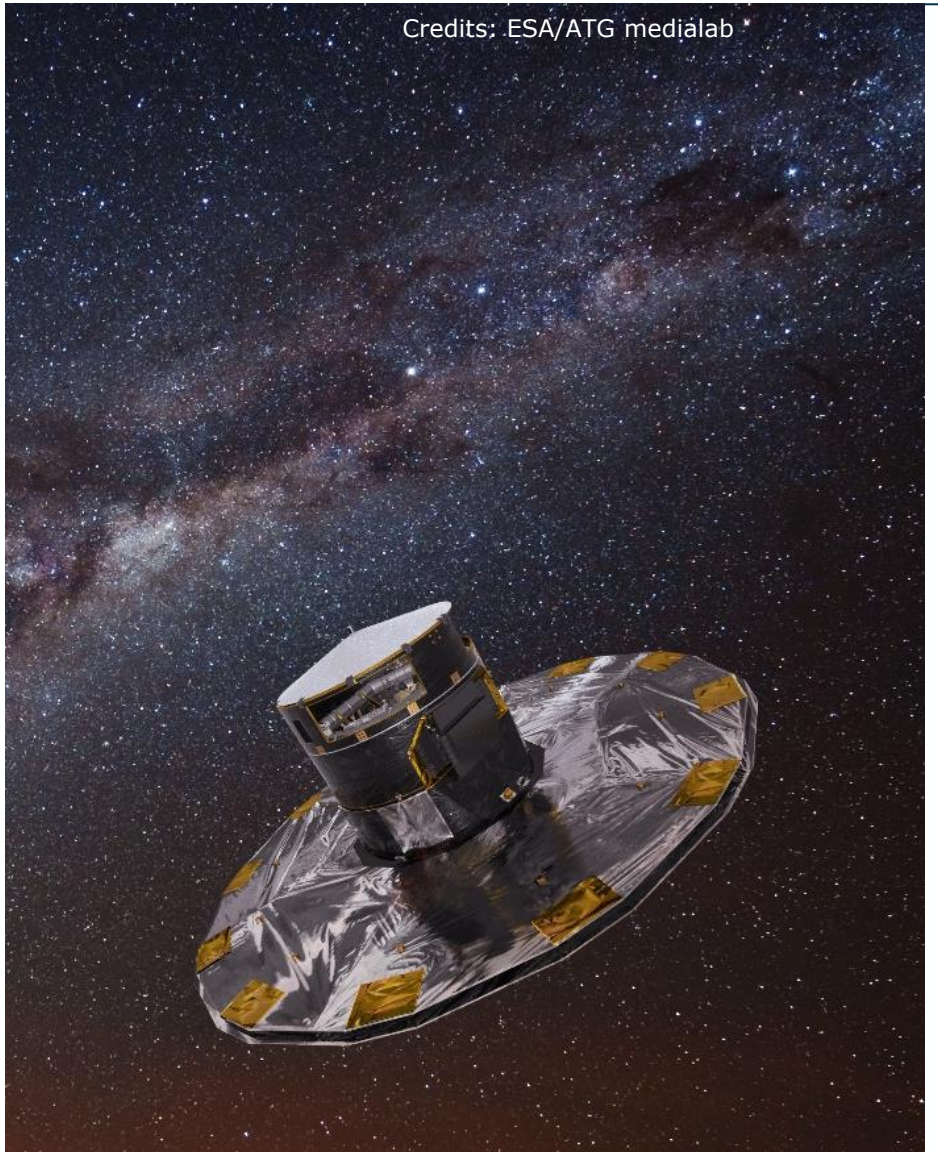
Gaia in ESA Datalabs: Exploiting large catalogues at scale

