Table 1: Modules for '**Improving the Reproducibility of Experimental Data Recording**'. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black).

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Separating data recording and analysis	Many biomedical laboratories currently use spreadsheets, with embedded macros, to both record and analyze experimental data. This practice empedes the transparency and reproducibility of both data recording and data analysis. In this module, we will describe this common practice and explain how it impedes the transparency and reproducibility of data recording and analysis. We will then outline alternative approaches that separate the steps of data recording and data analysis and explain how these alternative approaches can improve the reproducbility of biomedical research.	• Explain the difference between data recording and data analysis; • Understand why collecting data on spreadsheets with embedded macros impedes transparency and reproducibility; • List alternative approaches that separate data recording and data analysis to improve transparency and reproducibility.	15	• Discussion questions, including describing data recording approaches the trainee has previously used in research projects and the benefits and limitations of those approaches in terms of data transparency and reproducibility; • Short audio recording of two Co-Is giving their own answers to these discussion questions.
Principals and power of structured data formats	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 how using a structured data format when recording experimental data can improve the transparency and reproducibility of research; • Outline other benefits of using a structured format when recording data.	25	• Applied exercise: For example datasets, specify whether each is in a structured data format and, in cases where it is not, draft a structured format that could be used to record the data; • Video walking trainees through one solution to the applied exercise.

Table 1: Modules for '**Improving the Reproducibility of Experimental Data Recording**'. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
The 'tidy' data format: an implementation of a structured data format	The 'tidy' data format was outlined in a 200[x] paper and has since quickly gained popularity among statisticians and data scientists. By consistently using this data format, researchers have found they can employ combinations of simple, generalizable tools to perform complex tasks in data processing, analysis, and visualization. However, despite the power of this format, it is not yet widely known or used among laboratory scientists when they record experimental data. In this module, 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', including collaborations with statisticians.	• List which characteristics to check to determine if a dataset complies with the 'tidy' structured data format; • Explain the difference between the ideas of a structured data format (general concept) and the 'tidy' data format (one implementation of that general format that is now particularly popular in data analysis).	25	• Quiz questions: For example datasets, correctly identify which of the 'tidy' data principals the dataset has or lacks; • Video giving answers and explanations for quiz questions, including showing 'tidy' versions of each example dataset; • Link to paper that established the 'tidy' data format.
Designing templates for tidy data collection	This module will move from the principals of the 'tidy' data format to the practical details of designing a 'tidy' data format to use for a specific research project. We will describe common issues that prevent real datasets from experimental research projects from following a 'tidy' format and show how they can be avoided when deciding the format in which to record experimental data. 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 following a 'tidy' format; • Convert data from an 'untidy' to a 'tidy' format.	20	• Applied exercise: Take a dataset in an 'untidy' format, identify what characteristics keep it from being 'tidy', and convert design a 'tidy' form of the data; • Video providing a detailed solution to the applied exercise.

Table 1: Modules for **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Example: Creating a template for data collection	In this module, we will walk through an example of creating a template to collect data in a 'tidy' format for a laboratory-based research project. As an example, we will use a research project headed by one of our Co-Is on tuberculosis [more description of this project]. We will walk through the 'untidy' format initially used to collect data for this project, explain how this format differed from a 'tidy' format, and show how we changed the format to be 'tidy'. Finally, we will show examples of how the experimental data can easily be cleaned, analyzed, and visualized using reproducible tools once it is in a 'tidy' format.	• Understand how the principals of 'tidy' data can be applied to experimental data from a real research project; • Explain the advantages of using a 'tidy' data format for the example project.	15	• Discussion questions, including listing examples of how experimental datasets the trainee has previously worked with or is currently working with are 'untidy' and how they could be converted to a 'tidy' format; • Short audio recording of two Co-Is giving their own answers to these discussion questions.
Power of using a single '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 (raw and pre-processed), meta-data, code for data pre-processing, and research products further along the research pipeline (e.g., paper drafts, figures, code for data analysis). In this module, we will show how all research project files can be collected and saved in a single 'Project' directory. We will explain how using this practice from the start of a research project improves the reproducibility of the projects, as well as how this practice facilitates the use of later tools to improve reproducibility, including version control. Finally, we will list some of the common components and subdirectories that are useful 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 collecting all research data and other files related to a research project in a single 'Project' directory improves the reproducibility of a research project; • Describe how experimental data collection can be integrated with a research 'Project' directory.	20	abitem Quiz questions: These will test the trainee's understanding of what a 'Project' directory is, what common components it may include, and the benefits of structuring research project files—including raw and pre-processed experimental data—within a single 'Project' directory from the beginning of the research project; • Video with detailed answers and discussion of quiz questions.

