



Gaia Data Mining Platform - update

Nigel Hambly, Dave Morris & Stelios Voutsinas
IRIS Collaboration Meeting, Jan 2023, Edinburgh

Email: gaiadmp-support@roe.ac.uk



Gaia Data Release 3: 'A release of superlatives...'

... yet it represents

- *only 30%* of the likely end-of-mission mean catalogue
- *a few %* of the likely end-of-mission data release volume

DR1 Sep'16 (15 m)



DR2 Apr'18 (22 m)



EDR3 Dec'20 (34 m)



DR3 June 2022 (34 m)



DR4 end 2025 TBC (5.5 yr)



DR5 end 2030 TBC (≥ 10 yr up to end-of-mission)



July 2014

May 2017

Jan 2020

June 2022

Early 2025

Beyond the largest and most accurate astrometric and photometric survey to date (Gaia EDR3):

- Largest ever spectrophotometric survey
- Largest ever radial velocity survey
- First space-based all-sky survey of QSO galaxy hosts and of the surface brightness profiles of galaxies in the local universe
- Highest accuracy spectrophotometric-dynamical survey of asteroids
- For many classes of variable stars: largest survey ever
- Largest ever collection of astrophysical data for stars in the Milky Way
- Non-single star survey that surpasses all the work on non-single stars from the past two centuries

GAIA MISSION STATUS

2901 days in science operations
107,225 GB of science data gathered
203,059,434,235 transits observed

Gaia DR3 data volume

- Small, focused usage provisioned via “traditional” interactive/programmatic interfaces
 - Backed by relational DBMS technology
- Bulk download of science-ready data products is provisioned via a *Content Delivery Network*
 - 8.9 TB of gzipped eCSV (text) files; 25 TB uncompressed
 - Single thread download/decompress the lot in roughly 10 days
 - Largest single data sets examples:
 - MCMC posterior PDF samples from astrophysical parameter inference system: 10.1 TB
 - Blue/red photometer mean spectra: 8.0 TB (only 12% of catalogue sources at this release)
 - Astrophysical parameters: 2.4 TB
 - Main source catalogue: 2.1 TB
- DRs 4 & 5 detailed contents still under discussion but likely **10s** (DR4) to **100s** (DR5) times bigger than DR3
 - More spectra, epoch-resolved data, raw and/or intermediate data, ...

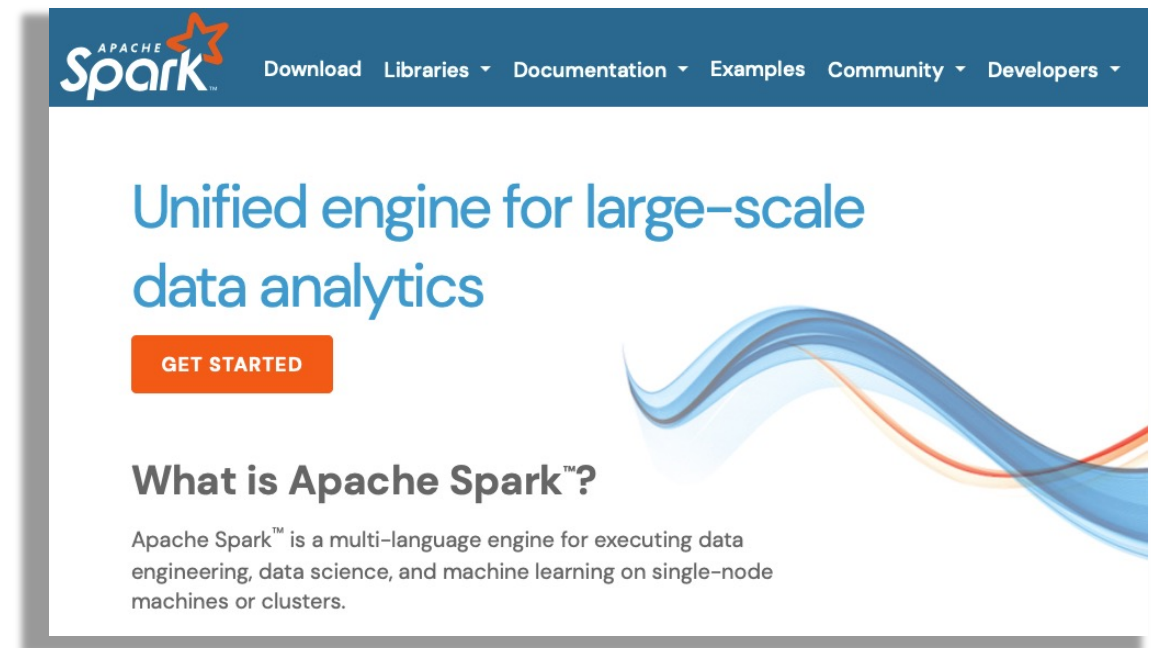
Advanced, scale-out usage scenarios

Community requirements gathered:

- Higher order, robust statistical aggregates
- Analysis of per-CCD photometry for short timescale variability
- Searches in Fourier-analysed time domain data
- Wholesale dataset trawls
- Pattern queries
 - some requiring Machine Learning techniques
- General CPU-intensive analysis
- Efficient searching for pairs (or higher multiples) of associated objects, e.g.
 - Lensed QSOs
 - Wide binaries
- Searches in time-resolved astrometric data, e.g. detect plane gravitational wave(s) or primordial stochastic GW background
 - Requires local plane coordinate residuals from epoch astrometry