Enrique Utrilla Molina, Aurora Technology for ESA/ESAC

ESA Datalabs 2022 workshop

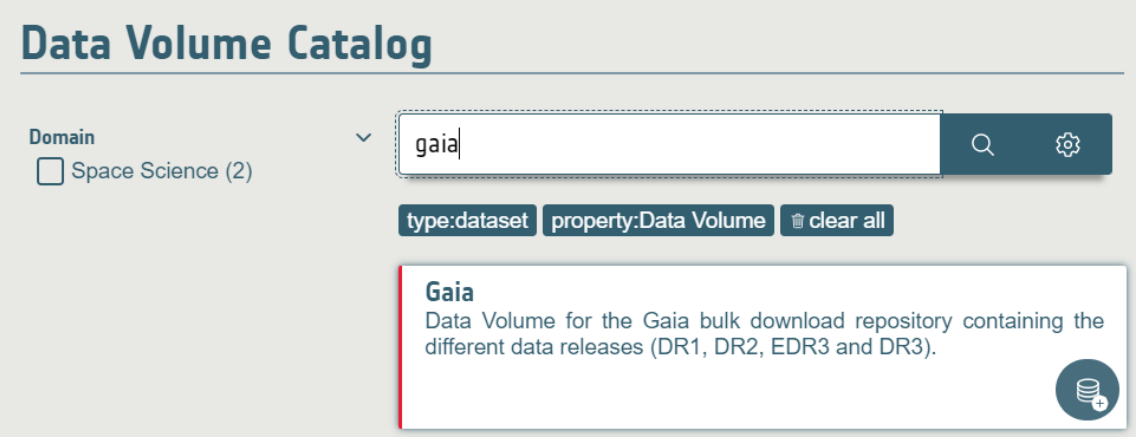
24/11/2022

The Gaia mission



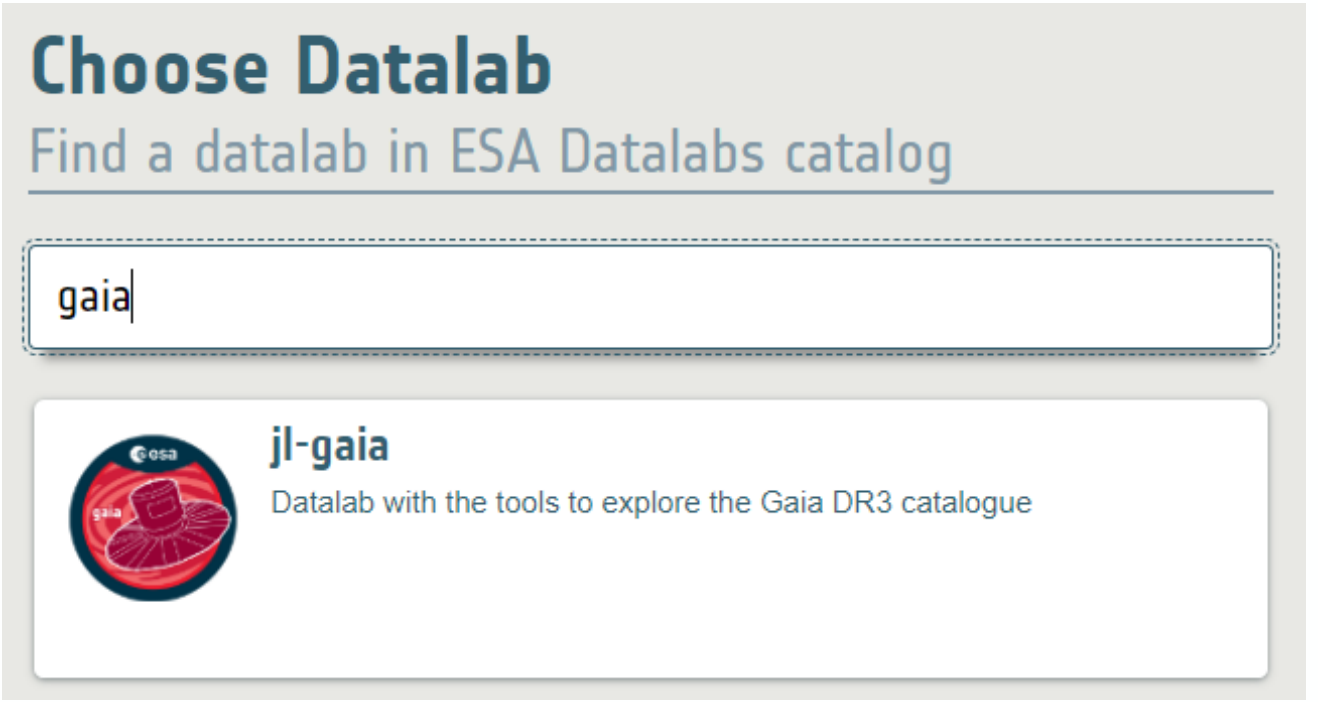
- Launched in 2013
- Orbiting L2
- Largest and most accurate astrometric and photometric survey to date (~1800 millions of sources in DR3)
- Largest ever low resolution spectroscopy survey (~220 million)
- Largest ever radial velocity survey (~34 million)
- Lots of derived data:
  - Astrophysical parameters
  - Variability
  - Binaries
  - Quasars and Galaxies
  - Asteroids
  - …
- More papers are being written based on Gaia than in Hubble

