



The NOAO Data Lab

Current Status and Future Vision

Stéphanie Juneau (NOAO)
on behalf of the DL team



National Optical Astronomy Observatory

Cerro Tololo Inter-American Observatory
Kitt Peak National Observatory
Community Science and Data Center



Current team:

Mike Fitzpatrick, Lead Developer

Matthew Graham, Scientist/Developer

Wendy Huang, Software Engineer

Stephanie Juneau, Data Scientist

David Nidever, Data Scientist

Robert Nikutta, Data Scientist

Pat Norris, Test Engineer

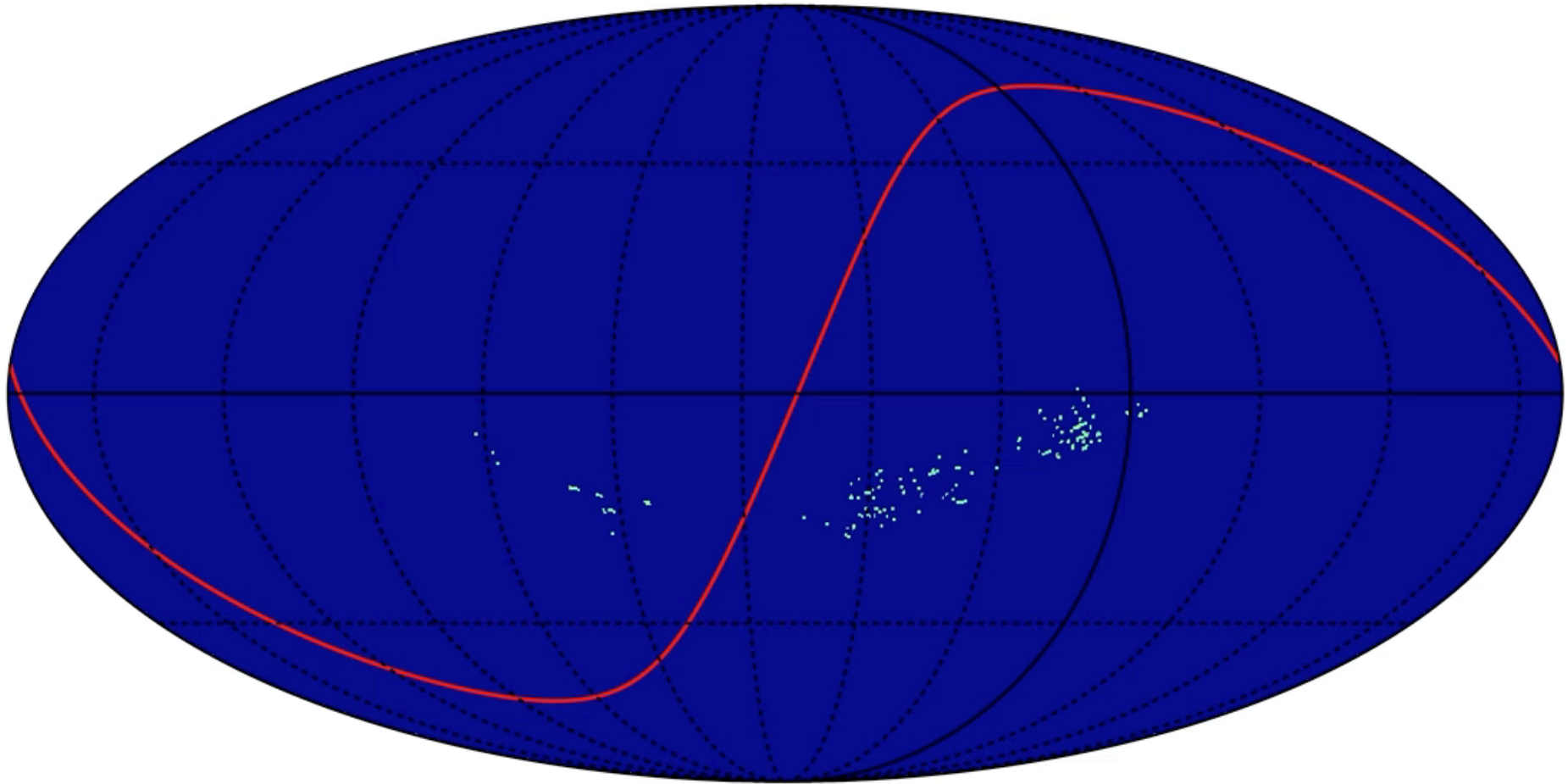
Knut Olsen, Project Scientist

Steve Ridgway, Scientist

Adam Scott, Database Architect

Pete Wargo, System Administrator

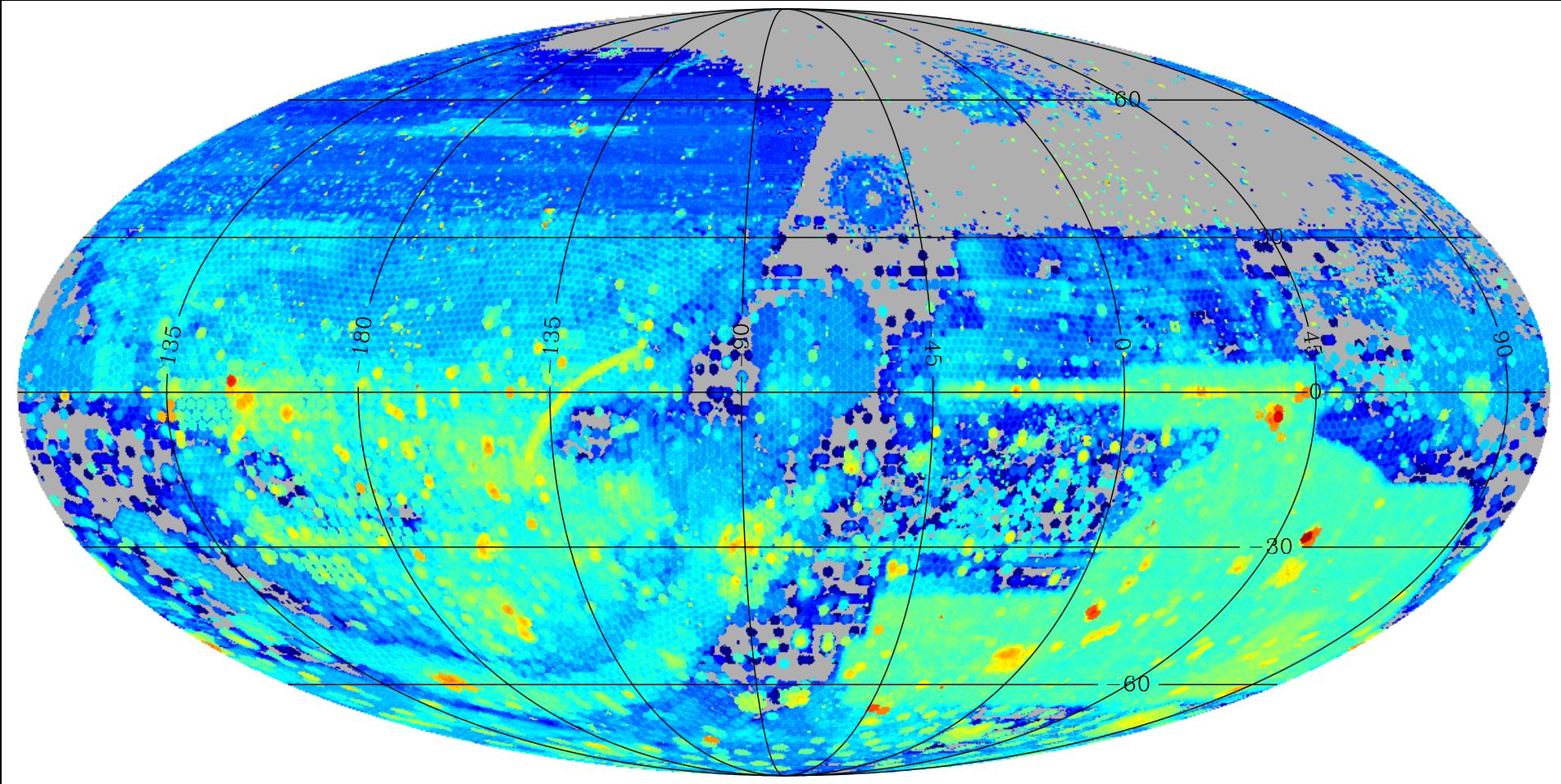
August 11, 2004



National Optical Astronomy Observatory
Cerro Tololo Inter-American Observatory
Kitt Peak National Observatory
Community Science and Data Center



DECam and Mosaic data in June 2017



Goal

Efficient exploration and analysis of large datasets with an emphasis on NOAO wide-field 4-m telescopes

Approach

- High-value catalogs from NOAO and external sources (e.g. SDSS, GAIA) and NOAO-based images linked to catalog objects
- Data discovery
- Developing intuition through interaction with selected catalog and image set of known objects
- Automation of analysis to aid discovery of unknown objects

