How to be a successful computational scientist.

C. Titus Brown

Best Practice #1: never show them your data.

No one can disagree with your results if you don't give them access to your data.

Be sure not to back up your data, as this makes it difficult to claim that you've lost it.

Only keep the intermediate stages of your data around. That way you have plausible deniability: "Oh, that's not the final data set. We smoothed out that inconsistency."

When people leave the lab, urge them to take their data with them. Then lose their e-mail addresses.

Best Practice #2: do not, under any circumstances, communicate clearly.

Do not document your code or data. If you must, use a language other than English. This minimizes the chance of someone discovering a flaw in your logic.

Make sure that no one author is responsible for the contents of a paper. This will make it impossible for people to ask detailed questions of the authors.

Keep as much information as possible in an opaque format like Word, that cannot be searched or opened by everyone.

Never write details in e-mails. Assume that someone will get ahold of your e-mail someday.

Best Practice #3: never release your source code.

Even if your logic is sound, your code may not be. This way, no one will ever find out.

Use multiple programming languages, even (or especially) if you don’t know them well. The only people who know more than one programming language don’t know enough science to argue with your results.

If no one has your source code, no one can use it to do their own science and scoop you.

Best Practice #4: judge computational science by results, not quality.

As long as you're producing good results, who cares if they're right?

Grants are based on *your* results, not on replicated or reproduced results.

Best Practices #1, #2, and #3 ensure that no one will ever know.

Best Practice #5: use as much data as possible.

This is self-evident: the more data you have, the more correct your hypothesis must be.

It costs money. You have to use it somehow!

And, after all, much of that data is noisy. You're heroic for pulling *any* signal out of it! What signal you do see has to be correct.

Plus, you didn't get wealthy by being sloppy. Wealth is just another sign that you're a good scientist.

...but seriously...

- It's hard to do good computational science.
- (I hope we've conveyed that.)
- But it can be a lot of fun.
- It's just another way to do science, after all.
- Just think about what could go wrong and plan accordingly!

The Trouble with Replication

- Most research is never replicated eactly, but rather reproduced (qual/quant similar results on different data). That's OK.
- However, on occasions when people have tried, most research (estimates range to over 88%) *cannot* be replicated. That's not OK.
- Consider: that last statement is about *cancer research*…
- **You cannot build on a shaky foundation.**

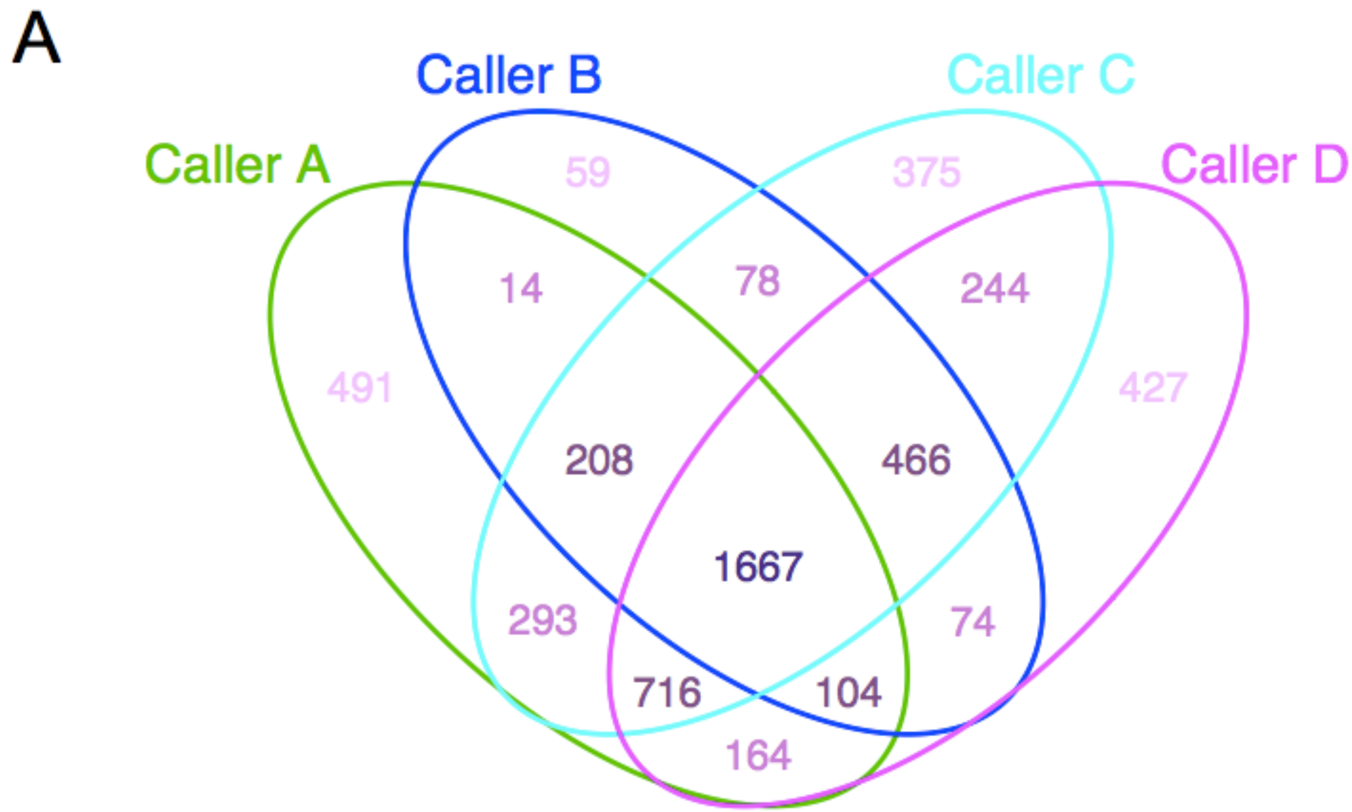
Towards responsible conduct of computational research: some proposed principles

1. Computation is part & parcel of the research.

2. All inputs to the computational process need to be recorded & relayed as part of publication, incl data (in as raw a state as possible) and analysis parameters.

3. The goal is to allow replication by reviewers and readers.

It's easy to be nihilistic about computing "correctness"



Kim & Speed, 2013

…but just because it’s not entirely *right* doesn’t mean it’s not useful.

- Document what you did computationally.
- Make sure that your data and analysis support your actual conclusions.
- Run it by some computational colleagues before submitting.

- …and most reviewers will miss this stuff anyway.

This is the end.

Wait! Not Yet!

- Terminate your instances.
- Hand in your lunch badges, unless you need 'em.
- Post-mortem discussion:
  - Good
  - Bad
  - Ugly
- Group photo!