Gaia resources in ESA Datalabs

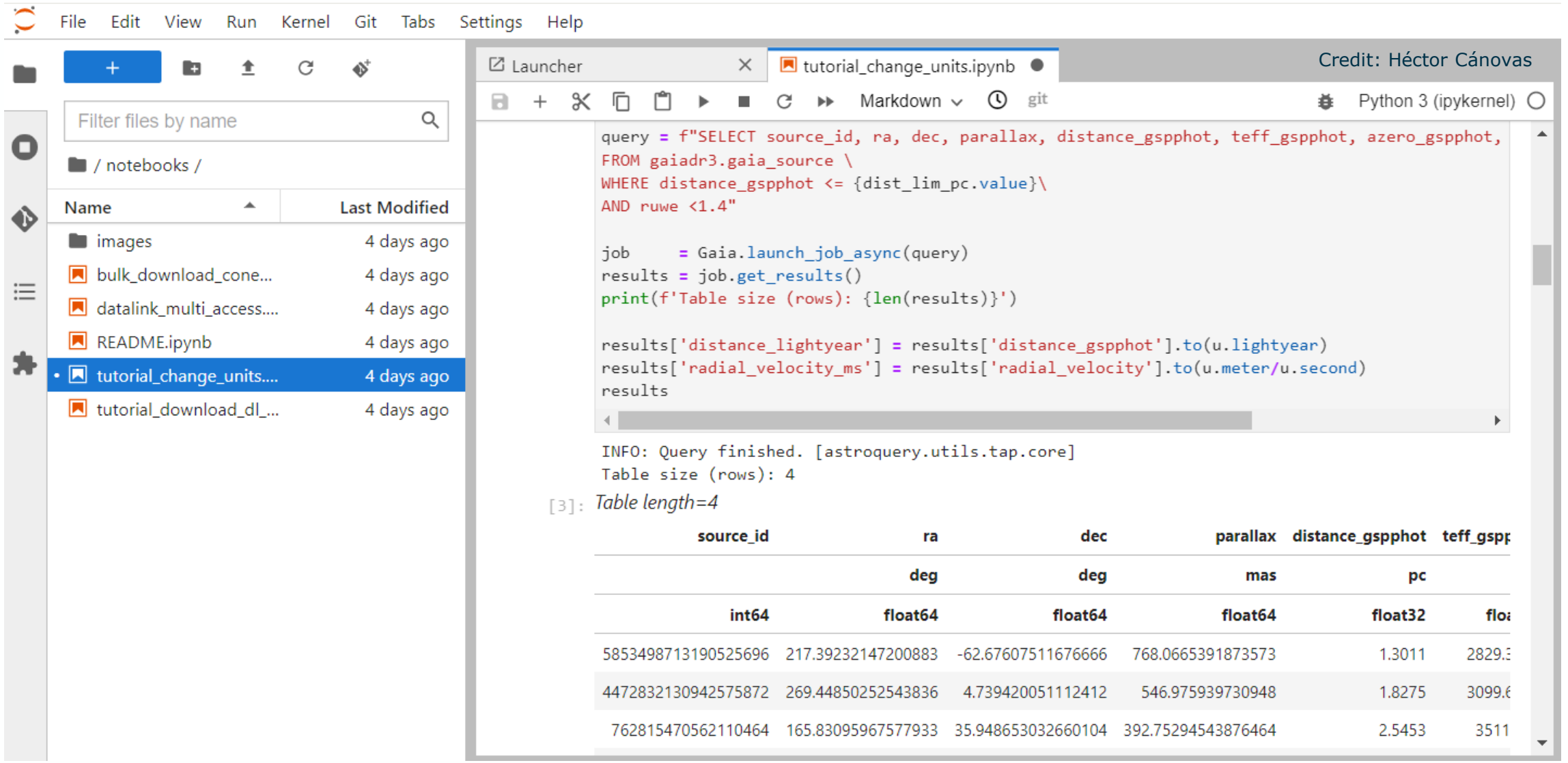


- Direct access to all released Gaia data
- Compressed CSV or ECSV format
- DR1, DR2, EDR3 and DR3

- Dedicated datalab for Gaia based on JupyterLab
- Tutorials notebooks
- Preconfigured with additional Gaia-specific utilities



