a variety of formats ranging from lectures where code is discussed and shown to live coding demonstrations where the instructor teaches as they interact with a computer and make time for learners to parallel their activities. The latter approach is far more active and produces a better experience and learning outcomes (37, 38). Beyond slowing down the delivery of content, this process of live coding has a few benefits. First, learners can see the instructor make mistakes and watch as the instructor diagnoses problems and works through the solutions. This also normalizes mistakes so that learners realize that even experienced programmers make mistakes. Second, in a live coding environment there are opportunities for learners to ask the instructor questions that pull the instructor off their script. This customizes the training the learner receives. The popularization of online video tutorials available on YouTube, Vimeo, and Twitch offer a virtualized version of the live coding instruction delivered in a workshop. Surprisingly the number of videos posted to these sites on topics related to reproducible research is relatively low. These videos generally fall into two categories. The first category is represented by short tutorials that cover the syntax of a command. Such videos are demonstrations of materials commonly found in reference materials. There is little integration with other tools or concepts. There is no opportunity for viewers to engage and practice the material. The second category is represented by longer videos that are recordings of someone working on a project. These videos are primarily demonstrations and performances rather than instruction. In rare cases, these are live streamed and viewers can comment and ask questions of the presenter. Future viewers can watch a recording of the exchanges. Similar to the short tutorials in the first category, there is no opportunity for the viewer to practice the concepts with new material. **Live coding online is an approach that has yet to be used to effectively teach reproducible data analysis skills.**

There is nothing inherently wrong with the materials that are available to scientists wanting to learn how to analyze data. The challenge is that too much is asked of these approaches. A self-learner struggles to advance because they do not know enough to know what to study - a grounding provided by workshops. Conversely, the workshop-based approach suffers because the learner does not continue to practice the material that was introduced during the workshop. **These approaches likely fail because they do not incorporate the extensive lessons from the cognition literature, which emphasize the value repeated practice over mass learning (39–42).**

Code Club. A common strategy for keeping up with the literature is participating in journal clubs, which involve group discussion of a pre-selected paper. In addition to staying current on the literature, journal clubs help strengthen skills in critical thinking, communication, and integrating the literature (43). Over the past 4 years, our research group has leveraged the similar problems of integrating the overwhelming scientific literature and learning data analysis skills. We have experimented with what we call a ***Code Club*** to improve data analysis skills in a community environment. Code Club sessions generally include a brief tutorial, a set of exercises related to the tutorial for learners to work through in a small group, and an opportunity to debrief and report back each group's solutions. Table 2 includes several examples of successful Code Club topics that we have done within a separate 1-hour long Code Club session.

Table 2. Examples of Code Club sessions facilitated within the Schloss lab's weekly group meetings

Title	Description
base vs. ggplot2	Given input data and a figure, recreate the figure using R's base graphics or ggplot2 syntax
Snakemake	Given a bash script that contains an analysis pipeline, convert it to a Snakemake workflow (can also be done with GNU Make)
DRYing code	Given script with repeated code, create functions to remove repetition
mothur and Vegan	Given a pairwise community dissimilarity matrix, compare communities using the adonis function in the Vegan R package
tidy data	Given a wide-formatted data table, convert it to a long, tidy-formatted data table using tools from R's tidyverse
GitFlow	Learners file and claim an issue to add their name to a README file in a GitHub-hosted repository and file a pull request to complete the issue

R with Google docs	Scrape a Google docs workbook and clean the data to identify previous Code Club presenters

As described in the letters of support from Drs. Lauring, Balunas, and Snitkin, this model has been used by other research groups with great success and enthusiasm. Because of the collaborative nature of the Code Club format, there is significant peer-to-peer instruction and customization of concepts, data sets, and questions that are relevant to the research group. However, often researchers who want to develop their data analysis skills feel isolated in a lab of traditional bench scientists or are in a small research group that lacks the critical mass to implement their own Code Club.