Large Catalogs – TB-scale databases

Pixel Data – images & spectra in NOAO Science Archive

Virtual Storage – 1 TB per user to minimize data transfer

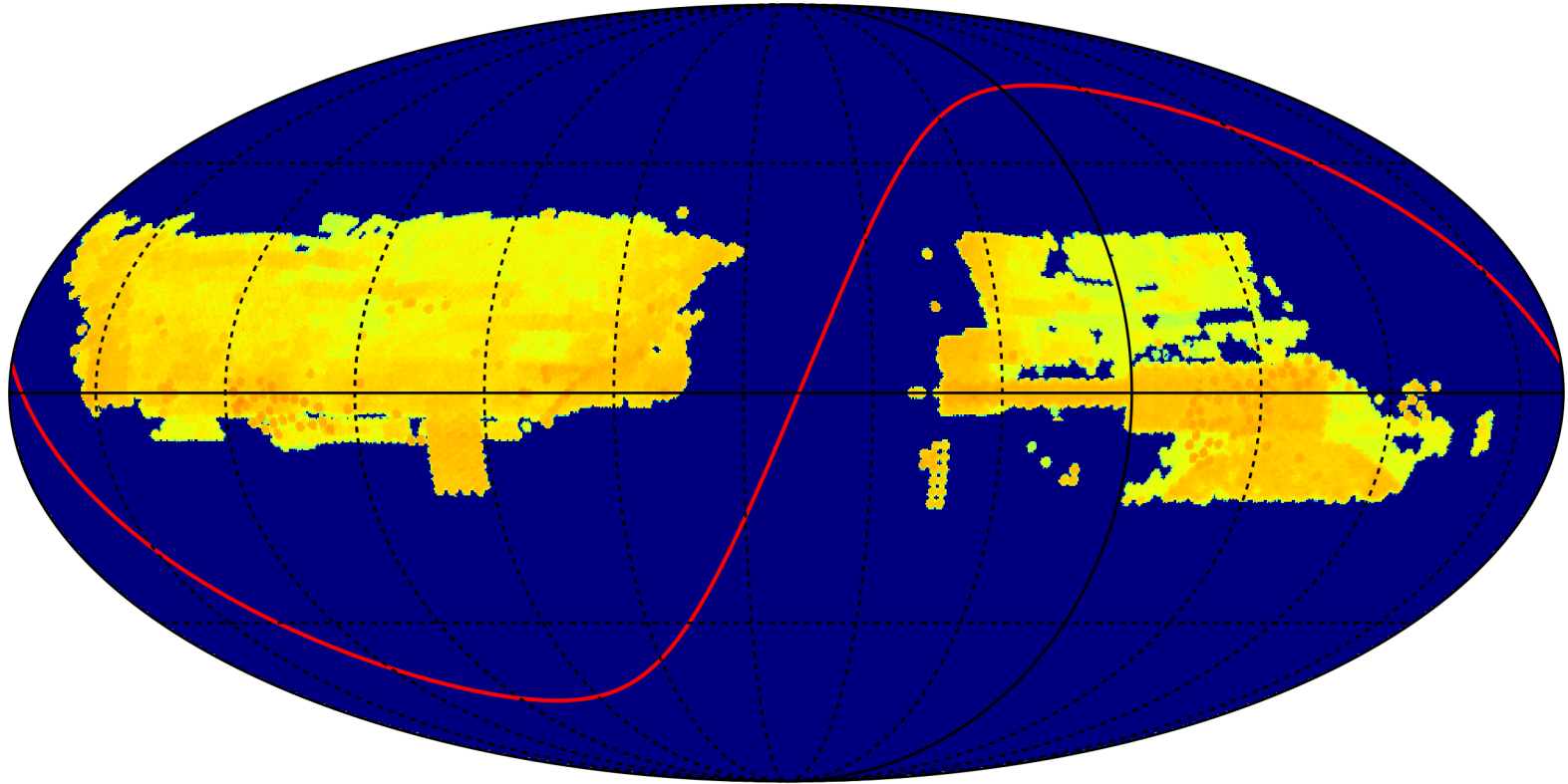
Visualization – data exploration

Compute Processing – workflows run close to the data

++ Access to published datasets, data publication,
exportable workflows, distributable software

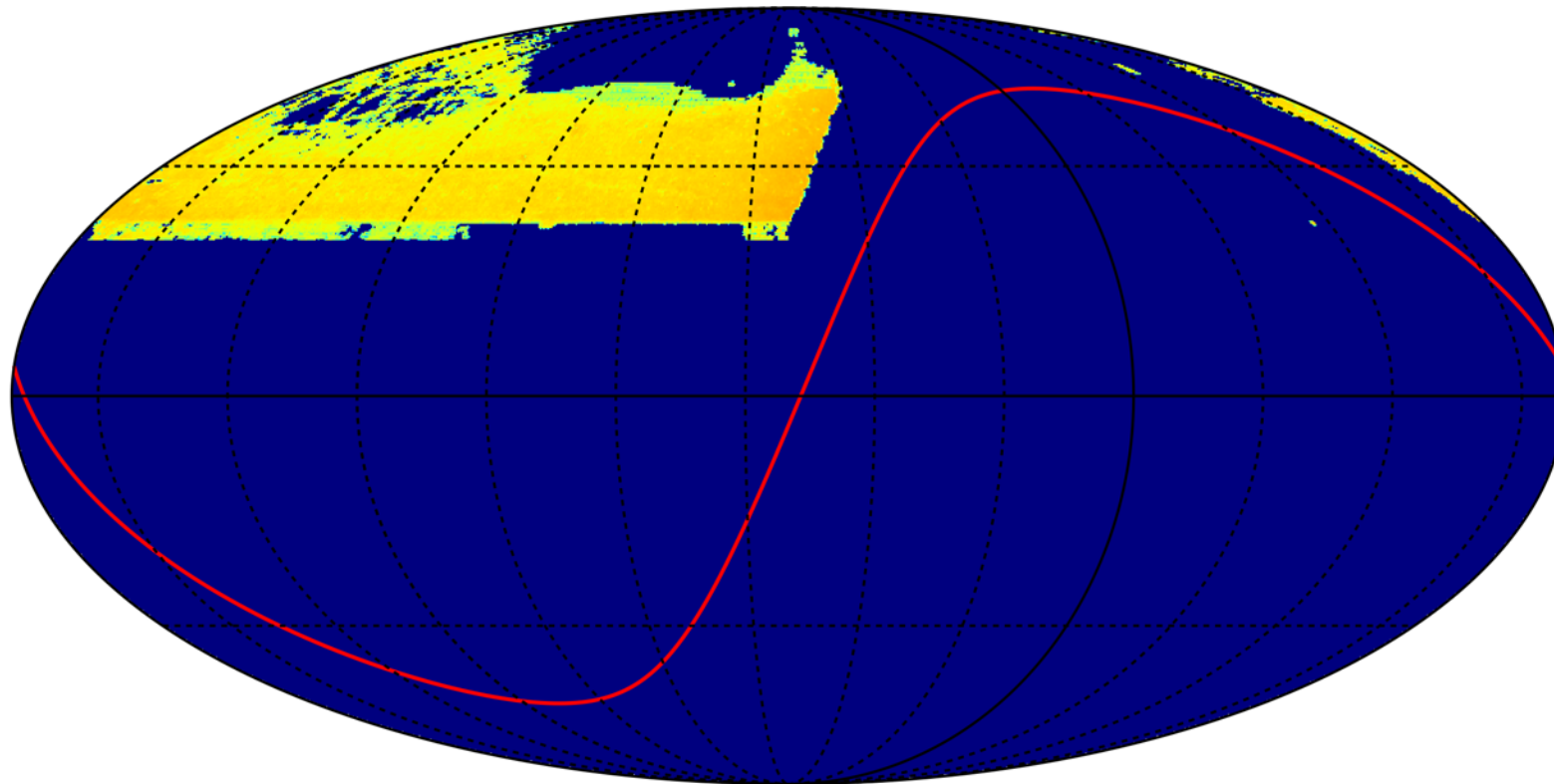
Data Lab 1.0 released in June 2017 (AAS)

