How to retrieve and use PS1 data

Currently the primary access to the PanSTARRS catalog is through database queries using the MAST CasJobs interface, and the primary access to the PS1 images is through the PS1 Image Cutout Service. There is also a simple web form to search the catalog for objects near a sky position and a VO-compatible cone search interface for simple positional searches. A VO TAP service also provides direct SQL query access to the database. Both catalog and image interfaces can be used from scripts. The DR2 search API is fully functional and can be used to search various tables with constraints from any of the table columns. The PS1 images are also accessible via the MAST Portal. Python notebooks are provided with examples of access via various interfaces.

The starting point for the PS1 data archive is at Pan-STARRS1 data archive home page.

Data Retrieval Quick Links	
Catalog and Database Access	
Search form	https://catalogs.mast.stsci.edu
VO Cone Search	https://gsss.stsci.edu/webservices/vo/CatalogSearch.aspx?CAT=PS1dr2OBJECTS&RA=*<ra>*&DEC=*<dec-degrees>*&SR=*<radius-degrees>*
CasJobs	https://mastweb.stsci.edu/ps1casjobs/
VO TAP	https://vao.stsci.edu/PS1DR2/tapservice.aspx
Image Access	
Image cutout service	https://ps1images.stsci.edu/cgi-bin/ps1cutouts
MAST portal	https://mast.stsci.edu

Contents

- Catalog Access Tools
  - Web search form
  - Catalogs API for scripted searches
  - Easy cross-match using catalogs API
  - VO cone search
  - SQL query interface: MAST CasJobs
  - SQL query interface: VO TAP
- Image Access Tools
  - PS1 Image Cutout Service
  - MAST Portal
- Moving Target Search

Catalog Access Tools

The PanSTARRS catalog is a large database of the objects and measurements of their properties including astrometry, photometry, and morphology. It includes copious metadata including links to the images that were used for the various object measurements. The total database size for the DR2 release is nearly 150 TB.

Web search form

A simple web form searches the object catalog for objects near a search position.

Features of the https://catalogs.mast.stsci.edu/ web form:

- The form includes a name resolver allowing positions to be specified using object names recognized by SIMBAD or NED.
- The form can search multiple tables from the catalog, including mean objects, stack objects, and detections. Additional tables will be added in the near future.
- A list of target names or positions can be uploaded to search multiple positions at once. (Currently this is relatively slow, but enhancements coming soon will make it much faster.)
- The user can customize the columns returned from the search and can add additional constraints on the search (e.g., using magnitudes or other columns from the catalogs).
  - We recommend using the *nDetections>=2* constraint when searching the mean objects catalog to reduce the number of spurious objects in the catalog. Such object are usually detected at only a single epoch.
- The search results from this form can be view interactively or downloaded as a comma-separated value file.
- **Note**: The old web form at https://archive.stsci.edu/panstarrs/search.php remains available but can only be used to search the DR1 catalog. It also lacks many of the filtering capabilities of the new interface. We recommend that all users of the old interface switch to the new interface. Eventually the old interface will be retired.

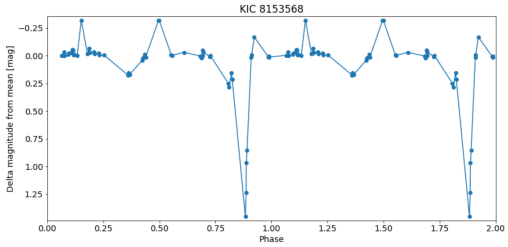
Catalogs API for scripted searches

The search form has an associated application program interface (API) that can be used for programmatic (scripted) queries to the catalog. The API includes access to all the functionality of the interface, including constraints, selection of columns, etc. It can return the data in various formats including CSV, JSON, VOTables. It also has a metadata query to get a list of the available columns for a particular table. The API can be used for larger queries than the web interface, which is limited to returning 50,000 rows of data.

It is simple to use the API either from a language such as Python or with other command-line tools such as curl.

```
In [25]:
period = 3.6071431
bjd0 = 54999.289794
t = ((dtab['obsTime']-bjd0) % period) / period
dmag = dtab['dmag']
w = np.argsort(t)
# extend to two periods
nw = len(w)
w = np.append(w,w)
t = t[w]
# add one to second period
t[-nw:] += 1
dmag = dmag[w]
xlim = [0,2.0]
ylim = np.array([dmag.max(),dmag.min()])
ylim = ylim + np.array([-1,1])*0.02*(ylim[1]-ylim[0])

pylab.rcParams.update({'font.size': 14})
pylab.figure(1,(12,6))
pylab.plot(t,dmag,'-o')
pylab.xlim(xlim)
pylab.ylim(ylim)
pylab.xlabel('Phase')
pylab.ylabel('Delta magnitude from mean [mag]')
pylab.title(objname)
pylab.tight_layout()
```



See our **Python Jupyter notebook for the MAST API** for simple examples of how to access the PS1 DR2 catalog via the API.

