4-2018

Evaluating Reproducibility in Computational Biology Research

Morgan Oneka
Grand Valley State University

Evaluating Reproducibility in Computational Biology Research

Morgan Oneka
Advisor: Dr. Greg Wolffe

Introduction and Background

The plan for my Honors Senior Project was to study five papers within the field of computational biology. Computational biology is defined broadly as the application of computational methods to solve biological problems, and this includes a large number of specific subfields. For this project, I selected one paper from each of the following five subfields:

Image Processing: A field of interest to the general computer science public, image processing seeks to utilize machine learning methods to automate image analysis. Applications in biology include detection of cancerous cells within a cell culture, or identification of specific biological structures from a PET or CAT scan.

Text Mining/Information Retrieval: With the amount of useful information available in the form of textbooks, research articles, and lab websites, researchers in this area develop new methodologies for automatically acquiring and summarizing data from these sources.

Systems Biology: Many fields of biology involve complex interactions between entities in a system: for example, interactions between organisms in an ecosystem, or between metabolites within the human body. Those who study systems biology develop models of these systems, and use these models to ask and answer questions about biological systems.

High Throughput Data Analysis: With genetic sequencing becoming less cost-prohibitive and more cost-effective, the amount of data generated by modern sequencing technologies has increased dramatically. Computational biologists active in this data-intensive area develop new algorithms for analyzing and interpreting an organism's genome or transcriptome.

Phylogenetics: As genetic information, such as sequenced genomes, becomes available for more organisms, evolutionary biologists have begun to study phylogeny from a genetic perspective rather than a purely phenotypic perspective. This area of computational biology develops new algorithms for inferring relationships between genetic samples, from drastically different organisms to different colonies of the same cancer cell line.

For my senior project, I studied these five papers with the intention of replicating the research each paper described. While I did not anticipate this would be easy by any means, the task was substantially more difficult than expected.

In preparation for this project, I also read a monograph entitled "Ten Simple Rules for Reproducible Computational Research" by Geir Kjetl Sandve. In this paper, published in PLOS Computational Biology, Sandve and his colleagues list what they consider the ten most vital rules one must follow when conducting computational-based research. I initially read this paper in hopes of letting it guide my own work throughout the semester, but as I analyzed others' publications, I realized many of these rules were not followed by the researchers.

Reproducibility is cornerstone of research in all fields of science, but it is especially important in

computational biology, which is still a relatively young field. Reproducible research is easier for peers to study, and thus provide commentary on, allowing for more effecive collaboration among scientists. Additionally, reproducible research can serve as an excellent training tool for those who are new to the field, such as myself.

In this paper, I will address the ten rules enumerated by Sandve, discuss how successfully these rules were adhered to by the papers I studied, and reflect upon how my ability to reproduce their results was affected by this.

The Rules

Rule 1

Rule 1 is "For Every Result, Keep Track of How It Was Produced." This rule requires that the results of all steps, including small pre- or post-processing steps, be included. Sandve also stipulates that every detail that may influence the execution of any step must be recorded—for example, when reporting the results of running a command line program, all parameters and inputs used should be specified.

This rule was, in large part, ignored. Vital steps were described in great detail, but smaller steps (such as data processing) were excluded. This omission creates obstacles when attempting to reproduce the published results, as small steps early on in the workflow can have drastic impacts later in the process, and early mistakes can spell disaster.

Rule 2

Rule 2 of the Sandve paper suggests that manual data manipulation be avoided. It is often tempting to do simple operations by hand—for example, manually adding two or three rows to a CSV file when data has accidentally been excluded —because it is easier than performing the task programmatically.

However, this poses several problems. One problem this introduces is the possibility of human error. A second problem the difficulty in properly describing this step in the published research—or forgetting to document the step entirely, leaving out a potentially crucial step in the analysis workflow.

Fortunately, across the board this was the most-followed rule. None of the papers studied appeared to have manually altered their data—all work was performed using scripts or familiar software packages.