With these factors in mind and the isolation many have felt due to the shutdown of research laboratories due to the COVID-19 pandemic, we have experimented with creating virtual Code Club sessions. Initially, the model included live learners on a Zoom call. The synchronous nature of that format made the model unsustainable. More recently, we have created an asynchronous model where a motivating question, tutorial, and set of exercises and their solutions are provided as a blog post with an accompanying video or "vlog" posted to the Riffomonas project YouTube channel (https://www.youtube.com/riffomonasproject). These Code Club sessions are released once a week and have received a positive response from its growing community. As the sessions gain wider reception, it is hoped that the community will make greater use of the commenting features to ask questions and propose topics for future sessions. **The development of additional Code Club sessions will significantly enhance the opportunity for learners to strengthen their reproducible data analysis skills through repeated and deliberate practice.**

The pedagogical benefits of Code Club over previous methods of instruction. As highlighted above, self-learners and those who participate in workshops struggle because the available resources fail to facilitate learners' ability to engage in the best metacognition practices. The basis of this proposal is that introducing Code Club activities will overcome those limitations. The primary benefit of the Code Club format is that because the sessions incorporate exercises for the learner to practice with, they are engaging in repeated practice, which is a more effective means of learning new material and developing automaticity than massed practice (39–42). By developing a large collection of sessions that learners can chose from, they will have many opportunities to cover similar content helping to make their practice deliberate rather than superficial. The design of the sessions requires that the amount of material be kept to at most 2 concepts. This minimalist design reduces the cognitive load for the learners, which can be significant in workshops (44). In fact, a key part of the design of each session is stripping out extraneous material to minimize the cognitive load for learners. Furthermore, in addition to the limited number of concepts that can be covered in a single session, it is not possible to completely cover those concepts. Therefore, it is unlikely that the learner will attain complete mastery the first time they see the concept. Interleaving concepts over multiple sessions without first attaining complete mastery is an effective way to help learners develop complete mastery because each time they see the concept, they need to practice their retrieval skills to apply what they already know (45, 46). Each time they do this they approach mastery and automaticity. The exercises that are provided with each session increase in difficulty. The first exercise asks the learner to do a simple modification of what was done in the tutorial to answer a related question. By the third question, the learners are asked to apply the concepts to a completely new question with the same or different data. The increasing difficulty of the exercises provides a self assessment to the learner (47). Even if the learner is only able to solve one or two of three exercises on their own, by trying the other exercises they are creating a mental model of how they think the exercise should be solved, which they can adjust once they see the solution (48). **Each Code Club session is intentionally designed to incorporate the cognition literature to enhance the development of reproducible data science skills.**

Teaching portfolio. Over the past 12 years Schloss has taught between 4 and 8 courses per year related to reproducible data analysis skills. Although most of these have been 3 day workshops, Schloss has also taught workshops lasting 2 hours to traditional courses that were a full semester. Schloss's teaching has largely been devoted to general reproducible research practices including R programming and the use of the mothur software package for analyzing microbial ecology data. More than 1,200 scientists have participated in these workshops. In addition, Schloss is a trained Carpentries Instructor and co-teaches one or two Software or Data Carpentry workshops per year. At the University of Michigan, Schloss directs the local chapter of the Carpentries organization where he helps coordinate 10 workshops that are taught each year on campus, pedagogical topics, and the development of additional workshop. Prior to 2020, these workshops were taught

in person. Prior to the COVID-19 pandemic Schloss had already transitioned to teaching his workshops via Zoom and was well prepared to start teaching the material virtually. In April 2020, Schloss taught a three day, R-based workshop to over 100 learners with the help of four teaching assistants. Between this large workshop and two other virtual workshops Schloss has already taught in 2020, he has significantly improved his ability to teach remotely. These experiences demonstrate that Schloss is connected to a large network of scientists who have participated in workshop style learning environments. Schloss will draw upon this network to recruit learners to continue their learning through virtual Code Club sessions. Furthermore, the depth of his experience shows that he has a unique familiarity with the strengths and weaknesses of different teaching approaches. **Schloss's network and deep experience will be significant assets for the proposed Research Eduction Program.**

