# Data detectives, self-love, and humility: a research parasite's perspective

Claire Duvallet

## Abstract

Secondary analysis solidifies and expands upon scientific knowledge through the re-analysis of existing datasets. However, researchers performing secondary analyses must develop specific skills to be successful, and can benefit from adopting some computational best practices. Recognizing this work is also key to building and supporting a community of researchers who contribute to the scientific ecosystem through secondary analyses. The Research Parasite Awards are one such avenue, celebrating outstanding contributions to the rigorous secondary analysis of data. As the recipient of 2019 Junior Research Parasite, I was asked to provide some perspectives on life as a research parasite, which I share in this commentary.

## Introduction

In sharp contrast with some unfounded fears that open data will lead to the rise of "research parasites" (Longo 2016), publicly available raw data adds substantial value to scientific research, allowing researchers to ask and answer new questions and to test hypotheses *in silico* without the need for costly new studies. In my PhD, I performed a meta-analysis of 28 gut microbiome studies, where I downloaded and reprocessed raw sequencing data to look at patterns of associations between the gut microbiome and 10 different diseases (Duvallet et al. 2017). When I first embarked on this project, I thought it would be straightforward to use existing data to answer some low-hanging-fruit questions in my field. But I soon learned that performing a meta-analysis is more than a "stamp collection" endeavor, and is rather a unique, difficult, and valuable contribution to the field (Glass 1976). As the recipient of the 2019 Junior Research Parasite, I was recognized for my contribution to the rigorous secondary analysis of microbiome data and given this opportunity to share some of the lessons I learned (Greene 2017).

Fig 1: What is a research parasite? [Casey's award timeline infographic]

## 1. Research parasites are data detectives

First, a research parasite must become a data detective. While databases and search engines ease the process of finding data, identifying a comprehensive list of datasets for niche research questions remains challenging. In my meta-analysis, I learned to supplement my database searches by chasing down references from papers and reviews. I became skilled at scrutinizing papers to find links to raw data and metadata, information which was often scattered throughout the paper and supplement. I eventually gained some intuition: if a ctrl-F for certain data-related keywords was not fruitful, I knew I should probably go ahead and email the authors. But even that required detective work in order to find up-to-date email addresses and, at times, contact information for the first author, who might be more inclined to reply to my email. Through every step of the process, I learned to notice strategic details and scour available information for any clues that might help me find what I needed.

## 2. Data without metadata is useless

Without all of the relevant technical and clinical metadata, it is impossible to re-analyze data and the entire dataset becomes useless. As a parasite, I identified three broad types of dataset-related metadata: (1) metadata linking the raw data IDs (e.g. file names, SRA run IDs) to their corresponding sample IDs, (2) metadata containing technical processing information (e.g. sequencing barcodes, sample replicate number, sampling date, etc) and (3) biological or clinical metadata (e.g. disease status, tissue type, etc). Here also, there are many tricks to learn, for example: looking at raw sequencing data to infer the presence or absence of primers or barcodes, making educated guesses about disease status from sample IDs, and looking in all the nooks and crannies of SRA or ENA sample descriptors.

Figure 2: Data and metadata components required for a successful re-analysis. (A) Raw sequencing data is usually labeled with an SRR Run ID or other processing ID. Raw data rarely contains information about the processing steps applied, and the research parasite must use other information to determine what processing needs to be done. (B) Technical metadata connects file names to the respective sample ID, and may also have other technical information like primer and barcode sequences. In cases where the raw sequencing data has not been separated into sample-wise files, barcodes are required to map sequences to samples. These are often the most difficult data to find. (C) Finally, study-related metadata is required to re-analyze samples. Metadata directly related to the analysis question is always necessary (i.e. disease status), but other metadata like subject ID, sampling day, replicate, etc may also be required to ensure that proper statistical comparisons are being made.

## 3. Organize yourself for success

As research parasites, we make mistakes and need to start analyses over from scratch many times. Intentional project and file organization can ensure that starting over is not a catastrophe each time. More specifically, separating source code from the data and producing intermediate files for each step of an analysis is key to unlocking parasitic productivity (Cookie Cutter Data Science, Wilson et al. 2016). Clear project organization makes it easier to remember what different files are, to implement automated pipeline workflows, and to write parallel and modular scripts.

Figure 3: a sample project structure, adapted from Cookie Cutter Data Science for my meta-analysis. In microbiome projects, OTU tables are the feature tables which result from processing raw sequencing data, and which serve as the input to all analyses. (A) Overall project structure; (B, C, and D) individual folder structures.

## 4. Documentation is clues for future-you

In a way, README's are a research parasite's lab notebook. Re-analyzing other people's datasets often means long and convoluted paths form raw data to insight. It's important to document every step along this path: how the data was downloaded and processed, including any strange things that happened and why certain decisions were made; how the scripts and data are organized and where all the (raw) data lives; and how others can use or build upon what has been done.

As a research parasite, the most likely beneficiary of thorough documentation is future-you, checking your work before submission, looking for code to do something you know you've already figured out, or going back to answer a question that woke you up in a cold sweat at 2 am. This was a lesson learned the hard way for me: I remember going back to finalize processing on datasets I'd downloaded over a year prior, and not finding any clues from myself to indicate what I'd done. In all of these cases, it was faster to re-download the data from scratch than to time-travel into the past and remember what I had been thinking.

## 5. The life-changing magic of Makefiles

In my opinion, the most revolutionary parasitic tactic of all is pipelining workflows with Makefiles. Broadly, Makefiles contain all of the instructions to reproduce every part of an analysis and are usually paired with a software program that reads the instructions in the Makefile to automatically re-run all parts of the analysis which need to be updated (Stallman 1991, Koster 2012). Makefiles are incredibly useful because they enable the re-making of an entire analysis and - more importantly - they are by definition a documentation of all the steps taken. They quite literally answer the question: "wait, how did I do that again?"