DECaLS DR3

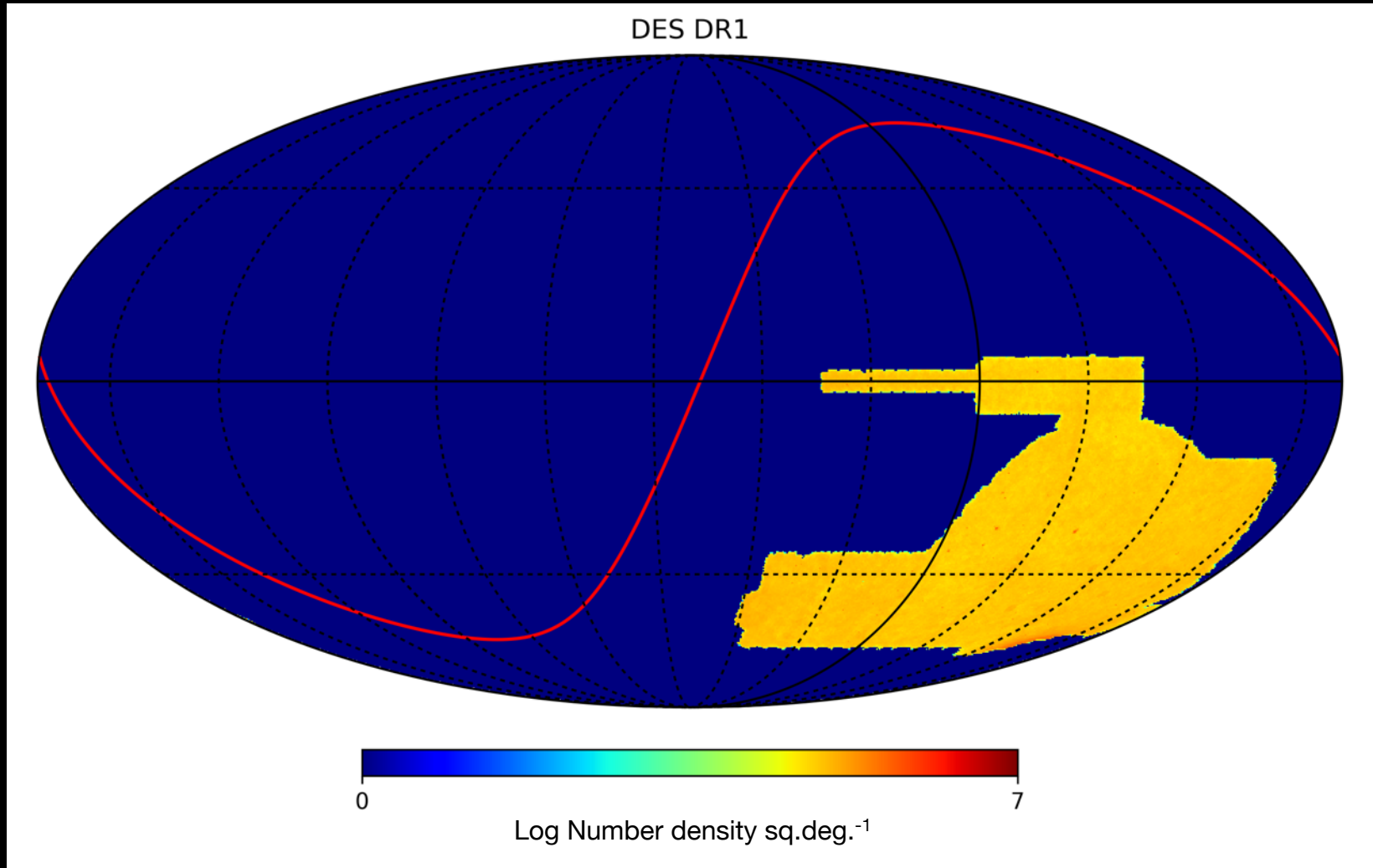


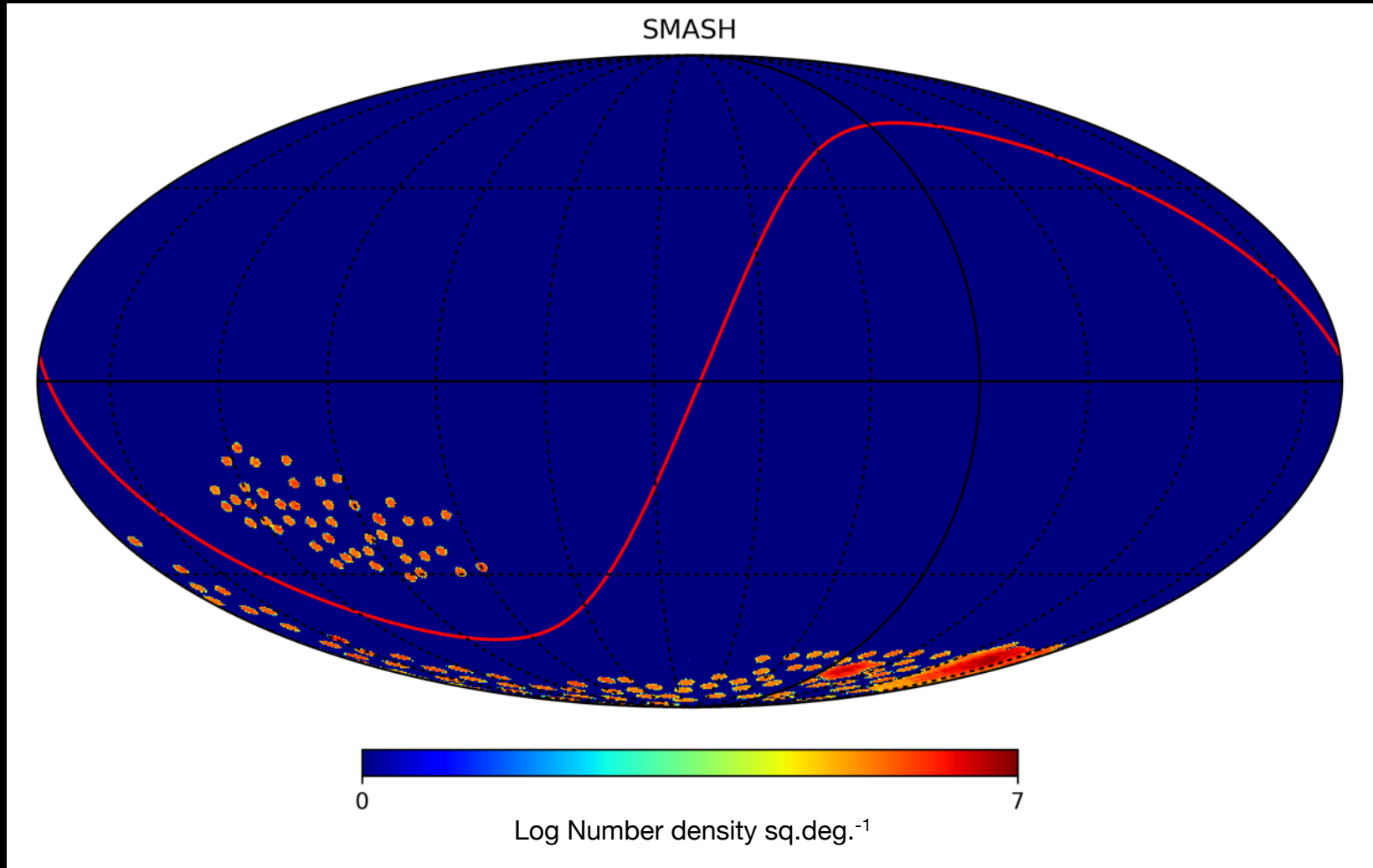
Log Number density sq.deg.^{-1}

DECaLS DR4

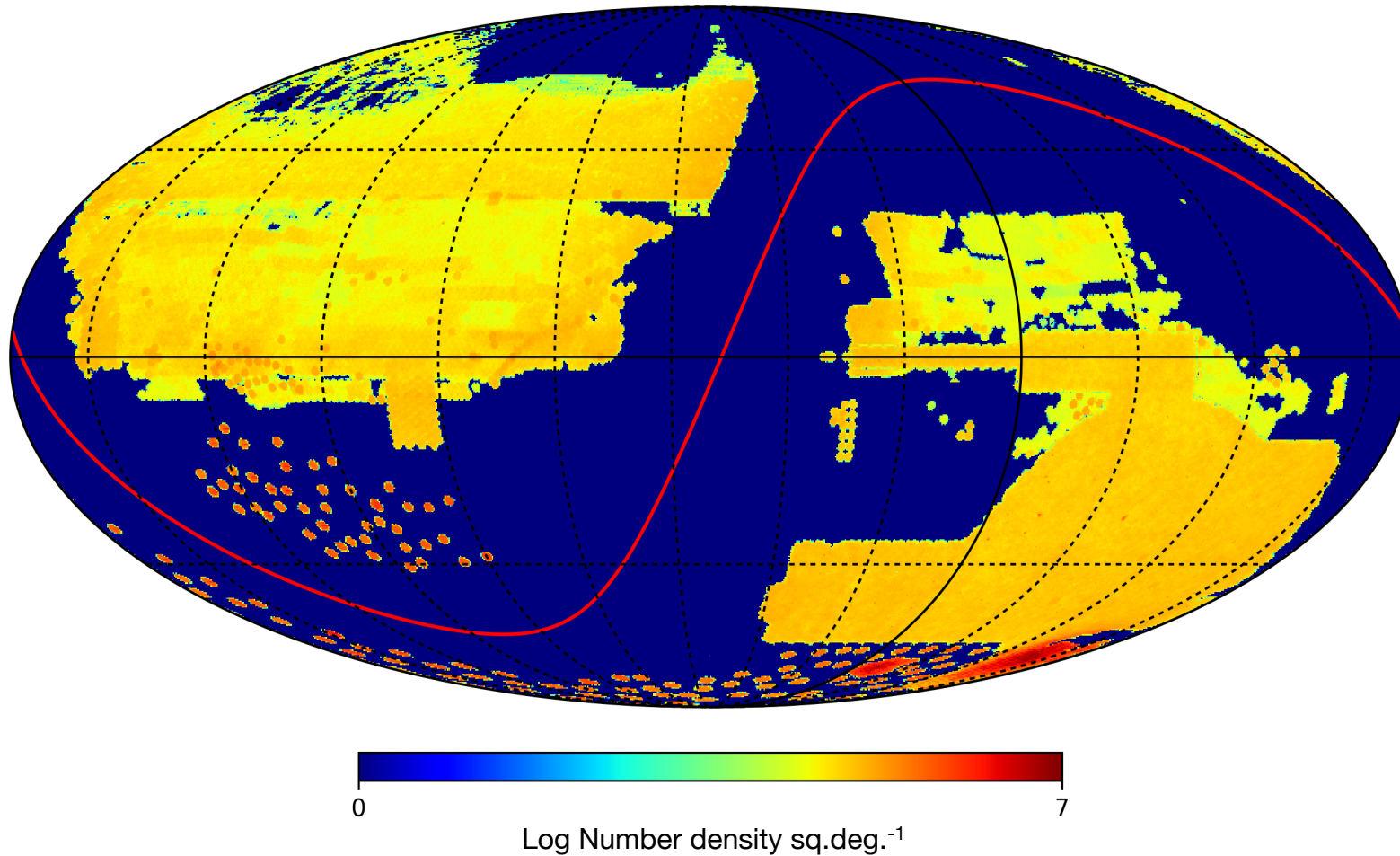


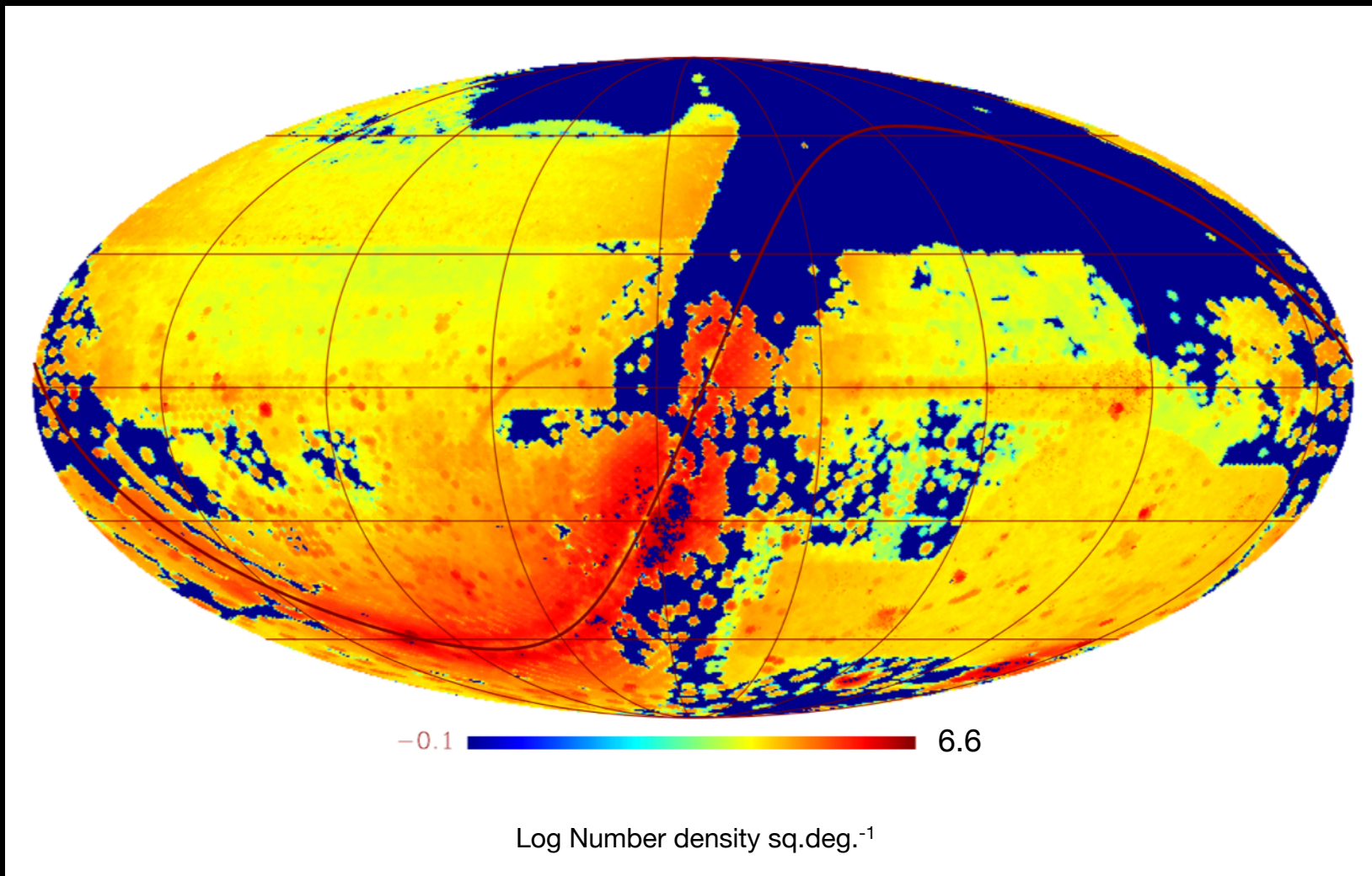
Log Number density sq.deg.⁻¹





Data Lab catalogs







Data Volume and Complexity

~500 TB (February 2017) of on-target imaging data ($t_{\text{exp}} > 30\text{s}$) currently from:

- Dark Energy Survey

- Legacy Surveys for DESI Targeting

- Community DECam and Mosaic programs and surveys

Hundreds of TB more coming

Total holdings at PB scale

Large catalogs, e.g.:

- Dark Energy Survey – 7 TB

- Complete DESI Targeting Survey – ~5 TB

- Community programs and surveys – up to several TB each





Data Volume and Complexity

NOAO Facilities Featured Surveys:

DESI imaging Legacy Survey (LS): ~860 million objects in DR4+5 (*now*)

SMASH: ~100 million objects in DR1 (*now*)

DES: ~400 million objects in DR1 (*AAS 01/2018*)

DECaPS: ~2 billion objects (*AAS 01/2018*)

NOAO All-Sky Source Catalog (NSC): ~2.5 billion objects (*AAS 01/2018*)

Additional Surveys:

select tables from **SDSS/BOSS** DR13 & DR14, **GAIA** DR1, DES SVA1, the **Allen NEO** catalog, and **USNO**-A2/B, *skinny* Pan-STARRS DR1, etc.