Teaching philosophy. Schloss believes that anyone can learn to analyze their own data. A data analysis is strongest when the person who designed the experiments and generated the data analyzes with the advice of experts in statistics and their sub-discipline. The best way to motivate learners to learn the concepts they are taught is by answering real world questions rather than using generic questions derived from simulated or overly abstract datasets (e.g. the mtcars or diamonds datasets, which are popular in R teaching materials, e.g. (49)). It would be dishonest to teach one set of methods and use a different set for our own professional work. Therefore, to answer these questions, Schloss teaches the approaches that his research group uses for their research. Schloss teaches learners in his classes as though they were trainees working in his lab that need skills to create the reproducible papers that his lab strives to publish. Just as Philip Bourne demonstrated in the anecdote above, the best data scientists acknowledge that they still have room to grow. With this in mind, Schloss uses an encouraging and growth-minded outlook that asks learners to do better with each new analysis they perform. This incremental approach may feel slow or incomplete. But this approach is far more effective than expecting scientists to take on a large set of skills at once.

Schloss's style of teaching can be seen in the current proposal and in the module he created for work funded under an RFA similar to the current RFA, RFA-GM-15-006. In that module, Schloss developed a series of 14 modules related to reproducible research practices for microbiome research (19). The nearly 14 hours of content was motivated by real world scientific questions, uses live-coding to demonstrate practices, and includes activities for learners to engage in to develop their own skills. The current proposal goes beyond the materials developed for the initial phase of the Riffomonas project and more fully integrates Schloss's teaching philosophy and the pedagogical goals outlined above. These materials are hosted as part of the Riffomonas Project (https://www.riffomonas.org). This name encapsulates how we have seen others and ourselves make the greatest gains in learning reproducible methods. "Riffing" involves taking a musical theme and either repeating it or adapting it to a new setting. The Riffomonas Project, seeks to help people learn concepts by showing how the concept can be employed to answer one question and then encouraging them to adapt the solution to a new question. By starting with solutions that they know work, they can dissect the solution to understand why it works and expand upon it to derive solutions to new problems. **This cycle helps a learner work through the levels of Bloom's Taxonomy as they interact with the concepts (50).**

<u>Significance of the Proposed Research Education Program</u>

Successful completion of the proposed Research Education Program will result in a library of resources that individuals or groups of researchers can use to engage in repeated practice of concepts important in conducting data analysis. ***This contribution is expected to be significant because it will address the problem of wasting the significant resources that are extended to participate in workshops only to be ineffective because learners do not have the additional resources for deliberate practice.*** It is likely that similar types of repeated practice activities would improve learning in areas where researchers also use workshops to engage in intensive learning activities including laboratory skills and safety training. Central to the proposed research is the problem that researchers participate in workshops with every intention of learning to program. The leave the workshop enthusiastic and feeling like they have learned a lot. Then they struggle to find opportunities to apply their skills. Because they fail to practice the material in the weeks following the workshop, they lose those skills. When another workshop is offered, they dutifully sign up again hoping that the outcome will be different. The materials developed for the proposed Research Education Program will provide opportunities to practice what was covered in the workshop, breaking the cycle of learning and forgetting.

Innovation

The *status quo* as it pertains to bench scientists developing data analysis skills is for them to take short and intensive workshops. This approach likely works well if they have an immediate need for these skills; however, this is rarely the case and the learner hopes to retain enough information from the workshop to apply it when they reach the data analysis portion of their project. The reality is that the bench scientist typically forgets the information by the time they are ready to use it. They have effectively crammed as much information as they could during the workshop hoping to retain it for later application. A consistent message from educational research is that cramming is ineffective, but that repeated and deliberate practice is essential to long term learning. ***The proposed Research Education Program is innovative, in our opinion, because it represents a substantive departure from the status quo by providing bench scientists with a library of resources to engage in repeated and deliberate practice of reproducible data analysis concepts.*** The Code Club concept is drawn from traditional Journal Clubs where a paper is presented, critiqued, and used to think of additional research questions. The Journal Club activities teach scientists best practices in experimental design, methods, and interpretation. Those activities build off of prior coursework to reinforce the concepts covered in the classroom. Similarly, the Code Club format seeks similar goals but with data analysis concepts. Analogous to a Journal Club presentation, Code Club resources will include a motivating research question and the data and data analysis concepts needed to answer that question. Learners will then have the opportunity to answer related questions using the concepts they just learned. With a high volume of resources, learners will see the same concepts multiple times over many sessions and in different contexts. This will build off of an initial workshop experience to deepen their understanding of the concepts and ability to integrate different concepts to answer their own questions. The result will be a better-trained scientific workforce that is able to ask better research questions of their data and answer the questions in a robust and reproducible manner.

