Introduction

JUSTIN KITZES

Think back to the first laboratory science course that you ever took, perhaps a high school or an undergraduate chemistry or biology lab. Imagine sitting down on the first day, in a new room, surrounded by new classmates, in front of a new teacher, and encountering all of the new sights and smells around you. Perhaps there were jars containing strange substances along the walls, oddly shaped glass and metal equipment, and safety gear to protect you from some mysterious danger.

As you entered this new physical and intellectual environment, preparing to learn the foundational knowledge and skills of a new field of science, what was the first thing that you were taught? Whatever it was, we suspect that it was *not* chemistry or biology. For most of us, the first instructions in a lab course were about how to perform basic tasks like cleaning the equipment, zeroing a balance, labeling a beaker, and recording every step that you performed in a lab notebook.

What did all of these seemingly menial tasks have to do with the science that you were supposed to be learning? Although it may not have been clear right away, these steps were all designed to ensure that, when you did conduct an experiment, you would be confident in the accuracy of your results and be able to clearly communicate what you did to someone else. Together, these two factors would permit someone else to perform the same experiment and achieve the same result, verifying your findings. None of your actual experimental results would have

been meaningful, or useful to others, had you not followed these basic procedures and principles.

Now jump forward again to the present, and consider the type of research work that you do today. Almost certainly, you are using methods, tools, and equipment that are significantly more complex than those that you encountered in your first lab course. If you are like most scientists today, your research is also slowly, or not so slowly, shifting away from the traditional "lab bench" of your discipline and into the rapidly expanding world of scientific computing. There is scarcely a scientific discipline today that is not being rapidly transformed by an infusion of new hardware, software, programming languages, messy data sets, and complex new methods for data analysis.

Unfortunately, however, many excellent and accomplished scientists never received even high school or undergraduate-level training in basic scientific computing skills. Many of us struggle along as best we can, trying to write code, work with uncomfortably large data sets, make correctly formatted figures, write and edit papers with collaborators, and somehow not lose track of which data and which analysis led to what result along the way. These are difficult tasks for someone well-versed in scientific computing, much less for scientists who are trying to pick up these skills on the fly from colleagues, books, and workshops.

In one sentence, this book is about *how to take the basic principles of the scientific method that you learned at the lab bench and translate them to your laptop*. Its core goal is to provide concrete advice and examples that will demonstrate how you can make your computational and data-intensive research more clear, transparent, and organized. We believe that these techniques will enable you to do better science, faster, and with fewer mistakes.

Within the world of scientific computing practice, the techniques that we explore in this book are those that support the goal of *computational reproducibility*. For the purposes of this book, we define computational reproducibility as follows:

> *A research project is computationally reproducible if a second investigator (including you in the future) can re-create the final reported results of the project, including key quantitative findings, tables, and figures, given only a set of files and written instructions.*

Thinking back to that first lab course, this would be equivalent to handing a notebook, a stack of equipment, and some raw materials to a classmate and asking them to arrive at the same result that you did.

There are many reasons why we believe that practicing computational reproducibility is perhaps the key foundational skill for scientific computing. Perhaps most importantly, working toward computational reproducibility will indirectly require you to follow many general scientific best practices for all of your digital analyses, including recording all steps in your research process, linking a final result back to the initial data and other inputs that generated it, and making all necessary data and inputs available to your colleagues.

Additionally, thinking explicitly about computational reproducibility helps to move the focus of research up a level from individual activities to the entire scientific workflow. This change in perspective is becoming increasingly important as our work becomes so complex that this overarching grand perspective is not always obvious.

Finally, the computational reproducibility of an individual research project can often be substantially increased or decreased by an individual investigator, meaning that the skills that we will discuss in this book can immediately be put into practice in nearly all types of research projects. This level of control contrasts, for example, with more complex issues such as scientific replicability (see chapter "Assessing Reproducibility"), which are more heavily dependent on coordination among many scientists or on institutional actions.

This book is designed to demonstrate and teach how many of today's scientists are striving to make their research more computationally reproducible. The research described in this volume spans many traditional academic disciplines, but all of it falls into what may be called the data-intensive sciences. We define these fields as those in which researchers are routinely expected to collect, manipulate, and analyze large, heterogeneous, uncertain data sets, tasks that generally require some amount of programming and software development. While there are many challenges to achieving reproducibility in other fields that rely on fundamentally different research methods, including the social sciences and humanities, these approaches are not covered here.