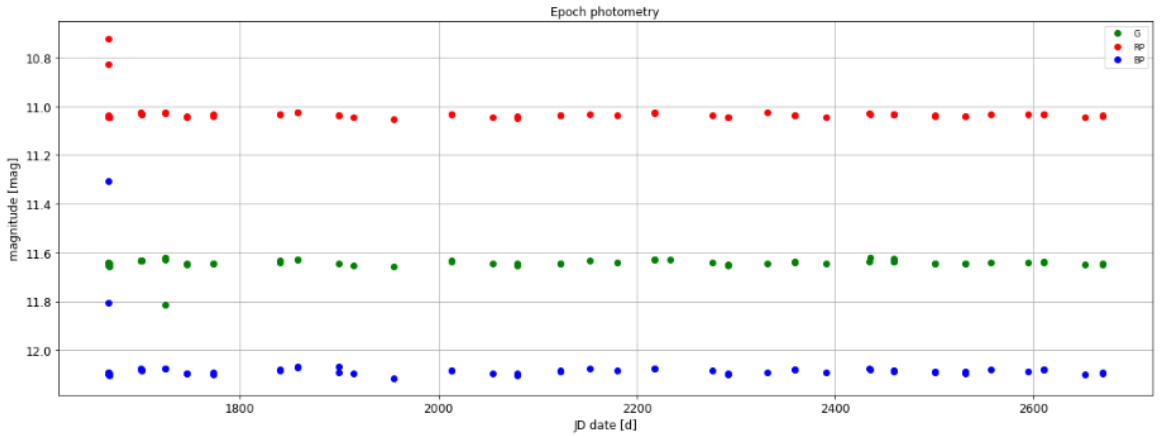
jl-gaia Datalab: astroquery.gaia TAP

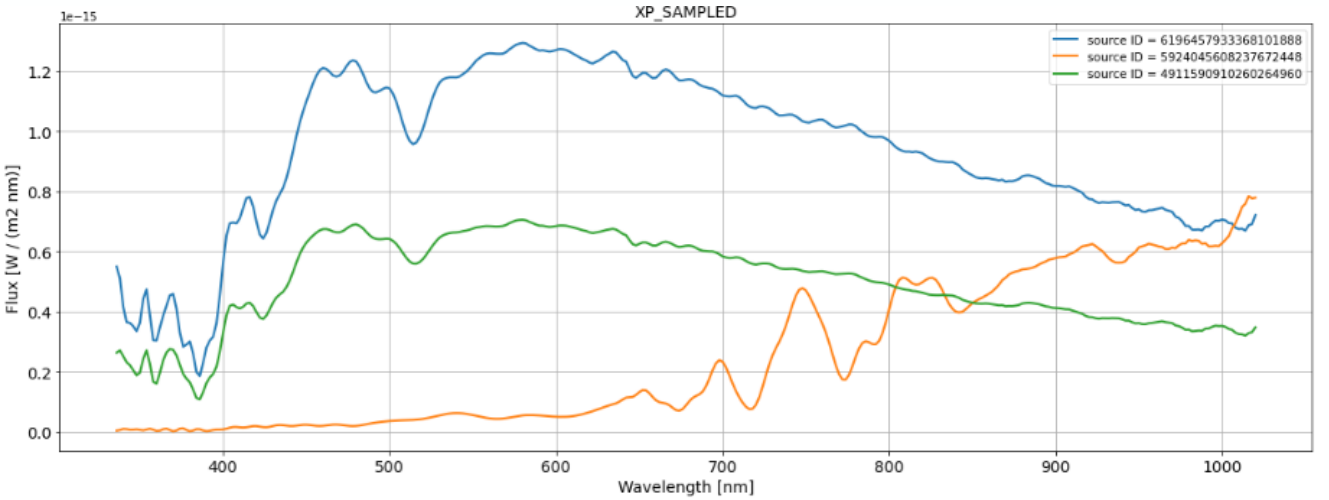


jl-gaia Datalab:astroquery.gaia DataLink

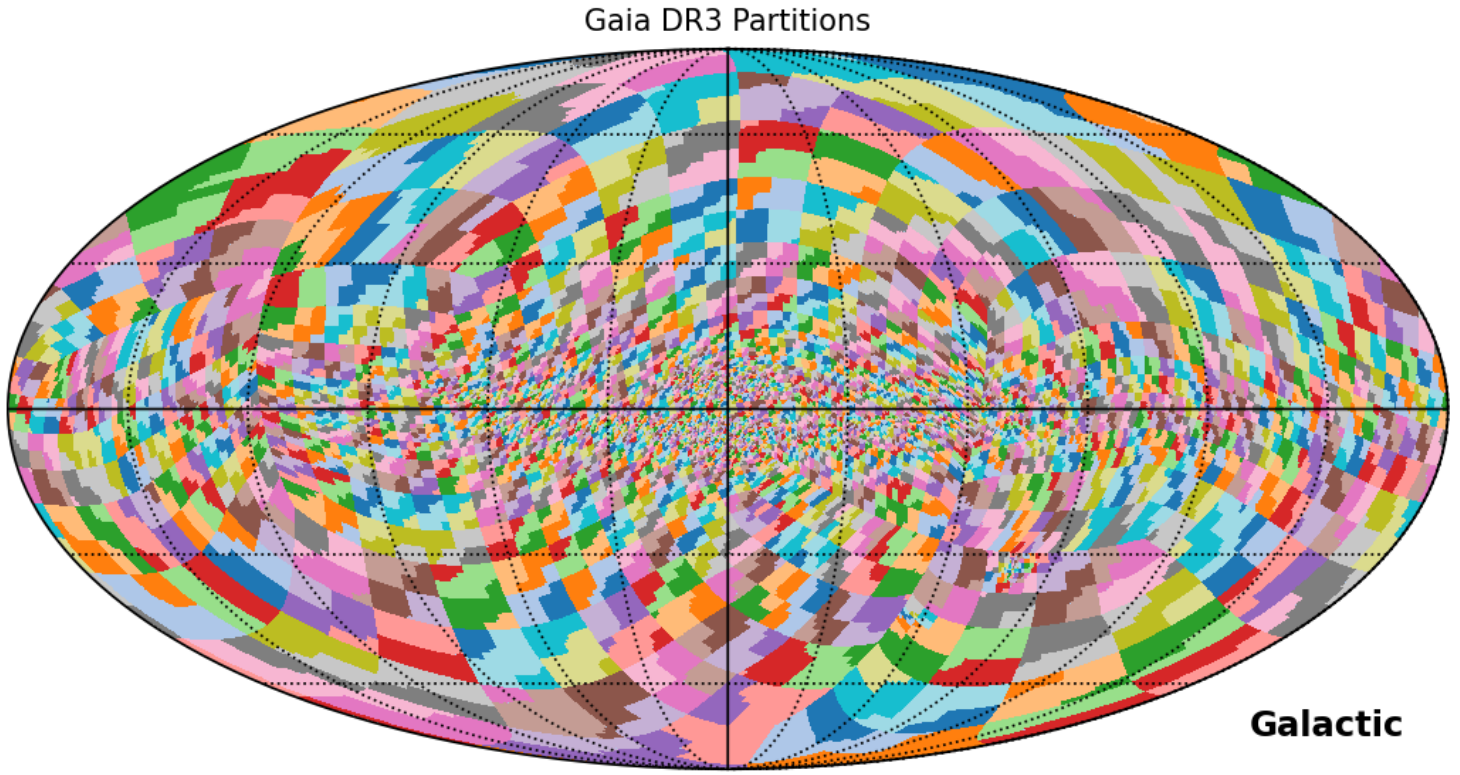
```
[5]: retrieval_type = 'ALL'          # Options are: 'EPOCH_PHOTOMETRY', 'MCMC_GSPPHOT', 'MCMC_MSC', 'XP_SAM
     data_structure = 'INDIVIDUAL'   # Options are: 'INDIVIDUAL', 'COMBINED', 'RAW'
     data_release   = 'Gaia DR3'     # Options are: 'Gaia DR3' (default), 'Gaia DR2'


     datalink = Gaia.load_data(ids=results['source_id'], data_release = data_release, retrieval_type=retrie
     dl_keys  = [inp for inp in datalink.keys()]
     dl_keys.sort()

     print()
     print(f'The following Datalink products have been downloaded:')
     for dl_key in dl_keys:
         print(f' * {dl_key}')
```

Credit: Héctor Cánovas





Direct file access



The complete Gaia catalogue files are directly accessible from ESA Datalabs

Most DR3 tables are partitioned following the same schema of contiguous HEALPix level 8 ranges

Similar number of entries in gaia_source, may not be homogeneous in other tables

Direct file access

Load utilities

```
[1]: run "~/notebooks/libraries/gaia_utils.py"
```

Set Gaia volume

```
[2]: dr3 = get_gaia_utils("DR3")
     dr3.set_gaia_volume("/home/media/data/user/gaia")
```

Spatial query

```
[3]: partitions = dr3.cone_search_partitions(14.2, 28.8)
     partitions
```

```
Using default radius value: 1.0 arcsec
[3]: ['034570-036209']
```