Approach: Produce Code Club sessions that highlight concepts important for performing rigorous and reproducible data science

Introduction

Most bench scientists struggle to apply modern tools that enable them to insure the reproducibility of their data analyses. The ***overall objective*** of this Research Education Program is to develop a collection of virtual Code Club sessions that researchers can use on their own or with colleagues to strengthen their reproducible data analysis skills. The materials will be targeted to biomedical scientists at any career stage. The ***central hypothesis*** is that completing Code Club sessions will improve the retention of concepts introduced in prior workshops and allow learners to more quickly develop their skills beyond those covered in a workshop. The ***rationale*** for this hypothesis is that although the workshop format is popular, its effectiveness is limited because a workshop asks learners to remember a large number of concepts and only provides superficial opportunities to practice the material. A workshop forces learners to engage in massed practice; an approach that is known to be ineffective (39–42). In contrast, Code Club sessions provide brief, regular opportunities to engage in repeated deliberate practice. By limiting each session to one or two concepts, we will lessen the cognitive load for learners and encourage them to practice their retrieval and application skills (44–46). The framework of the Code Club is based on Schloss's considerable experience helping bench scientists learn to do their own data analyses and observing the experiences of colleagues who have run their own Code Clubs. The ***outcome*** of this Research Education Program will be a validated collection of materials for more than 100 Code Club sessions that cover multiple areas of reproducible data analysis.

Design

Format. Each Code Club session will have the same structure and will be motivated by a question. For example, "what is the half-saturation constant for β-galactosidase?". After a brief introduction stating the question, the host will provide a 20-25 minute tutorial on that concept in which they will answer the question. Depending on the relative balance of concepts that have been covered in other Code Club sessions (see next paragraph), we will select one or two concepts to demonstrate for this question. For example, the host could focus on linear regression for this session. The tutorial would cover the assumptions that must be true to fit data to a linear regression, how to evaluate the quality of the fit, and how to perform the fit in R. To answer the question, the host would discuss the strengths and weaknesses of using a double-reciprocal version of the Michaelis–Menten equation (i.e. the Lineweaver-Burk plot) to determine the relevant constants. Next, viewers will be encouraged to pause the video to complete three exercises. The first question will ask the viewer to

adapt their code to a related question (e.g. determine the half-saturation constant for β-glucoronidase). The second question will ask the viewer to adapt their code to answer a related, but more distant question (e.g. fit data using the Hanes-Woolf or Eadie–Hofstee transformations). The third question will have the viewer apply the concept in a different context (e.g. construct and apply a linear calibration curve for a Bradford assay). To conclude the Code Club session, the host will spend 15 minutes sharing their solutions to the exercises. The goal for each Code Club is not to achieve mastery, but to supplement prior knowledge, provide an opportunity to practice concepts in diverse settings, and to give an opportunity for self assessment. To emphasize this point, a subsequent session might use linear regression to solve a different problem or might build off of this session by discussing logistic regression. Another session might cover the same question, but demonstrate how to fit the Michaelis-Menten equation using non-linear regression. Schloss will be the primary host, but will periodically invite other hosts and co-hosts to diversify the delivery of the message and provide more context for biological questions.