Here is a **simple 1-line command using curl** to get a csv table of source information from the mean catalog. This searches the PS1 DR2 mean catalog for sources within 0.5 degrees of the position of NGC 6822. It restricts the result to objects with nDetections >= 2 and increases the default limit on the number of sources returned to 500,000. This query executes in less than one minute:

```
curl 'https://catalogs.mast.stsci.edu/api/v0.1/panstarrs/dr2/mean.csv?
ra=296.24059&dec=-14.80343&radius=0.5&pagesize=500000&nDetections.gte=2' >
ngc6822.csv
```

Easy cross-match using catalogs API

The MAST catalogs interface has a cross-match capability that can be easily used to match lists of source positions. See Easy cross-match with PS1 using a list of source positions for an example.

VO cone search

- Mean object search https://gsss.stsci.edu/webservices/vo/CatalogSearch.aspx?CAT=PS1dr2OBJECTS&RA=<*ra*>&DEC=<*dec-degrees*>&SR=<*radius-degrees*>
- Detection search https://gsss.stsci.edu/webservices/vo/CatalogSearch.aspx?CAT=PS1dr2DETECTIONS&RA=<*ra*>&DEC=<*dec-degrees*>&SR=<*radius-degrees*>
- A generic description of this multi-catalog service is available at https://gsss.stsci.edu/Software/WebServices.htm.

SQL query interface: MAST CasJobs

- https://mastweb.stsci.edu/ps1casjobs
- The CasJobs interface (which is familiar to users of SDSS) includes built-in Help along with some sample PS1 queries. It also includes links to SQL tutorials.

Query Pan-STARRS DR2 catalog using CasJobs

This script shows how to query the Pan-STARRS DR2 catalog using a Python interface to Casjobs. The examples show how to do a simple cone search, how to manipulate the table of results, and how to get a light curve from the table of detections.

This relies on the mastcasjobs Python module. Follow the installation instructions given here: https://github.com/rlwastro/mastcasjobs.

You must have a MAST Casjobs account (see https://mastweb.stsci.edu/ps1casjobs to create one). Note that MAST Casjobs accounts are independent of SDSS Casjobs accounts.

For easy startup, set the `CASJOBS_WSID` and `CASJOBS_PW` environment variables with your Casjobs account information. You can get your WSID by going to https://mastweb.stsci.edu/ps1casjobs/changedetails.aspx after you login to Casjobs. Your password is what you enter when logging into Casjobs.

This script prompts for your Casjobs WSID and password if the environment variables are not defined.

You can also specify your wsid and password directly in the MastCasJobs initialization using the userid and password keyword parameters.

This notebook is available for download.

```
In [1]:
%matplotlib inline
import mastcasjobs
from astropy.io import ascii
from astropy.table import Table

import sys
import os
import re
import numpy as np
import pylab
import json
```

- The MAST CasJobs interface also provides access to other catalogs including GALEX, Kepler, and the Hubble Source Catalog.
- Note that CasJobs queries can also be executed from Python. See our Python Jupyter notebook for PS1 CasJobs queries for examples of one approach to accessing CasJobs from Python. Note that this notebook does the same queries as the MAST API notebook, so a comparison between the two may help in understanding the relationship between the two interfaces.

SQL query interface: VO TAP

- The VO Table Access Protocol (TAP) interface allows database queries through a SQL-based language. The MAST PS1 TAP interface can be used to query the PS1 database. See the TAP documentation for details about how TAP services work.
- A Jupyter notebook shows how to query the database from Python.
- The PS1 TAP service is also accessible from standard VO tools such as TOPCAT. In TOPCAT, use the VO Table Access Protocol menu to get a list of list of services, and then search for PS1DR2 to find the TAP service.

Image Access Tools

The PS1 images are available for the five PS1 filters for combined multi-epoch stacked images. and, with DR2, to the single epoch "warp" images.

PS1 Image Cutout Service

- The PS1 Image Cutout Service provides color images, JPEG and FITS cutouts, an interactive image display, and full FITS image retrievals.
- The image cutout service is programmatically accessible to allow downloading either cutouts or full images via a script. There is a simple Python Jupyter notebook that shows how to download images.

MAST Portal

- PanSTARRS images are also accessible via the MAST Portal, which provides both a flexible interactive interface (which integrates the PS1 images with the other MAST holdings include Hubble data).
- The Portal also has a powerful API that provides programmatic access to MAST data holdings.

Moving Target Search

A Python Jupyter notebook that demonstrates how to search for both images and catalog sources that match solar system moving targets is available on the How to search for moving targets in PS1 images and catalogs page.

```
# show positions at the times of PS1 observations
jpl_mjd = Time(jpltab['datetime_jd'],format='jd').mjd
ra_interp = np.interp(cadc_tab['MJD'], jpl_mjd, ratab)
dec_interp = np.interp(cadc_tab['MJD'], jpl_mjd, jpltab['DEC'])
plt.plot(ra_interp, dec_interp, 'o', label=f"{len(cadc_tab)} PS1 observations")

plt.xlabel("Unwrapped RA [deg]")
plt.ylabel("Dec [deg]")
plt.legend(loc="best")
plt.title(f"{fullname} from {emin.isot[:10]} to {emax.isot[:10]}")
plt.tight_layout()
if saveplots:
    plt.savefig("sky_path.png", facecolor="white")
```

```
Searching for rickwhite from 2009-06-03T00:00:00.000 to 2015-02-26T00:00:00.000 (2094 days)
Got positions from Horizons with 2095 rows for 174361 Rickwhite (2002 TV315)
CPU times: user 5.68 s, sys: 18 ms, total: 5.7 s
Wall time: 5.63 s
```