Find file paths

```
[4]: files = dr3.get_files(partitions)
     files
```

```
[4]: ['/home/media/data/user/gaia/gdr3/gaia_source/GaiaSource_034570-036209.csv.gz']
```

Load data from file

```
[5]: %%time
     data = dr3.load_file(files[0])
```

```
CPU times: user 1min 43s, sys: 7.19 s, total: 1min 51s
Wall time: 1min 51s
```

```
[6]: data
```

[6]: *Table length=536396*

solution_id	designation	source_id	random_index	ref_epoch
				yr
int64	str27	int64	int64	float64	
1636148068921376768	Gaia DR3 304080935778333056	304080935778333056	628370977	2016.0	27.
1636148068921376768	Gaia DR3 304080935778333184	304080935778333184	1651938970	2016.0	27.5
1636148068921376768	Gaia DR3 304080935778336000	304080935778336000	792769992	2016.0	27.5
1636148068921376768	Gaia DR3 304080935778339072	304080935778339072	1489924736	2016.0	27.
1636148068921376768	Gaia DR3 304081008792792832	304081008792792832	1431357637	2016.0	27.5
1636148068921376768	Gaia DR3 304081008793232000	304081008793232000	72989943	2016.0	27.5
1636148068921376768	Gaia DR3 304081073217310336	304081073217310336	231103464	2016.0	27.
1636148068921376768	Gaia DR3 304081107577050752	304081107577050752	67591202	2016.0	27.

GaiaXPy

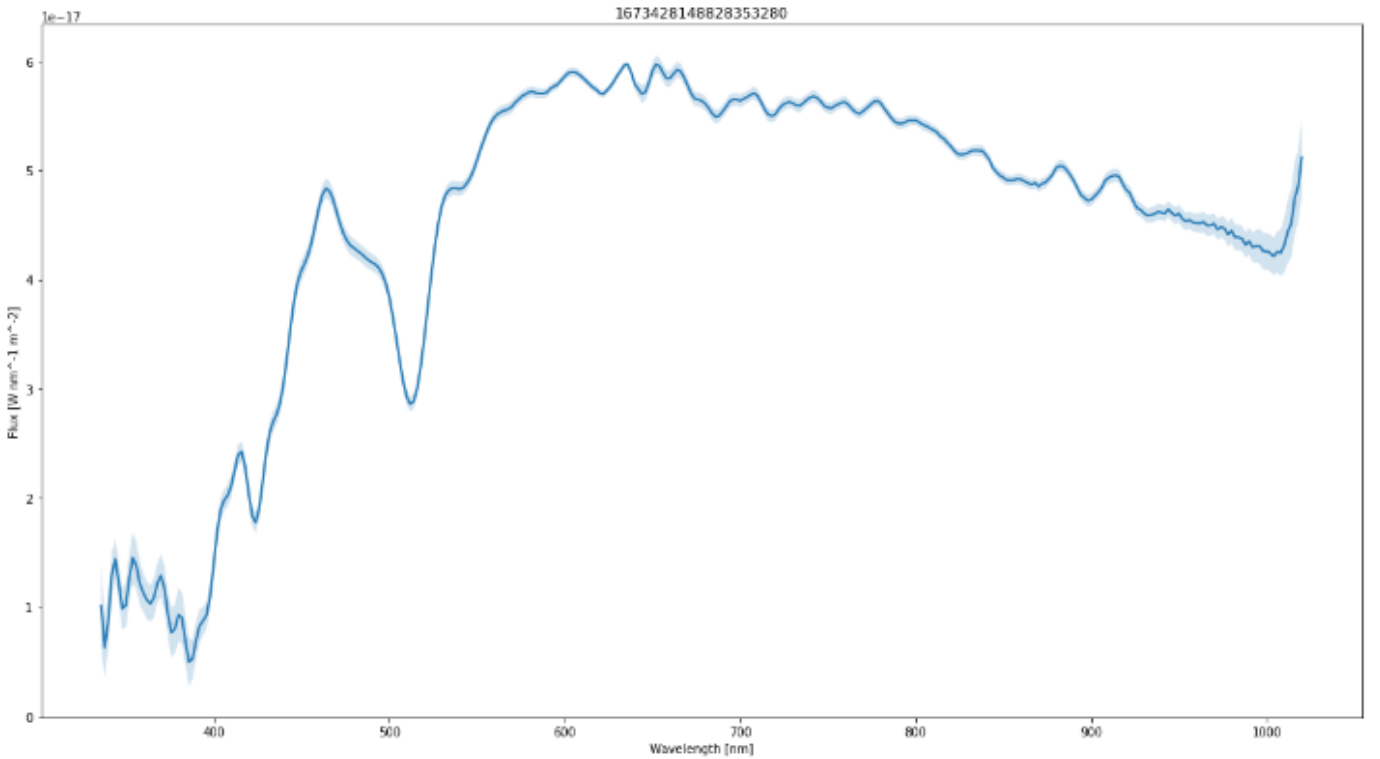
- XP (low resolution) spectra in DR3:
  - ~220 million continuous mean spectra
    - Internally calibrated
    - Coefficients of Hermite functions
    - Encoded in pseudo-wavelengths
  - ~35 million sampled mean spectra
    - Subset of continuous spectra
    - Externally calibrated
    - W/m2 per nm

GaiaXPy is a tool to calibrate and change the sampling of the continuous spectra as needed

https://gaia-dpci.github.io/GaiaXPy-website/

[1]:
```
from gaiaxpy import calibrate, plot_spectra
import numpy

source_ids=["1673428148828353280"]
calibrated_df, sampling = calibrate(source_ids, save_file=False)
plot_spectra(calibrated_df, sampling=sampling, multi=False, show_plot=True)
```

```
0/1 [00:00<?, ?spec/s]
```