Each Code Club session will consist of a blog post hosted as webpages on the Riffomonas project website (https://www.riffomonas.org/code_club) and as a video hosted on the Riffomonas project YouTube channel (https://www.youtube.org/riffomonasproject). All materials will be released under the Creative Commons By Attribution (CC-BY v4.0) license. The materials will be developed and disseminated keeping in mind the best practices to comply with revised Section 508 Standards. We will regularly review and adopt the best practices described in the standards described in the Guide to Accessible Web Design & Development (https://www.section508.gov/content/guide-accessible-web-design-development) and the W3C, WCAG, and ARIA authoring best practices. The videos will be captioned using the Google speech-to-text algorithm and if the quality is poor, we will use a transcription service and upload our own captions. This is the approach we used previously for the microbiome-based reproducible research tutorial series. We are committed to facilitating the learning of all scientists.

Topic areas. The concepts that will be covered in each Code Club session will be selected from topics that are relevant to insuring reproducible and robust data analyses. The topic areas will include (the estimated number of sessions per topic area are shown in parentheses):

- **Scripting of analyses (n=60 sessions).** Scripting is a critical topic area because it allows the analyst to show the code that was used to transform raw data into the final results. Because this generally involves applying programming skills, this topic area has the most concepts that need to be covered. Sessions will teach learners best practices for using R and bash scripts to clean, process, and validate raw data, visualize data, and statistically analyze and model data. In addition, best practices for building and working with spreadsheets to facilitate scripting of analyses will be covered.

- **Automation (n=10 sessions).** Related to scripting analyses, automation is an important consideration since an automated pipeline details how scripts are integrated with each other to complete a full analysis and how to track the data and code dependencies across a project. Concepts will be demonstrated using bash scripts, GNU Make, and snakemake.

- **Project organization and documentation (n=10 sessions).** A project that is automated using elegant code is worthless if the project is poorly documented or organized. Concepts related to this topic area will detail the value of project structure, self-documenting file and directory names, and providing a guide to the reader to navigate the project.

- **Version control (n=10 sessions).** The ability to see how a data analysis has evolved over its life is possible if good version control practices are used. Concepts related to both creating that historical thread for a project and looking back over the historical thread will be covered. git and GitHub will be used to demonstrate the concepts for this topic area.

- **Literate programming (n=10 sessions).** A literate programming document embeds code within a written narrative and insures that any results in a document are directly linked to the code responsible for producing that result (32, 51). Concepts and applications using R Markdown will be used to cover this topic area.

- **FAIR (Findable, Accessible, Interoperable, and Reusable) principles (n=4 sessions).** The ability to reproduce and build off of an existing analysis is central to the philosophy of the Riffomonas project. To achieve that goal, we will highlight the FAIR principles throughout the series of Code Club sessions. We

will also emphasize the value of open science, data and code accessibility, and the creation of containers and machine images for facilitating these principles (https://datascience.nih.gov/sites/default/files/NIH_Strategic_Plan_for_Data_Science_Final_508.pdf)

Although we will limit the number of learning objectives to one or two concepts per session, they will be demonstrated in the context of other concepts to emphasize how the concept of the session can be integrated with other concepts. For example, a session discussing licensing might involve using version control to put a copy of a license in a git repository hosted on GitHub. There are many tools available to demonstrate the concepts in each of these topic areas. We have selected a set of tools that are widely used. As we develop the Code Club sessions we will reassess that the tool we are demonstrating is still preferred. In each case we will also present the strengths and weaknesses of different tools.

Motivating questions. Using real questions and situations faced by scientists is an important component of the design of the Code Club sessions. Some questions will be related to the preparation of reports, presentations, and manuscripts while others will be driven by a biological question. For these latter questions, we will initially rely on our own interests and the suggestions of colleagues to insure that we have a broad range of topics that cover the biomedical sciences (see letters of support). At a minimum, we foresee using questions inspired from genomics, microbiome science, physiology, bacteriology, viral evolution, biochemistry, immunology, epidemiology, psychology, and neurobiology. We will always use data that are freely available and if the question is from a domain outside of our area of expertise (i.e. microbiology), we will consult with an expert to review our materials.