This book is based on a collection of 31 contributed case studies, each authored by a leader in data-intensive research. Each case study presents the specific approach that the author used to attempt to achieve reproducibility in a real-world research project, including a discussion of the overall project workflow, key tools and techniques, and major challenges. The authors include both junior and senior scholars, ranging from graduate students to full professors. Many of the authors are affiliated with one of three Data Science Environments, housed at the

University of California Berkeley, the University of Washington, and New York University. We are particularly grateful to the Gordon and Betty Moore Foundation and the Alfred P. Sloan Foundation for supporting these environments, which provided the intellectual space and financial support that made this book possible.

In addition to these contributed case studies, this book also includes synthesis chapters that introduce, summarize, and synthesize best practices for data-intensive reproducible research. Part I of the book introduces several important concepts and practices in computational reproducibility and reports on lessons learned from the 31 case studies. In "Assessing Reproducibility," Rokem, Marwick, and Staneva outline the factors that determine the extent to which a research project is computationally reproducible. In "The Basic Reproducible Workflow Template," Kitzes provides a step-by-step illustration of a core, cross-disciplinary reproducible workflow, suitable as a standalone first lesson for beginners and as a means of framing the subsequent case study chapters.

These preliminary discussions are followed by "Case Studies in Reproducible Research," by Turek and Deniz, which describes the format of the contributed case studies and summarizes some of their key features. In "Lessons Learned," Huff discusses common themes across the case studies, focusing on identifying the tools and practices that brought the authors the most reproducibility benefit per unit effort and the universal challenges in achieving reproducibility. Ram and Marwick's "Building toward a Future Where Reproducible, Open Science is the Norm," includes a broad discussion of reproducible research in modern science, highlighting the gaps, challenges, and opportunities going forward. Finally, an extended "Glossary" by Rokem and Chirigati defines, describes, and discusses key concepts, techniques, and tools used in reproducible research and mentioned throughout the case studies.

Part I of the book can be read as a standalone introduction to reproducible research practices in the data-intensive sciences. For readers wishing to learn more about the details of these practices, Part II and Part III of the book contain the 31 contributed case studies themselves, divided into high-level case studies that provide a description of an entire research workflow, from data acquisition through analysis (Part II), and low-level case studies that take a more focused view on the implementation of one particular aspect of a reproducible workflow (Part III).

This book unavoidably assumes some background on the part of readers. To make best use of this book, you should have some experience with programming in a scientific context, at least to the point of writing

a few dozen lines of code to analyze a data set. If you are not yet comfortable with this task, many good books and courses on basic programming skills are currently available. We would particularly recommend the online lessons and in-person trainings provided by the groups Software Carpentry and Data Carpentry. In addition to basic programming, we presume that you have at least some familiarity with the basic principles of scientific research, and that you are either a published author of scientific papers yourself or are aspiring to be one shortly.

For those who are relatively new to computational research and reproducibility, we suggest beginning by carefully reading the chapters in Part I of the book and attempting to follow along with the basic workflow template described in the chapter "The Basic Reproducible Workflow Template," either exactly as presented or as adapted to a new research project of your own choosing. The case studies can then be skimmed, with particular attention paid to the high-level workflows in Part II. The "Glossary" chapter should be referred to regularly when encountering unfamiliar terms and concepts.

For those with more experience in computational research, particularly those who are interested in adapting and advancing their own existing research practices, we recommend focusing first on the chapter "Case Studies in Reproducible Research" and then reviewing all of the case studies themselves. We suggest reading the high-level case studies first, followed by the low-level case studies, with an eye towards identifying particular strategies that may be applicable to your own research problems. The "Lessons Learned" and "Building toward a Future Where Reproducible, Open Science is the Norm" chapters will be useful in providing a synthesis of the current state of reproducible research and prospects and challenges for the future.

Regardless of your current background and skill set, we believe that you will find both inspiration and concrete, readily applicable techniques in this book. It is always important to remember that reproducibility is a matter of degrees, and these examples will demonstrate that while achieving full reproducibility may sometimes be difficult or impossible, much can be gained from efforts to move a research project incrementally in the direction of reproducibility.

Let's get started.