Gaia DR3 contents (selected tables, compressed CSV format)

Table	Rows	Size
gaia_source	1,811,709,771	757 GB
astrophysical_parameters	1,590,932,717	258 GB
astrophysical_parameters_supp	473,020,612	202 GB
mcmc_samples_gsp_phot	449,297,716	3.3 TB
mcmc_samples_msc	348,711,151	1.4 TB
xp_continuous_mean_spectra	219,197,643	3.7 TB
xp_sampled_mean_spectra	34,468,373	115 GB

Total DR3 size ~8.9 TB, and DR4 expected to be much bigger

Work smarter, not harder…

- **Work with smaller datasets:**
  - Filter by position/partition or some other parameter
  - Random sampling of the sources:
    - `random_index` column in `gaia_source`

- **Don't use "`select *`"**

- **Use `gaia_source_lite` if possible**

But sometimes full table scans over the whole dataset (or al large part of it) are needed…

…or work smarter AND harder

- **Quantity sometimes is a quality on its own:**
  - Very detailed histograms and statistics
  - Detection of outliers that are not so random
  - Machine Learning
  - …

- **For large workloads, vertical scaling is not enough → parallelization**

- **Nevertheless, brute force is not a substitute for algorithm optimization**

A glimpse into the future

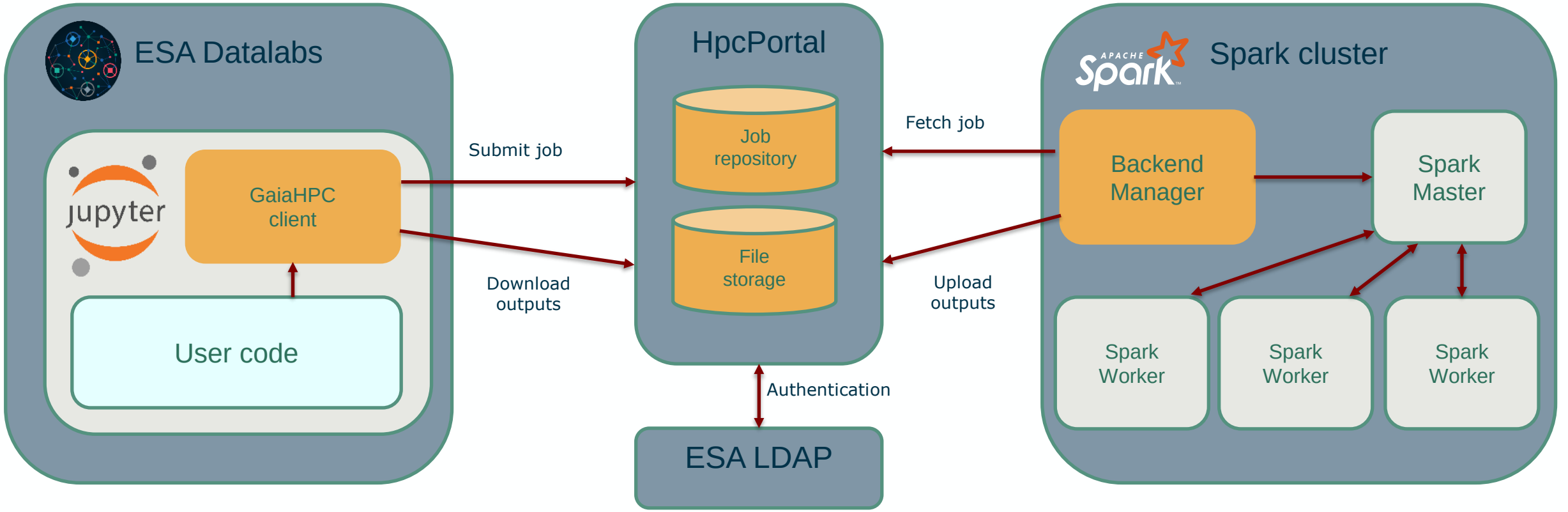
Apache Spark: Large scale analytics

- In Gaia, we use an Spark cluster (among other tools) for validation of the catalogue data
  - 2 clusters (PRE & OPS) with ~340 CPU cores each, with JupyterLab frontend

- Open source
- Automatic distributed processing over a large number of processing nodes
- APIs for Java, Scala, Python and R
- ANSI SQL-compatible
- Different storage formats: CSV, JSON, ORC, Parquet, …
  - Recommended: Parquet

GaiaHPC: Architecture

Delegate computations to an external backend, e.g. an Apache Spark cluster
Proof of Concept: Not yet released in the public Gaia datalab template



