Building bridges between data providers and users: best practices and lessons learned

foo bar[*,a]

[a]*rOpenSci, Museum of Paleontology, University of California, Berkeley, CA, USA*

Abstract

Corresponding Author:

foo bar

rOpenSci, Museum of Paleontology, University of California, Berkeley, CA, USA

Email address: stuff@ropensci.org

[*]Corresponding author
Email address: `stuff(at)ropensci.org` (foo bar)

abstract text ...

Introduction

intro text ...

OUTLINE (from chat btw Carl/Scott)

- use cases necessitate FTP vs. csv dump vs. rest API vs etc.
- within APIS: what are best practices
- discoverability - how do you find out if an API has data (taxonomic, geospatial, etc.) you want?
- data formats: tabular vs. JSON/XML, etc.
- combining data: e.g., using identifiers instead of names
- best practices in relational databases, briefly
- ropensci filling gaps btwn data providers and scientists
    - communicating from data provdiers to sci.
    - communicating from sci. to data provdiers

Overview of the landscape

There is an increasing amount of data available to researchers. Leveraging that data efficiently and reproducibly ideally requires software. However, researchers have limited time - thus, most rightly focus on the science. Furthermore, we want researchers to focus on science. Given this, there are a number of issues that others must handle:

- We need well made open source software to help researchers leverage data
- We need people as bridges between data providers and data users (researchers)
- Slack needs to be tacken up to help small scientific databases
- We need best practices for working with data

Matching the data format to the use case

There are innumerable data formats, and there is no one data format that is best for every situation. Ideally, one leverages the right data format for their use case. The following use cases highlight some of the diversity and what data formats they best match.

- Large amount of data: This varies surely, and depends on connection speeds, computers available, etc. - but for sake of argument lets say that > 1 GB is large data for this use case. It doesn't make a lot of sense to provide this through an API, and makes sense to provide as a compressed format. Data of this type makes sense to provide via FTP or similar (Amazon S3, http file server, etc.).

- Small amount of data: Probably the majority of data use cases are "small data". For example, a spreadsheet with 100 or 100,000 rows is small data. The delivery mechanism can be more flexible with this kind of data. You can definitely serve this data over FTP, but can also simply provide csv/tsv files, and can serve the data over an API.

- Data constantly changing: This is a good use case for delivering data via an API. APIs connect to an underlying database that can change as much as the data providers need. However, the API ideally changes only very slowly so that clients can depend on the interface. It's easier to update data incrementally over an API than if there's small changes in lots of files on an FTP server.

Discoverability

Discovering what kind of data you can get from a data source before actually getting that data is an imporatnt one. For example, say there's a database with 1000 GB of data. You don't want to have to download that database to your own machine, then search through it to find the data you want. Ideally there's a fast way to query a database (perhaps it's metadata) before delving into the data fetching process that will likely take longer.

Unfortunately, most datasets are very lacking in metadata. However, when metadata is well filled out, it makes data discovery much easier. For example, ...

Suggestions?: do x. y, and z

Bridges between data providers and users

Acknowledgments

This project was supported in part by the Alfred P Sloan Foundation (Grant No. G-2014-13485), and in part by the Helmsley Foundation (Grant No. 2016PG-BRI004).

References