Makefiles also make the review process much easier, saving time and freeing research parasites from tedious time-consuming details. With a Makefile, it is easy to go back, find, and double-check individual steps in an entire analysis, as opposed to digging through folders of spaghetti code without guidance. Adding new analyses also becomes easier, as new scripts can be seamlessly plugged into the Makefile, using intermediate analysis files already created from other steps in the workflow. And most importantly, Makefiles take care of the details, automatically updating figures and tables when their underlying inputs change, saving research parasites many hours of tedious detail-checking.

Figure 4: Schematic of a subset of the steps involved in the workflow for my meta-analysis. Each box contains a description of the data or analysis at that step, and arrows indicate progression through the analysis workflow. Boxes are colored according their respective location in the project structure. Dependencies between input data, intermediate files, and scripts are encoded in the Makefile, which automatically re-runs any necessary steps when input data or files are updated. For example, if the script to perform univariate steps is updated, all steps which depend on that box get re-run and updated; i.e. Figure 1, 2, and 3a are all re-made. In contrast, if the code to run the random forests is updated, only Figures 1 and S1 are updated.

## 6. Computational best practices as an act of radical self-love

Documentation and pipelining are two manifestations of an important philosophy that is crucial to the research parasite's success: implementing best practices in computational research is an act of radical self-love. Acting in self-love means approaching all decisions not for current-you, but instead for the person who will bear the consequences of your decisions: future-you, long after you've forgotten anything you've thought today. It isn't possible to enumerate all of the possible best practices or even to identify which ones are practical to implement as PhD students, but I found that as long as I made decisions for future-me rather than only current-me, my life as a parasite drastically improved.

## 7. Democratized software catalyzes data generation by non-specialists

Throughout my PhD, I realized that being a parasite is not just about data: developing software tools that encourage non-bioinformaticians to analyze their own datasets is also key to a future where secondary analyses are the norm. In the microbiome field, we've seen a recent development of incredibly accessible bioinformatics tools that democratize access to computational analyses (Bolyen et al 2019). I hypothesize that these easy-to-use analysis tools have contributed to an increase in non-specialists generating microbiome sequencing data.

For example, many microbiome-related clinical trials now include microbiome analyses as secondary endpoints or to raise the impact of a study. These analyses would be inaccessible without the easy-to-use software being developed by the computational microbiome community. Interestingly, these studies rarely dive into the full potential of their data because they are focused on their primary clinical endpoints. That leaves a lot of room for research parasites to come in and ask different questions of those datasets, which I think is a great byproduct of developing accessible software suites.

## 8. Practice humility and broaden your perspective

Finally, the biggest lesson I learned from my life as a parasitic PhD student was humility. As a research parasite, it's easy to get self-righteous and angry whenever the data or metadata isn't easily accessible, but at the end of the day it's not productive. Especially as junior PhD students, we must recognize that what we view as just additional N's in our analyses, clinicians see as real people who gave their time and samples to advance research, and in some cases, as patients who can be deeply suffering.

Humility and empathy is also important to recognize that while we stand on our soapboxes advocating for open and reproducible science, it's actually really difficult to do well. An example is data sharing: parasites might like to grumble at how hard it is to find and download raw data, but depositing data is itself a complicated process. Data generation is a collaborative, lengthy process, and the person depositing the data is rarely involved in every step of the data generation. Even after downloading and processing these 28 datasets, I myself was incredibly confused when I needed to deposit data for a different project that I worked on, and I'm sure I missed some important information that will make a future parasite grumble herself. Thus, not only do we need to require data sharing, but we also need to make it easier, more accessible, and more amenable to improvements and feedback, in addition to encouraging and rewarding those who do share their data (e.g. like the Research Symbiont award, a partner to the Research Parasite Award: https://researchsymbionts.org/). And as we're doing our parasitic research, we need to keep in mind that we are just one small part of an entire ecosystem, and that as parasites, we depend on our hosts to survive.

## References

Evan Bolyen, Jai Ram Rideout, Matthew R. Dillon, Nicholas A. Bokulich, et al. "Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2." Nature biotechnology 37, no. 8 (2019): 852-857.

Cookie Cutter Data Science. https://drivendata.github.io/cookiecutter-data-science/. Accessed September 28, 2019.

Claire Duvallet, Sean M. Gibbons, Thomas Gurry, Rafael A. Irizarry, and Eric J. Alm. "Meta-analysis of gut microbiome studies identifies disease-specific and shared responses." Nature communications 8, no. 1 (2017): 1784.

Casey S. Greene, Lana X. Garmire, Jack A. Gilbert, Marylyn D. Ritchie, and Lawrence E. Hunter. "Celebrating parasites." Nature genetics 49, no. 4 (2017): 483.

Gene V. Glass. "Primary, secondary, and meta-analysis of research." Educational researcher 5, no. 10 (1976): 3-8.

Johannes Köster, and Sven Rahmann. "Snakemake—a scalable bioinformatics workflow engine." Bioinformatics 28, no. 19 (2012): 2520-2522.

Dan L. Longo and Jeffrey M. Drazen. "Data sharing." (2016): 276-277.

Stallman RM, McGrath R. , GNU Make—A Program for Directing Recompilation , 1991 http://www.gnu.org/software/make/

Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, Tracy K. Teal. 2019. "Good enough practices in scientific computing." https://arxiv.org/abs/1609.00037