Andrew Bray
University of Massachusetts, Amherst
Amherst, MA
760-519-5979
andrew.bray@gmail.com

Implementing Reproducible Research

Victoria Stodden, Friedrich Leisch, Roger D. Peng.

Publisher Address: CRC Press, 2013. ISBN 978-1-4665-6159-5. xix+428 pp.

The concept of reproducibility is closely related to that of replication. Reproducibility is defined as, “the calculation of quantitiative scientific results by independent scientists using the original datasets and methods” (p. vii). Replication, then, goes the additional step of collecting new data. If replication is one of the foundational principles of science, ensuring a reproducible analysis is its necessary precursor and the focus of this important and timely new book.

In the preface, the editors specify that their objective is not to convince the reader *why* reproducibility is important, but rather *how* it can be achieved. They organize the book into three parts, Tools, Practices and Guidelines, and Platforms, that mirror the three directions of research in reproducibility. One contributor then usefully divides tools for reproducible research into three general categories: tools for literate programming (Ch. 1: knitr in R), workflow management systems (Ch. 2: VisTrails), and tools for environment capture (Ch. 3: Sumatra, Ch. 4: CDE). I was most familiar with knitr, an R package used to blend computing and text, but I still learned alot from the author’s discussion of the design elements that specifically enable reproducible analyses. The remaining tools were new to me and address the added challenges when faced with a more complex workflow and software that involves many dependencies. Note that these chapters are not full tutorials; the reader will not emerge as a well-versed user, but hopefully with a sense of which tools they would like to invest time in learning.

The remainder of the book is a mix of more general and technical pieces, some of which are quite domain specific. Chapters 5, 7, and 13 discuss methods from physical, biological science, and machine learning, respectively, though their approach to computation may be of interest to those in other fields. Chapters 6, 9, and 15 discuss arguments and techniques for practicing open science, an important consideration if the goal is to allow anyone to reproduce a study without barriers. Chapter 8 provides the prospective from within industry of a large-scale data anlysis project. Chapter 10 describes how cloud-based virtual machines (VM) can enable anyone to recreate a computational experience in an environment identical to that of the original authors. Chapter 12 reviews the tenets of traditional intellectual property law and discusses the difficulty this poses for reproducibility. Chapter 14 introduces the reader to runmycode.org, a user-friendly web-based platform that allows one to tweak-the-knobs, so to say, on an analysis.

This book should have broad appeal; the more general chapters from this book should be of interest to any empirical scientist. Most contributors assume some familiarity with scientific computing, and for the most technical chapters, familiarity with UNIX-based platforms will be useful. I can imagine this book serving as a guide to improved reproducibility within a research lab or as terrific material to kickstart discussion for a university working group on reproducibility. As a compendium of best practices, it would also be a valuable read for new graduate students who are in the process of forming their research habits.

As a cover-to-cover read, the book can be repetitive (literate programming, version control, and topics in open science are covered in multiple chapters). It can also be disorganized, with sections varying greatly in level of technical background (e.g. Chapter 4 assumes familiarity with Linux while Chapter 8 reverts to a primer on the basics of computing). Additionally, some readers might be surprised to find that, for a book published as part of *The R Series*, Python gets as much coverage as R, and much of the book is language-agnostic. Considering the many arguments that this book makes for open science, it was reassuring to find that full pdfs of every chapter (including a missing one by Malik et al.) can be downloaded for free through the Open Science Framework (implementingrr.org). One can't help but feel this to be the more natural format for this material: dynamic, modular, and with no barriers to access.

Despite some weaknesses of the book format, *Implementing Reproducible Research* still introduces some extremely useful tools and practices from leaders in the field. On top of that, it also contains exciting visions for the future of scientific research. One is the idea of learning from research *processes* instead of just the end result, which becomes feasible when work is reproducible and transparent. Part of this will be shift away from publishing static papers to publishing full research environments. Another is the promise of collaborative science, from the distributed replication studies of The Reproducibility Project (Ch. 11) to the reevaluation of traditional peer review practices in light of the paradigm epitomized by GitHub's pull-request.

The challenge of reproducibility in the computational era is being confronted across the sciences, with each field developing its own tools and best practices. This book is an very important step in bringing together a broad group of scientists to share what has been learned.

Andrew Bray
University of Massachusetts, Amherst