Interlude — Good practices for scientific computing

https://doi.org/10.1017/9781009212601.020 Published online by Cambridge University Press

https://doi.org/10.1017/9781009212601.020 Published online by Cambridge University Press

About this interlude

Before moving on from the applied focus of Part II to the theoretical aspects of working with network data in Part III, here we take a brief detour to discuss practical aspects of doing data-driven and computationally driven research. Our advice is applicable to network data, of course, but is more general, and we hope useful to any computational research you may conduct.

Chapters 17 and 18 tackle a major bugbear of computer work: keeping track of what you have done and what you have changed and, just as importantly, why. Record-keeping is critical for good science but computer work is different in some fundamental ways compared to the traditional laboratory setting. We discuss ways to cope.

Computational work is driven by computer code. You will read, write, and run code as you work. You want code to be reliable and understandable above all else, with computational efficiency being a further desirable property. What are good strategies for producing quality scientific code? Software engineering principles help, but coding for software and coding for science can be quite different. Turn to Ch. 19 to learn more.

In general, this interlude is structured around advice for computational work that is relatively independent of specific tools. Computer software is a high-churn environment, with packages and programs always coming and going. Any advice we give for very specific software is sure to be soon obsolete. That said, we round out the interlude by discussing specific tools to help you work in Ch. 20.

https://doi.org/10.1017/9781009212601.020 Published online by Cambridge University Press

https://doi.org/10.1017/9781009212601.020 Published online by Cambridge University Press