Rule 3

Rule 3 requires researchers to archive exact versions of all external programs that are used. Software is frequently changing, especially popular open source software that is open to the public to contribute to. Version changes can introduce new features, remove old ones, or change the way certain processes are executed.

None of the authors whose papers I studied archived the exact versions of the software they used—at least not anywhere it could be found—but most authors were diligent enough to include the specific version of each external library they used, or at the minimum the ones that were most vital to their research.

Given that most libraries have their own source control and make previous versions easily accessible, this rule feel almost too strict. Having access to a physical copy of the library may be necessary for software whose previous versions are not available online, but since this is not the case for most mainstream libraries, simply reporting the specific version used seems sufficient.

Rule 4

Rule 4 insists that researchers version control all custom scripts. Many of the papers I read kept all major versions of their software available online.

However, some did not. Typically, they only hosted the most recent version. Additionally, "smaller" scripts—such as scripts used to clean data—were not saved anywhere publically.

The effect of this on reproducibility can vary depending on the importance of the script. Simple data cleaning steps—for example, removing rows that have an empty entry—are easily reproduced and do not require access to a stored version. However, larger and more important tasks—for example, software that implements a neural network for purposes of prediction/classification—are much more challenging to reproduce in the exact fashion used in the published research.

Rule 5

Rule 5 of the Sandve paper is "Record All Intermediate Results".

None of the authors whose papers I studied followed this rule in its entirety. While certain

intermediary steps were recorded for nearly every paper, it was certainly not the case that every small intermediate step had its output recorded.

Problematically, even when intermediary results were recorded, they often were not in a standard format. Several papers included intermediate data captured in Microsoft Excel files annotated with comments, others provided a simple .csv file and an accompanying .txt file.

Rule 6

Rule 6 requests that researchers include their random seeds whenever they use random number generators. Generating random numbers is, well, *random*, but providing the seed number used to initialize the generator will allow others to reproduce the same randomness when running the program themselves.

Few papers I read had workflows that depended on random number generation, but those that did had not recorded their random seeds. This is an issue because, even if all other nine rules are followed, a person reproducing an experiment can theoretically obtain vastly different results if the random seeds differ. This can cause confusion, as the reproducer may not be able to tell if the disparity in outcomes is due to an implementation error or if it is caused by a difference in the way randomness was used.

Rule 7

Rule 7 stipulates that researchers always store the raw data represented in plots/graphs, and also the code used to generate those plots if pertinent. Graphs are a vital way of communicating the results of an experiment, and one way to verify that a reproduced workflow yields correct results is to compare reproduced graphs to the original graphs.

This rule was seldom followed. Although a few of the papers I sampled throughout my project contained the raw data behind their plots and graphs, many did not.

When the raw data or details about specific commands used to create the graph are not included, then those repeating research may not be able to reproduce the original chart, and thus will have to forgo a useful means of validating the fidelity of their workflow.

Rule 8

Rule 8 insists that researchers generate hierarchical analysis output. In other words, instead of merely recording a "summary table" or other form of highly aggregated data, researchers should also record more detailed information about each value that appears in the summarized data. A simple example of this would be that an author who provides a table of mean values, also provides the data behind that mean.

This rule was not followed. Most papers I read provided access to the raw data, which technically could be used to find any hierarchical data, but data that had been processed to the point directly before the data had been summarized was not available.

Rule 9

Rule 9 states that researchers should connect textual statements to underlying results. That is, when the author of a research paper makes a claim, that claim is saved alongside the data, and any relevant papers or theories that contributed to the conclusion are also saved alongside the textual interpretation.

Some authors did provide this. At least one of the papers I read provided supplementary information that included a Microsoft Excel spreadsheet with data accompanied by comments explaining why certain conclusions were drawn.

Many, however, did not, and this can be a problem for those who are attempting to use the same workflow but who are not as familiar with the scientific questions being asked. A person with limited background or experience in an area may struggle to understand why a certain conclusion is supported by the provided data.

