



query astronomical databases and archives from python

Brigitta Sipőcz



@bsipocz



@AstroBrigi

# Astropy affiliated packages

These can be:

- Functionality under development, may become part of core  
(e.g. regions, reproject, specutils, wcsaxes)
- More specialized functionality  
(e.g. astroquery, gammapy, sncosmo)
- Packages with incompatible licenses

Adhere to Astropy **coding, testing, and docs guidelines**

**Use Astropy** wherever possible, share resources

We provide infrastructural help (package template, astropy-helpers, ci-helpers)

# astroquery vs astropy

- **astroquery** is an *astropy affiliated package*
- Vision: uniform pythonic interface for data consumers
  - de-facto gateway to data. Some providers do support modules already.
- **astroquery** will always remain independent from core astropy: it has to keep up with rapid changes on servers
- Latest release: 0.3.6 (0.3.7 very soon)
  - The dev version is almost always best to use; releases are based on new modules rather than upgrades to old

# astroquery

- Maintained by Adam Ginsburg & Brigitta Sipőcz

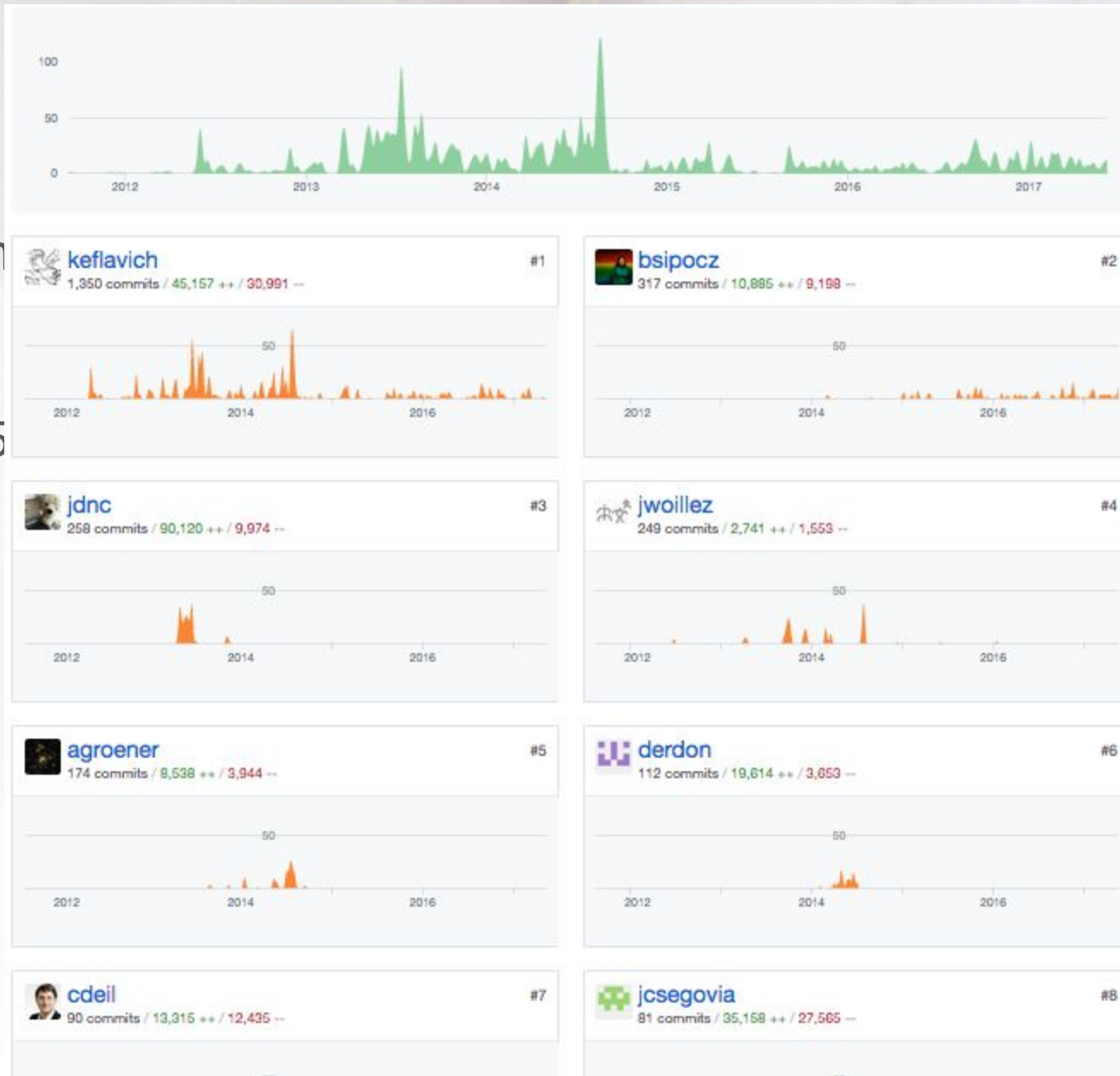
50+ contributors (including 2 GSoC students)

