the WINNOWER

BIOLOGICAL SCIENCES

Avoid having to retract your genomics analysis.

YANNICK WURM [1]

1. Queen Mary University of London

CORRESPONDENCE:
y.wurm@qmul.ac.uk

DATE RECEIVED:
July 15, 2015

DOI:
10.15200/winn.143696.68941

ARCHIVED:
July 15, 2015

KEYWORDS:
reproducibility, retraction, code, genomics, analysis, data analysis

CITATION:
Yannick Wurm, Avoid having to retract your genomics analysis., *The Winnower* 2:e143696.68941 , 2015 , DOI: 10.15200/winn.143696.68941

AVOID HAVING TO RETRACT YOUR GENOMICS ANALYSIS.

BIOLOGY IS A DATA-SCIENCE

The dramatic plunge in DNA sequencing costs means that a single MSc or PhD student can now generate data that would have cost $15,000,000 only ten years ago. We are thus leaping from lab-notebook-scale science to research that requires extensive programming, statistics and high performance computing.

This is exciting & empowering - in particular for small teams working on emerging model organisms that lacked genomic resources. But with great powers come great responsibilities... and risks of doing things wrong. These risks are far greater for genome biologists than, say physicists or astronomers who have strong traditions of working with large datasets. In particular:

- biologist researchers generally learn data handling skills *ad hoc* with little knowledge of best practices;
- PIs - having never themselves handled huge datasets - have difficulties critically evaluating the data and approaches;
- new data are often messy with no standard analysis approach; even so-called "standard" analysis methodologies generally remain young or approximative;
- analyses intending to identify biologically interesting patterns (e.g., genome scans for positive selection, GO/gene set enrichment analyses) will enrich for technical artifacts and underlying biases in the data;
- data generation protocols are immature & include hidden biases leading to confounding factors (when things you are comparing differ not only according to the trait of interest but also in how they were prepared) or pseudoreplication (when one independent measurement is considered as multiple measurements).

INVISIBLE MISTAKES CAN BE COSTLY

Crucially, data analysis problems can be invisible: the analysis runs, the results seem biologically meaningful, and are wonderfully interpretable but they may in fact be completely wrong.

Geoffrey Chang's story is an emblematic example. By the mid-2000s this young superstar professor crystallographer had won prestigious awards and published high-profile papers providing 3D-structures of important proteins. For example:

- **Science** (2001) Chang & Roth. *Structure of MsbA from E. coli: a homolog of the multidrug resistance ATP binding cassette (ABC) transporters.*
- **Journal of Molecular Biology** (2003) Chang. *Structure of MsbA from Vibrio cholera: a multidrug resistance ABC transporter homolog in a closed conformation.*
- **Science** (2005) Reyes & Chang. *Structure of the ABC transporter MsbA in complex with ADP vanadate and lipopolysaccharide.*
- **Science** (2005) Pornillos et al. *X-ray structure of the EmrE multidrug transporter in complex with a substrate.* 310:1950-1953.
- **PNAS** (2004) Ma & Chang *Structure of the multidrug resistance efflux transporter EmrE from Escherichia coli.*

But in 2006, others independently obtained the 3D structure of an ortholog to one of those proteins. Surprisingly, the orthologous structure was essentially a **mirror-image** of Geoffrey Chang's result.

Rigorously double-checking his scripts, Geoffrey Chang then realized that: *"an in-house data reduction program introduced a change in sign [..,]"*.

In other words, a simple +/- error led to plausible and highly publishable but dramatically flawed results. He retracted all five papers.

Devastating for him, for his career, for the people working with him, for the hundreds of scientists who based follow-up analyses and experiments on the flawed 3D structures, and for the taxpayers or foundations funding the research. A small but costly mistake.

APPROACHES TO LIMIT THE RISKS

A +/- sign mistake seems like it should be easily detectable. But how do you ensure that experiments requiring complicated data generation and complex analysis pipelines with interdependencies and sophisticated data structures yield correct results?