Rule 10

Rule 10 requests that researchers ensure public access to data, scripts, runs, and results. Any code, debugging information, or program output should be accessible to anyone who is interested in accessing it.

Every paper that I read made at least some of this information publicly available. All authors provided at least a link to the data that was used in the paper, and many provided source code, either as supplementary information on the journal website or on their lab webpage.

However, no author provided all of the required information. Frequently, run information and raw data was not provided.

Reflection

What effects do breaking the rules have?

Clearly, the ten rules Sandve proposes were not followed often. Many of the papers I reviewed did not fully follow any of the ten rules.

This made reproducing the results presented in each paper challenging, especially in the short amount of time allotted for each project. In most cases, I was unable to completely replicate the entire workflow.

Why were the rules broken?

Of course, scientists obviously did not make a conscious decision to create a workflow that was difficult to reproduce. The occurrence was likely the result of many factors.

One aspect that could have contributed to the failure to abide by the rules is possible restrictions placed on data distribution due to privacy concerns. Much biological data, especially that relevant to human health information, requires permission and often specific credentials to be used, and those with access to the data are usually prohibited from sharing its contents or even information about it. This makes it difficult or impossible for scientists to follow any rules that require them to share output or even scripts, as code can also reveal too much about the data.

Another factor that could make it difficult to focus on reproducibility is a sheer lack of time on the part of the researchers. Often, research is conducted with a specific deadline in mind, whether that be submission to a journal or the end of a grant. Focus on producing results is clearly the main objective, so it is understandable that assuring reproducibility may be pushed to the wayside.

Finally, it could simply be that reproducibility is not a priority to begin with. Some investigators may not anticipate that others would want to reproduce their workflow, so they do not even consider taking steps to make their work amenable to that process.

Additional Rules

The rules that Sandve proposes are comprehensive, and I believe that if they were followed, it would make for more reproducible and more understandable research. However, after experiencing the process, I do have some suggestions for additional rules.

Keep a "Lab Notebook"

It is common for lab biologists to maintain a lab notebook, which documents every step they take in great detail. This not only allows anyone who attempts to run the same experiment to understand exactly what workflow to follow, but it helps the biologist keep track of where they are in the process.

If computational biologists kept a "lab notebook" detailing each step taken during a project, this would eliminate the extra time needed to make the end results more reproducible, as one would simply need to transcribe the lab notebook into a publicly available format. This could help eliminate the lack of reproducibility that results from a lack of time.

Containers and Virtual Machines

Another paper I read in preparation for my project was called "Practical Computational Reproducibility in the Life Sciences.", which made several suggestions beyond those in Sandve's paper. One that I believe should be adopted as a "rule" for reproducible research is the use of containers or virtual machines.

Code sometimes runs differently depending on the operating system on which it is run, and the specific packages installed. Because of this, someone faithfully following the workflow described in a paper may still obtain different results simply because the program was run on a differently-configured machine. Containers or virtual machines provide a way for researchers to share and distribute the exact conditions under which a program was run, down to the operating system. This assures that the code will run exactly the same no matter who is running the code.

Conclusion

Reproducibility in computational biology still has a long way to go. However, it is important that computational biologists strive to improve the reproducibility of their work. Reproducibility is vital to the growth of the field, as it allows new scientists to learn from their predecessors and makes it simpler for scientists to understand their peers'

work, allowing for increased collaboration. As research in computational biology progresses, we must make reproducibility a priority, as it will be one of the important factors that helps this new field grow.

Sources

Gruning, B., Chilton, J., Köster, J., Dale, R., Goecks, J., & Backofen, R. (2017). Practical computational reproducibility in the life sciences, 1–12. https://doi.org/10.1101/200683

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten Simple Rules for Reproducible Computational Research. PLoS Computational Biology, 9(10), 1–4. https://doi.org/10.1371/journal.pcbi.1003285