Most modules have been implemented independently

Referred 30+ times on ADS

# Astroq

- Main



# Astroq

- Main



# Modules

- **ESASky**: Chandra, XMM, Gaia, Herschel, Planck, ...
- **Gaia**: access to public tables, cone search, sql queries
- **MAST**: HST, Kepler, GALEX, SWIFT, and many others
- **ESO**: Instrument and Phase-3 survey queries
- **ALMA, Simbad, VizieR, SDSS, SkyView** and many more



# Modules

- SIMBAD Queries (`astroquery.simbad`)
- VizieR Queries (`astroquery.vizier`)
- ESASky Queries (`astroquery.esasky`)
- IRSA Dust Extinction Service Queries (`astroquery.irsa_dust`)
- NED Queries (`astroquery.ned`)
- Splatalogue Queries (`astroquery.splatalogue`)
- Vamdc Queries (`astroquery.vamdc`)
- IRSA Image Server program interface (IBE) Queries (`astroquery.ibe`)
- IRSA Queries (`astroquery.irsa`)
- UKIDSS Queries (`astroquery.ukidss`)
- MAGPIS Queries (`astroquery.magpis`)
- NRAO Queries (`astroquery.nrao`)
- Besancon Queries (`astroquery.besancon`)
- NIST Queries (`astroquery.nist`)
- NVAS Queries (`astroquery.nvas`)
- GAMA Queries (`astroquery.gama`)
- ESO Queries (`astroquery.eso`)
- xMatch Queries (`astroquery.xmatch`)
- Atomic Line List (`astroquery.atomic`)
- ALMA Queries (`astroquery.alma`)
- Skyview Queries (`astroquery.skyview`)
- NASA ADS Queries (`astroquery.nasa_ads`)
- HEASARC Queries (`astroquery.heasarc`)
- Gaia TAP+ (`astroquery.gaia`)
- VO Simple Cone Search (`astroquery.vo_conesearch`)
- MAST Queries (`astroquery.mast`)

These others are functional, but do not follow a common & consistent API:

- Fermi Queries (`astroquery.fermi`)
- SDSS Queries (`astroquery.sdss`)
- ALFALFA Queries (`astroquery.alfalfa`)
- Spitzer Heritage Archive (`astroquery.sha`)
- LAMDA Queries (`astroquery.lamda`)
- OGLE Queries (`astroquery.ogle`)
- Open Exoplanet Catalogue (`astroquery.open_exoplanet_catalogue`)
- CosmoSim Queries (`astroquery.cosmosim`)
- HITRAN Queries (`astroquery.hitran`)



# Usage

- Every remote service is different, but:
- Common API: most modules provide a **query\_object** and/or **query\_region** interface (it depends on what the remote service supports)
- Some services include data retrieval features
- Check the API documentation

# Usage

```
>>> from astroquery.mast import Observations
```

```
>>> Observations.query_object("HD 189733")
```

```
<Table length=916>
```

dataproduct_type	obs_collection	instrument_name	...	objID	distance
str10	str5	str13	...	str11	float64
cube	SWIFT	UVOT	...	15000375980	0.0
cube	SWIFT	UVOT	...	15000152918	0.0
cube	SWIFT	UVOT	...	15000152920	0.0
cube	SWIFT	UVOT	...	15000368570	0.0
cube	SWIFT	UVOT	...	15000152922	0.0
cube	SWIFT	UVOT	...	15000375986	0.0
...	...	...	...	...	...
image	HLA	ACS/SBC	...	2017623809	0.0
image	HLA	ACS/SBC	...	2017623810	0.0
image	HLA	ACS/SBC	...	2017623811	0.0
image	HLA	ACS/SBC	...	2017623812	0.0
image	HLA	ACS/SBC	...	2017623813	0.0
image	HLA	NICMOS/NIC3	...	2018016590	630.689611321
image	HLA	NICMOS/NIC3	...	2018016591	630.689611321

# Outputs

- Most queries return astropy Tables or list of astropy Tables
  - pprint and show\_in\_browser are good quick look tools