Using the NOAO Data Lab

datalab.noao.edu

NOAO Data Lab

[Login](#) | [Sign up](#)



[About](#)

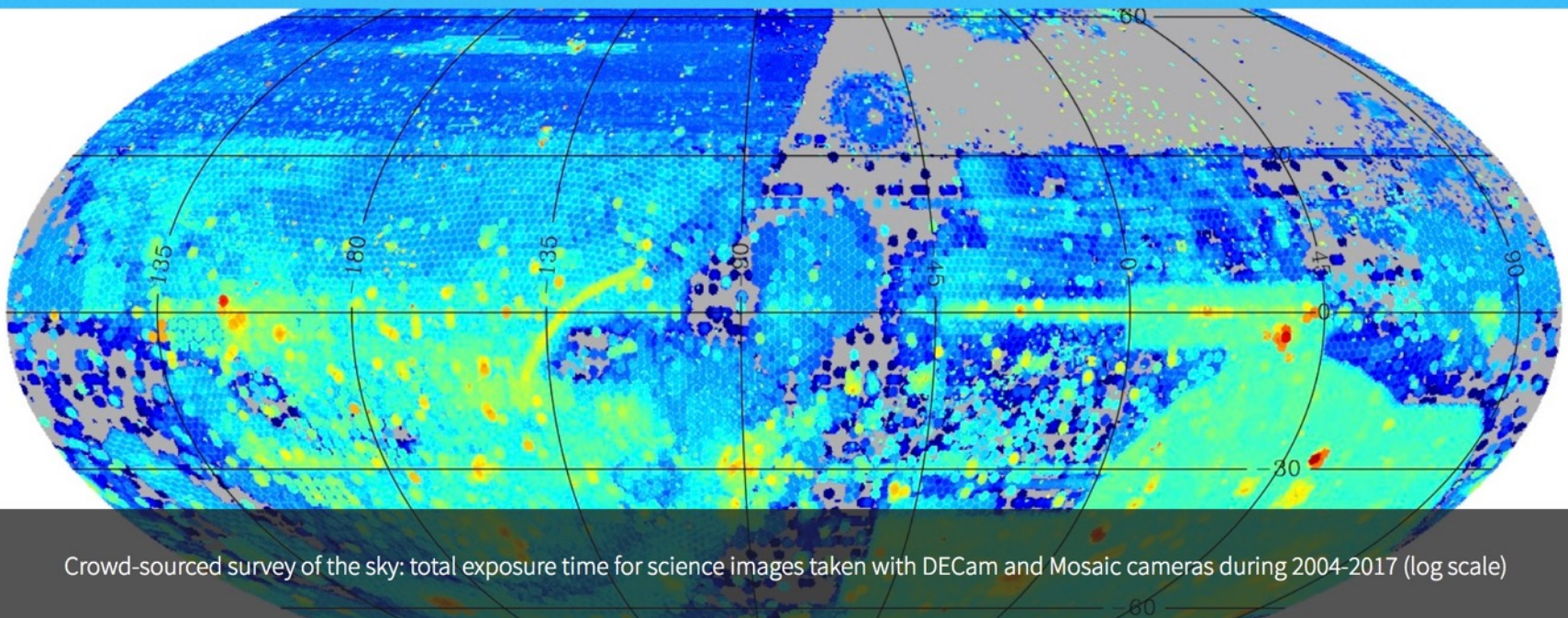
[Quick Start](#)

[Tools](#)

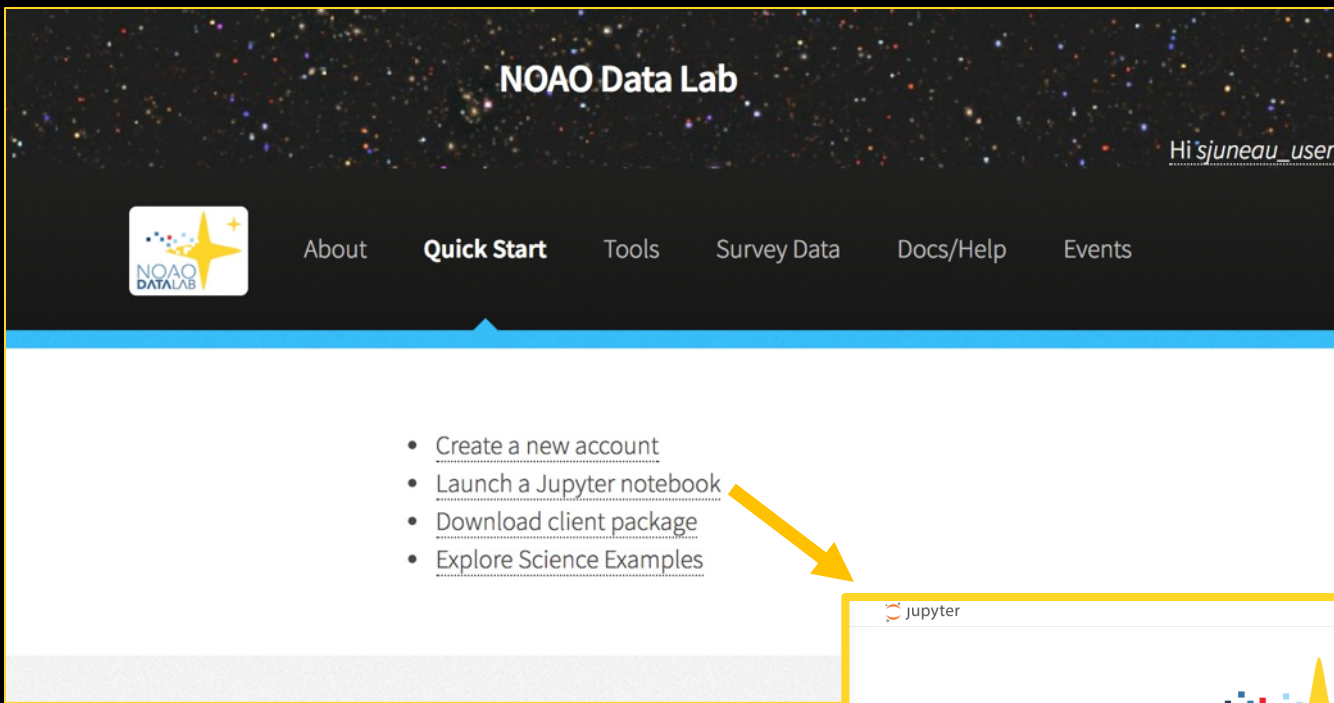
[Survey Data](#)

[Docs/Help](#)

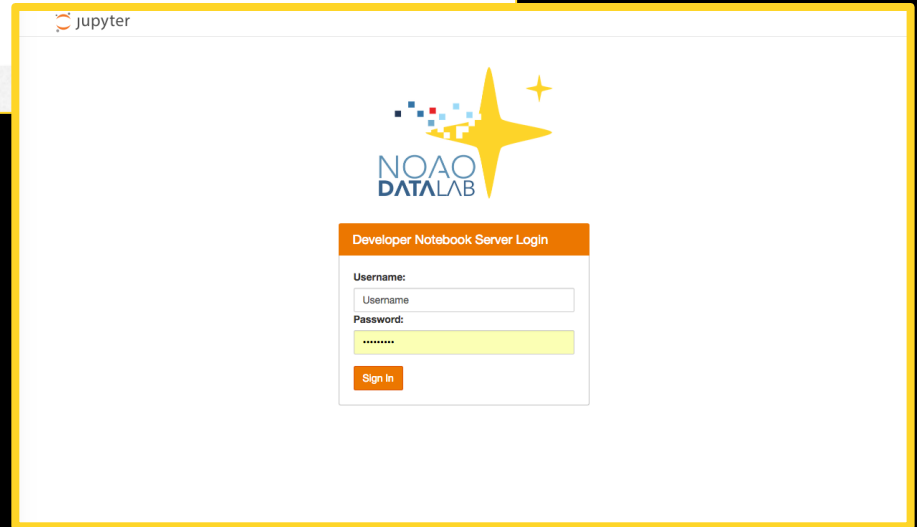
[Events](#)



Crowd-sourced survey of the sky: total exposure time for science images taken with DECam and Mosaic cameras during 2004-2017 (log scale)



1) User logs in to Data Lab



2) Launch Jupyter Notebook server

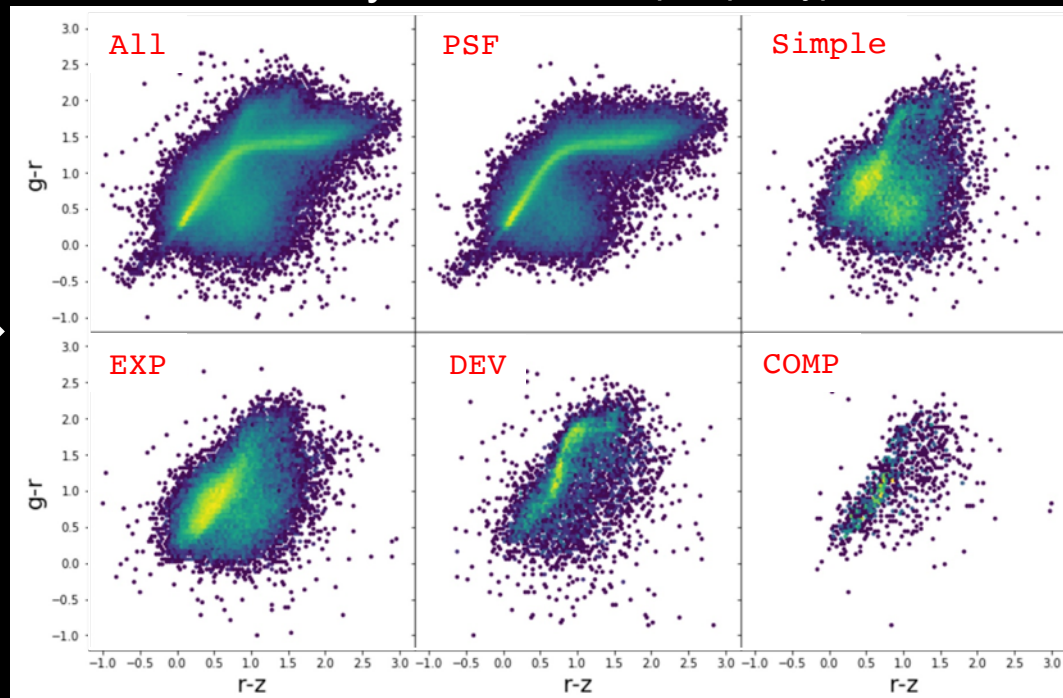


Query to database: *magnitudes and object shape (type)*

```
query = ""  
SELECT dered_mag_g as gmag, dered_mag_r as rmag,  
       dered_mag_z as zmag,  
       dered_mag_w1 as w1mag, dered_mag_w2 as w2mag, type,  
       snr_g, snr_r, snr_z, ra, dec  
FROM ls_dr3.tractor_primary  
WHERE (snr_g>3 and snr_r>3 and snr_z>3)  
LIMIT 200000""  
  
# dered_mag_g,r,z = AB mag in DECam g,r,z bands corrected  
#                  for Galactic reddening  
# dered_mag_w1,w2 = AB magnitudes in WISE bands W1 & W2  
#                  corrected for Galactic reddening  
# type            = object type (PSF, SIMP, EXP, DEV, COMP)  
# snr_g,r,z       = signal-to-noise ratios (S/N) in g,r,z bands  
# ra,dec          = celestial coordinates  
#  
# WHERE: requirement that S/N>3 in each DECaLS band  
# LIMIT: returns 200,000 rows that satisfy the query
```