Table 1: Modules for **'Improving the Reproducibility of Experimental Data Recording'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Creating 'Project' templates	Researchers can gain even more benefits, in terms of improving both the reproducibility and efficiency of research, by using a consistent structure for the 'Project' directories for all of the research projects for a research group. We will describe the benefits of using a consistent structure for 'Project' directories across different research projects within a research group, including how this practice can facilitate the re-use of code for pre-processing, analyzing, and visualizing data. Further, we will demonstrate how RStudio can be used to create a template of a research group's 'Project' directory structure, so a new project can be initialized with a skeleton directory that follows the 'Project' directory format established by the research group.	• Explain how using a consistent structure for research 'Project' directories can improve the reproducibility and efficient of research projects within a research group; • Understand how RStudio can be used to create a template to use to create consistently-structured research 'Project' directories.	25	• Discussion questions, including descriptions of how the trainee has saved and tracked research project files for previous research projects and what barriers, if any, these practices introduced in terms of the reproducibility and efficiency of research; • Short audio recording of two Co-Is discussing how they have saved and tracked research project files in previous projects and what barriers to reproducibility these practices introduced.
Example: Creating a 'Project' template	In this module, we will walk through a real example 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. [Further description of the real research project]	• Create a 'Project' template in RStudio to use to initialize consistently-formatted 'Project' directories to store all files related to a research project; • Initialize a new 'Project' directory using this template.	15	• Applied exercise: We will provide a description of the components and subdirectories that a research group has decided to include in their 'Project' template. The trainee will need to use RStudio to create and save a 'Project' template that meets these specifications; • Video demonstrating a detailed solution to the applied exercise.

Table 1: Modules for '**Improving the Reproducibility of Experimental Data Recording**'. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Harnessing version control for transparent data collection	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 any changes can be reverted to earlier versions. However, this practice leads to an explosion of research project 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 in a way that allows them to identify and undo any previous changes while maintaining a single version of each file. Further, version control requires short messages describing each change made to each file, which improves the transparency and reproducibility of both the recording of experimental data and also the later steps of pre-processing, analyzing, and visualizing the data. In this module, we will explain what version control is and how it can be used in research projects. We will highlight how version control can improve the transparency and reproducibility of research. Finally, we will give examples of version control tools that are popular for research.	• Describe version control and what it does; • List how using version control improves the transparency and reproducibility of research.	10	• Discussion questions, including discussion of how the trainee has managed evolving research project files in previous projects and any barriers those practices introduced in conducting efficient and reproducible research; • Short audio recording of two Co-Is giving their own answers to these discussion questions.

Table 1: Modules for '**Improving the Reproducibility of Experimental Data Recording**'. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Using git and GitLab to enhance the reproducibility of collaborative research	Once a researcher has learned to use git on their own computer for local version control, they can begin using version control platforms like GitLab and GitHub to collaborate with others in their research group while keeping the project under version control. These platforms allow the 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. In this module, we will describe why a research team may want to use a version control platform like GitLab to work collaboratively on a project. Further, 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.	• Explain the benefits of using a version control platform like GitLab, rather than Dropbox, to share project files for collaborative research projects, particularly in terms of increasing the transparency and reproducibility of a research project; • Describe the difference between git and GitLab; • Understand how to set up and use git through RStudio's interface; • Understand how to connect with GitLab through RStudio and how to use these version control and collaboration tools to improve the reproducibility of research projects.	20	• Applied exercise, with detailed instructions for each step: Use RStudio to initialize version control for a directory and to make several tracked changes. Create a matching GitLab repository and use RStudio to connect your local and GitLab versions of the directory. • Video walking trainees through a detailed solution to the exercise.

