If and when a paper on astroquery is written, we'll use this article (and its related github repo: https://github.com/adamginsburg/astroquery-paper) as the basis. However, it is not even a work in progress yet.

1 Introduction

Sharing data is a critical component of astronomical research. Astronomy has historically been a leading field in data sharing, motivated at least in part by questions that cannot be answered with single instruments. In the past few decades, blind surveys have played a huge role in advancing our understanding of the universe.

Data sharing has taken on a variety of forms. The most prominent are the major observatory archives: MAST, NOAO, ESO, IPAC, CADC, CDS (hosting Vizier and SIMBAD), NRAO, CXC, HEASARC, ESA are the main organizations hosting raw and processed data from ground and space based telescopes. These data archive also serve as the primary means for serving data to users when the data are taken in queue mode, i.e., when the data are taken while the observer is not on-site.

In addition to observatories and telescopes, individual surveys often share their full data sets. In some cases, these data sets are shared via the observatory that acquired them - for example, the all-sky data acquired with Planck, WMAP, and COBE delivered a variety of data products as part of the mission. Other surveys, particularly ground-based surveys, serve their own data. Examples include SDSS, UKIDSS, ...

Individual teams and small groups will often share their data. These services do not follow any particular standard and can be widely varied in the type and amount of data shared.

Finally, there are other data types relevant to astronomy that are not served by the typical astronomical databases. Examples include molecular and atomic properties.

Astroquery arose from a desire to access these databases from the command line in a scriptable fashion. Script-based data access provides astronomers with the ability to make reproducible analysis scripts in which the data are acquired and processed into scientifically relevant results with minimal overhead.

The Virtual Observatory has some overlap with astroquery, but it does not provide the simple tools many astronomers find necessary in day-to-day work. Additionally, many of the services noted above do not support VO standards or protocols and are therefore inaccessible to the VO.

2 The Software

Astroquery consists of a collection of modules that mostly share a similar interface, but are meant to be used independently. They are primarily based on a

common framework that uses the python `requests` package to perform HTTP requests to communicate with web services.

For new development, there is a `template` that lays out the basic framework of any new module. All modules are based on having a single core `class` that will have some number of `query_*` methods. The most common query methods are `query_region`, which usually provide a "cone search" functionality, i.e., they search for data within a circularly symmetric region projected on the sky.

An example using the SIMBAD interface is shown below (see `http://astroquery.readthedocs.io/en/latest/simbad/simbad.html`):

```
from astroquery.simbad import Simbad
result_table = Simbad.query_region("m81")
```

In this example, `Simbad` is an instance of the `astroquery.simbad.SimbadClass` class. The returned result, stored in the variable `result_table`,

3 The development model

Astroquery has received contributions from 53 people as of June 2017. While the primary maintenance burden is shouldered by 2-3 people at any given time, most individual modules have been implemented independently by interested volunteers. Additionally, astroquery has received support from the Google Summer of Code program, with three students (coauthors Madhura Parikh, Simon Liedtke, and Ayush Yadav) from 2013-2017.