Example Workflow

Analysis: *color-color plot per type*



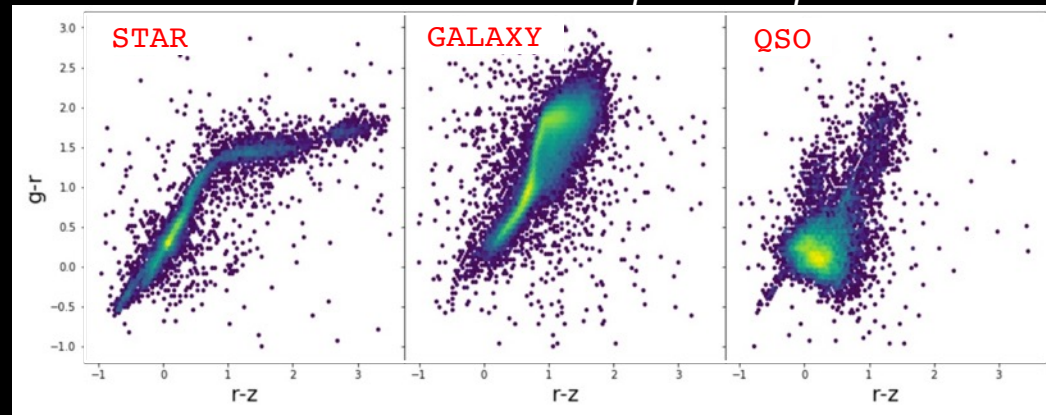
Machine-Learning:

Confusion matrix (spectroscopic training set)

GALAXY	0.982	0.008	0.001
QSO	0.087	0.878	0.035
STAR	0.018	0.012	0.97
	GALAXY	QSO	STAR

Joint query:

cross-match with SDSS spectroscopic class



Roles for the Data Lab with DESI

Host DESI imaging Legacy Surveys (DECaLS, BASS/MzLS)

→ Databases (ls_dr3, ls_dr4, ls_dr5)

→ Images in NOAO Science Archive (raw + processed)

now!

Host DESI targets

→ Database for final, public set of targets

Host DESI redshifts

→ Database for public releases of redshift catalogs

→ Tools for spectra visualization/analysis

future

Create example Notebooks & workflows

Users can work with all data products



Roles for the Data Lab in the LSST Era

Doing LSST pathfinder science today, e.g.

- Explore ways to perform star/galaxy separation, including machine learning techniques
- Automated search for dwarf galaxies & streams in all-sky data
- Identify variable sources in all-sky photometry, retrieving time series of variable stars, QSOs, etc.
- Cross-match large catalogs, e.g. LSST Stack-reduced crowded field vs. DAOPHOT/DoPHOT as part of Q.A.
- Upload table of moving objects, retrieve image cutouts from Archive, stack and analyze





Roles for the Data Lab in the LSST Era

As a complementary tool for LSST science

- Unique datasets for comparison with LSST + users can publish datasets
- Legacy code close to Data Lab data + users can upload personal data
- Different technology on the backend thus a diversity of solutions
- User friendly platform for large data analysis (similar to LSST DAC; with support for greater diversity of datasets)
- “Skinny” version of LSST main catalog table* (subset of columns, but all rows) to perform cross-match + preliminary analysis [*needs feasibility tests]



Easy access to data for entire astronomy community

→ Databases: Tables, Images, Spectra

User-friendly yet powerful analysis tools

→ Quick start analysis

→ Automated & sophisticated workflows

Data Publication Service

→ User contributed datasets

Interactive interface with advanced visualization

→ connected exploration & analysis, drag-and-drop workflow

Data Lab software package

→ widely distributed, user-contributed developments



Data Lab Future Visions (cont'd)

Machine-Learning algorithms

→ Running in background on all the datasets

Education & Public Outreach

→ Astronomy/Data Science activities for classrooms

→ Art/Science Collaborations

Citizen-science projects



National Optical Astronomy Observatory
Cerro Tololo Inter-American Observatory
Kitt Peak National Observatory
Community Science and Data Center



Combining increasingly larger datasets including multi-wavelength cross-analysis & combining with simulations/simulated data

Interface between different Data Centers

- different technologies
- different data models and/or formats
- cannot always have co-located data (e.g., full LSST)

Balancing public (astro/cosmo community) and private (survey team) needs for data access and analysis tools

Try it out and get in touch!

Web: datalab.noao.edu

Email: datalab@noao.edu

GitHub: <https://github.com/noao-datalab>

Twitter: @NOAODataLab





Extra Slides: Data Lab Info & Tutorial



National Optical Astronomy Observatory
Cerro Tololo Inter-American Observatory
Kitt Peak National Observatory
Community Science and Data Center



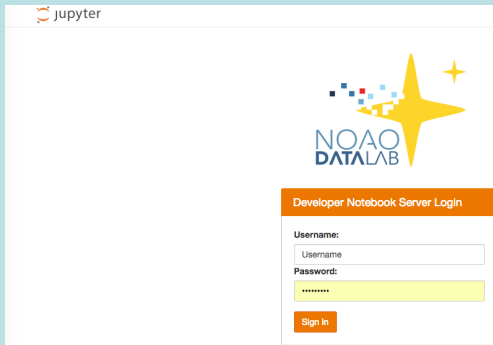
Summary of Current Functions

Function	Method
Sky exploration	Image discovery tool Catalog overlay tool Catalog visualization tool (prototype)
Authentication	Web interface datalab command Python authClient, DL interface
Catalog query	Web interface datalab command line (CLI) Python queryClient, DL interface TOPCAT
Image query	Simple Image Access (SIA) service
Query result storage	myDB Virtual storage space
File transfer	datalab command and Virtual storage space
Analysis	Jupyter notebook server

Example: Detecting a faint dwarf galaxy

User logs in to Data Lab

1



Launches Jupyter Notebook

Queries database for blue stellar objects in SMASH DR1 Field

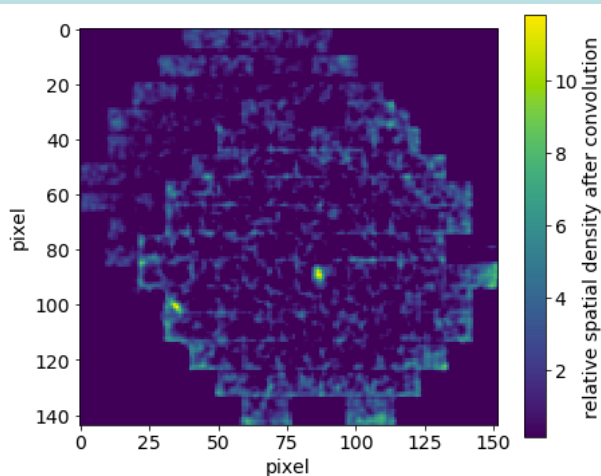
2

```
field = 169 # SMASH field number to query
depth = 1 # depth (=no short exposures please)

# Create the query string; SQL keyword capitalized for clarity
query_template = \
"""SELECT ra,dec,gmag,rmag,imag FROM smash_dr1.object
WHERE fieldid = '%d' AND
depthflag > %d AND
abs(sharp) < 0.5 AND
gmag BETWEEN 9 AND 25 AND
(gmag-rmag) BETWEEN -0.4 AND 0.4"""

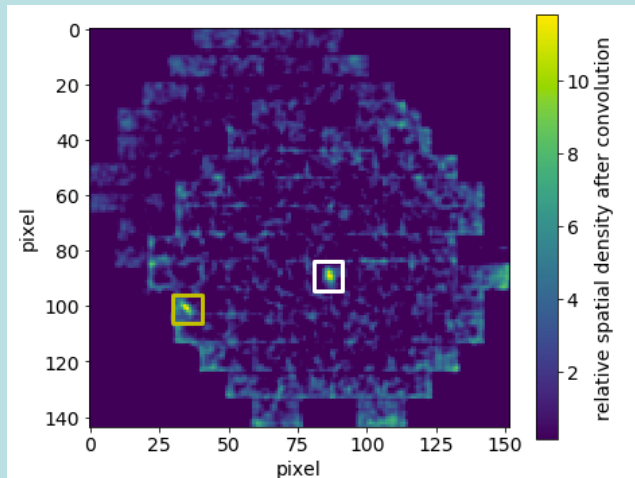
query = query_template % (field, depth)
```

Applies filter to spatial distribution



3

Runs automatic peak detection



4

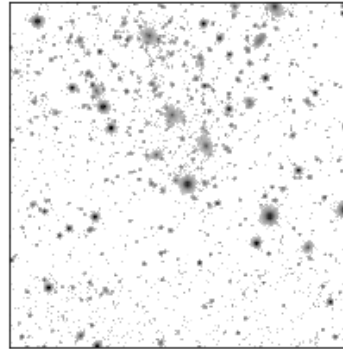
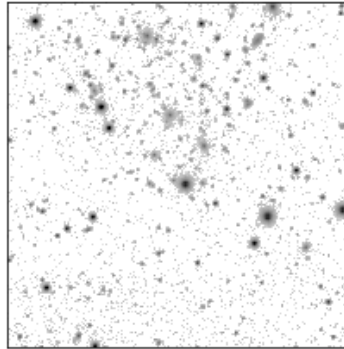
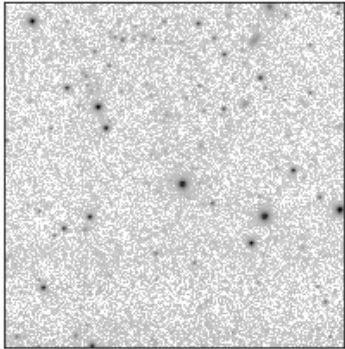
Queries peak locations for image cutouts

5

g band

r band

i band



Stores all results in virtual storage...