Table 2: Modules for **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black).

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Principals 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, peak finding and quantification for LC / MS metabolomics data). While often proprietary, point-and-click software is available for this pre-processing, use of such software can limit the transparency and reproducibility of this pre-processing stage of the analysis, and point-and-click software is often time-consuming to use for repeated tasks over large research projects. In this module, we will explain how using scripts to apply open source software for this pre-processing step can improve the transparency, reproducibility, and transparency of research.	• Define pre-processing of experimental data and give some examples; • Describe how the use of proprietary software for pre-processing experimental data limits transparency and reproducibility; • Understand what an open source code script is and how it can be used as an alternative in pre-processing experimental data; • List some popular packages in R that can be used to pre-process biomedical experimental data.	15	• Discussion questions, including discussion of which steps are commonly used to pre-process experimental data in the trainee's research area; • Short audio recording of two Co-Is giving their own answers to these discussion questions; • List of some popular R packages for pre-processing different types of biomedical experimental data.
Introduction to R code scripts	In this module, we will explain the difference between interactive software use and the use of code scripts, using examples from R. 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.	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, similar example dataset. Hints on useful R functions will be provided to help trainees new to the R language; • Video providing a detailed walk-through of a solution to the applied exercise.

Table 2: Modules for **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
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 principal 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 challenges in working with data; • Describe the difference between 'base R' and R's 'tidyverse'.	15	• Quiz: Questions will test the trainee's understanding of what R's 'tidyverse' is and why it is a powerful yet user-friendly tool for improving the reproducibility, transparency, and efficiency of research projects. • Video with detailed answers and explanations for the quiz questions; • Links to further free sources for developing more 'tidyverse' coding skills.
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. These complex data formats are necessary for computational efficiency and to aid the structure of the pre-processing software, but the 'untidy' formats add a critical barrier for researchers who wish to explore and visualize the data. In this module, we will describe the complex data formats are often used in open source software for pre-processing experimental data, explain why use of these complex formats is often necessary, and outline how these complex formats create hurdles in implementing reproducibility tools among laboratory-based scientists.	• Explain why R software for pre-processing biomedical data often stores the 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: Determine trainee's understanding of why complex data formats are often used within steps of experimental data pre-processing in open-source software; • Video providing detailed answers to quiz questions.

Table 2: Modules for **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
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. This is necessary for computational efficiency of the pre-processing, but creates a hurdle for using many common tools taught to improve research reproduciblity, including R's 'tidyverse' tools. With the rising popularity of the 'tidyverse' collection of R tools, which require data to be in a 'tidy' format, R users have recognized that the use of complex, 'untidy' data formats can complicate reproducible code for data pre-processing, analysis, and visualization. Very recently, some researchers have developed tools (the broom and biobroom R package extensions) that can extract a 'tidy' dataset from data stored in a complex, list-based format. These tools create a clean, simple connection between the complex data formats often used in pre-processing or modeling 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 well as provide tips on how to explore and extract data stored in R in this format. We will then demonstrate how the new *broom* and *biobroom* packages can be used to extract to use to convert output from pre-processing software to 'tidy' data formats for futher steps of reproducible data visualization and analysis. 'tidy' versions of pre-processed experimental data from their complex data formats, to allow user-friendly data analysis and visualization using the widely-taught general 'tidyverse' tools.	• 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 why converting data from a complex format to a 'tidy' format can improve the transparency and reproducibility of a research project.	15	• Applied exercise: We will provide example data in a complex, list-based format. The trainee will explore this data based on step-by-step instructions and will extract specified elements from the data format as well as practice using *broom* and *biobroom* R packages to extract 'tidy' data from complex data formats.; • Video providing a detailed walk-through of completing this exercise, with explanations for specific steps.