We can **take inspiration from software developers in internet startups**: similarly to academic researchers, they form small teams of qualified people to do great things with new technologies. Their approaches for making software robust can help us make our research robust.

An important premise is that humans make mistakes. Thus (almost) all **analysis code includes mistakes** (at least initially; this includes unix commands, R, perl/python/ruby/node scripts, *et cetera*). Increasing robustness of our analyses thus requires becoming better at detecting mistakes - but also ensuring that we make fewer mistakes in the first place. Many approaches exist for this. For example:

- Every additional chunk of code can contain additional mistakes. Write less code, you'll make fewer mistakes. For this we should try to reuse our own code and that of others (e.g., by using bio* libraries).
- Every subset/function/method of every piece of code should be tested on fake data (edge cases) to ensure that results are as expected (see unit and integration testing). It can be defendable to write the fake datasets and tests even before writing analysis code.
- Continuous integration involves tests being automatically rerun (almost) instantly whenever a change is made anywhere in the analysis. This helps detect errors rapidly before performing full analyses.
- Style guides define formatting and variable naming conventions (e.g., for ruby or R). Respecting one

makes it easier for you to go back over your analysis two years later (e.g., for paper revisions or a new collaboration); and for others to reuse and improve it. Tools can automatically test whether your code is in line with the style guide (e.g., RLint, Rubocop, PyLint).
- Rigorously tracking data and software versions and sticking to them reduces risks of unseen incompatibilities or inconsistencies. A standardized project structure can help.
- Code reviews: having others look through your code - by showing it to them in person, or by making it open source - helps to learn how to improve code structure, to detect mistakes and to ensure that our code will be reusable by ourselves and others.
- There are specialists who have years of experience in preventing and detecting mistakes in code or analyses. We should hire them.
- Having people independently reproduce analyses using independent laboratory and computational techniques on independently obtained samples might be the best validation overall...

This list overlaps at least in part with what has been written elsewhere and my coursework material. In my lab we do our best to follow best practices for the bioinformatics tools we develop and our research on social evolution.

Additionally, the essentials of experimental design are long established: ensuring sufficient power, avoiding confounding factors & pseudoreplication (see above & elsewhere), and using appropriate statistics. Particular caution should be used with new technologies as they include sources of bias that may not be immediately obvious (e.g. Illumina lane, extraction order...).

THERE IS HOPE

There is no way around it: analysing large datasets is hard.

When genomics projects involved tens of millions of $, much of this went to teams of dedicated data scientists, statisticians and bioinformaticians who could ensure data quality and analysis rigor. As sequencing has gotten cheaper the challenges and costs have shifted even further towards data analysis. For large scale human resequencing projects this is well understood. Despite the challenges being even greater for organisms with only few genomic resources, surprisingly many PIs, researchers and funders focusing on such organisms suppose that individual researchers with little formal training will be able to perform all necessary analysis. This is worrying and suggests that important stakeholders who still have limited experience of large datasets underestimate how easily mistakes with major negative consequences occur and go undetected. We may have to see additional publication retractions for awareness of the risks to fully take hold.

Thankfully, multiple initiatives are improving visibility of the data challenges we face (e.g., 1, 2, 3, 4, 5, 6). Such visibility of the risks - and of how easy it is to implement practices that will improve research robustness - needs to grow among funders, researchers, PIs, journal editors and reviewers. This will ultimately bring more people to do better, more trustworthy science that will never need to be retracted.

ACKNOWLEDGEMENTS

This post came together thanks to the SSI Collaborations workshop, Bosco K Ho's post on Geoffrey Chang, discussions in my lab and through interactions with colleagues at the social insect genomics conference and the NESCent Genome Curation group. YW is funded by the Biotechnology and Biological Sciences Research Council [BB/K004204/1], the Natural Environment Research Council [NE/L00626X/1, EOS Cloud] and is a fellow of the Software Sustainablity Institute. Cross-posted at The Winnower

June 2, 2015