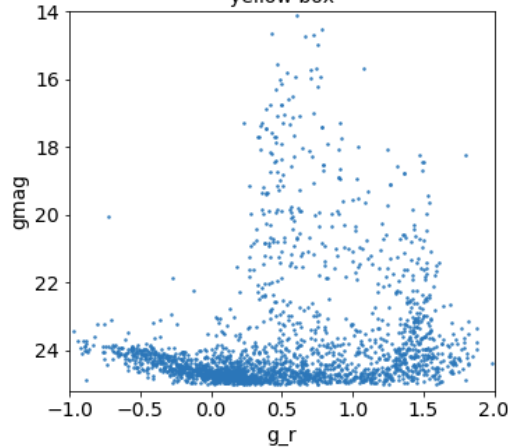
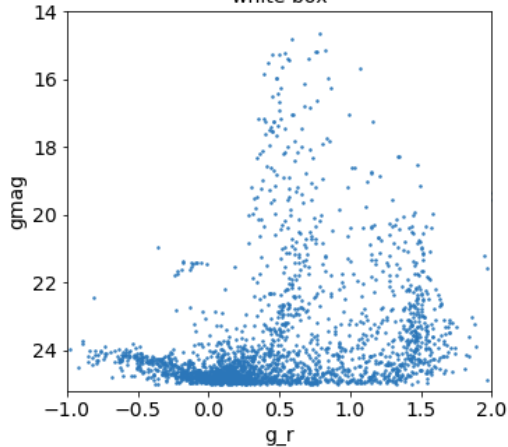
7

...and repeat!

Queries peak locations for full photometry

white box

yellow box

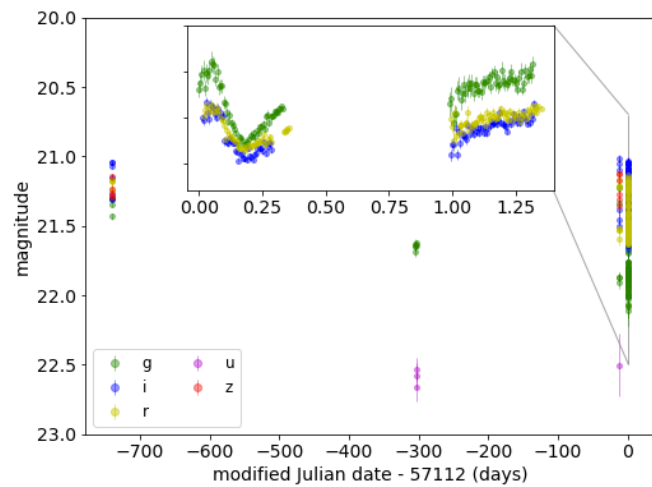


6

Example: Detecting variables

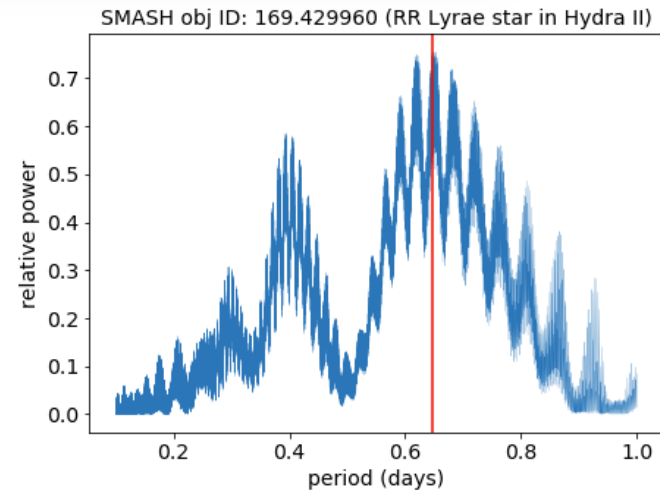
Retrieve light curve of candidate Hydra II variable

1



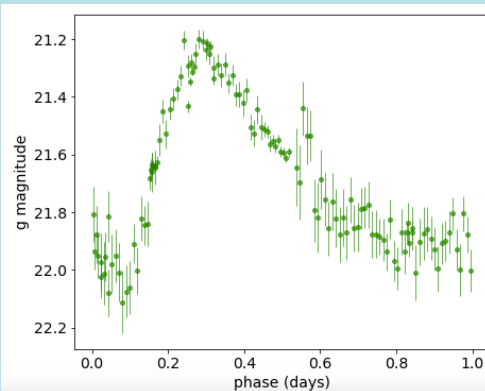
Apply Lomb-Scargle

2



Fold light curve

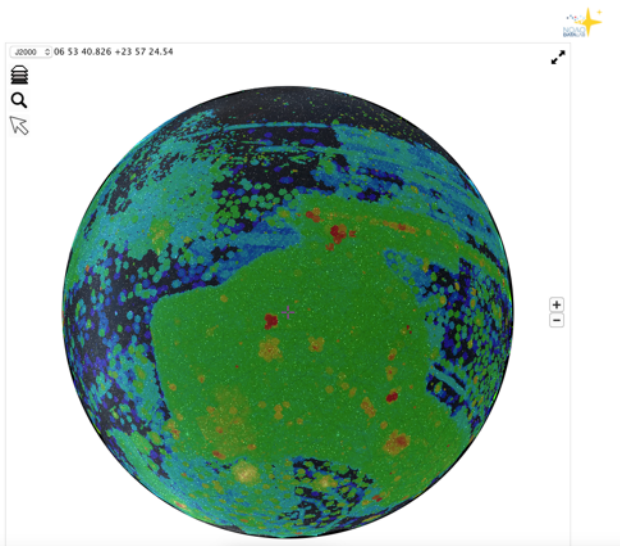
3



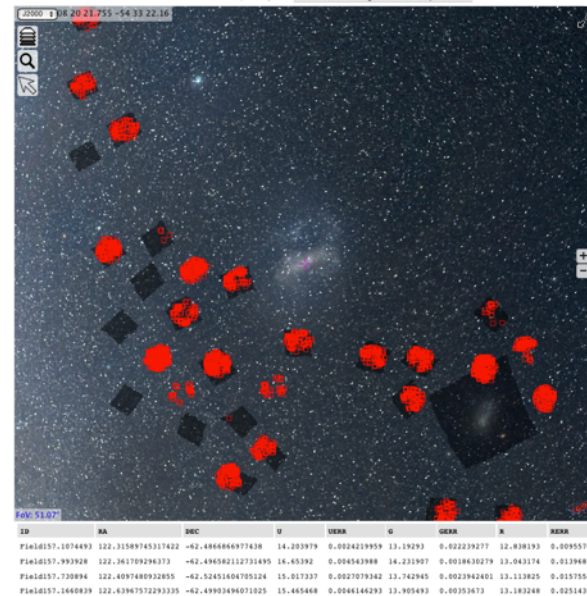
Identify more variables through statistical techniques!

4

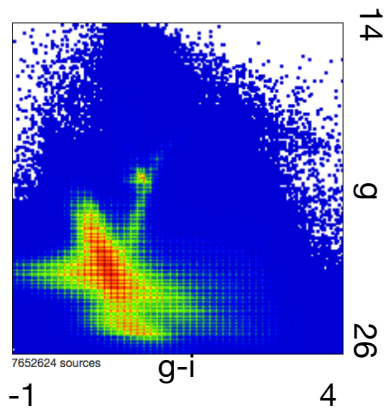
Images



Catalogs

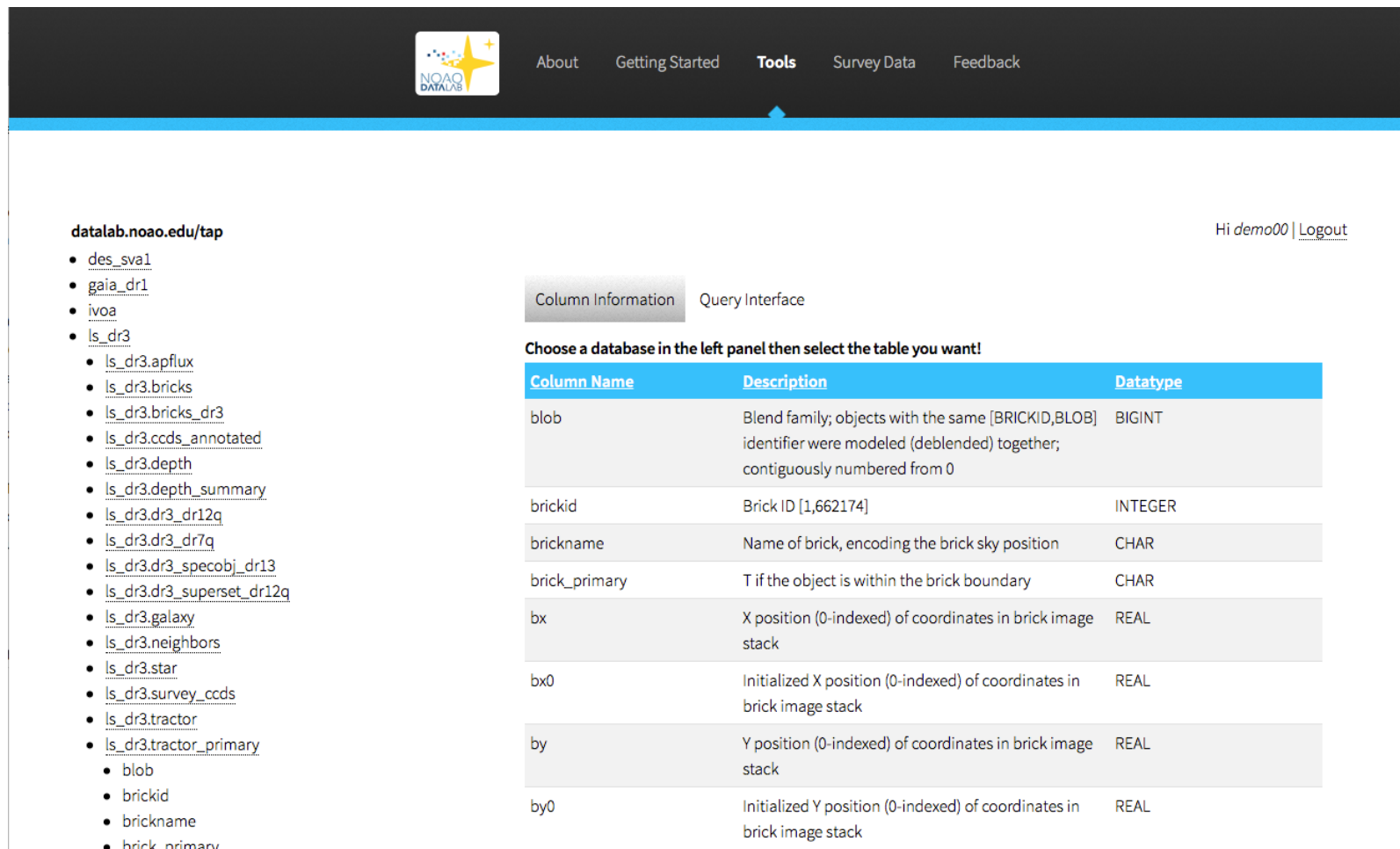


Catalog visualization (prototype)