# Outputs

```
>>> from astroquery.esasky import ESASky
>>> esa = ESASky()
>>> from astropy.coordinates import SkyCoord
>>> HD189_coords = SkyCoord.from_name("HD 189733")
>>> results = esa.query_object_catalogs(HD189_coords)
>>> results
TableList with 6 tables:
'0:GAIA DR1 TGAS' with 10 column(s) and 1 row(s)
'1:TYCHO-2' with 7 column(s) and 1 row(s)
'2:XMM-OM' with 12 column(s) and 5 row(s)
'3:XMM-EPIC' with 6 column(s) and 17 row(s)
'4:GAIA DR1' with 7 column(s) and 1 row(s)
'5:HIPPARCOS-2' with 7 column(s) and 1 row(s)
>>> results['GAIA DR1']
<Table masked=True length=1>
      name          ra [1]      dec [1]  ... phot_g_mean_mag [1] phot_variable_
      object        float64    float64  ...          float64          object
-----
1827242811876888960 300.182122759 22.7098106295 ...          7.36159117116          NOT_AVAIL
```



# Examples

```
>>> from astroquery.gaia import Gaia
>>> Gaia.query_object(HD189_coords, radius=0.1*u.deg)
<Table masked=True length=2000>
      dist      solution_id      ...      ecl_lat
      float64      int64      ...      Angle[deg]
      float64      int64      ...      float64
-----
0.0010468984575019045 1635378410781933568 ... 42.175189082462374
0.0028771206906004925 1635378410781933568 ... 42.177988310771603
0.0029722713621037917 1635378410781933568 ... 42.178734031692073
0.0031701332304065037 1635378410781933568 ... 42.1735379193337903
0.0032730182765045098 1635378410781933568 ... 42.173982011363421
0.0036808051913854874 1635378410781933568 ... 42.179609777084494
0.0037659540796916736 1635378410781933568 ... 42.174637716131102
0.00393352730192182 1635378410781933568 ... 42.179246124278293
0.0041547040512449861 1635378410781933568 ... 42.180339910394032
      ...      ...      ...      ...
0.058013367963250434 1635378410781933568 ... 42.224304250037612
0.058015566683985303 1635378410781933568 ... 42.216412721627407
0.058065711060170729 1635378410781933568 ... 42.233357172705091
0.058080000710589046 1635378410781933568 ... 42.173231344872626
0.058109571819623611 1635378410781933568 ... 42.132997286271426
0.058112546881472278 1635378410781933568 ... 42.218960516644238
0.058124280157692332 1635378410781933568 ... 42.195878523623357
0.058124742678666359 1635378410781933568 ... 42.124055098059564
0.058163052235223049 1635378410781933568 ... 42.150872274389798
0.058164690957779576 1635378410781933568 ... 42.225204312039821
```

# Examples

```
>>> from astroquery.simbad import Simbad
>>> result_table = Simbad.query_object("m1")
>>> result_table.pprint(show_unit=True)
```

MAIN_ID	RA	DEC	RA_PREC	...	COO_QUAL	COO_WAVELENGTH	COO_BIBCODE
	"h:m:s"	"d:m:s"		...			
-----							
-							
M 1	05 34 31.94	+22 00 52.2	6	...	C	R	
2011A&A...533A..10L							



# Examples

```
>>> from astroquery.gaia import Gaia
>>>
>>> job = Gaia.launch_job("select top 100 \
>>> solution_id,ref_epoch,ra_dec_corr,astrometric_n_obs_al,matched_observations,duplicat
>>> from gaiadr1.gaia_source order by source_id")
>>>
>>> print(job)

Jobid: None
Phase: COMPLETED
Owner: None
Output file: sync_20170223111452.xml.gz
Results: None

>>> r = job.get_results()
>>> print(r['solution_id'])

solution_id
-----
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
1635378410781933568
...
1635378410781933568
1635378410781933568
1635378410781933568
```

## Use cases

- Download catalogues to make reproducible analysis scripts
- Download data from archives
- Combine image and catalogue services to create finder charts

# Future

- More modules
  - E.g. NOAO, LCO, VISTA are work in progress
- Cleaner documentation, better tests
- You can help!

# Contributing

- Bug and documentation fixes
- Creating new modules (there is a template):
  - Best case: use the public API
  - Worst case: Simple web scraping

# Getting help

- Python Users in Astronomy - Facebook group
- StackOverflow - use 'astroquery' or 'astropy' tag
- Gitter - <https://gitter.im/astropy/astroquery>