Dissemination Plan

To disseminate the materials generated as part of the proposed research, we will pursue several avenues beyond linking the materials to the NIGMS clearinghouse web site:

Recruitment through workshops. The primary direct approach that we will use to recruit people to participate in Code Clubs is through workshops that Schloss helps facilitate. Each year, Schloss teaches 6 3-day workshops that are attended by researchers from around the world. In addition, at the University of Michigan, he teaches his Microbial Informatics class that is taught as a 3 day workshop and he co-teaches 2 2-day Carpentries workshops. Schloss is the director of the University of Michigan's Carpentries Partner Organization. Including the workshops that Schloss co-teaches, the local Carpentries organization teaches 10 workshops per year. Through these diverse teaching venues, Schloss has the ability to annually draw from a population of more than 120 scientists from outside the University of Michigan and more than 200 scientists from the University of Michigan. We will make every effort to recruit scientists that participate in these activities. Our surveys of past workshop participants indicates that our learners are primarily graduate students (~40%) and postdocs (~40%), but research staff (~15%), faculty (~5%), and occasionally undergraduates (~1%) also participate. An equal number of women and men participate in these workshops.

Advertising of materials. We will use social media (e.g. YouTube and Twitter) to promote the Code Club materials. Schloss currently has over 8,000 followers on Twitter, where he has a reputation for discussing data analysis issues. We will advertise new Code Club sessions through his Twitter account. We will also use search engine optimization (SEO) strategies targeting YouTube's recommendation algorithm to grow a new viewership base. Both avenues will create enthusiasm in the biomedical research community and beyond to foster their interest in the sessions.

Publications and presentations. We anticipate publishing at least two manuscripts for this project. The first will announce the availability of the resources, similar to what we previously did for the Riffomonas project's reproducible research tutorial series (19). The second will report the results of our study into the benefits of engaging in Code Club materials over workshop-based learning alone. Finally, it is likely that there will also be opportunities to give seminar, conference, and webinar presentations describing the project to interested groups.

Availability of materials. All instructional materials will be made freely available through the Riffomonas project website at (https://www.riffomonas.org/code_club) and all videos will be hosted on YouTube under the Riffomonas project channel (https://www.youtube.com/riffomonasproject). All materials related to the project will be maintained as a public GitHub project repository (https://www.github.com/riffomonas/code_club). In fact,

the development of this proposal is available at www.github.com/riffomonas/2020_RR_R25. All content will be released under a Creative Commons by Attribution (CC-BY-4.0) license.

Overall, we have a structure in place to disseminate the Code Club materials developed in the proposed plan to a large number of researchers and a plan to expand their reach beyond our current network.

Evaluation Plan

The central hypothesis of the proposed Research Education plan ***is that completing Code Club sessions will improve the retention of concepts covered in prior workshops and allow learners to more quickly develop their skills expand beyond those covered in a workshop.*** To test this hypothesis, we will establish three groups of learners in our study. The first are those that participated in a programming workshop and watched at least one Code Club session. The second are those who only participated in a workshop. The third are those that only participate in Code Club sessions. We will follow up with learners during the week after participating in the workshop to establish a baseline and 2 and 6 months after the workshop. For those in the third group who did not participate in a workshop, we will assess their baseline as soon as they are recruited into the study. We will partner with the Center for Research on Learning and Teaching at the University of Michigan to create survey and assessment instruments (see attached letter from Malinda Matney, PhD). We will finalize the survey and assessment tools in the first year while we are refining the style and building the collection of Code Club materials. The survey and assessment tools will then be deployed in the second and third years of the project (see timeline below).

Surveying participation. Learners will take a survey to record demographic information. It will be important to determine whether factors like gender, race, or career stage impact whether one is more likely to participate in Code Club sessions. We will also survey the learners to measure covariates that we expect to be important to account for retention and growth including the type of workshop they participated in, how many Code Club sessions they participated in, the number of hours they engaged in practice and application of the content, and their career stage.