Table 2: Modules for **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Example: Converting from complex data types to 'tidy' data formats	We will provide a detailed example of a case where data pre-processing in R has resulted in data in a complex, 'untidy' format, and where tools can be used to extract data in a 'tidy' format, which then can easily integrate with general R 'tidyverse' tools for data analysis and visualization. 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. This example will use real experimental data from research on the immunology of tuberculosis [more details on this project].	• List R package extenstions that can be used to extract 'tidy' data from complex, 'untidy' R data formats; • Describe how these tools can be used in research projects to shift from data pre-processing to analysis and visualization of the processed data.	20	• Applied exercise: Trainees will be given an example dataset in a complex, 'untidy' data format in R and will be instructed in how to convert it to a 'tidy' format and then create some straightforward plots of the data based on this 'tidy' dataset; • Video demonstrating a detailed solution to the applied exercise.
Introduction to reproducible data pre-processing protocols	Reproducibility tools can be used to create reproducible data pre-precessing protocols—documents that combine code and text in a 'knitted' document ... . In this module, we will describe how reproducible data pre-processing protocols can be leveraged early in a research project to improve the reproducibility of the pre-processing of experimental data and to ensure transparency, consistency, and reproducibility across the research projects conducted by a research team.	• Describe a reproducible data pre-processing protocol; • Explain how reproducible data pre-processing protocol can be used to improve the reproducibility of research projects at the data pre-processing phase; • List other benefits of using reproducible data pre-processing protocols, including improving efficiency and consistency of data pre-processing across a research groups research projects.	15	• Discussion questions: Including discussion of how reproducible data pre-processing protocols can make biomedical research more reproducible at the data pre-processing stage; • Short audio recording of two Co-Is giving their own answers to these discussion questions.

Table 2: Modules for **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Introduction to RMarkdown as a tool for creating reproducible data pre-processing protocols	RMarkdown can be used to create documents that combine code and text in a 'knitted' document, and it has become a popular tool among statisticians and data scientists for improving the computational reproducibility and efficiency of their research. This tool can also be used earlier in the research process, however, to develop well-documented code to pre-process raw experimental data. In this module, we will show trainees the types of documents that can be created and run using RMarkdown. We will describe how RMarkdown is used among statisticians to improve the reproducibility, efficiency, and transparency of data analysis, as well as describe how it can be leveraged earlier in a research project to improve the reproducibility of the pre-processing of experimental data. We will also provide detailed instructions on how to use RMarkdown in RStudio to create documents that combine code and text. We will explain how these documents can be converted into different final file formats (PDF, HTML, Microsoft Word). 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; • Describe the documents that can be created using RMarkdown; • 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; • Render the document in multiple file formats; • Apply the document to several different datasets that follow the same format.	15	abitem Applied exercise: Trainees will be asked to create, save, and render their own RMarkdown document through RStudio; • Video providing a detailed walk-through of a solution to the applied exercise.

Table 2: Modules for **'Improving the Reproducibility of Experimental Data Pre-Processing'**. The color of each module's title indicates whether the module focuses on **Principals** (blue), **Implementation** (red), or **Case study examples** (black). *(continued)*

Module title	Description of module content	Objectives (After taking the module, the trainee can ...)	Video Length	Extra educational materials
Example: Creating a reproducible data pre-processing protocol	We will provide an example of creatin a reproducible protocol for the automated gating of flow cytometry data for a project on the immunology of tuberculosis [more details on project]. 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 a research project. We will walk the trainees through the final pre-processing protocol, how we apply this protocol to new experimental data, and how we developed the protocol initially.	• Explain how a reproducible data pre-processing protocol can be integrated into a real research project; • Describe what is included in a data pre-processing protocol; • 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: These will test the trainees understanding of how and why we created well-documented and reproducible data pre-processing protocols for this project, as well as how this helps increase the transparency and reproducibility of the research project; • Short audio recording of discussion with the head of this example research project on how this reproducible data pre-processing fits into her research project and how use of this protocol differs from previous data pre-processing practices in the group.