GaiaHpc: SQL query on Spark

```
[1]: from gaiahpc import GaiaHpcClient
```

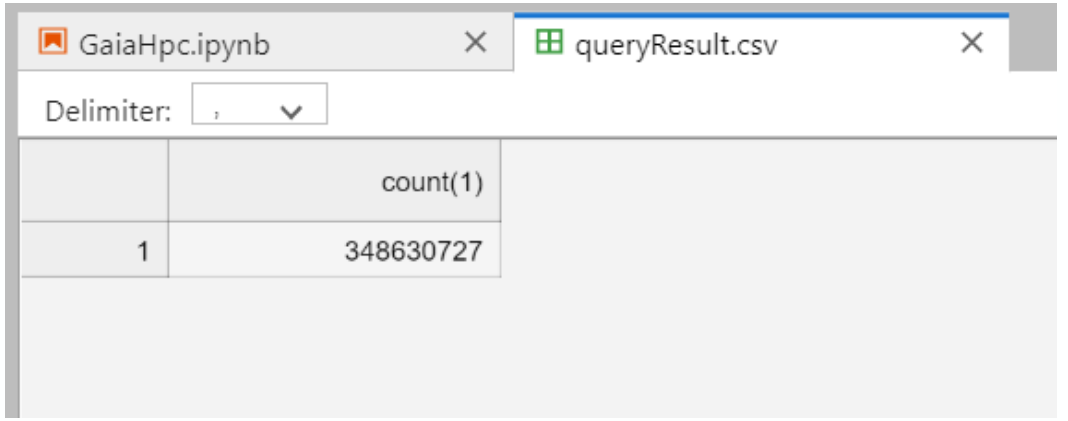
```
[2]: client = GaiaHpcClient("eutrilla")
```

```
Password for user eutrilla:  ···········
```

```
[3]: output_files = client.sql("""
SELECT COUNT(*)
FROM gaiadr3.gaiasource
WHERE photGMeanMag <= 18.25
      AND hasMcmcMsc = true
""")
output_files
```

```
2022/11/21 15:12:38 - Created job 44

2022/11/21 15:12:43 - Status: RUNNING
2022/11/21 15:13:58 - Status: FINISHED

2022/11/21 15:13:58 - Found 1 files:
2022/11/21 15:13:58 -    - Downloaded /media/user/Job_44/queryResult.csv

2022/11/21 15:13:58 - Job 44 deleted from the server

2022/11/21 15:13:58 - Operation completed in 0:01:20.188377
```

```
[3]: ['/media/user/Job_44/queryResult.csv']
```

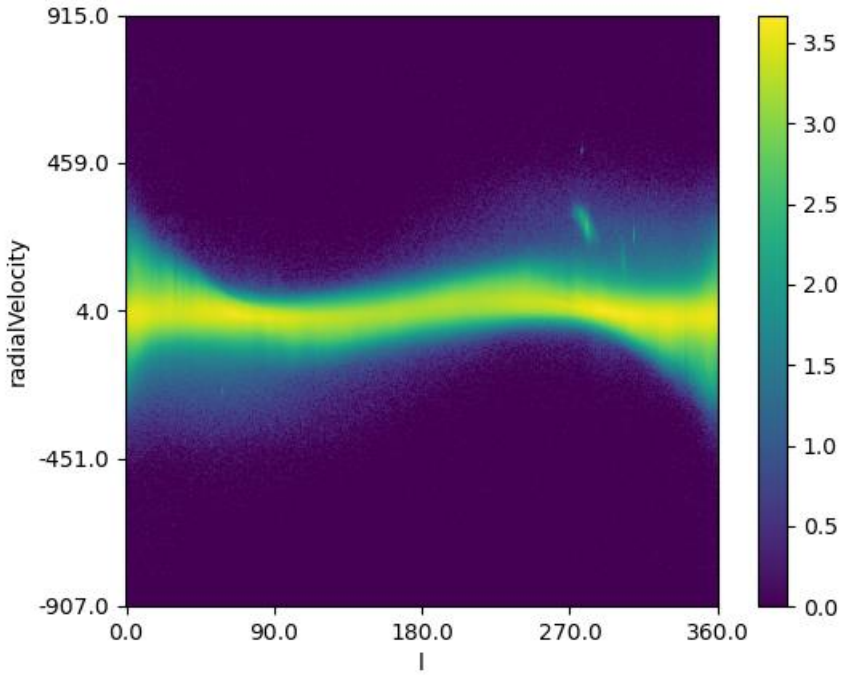


GaiaHpc: Other types of jobs

DensityMap

```
[4]: from gaiahpc.models import JobRequest
densityMap = JobRequest(type="DensityMap",
                        parameters = {"table": "gaiadr3.gaiasource",
                                      "columns":"l, radialVelocity"})
```

```
[5]: client.run(densityMap)
```

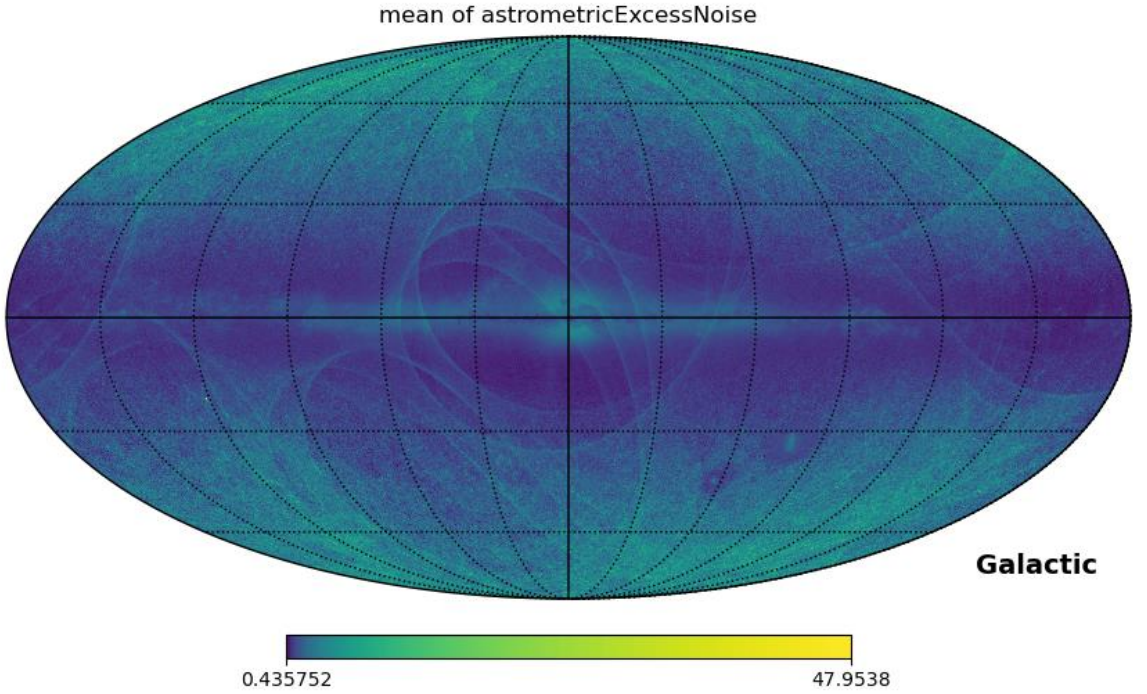



```
Operation completed in 0:04:30.417462
```