Evaluating efficacy of Code Club Materials. Learners will complete an evaluation tool that asks the learners to answer a series of questions and perform a series of tasks. These tasks will be brief to minimize the time and effort required by the learner. This evaluation will be a mixture of problem types that ask the learner to:

- Modify a series of steps to achieve a solution to a new question
- Debug a series of steps to achieve a solution with an actual pipeline, using blanked out commands and/or arguments, or rearranging steps to achieve a solution
- Generate the code to convert data to the specified output

Evaluating the materials. Beyond assessing the learners, we need to assess the reception of the learning materials. We will assess the materials with several tools. First, every Code Club session will include an anonymous survey asking learners to evaluate the materials for the clarity of their presentation, relevance, and clarity. The data will be aggregated using Google Forms. Second, we will use the built in analytic tools for YouTube to track the number of views, time spent on each video, viewer demographics, how viewers found the video, likes vs dislikes, and comments. Third, we will use Google's analytics tools to track how learners find each Code Club session's blog post, how long they spend on the site, and where they go after visiting the site. We will track all of these metrics and adjust accordingly throughout the funding period. We anticipate that we will make the most significant changes to the style and strategy in the first year.

This evaluation plan demonstrates that we have a comprehensive plan to evaluate our materials, the effectiveness of the materials, and the types of people engaging in the material.

Principal Investigator

As indicated by his Biosketch and the numerous letters of support, Schloss is a respected member of the microbiome research community and is an excellent teacher who is anxious to utilize innovative teaching methods to communicate complex materials. Over the past 12 years, Schloss has been the PI on 9 research grants funded by NIH and other agencies including 4 R01 and U01 projects related to the microbiome and an R25 related to developing instructional modules for engaging in reproducible research practices. He has

served as a co-Investigator on 16 additional projects during that time. Over the course of his career he has published 103 manuscripts, which covered topics including microbial ecology and the microbiome, science policy and communication, and reproducible practices. The R25 that Schloss was awarded under RFA-GM-15-006, "Development of reproducible informatics skills among microbiome researchers (R25GM116149)" successfully yielded two peer-reviewed publications (19, 24). The most watched video from that series has been viewed more than 1,300 times. Beyond the funding period of that project, Schloss has continued to develop and post educational content to the Riffomonas project website at https://www.riffomonas.org. At the University of Michigan, Schloss has developed two courses: *Symbiosis* and *Microbial informatics*. The latter is a course that is designed to teach microbiologists in MS and PhD programs and postdocs how to use R. Initially designed as a semester-long, 3 credit course, Schloss revamped the course to a 3 full-day workshop to better serve more diverse researchers who could not commit to a semester-long course. Over the 5 years that he has offered the course in this format, the class has grown and he has diversified its content from focusing on microbiome-related data sets to data sets that appeal to a broader audience. Although this course touches on the content of the proposed teaching materials, it has focused on developing R programming skills and not broader data analysis practices. This course and Schloss's willingness to experiment with the content is indicative of his innovative approach to teaching. Finally, over the past 12 years Schloss has offered numerous other workshops each year describing how microbiologists can use mothur and R to analyze data from their research projects. This experience has given him a unique perspective into the needs and competencies of the biomedical research community. **Together, these data and experiences indicate Schloss is "actively engaged in research in an area related to the mission of NIH, and can organize, administer, monitor, and evaluate the research education program."**

Institutional Environment and Commitment

We have secured institutional support for this project on multiple levels. First, as indicated by the letter of support from Dr. Bethany Moore, Interim Chair of the Department of Microbiology & Immunology at the University of Michigan School of Medicine, Schloss has the support of the university to gain access to adequate staff, facilities, and educational resources to make the planned research education program successful. Second, Schloss has interacted with the Center for Research on Learning and Teaching (CRLT) at the University of Michigan to plan the assessment program for this project (see letter of support from CRLT). The CRLT provides a mixture of complimentary and fee-based services, but does not participate in projects as personnel on grant proposals. The support provided by CRLT will insure that we are utilizing the best practices to evaluate the teaching modules. Third, as indicated by the letters of support from other researchers at the University of Michigan and across the United States, Schloss has the support and commitment of other investigators to implement this project. They all see the value of developing instructional materials such as those described in this proposal. **The multiple levels of commitment and broad support that this proposal enjoys speaks to its importance and the unique qualifications of Schloss to lead the project.**