Querying the catalogs

- Through the Data Lab website:



The screenshot shows the NOAO DataLab website interface. At the top, there is a navigation bar with the NOAO DataLab logo and links for "About", "Getting Started", "Tools", "Survey Data", and "Feedback". The "Tools" link is highlighted with a blue arrow. Below the navigation bar, the page content is divided into two main sections. On the left, there is a list of databases under the heading "datalab.noao.edu/tap". The list includes "des_sva1", "gaia_dr1", "ivoa", and "ls_dr3". Under "ls_dr3", there is a sub-list of tables: "ls_dr3.apflux", "ls_dr3.bricks", "ls_dr3.bricks_dr3", "ls_dr3.ccds_annotated", "ls_dr3.depth", "ls_dr3.depth_summary", "ls_dr3.dr3_dr12q", "ls_dr3.dr3_dr7q", "ls_dr3.dr3_specobj_dr13", "ls_dr3.dr3_superset_dr12q", "ls_dr3.galaxy", "ls_dr3.neighbors", "ls_dr3.star", "ls_dr3.survey_ccds", "ls_dr3.tractor", and "ls_dr3.tractor_primary". Under "ls_dr3.tractor_primary", there is a further sub-list: "blob", "brickid", "brickname", and "brick_primary". On the right, there is a "Query Interface" section. It has two tabs: "Column Information" (selected) and "Query Interface". Below the tabs, there is a prompt: "Choose a database in the left panel then select the table you want!". Below this prompt is a table with three columns: "Column Name", "Description", and "Datatype". The table lists the following columns: "blob" (BIGINT), "brickid" (INTEGER), "brickname" (CHAR), "brick_primary" (CHAR), "bx" (REAL), "bx0" (REAL), "by" (REAL), and "by0" (REAL). The "Description" column provides details for each column, such as "Blend family; objects with the same [BRICKID,BLOB] identifier were modeled (deblended) together; contiguously numbered from 0" for "blob".

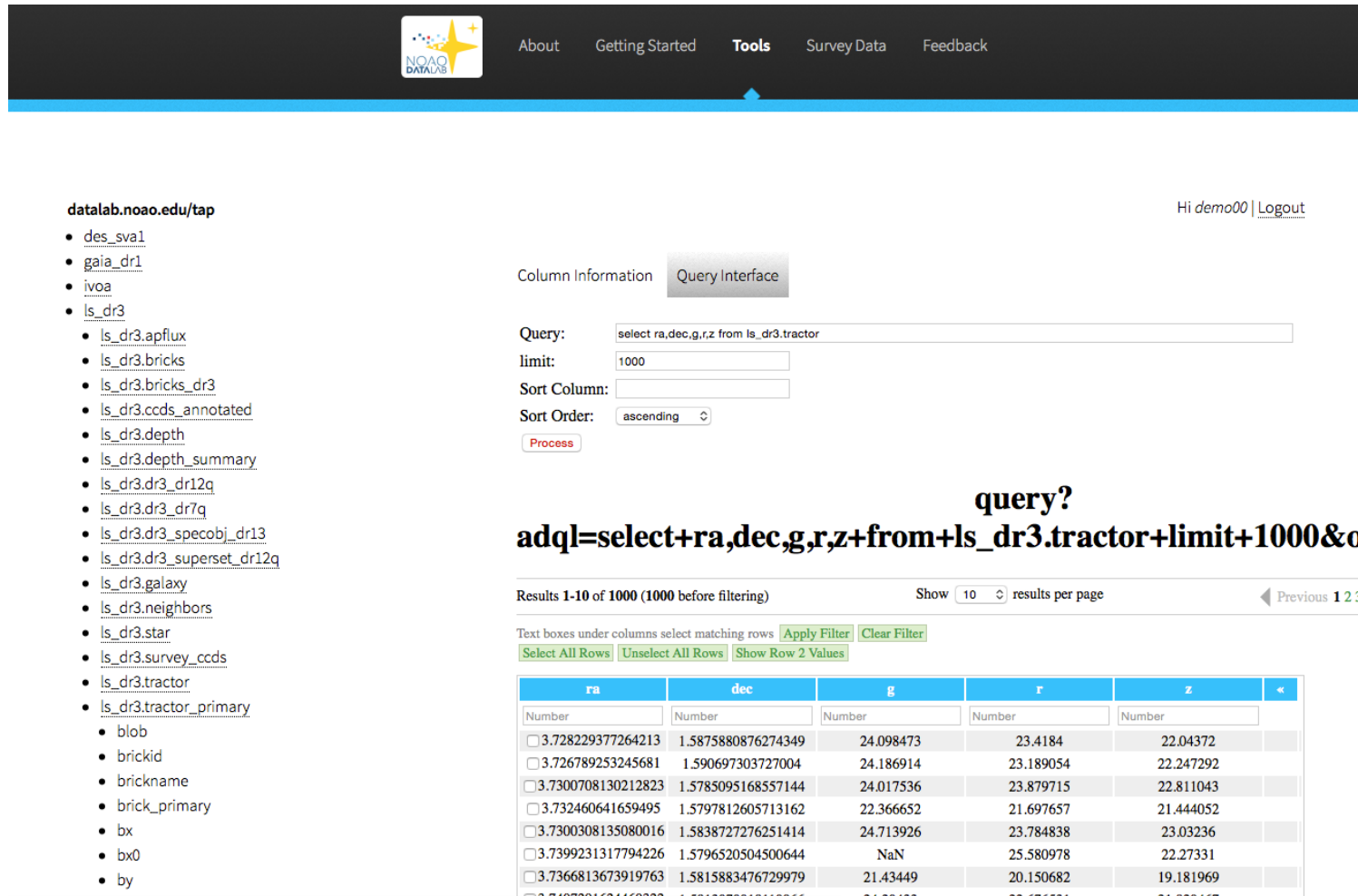
Hi demo00 | [Logout](#)

Column Information Query Interface

Choose a database in the left panel then select the table you want!

Column Name	Description	Datatype
blob	Blend family; objects with the same [BRICKID,BLOB] identifier were modeled (deblended) together; contiguously numbered from 0	BIGINT
brickid	Brick ID [1,662174]	INTEGER
brickname	Name of brick, encoding the brick sky position	CHAR
brick_primary	T if the object is within the brick boundary	CHAR
bx	X position (0-indexed) of coordinates in brick image stack	REAL
bx0	Initialized X position (0-indexed) of coordinates in brick image stack	REAL
by	Y position (0-indexed) of coordinates in brick image stack	REAL
by0	Initialized Y position (0-indexed) of coordinates in brick image stack	REAL

- Through the Data Lab website:



The screenshot shows the NOAO Data Lab website interface. On the left, there is a navigation menu with links to various data catalogs, including `ls_dr3` and its sub-catalogs like `ls_dr3.apflux`, `ls_dr3.bricks`, and `ls_dr3.tractor`. The main content area features a "Query Interface" tab. The query entered is `select ra,dec,g,r,z from ls_dr3.tractor`. The limit is set to 1000, and the sort order is ascending. A "Process" button is visible below the query fields. Below the query interface, there is a section titled "query?" with the text `adql=select+ra,dec,g,r,z+from+ls_dr3.tractor+limit+1000&o`. The results section shows "Results 1-10 of 1000 (1000 before filtering)" and a table with columns for `ra`, `dec`, `g`, `r`, and `z`. The table contains 10 rows of data, each with a checkbox for selection. The table is as follows:

ra	dec	g	r	z
<input type="checkbox"/> 3.728229377264213	1.5875880876274349	24.098473	23.4184	22.04372
<input type="checkbox"/> 3.726789253245681	1.590697303727004	24.186914	23.189054	22.247292
<input type="checkbox"/> 3.7300708130212823	1.5785095168557144	24.017536	23.879715	22.811043
<input type="checkbox"/> 3.732460641659495	1.5797812605713162	22.366652	21.697657	21.444052
<input type="checkbox"/> 3.7300308135080016	1.5838727276251414	24.713926	23.784838	23.03236
<input type="checkbox"/> 3.7399231317794226	1.5796520504500644	NaN	25.580978	22.27331
<input type="checkbox"/> 3.7366813673919763	1.5815883476729979	21.43449	20.150682	19.181969



Querying the catalogs

- Through the datalab command:

```
[kolsen@gp02 ~]$ datalab login user=demo00 password=  
Welcome to the Data Lab, demo00  
[kolsen@gp02 ~]$ datalab query sql="select * from usno.a2 limit 10"  
id,raj2000_,dej2000_,actflag,mflag,bmag,rmag,epoch,raj2000,dej2000  
0150-00069690,00:14:47.196,-68:49:48.92, .,19.6,17.9,1981.81,3.696648,-68.830256  
0150-00070481,00:14:54.972,-68:49:58.22, .,19.8,18,1981.81,3.72905,-68.832839  
0150-00069562,00:14:45.900,-68:49:37.66, .,18,17.8,1981.81,3.69125,-68.827128  
0150-00069750,00:14:47.844,-68:49:29.41, .,19.4,18,1981.81,3.699348,-68.824837  
0150-00070904,00:14:59.041,-68:49:25.26, .,20.2,18,1981.81,3.746003,-68.823684  
0150-00072260,00:15:12.458,-68:54:06.12, .,18.9,17.1,1981.81,3.801909,-68.9017  
0150-00072812,00:15:17.694,-68:54:09.03, .,16.4,15.2,1981.81,3.823725,-68.902509  
0150-00072863,00:15:18.280,-68:53:21.92, .,17.7,16.5,1981.81,3.826164,-68.889423  
0150-00073055,00:15:20.016,-68:53:23.36, .,18.7,17.5,1981.81,3.8334,-68.889823  
0150-00074055,00:15:29.570,-68:54:38.01, .,19.3,18,1981.81,3.873206,-68.910559  
[kolsen@gp02 ~]$ datalab query sql="select * from usno.a2 limit 10" out="mydb://usno_test2"  
[kolsen@gp02 ~]$ datalab query sql="select * from usno.a2 limit 10" out="vos://foo2.csv"  
[kolsen@gp02 ~]$ █
```