HealpixMap

```
[7]: excessNoiseDistribution = JobRequest(type="HealpixMap",
                                          parameters = {"table": "gaiadr3.gaiasource",
                                                        "column": "astrometricExcessNoise",
                                                        "stat": "mean"})
```

```
[9]: client.run(excessNoiseDistribution)
```




```
Operation completed in 0:04:10.421384
```

GaiaHpc: Other types of jobs

SimilarSpectra

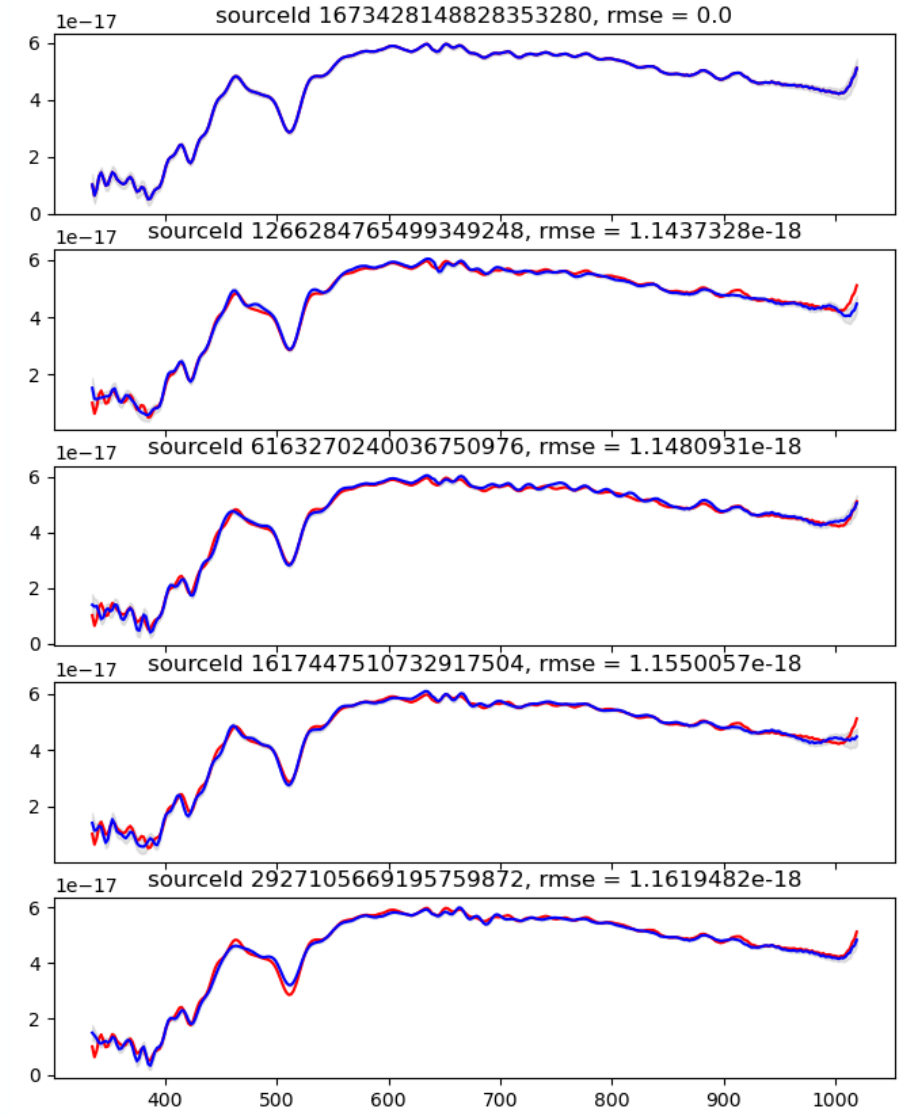
[10]:
```
from gaiahpc.models import JobRequest
similarSpectra = JobRequest(type="SimilarSpectra",
                            parameters = {"limit": "5",
                                          "prototype_file": "my_prototype",
                                          "flux_column": "flux"})
```

[12]:
```
client.run(similarSpectra, inputs={"my_prototype":"./prototype.csv"})
```

```
Operation completed in 0:02:35.290411
```

Blue: spectrum
Red: reference prototype

The first spectra found is identical to the prototype:
In this example, the prototype was downloaded from an existing source, so it has successfully found the original source



Summary

- ESA Datalabs provides a collaborative platform to bundle together:
  - 3rd party libraries
  - Our own tools and scripts
  - Tutorials
  - User code

- It supports the tools for all preferences:
  - JupyterLabs + astropy/astroquery
  - But also other tools such as Topcat

- Can be extended to tap into large-scale data processing platforms

Questions?