Expected Outcomes

By the end of the proposed project, we expect to find that scientists who engage in a workshop retain and grow their skills better if they also participate in a weekly Code Club session. This expectation derives from the well-validated benefits of repeated practice for improving automaticity. To achieve this outcome, we will create a library of more than 100 recorded Code Club sessions that scientists and the general public can use to improve their reproducible data analysis skills. By consistently releasing a session each week and following our broad dissemination plan, we will build a large subscriber base of more than 1,000 individuals; the Riffomonas project YouTube channel currently has 193 subscribers and has increased the number of subscribers by 56 individuals since Schloss started posting the recent Code Club materials (as of June 15, 2020). Subscribers represent the core group of individuals who will engage with the materials, participate by leaving comments where they ask questions and make suggestions for future concepts they would like to see us cover. **Ultimately, we expect to significantly enhance the reproducible data science skills of a diverse range of scientists at every career level in ever sub-discipline of biomedical research.**

Potential Problems & Alternative Strategies

We are confident that the proposed Research Education Program Plan will be successful; however, there several potential problems that we may face. First, we may find that video is not an attractive medium for scientists to engage with the Code Club sessions and that they prefer the accompanying blog posts. We believe this problem is unlikely because of the popularity of live coding. However, if this is the case, we would pivot to text-based content and cease producing the video. Second, we may find that the number of people

engaging the Code Club materials grows at a slower than expected pace. We are optimistic that this will be unlikely because there is a large population of scientists interested in learning data science skills. Furthermore, we will be consistently producing content, which has been shown to be key to building viewership. If this does become a problem, we will conduct focus groups and interviews to assess what is causing slow adoption. We will also ask scientists with a strong social media presence to help evaluate the materials and promote our content. Finally, it is possible that we will be unable to keep up with the pace of producing one video per week. Considering each Code Club session takes about 8 hours for one person to develop, record, edit, and release, this is unlikely since Schloss is devoting 20% of his effort (i.e. 8 hours per week) and will be working with a postdoctoral research fellow (25% effort) to develop the material. This should be a sufficient amount of effort to cover the proposed pacing, especially since we anticipate finding efficiencies as we go forward. One solution might be to produce multiple weeks' worth of Code Club sessions simultaneously and deploy them once per week. Although less desirable, we could also shift to releasing the sessions every two weeks. **We have developed a robust frameworks for developing, disseminating, and evaluating the Code Club sessions, which will yield a successful outcome.**

Timeline and Benchmarks for Success

The proposed Research Education Program Plan will consist of developing 52 Code Club sessions per year that are released weekly to the Riffomonas Project website and YouTube channel (https://www.youtube.com/riffomonasproject) for the first two years of the program. If we continue to develop videos beyond this time, it will be supported by independent sources of funding. In the first year we will also develop and refine our survey and evaluation tools. In the second and third years we will recruit learners to the study from our workshop attendees, social media network, and those who find the materials through search engines. In the third year we will continue to recruit learners and report the results of our surveys and evaluations.

	2021			2022				2023				2024
Task	**Q2**	**Q3**	**Q4**	**Q1**	**Q2**	**Q3**	**Q4**	**Q1**	**Q2**	**Q3**	**Q4**	**Q1**
Deploy sessions												
Maintenance and refinement												
Develop assessment tools												
Recruit learners												
Evaluate learners												
Summarize and report results												

Future directions

Similar to how Journal Clubs are used to improve research skills and develop new research directions, Code Clubs have the potential to improve reproducible data analysis skills and inspire scientists to push their sub-disciplines further. If the proposed Research Education Program Plan is successful, then we will be likely to have inspired "copy cats" to produce a similar type of content within the data science arena. Hopefully, we can build enthusiasm around the Code Club concept and build a community of people helping each other to learn to engage in more reproducible practices. In the future, the need for us to produce all of the content ourselves would lessen and we would receive Code Club session proposals from the community who could produce the sessions under the Riffomonas Project. We expect that we will be able to adapt the Code Club concept to other components of research including writing and bench skills to complement programs that align with the goals of other educational programs relevant to NIGMS's training portfolio.