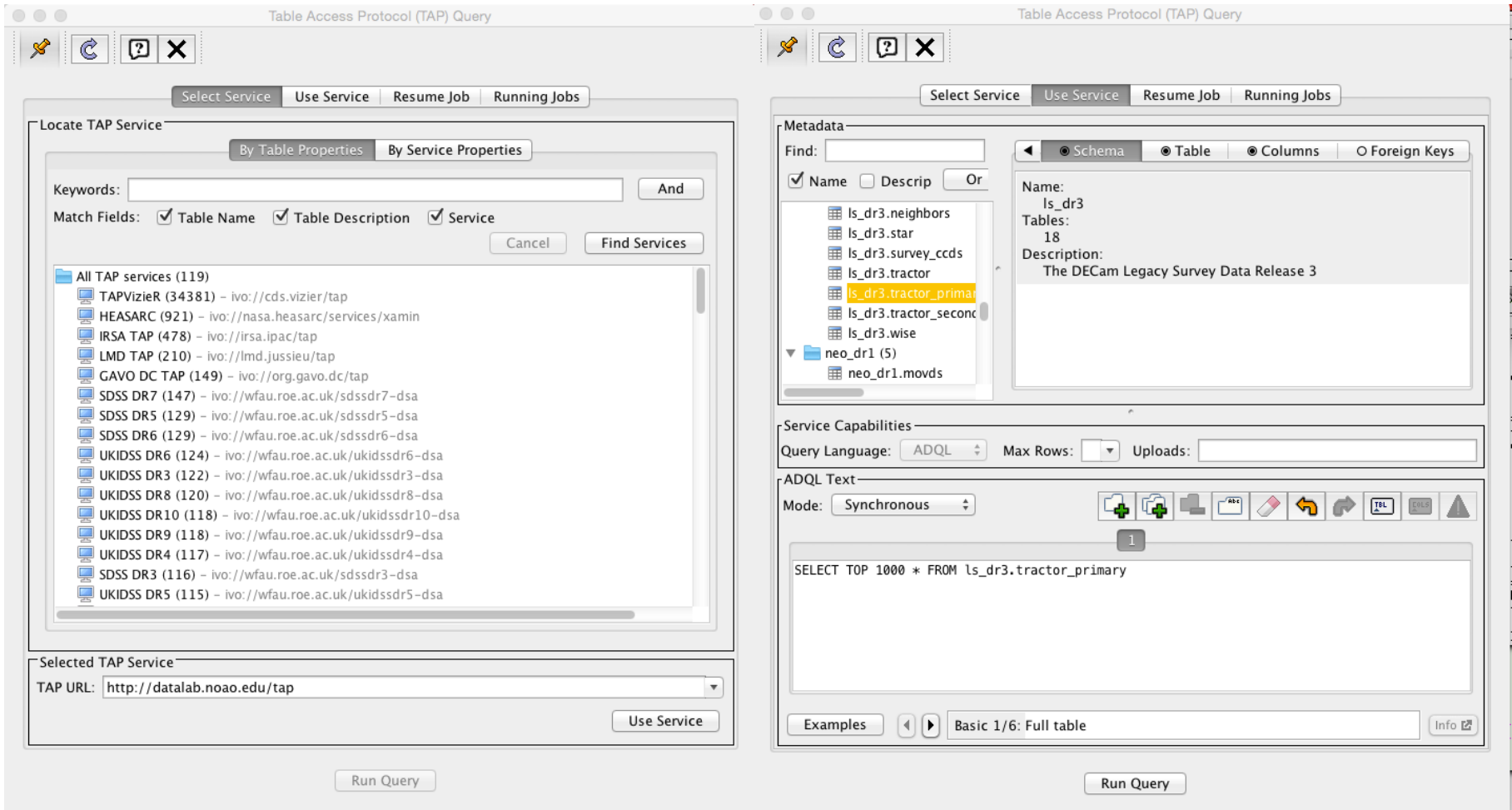
- Through the Python queryClient module:

```
In [4]: from dl import authClient, queryClient
        from getpass import getpass
        token = authClient.login(raw_input('Enter username: '),getpass('Enter password: '))
```

```
In [29]: %%time
        query="SELECT id,ra,dec,gmag,rmag FROM smash_dr1.object WHERE fieldid=169 LIMIT 100"
        try:
            response = queryClient.query(token, sql = query, fmt = 'csv')
        except Exception as e:
            print e.message
            raise
        print response[:205]
```

```
id,ra,dec,gmag,rmag
169.458572,185.342365895208,-32.1201617232873,24.8856,24.6991
169.460663,185.348188180985,-32.1200524648251,24.665,24.5361
169.1065651,185.353177442806,-32.1208638198927,25.0639,24.6239
CPU times: user 7.4 ms, sys: 956 µs, total: 8.36 ms
Wall time: 53 ms
```

- Through TOPCAT:



The image displays two screenshots of the Table Access Protocol (TAP) Query interface, showing the process of locating a service and querying its metadata.

Left Screenshot: Locate TAP Service

- Keywords:** (Empty text field)
- Match Fields:** Table Name Table Description Service
- All TAP services (119):**
 - TAPVizieR (34381) - ivo://cds.vizier/tap
 - HEASARC (921) - ivo://nasa.heasarc/services/xamin
 - IRSA TAP (478) - ivo://irsa.ipac/tap
 - LMD TAP (210) - ivo://lmd.jussieu/tap
 - GAVO DC TAP (149) - ivo://org.gavo.dc/tap
 - SDSS DR7 (147) - ivo://wfau.roe.ac.uk/sdssdr7-dsa
 - SDSS DR5 (129) - ivo://wfau.roe.ac.uk/sdssdr5-dsa
 - SDSS DR6 (129) - ivo://wfau.roe.ac.uk/sdssdr6-dsa
 - UKIDSS DR6 (124) - ivo://wfau.roe.ac.uk/ukidssdr6-dsa
 - UKIDSS DR3 (122) - ivo://wfau.roe.ac.uk/ukidssdr3-dsa
 - UKIDSS DR8 (120) - ivo://wfau.roe.ac.uk/ukidssdr8-dsa
 - UKIDSS DR10 (118) - ivo://wfau.roe.ac.uk/ukidssdr10-dsa
 - UKIDSS DR9 (118) - ivo://wfau.roe.ac.uk/ukidssdr9-dsa
 - UKIDSS DR4 (117) - ivo://wfau.roe.ac.uk/ukidssdr4-dsa
 - SDSS DR3 (116) - ivo://wfau.roe.ac.uk/sdssdr3-dsa
 - UKIDSS DR5 (115) - ivo://wfau.roe.ac.uk/ukidssdr5-dsa
- Selected TAP Service:** TAP URL: <http://datalab.noao.edu/tap>

Right Screenshot: Metadata

- Find:** (Empty text field)
- Metadata View:**
 - Name Descrip Or
 - ls_dr3.neighbors
 - ls_dr3.star
 - ls_dr3.survey_ccds
 - ls_dr3.tractor
 - ls_dr3.tractor_primary** (highlighted)
 - ls_dr3.tractor_second
 - ls_dr3.wise
 - neo_dr1 (5)
 - neo_dr1.movds
- Service Capabilities:**
 - Query Language: ADQL
 - Max Rows: (Dropdown)
 - Uploads: (Text field)
- ADQL Text:**
 - Mode: Synchronous
 - Query: `SELECT TOP 1000 * FROM ls_dr3.tractor_primary`

In [14]:

Slide Type -

```
bands = list('gri')
images = download_deepest_images(tbl['ra'][1], tbl['dec'][1], fov=0.07, bands=bands) # FOV in de
```

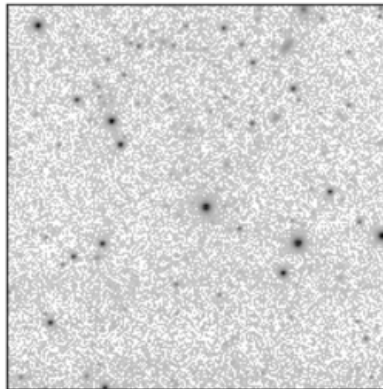
```
The full image list contains 2514 entries
Band g: downloading deepest stacked image...
Band r: downloading deepest stacked image...
Band i: downloading deepest stacked image...
Downloaded 3 images.
```

In [15]:

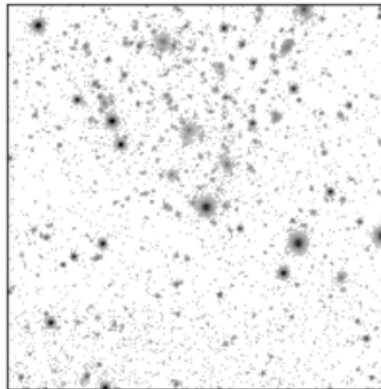
Slide Type -

```
plot_images(images, bands=bands)
```

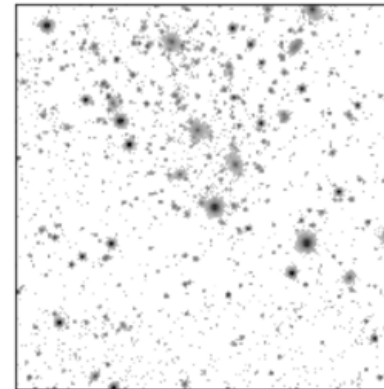
g band



r band



i band



- myDB:

```
In [29]: query = "select * from usno.b1 limit 1000"
try:
    response = queryClient.query (token, adql=query, fmt='csv',
                                  out='mydb://mags3')
    #queryClient.list (token, table='mydb://mags3')
except Exception as e:
    # Handle any errors in the query. By running this cell multiple times with the same
    # output file, or by using a bogus SQL statement, you can view various error messages.
    print (e.message)
else:
    if response is not None:
        print (response)           # print the response
    else:
        print ("OK")
```

<http://dlsvcs.datalab.noao.edu/query/list?table=mydb://mags3>

OK

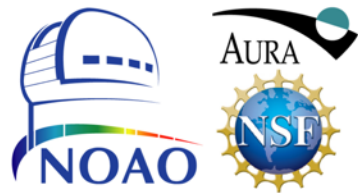
```
dataLab query sql="select * from usno.a2 limit 10" out="mydb://usno_test2"
```

- Virtual storage:

```
try:
    response = queryClient.query (token, adql=query, fmt='csv',
                                  out='vos://mags.csv')
except Exception as e:
    # Handle any errors in the query. By running this cell multiple times with the same
    # output file, or by using a bogus SQL statement, you can view various error messages.
    print (e.message)
else:
    if response is not None:
        print (response)           # print the response
    else:
        print ("OK")

# Remove the file we just created, but list it first to show it exists
storeClient.ls (token, name='vos://mags.csv')
storeClient.rm (token, name='vos://mags.csv')
```

```
datalab query sql="select * from usno.a2 limit 10" out="vos://foo2.csv" ■
```



- File transfer:

```
[kolsen@gp02 ~]$ datalab put fr=/etc/hosts to=vos://  
(1 / 1) /etc/hosts -> vos://hosts  
[kolsen@gp02 ~]$ datalab get fr=vos://hosts to=/tmp/myhosts  
(1/1) [=====] [ 281B] hosts
```

Try it out and get in touch!

- Web: datalab.noao.edu
- Email: datalab@noao.edu
- GitHub: <https://github.com/noao-datalab>
- Twitter: @NOAODataLab