The *Gaia Data Mining Platform*

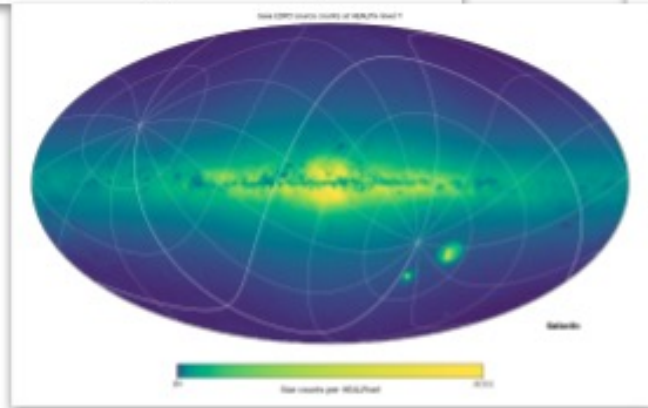
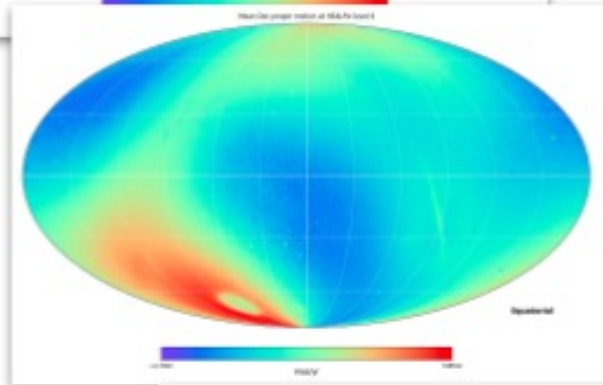
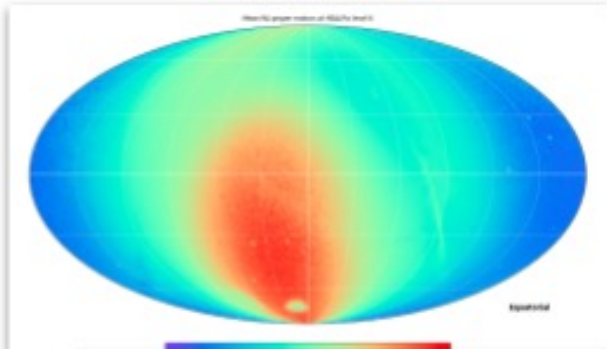
- The (obvious) solution to large-scale analysis: code-to-data platforms
 - Bring end-user code to lots of CPU co-located with the data
 - Employ distributed computing to mitigate increases in data volume and scale of processing
- The UK Gaia DMP
 - Deployed on the STFC IRIS Cloud
 - Employs Apache Spark ecosystem
 - Python notebook interface (Apache Zeppelin)
 - Friendly APIs to access distributed processing
 - Familiar libraries for vectorized operations
 - Machine Learning and many other libraries



Gaia DMP current status

The screenshot shows the Gaia DMP web interface. At the top, there is a dark red header with the Gaia DMP logo, navigation links for 'Notebook' and 'Job', a search bar, and a user profile for 'Nigel'. The main content area features a large 'Welcome to the Gaia Data Mining Platform!' message, followed by 'Powered by Apache Spark & Zeppelin'. A paragraph describes the platform as a science tool for large-scale data exploitation. Below this, there are sections for 'Notebook' (with links to import or create notes and a list of example notebooks), 'Help' (with a link to documentation), and 'Community' (with contact information and a GitHub link). A large 'gaia' logo is on the right. At the bottom, a row of logos includes 'Gaia in the UK', 'UKRI Science and Technology Facilities Council', 'iris', 'Gaia DPAC', 'Apache Spark', and 'Apache Zeppelin'.

- Deployed on Arcus OpenStack Cloud at Cambridge
- \approx 10 registered users
 - Mixture of postgrads, young postdocs and RSEs



```
from gaiautils.galactic import parallel
import pandas as pd
import numpy as np

def find_relevant_continuous_spectra(data_frame = DataFrame, template_df = DataFrame):
    """
    Given data frames defining a large set of RP spectra in continuous representation,
    and a single template example also in continuous representation in a data frame,
    search the former for cases of the latter. The selection condition is that the
    relevant basis coefficients in both RP and RP are similar within 1-sigma.

    Parameters:
    -----
    data_frame : DataFrame
        the data frame encapsulating the set of RP continuous representation spectra to be searched
    template_df : DataFrame
        the template, also in RP continuous representation encapsulated in a data frame.

    Returns:
    -----
    a new data frame containing all matches to the template.

    If convenience reference to template as a Row object:
    template_row = template_df.iloc[0][0]

    If extract the template names including only those bases relevant to the representation:
    template_relevant_bases = template_row[template_relevant_bases]
    template_relevant_bases = np.array(template_row[template_relevant_bases])
    template_relevant_bases_errors = np.array(template_row[template_relevant_bases_errors])
    template_relevant_bases_errors = np.array(template_row[template_relevant_bases_errors])
    template_relevant_bases_errors = np.array(template_row[template_relevant_bases_errors])
    template_relevant_bases_errors = np.array(template_row[template_relevant_bases_errors])

    If propagate the variance on the basis coefficients:
    template_relevant_bases_errors = template_relevant_bases_errors + template_relevant_bases_errors
    template_relevant_bases_errors = template_relevant_bases_errors + template_relevant_bases_errors
    template_relevant_bases_errors = template_relevant_bases_errors + template_relevant_bases_errors

    If define a weighted pandas DataFrame for comparison of other spectra against it:
    template_df_weighted =
    """
```

```
Assemble training and reserve test sets
import numpy

# define training (87%) and test (13%) sample splits (seeded randomness for repeatability)
good_87pc, good_13pc = all_good_training_df.sample(frac=[0.87, 0.13], 42)
bad_87pc, bad_13pc = all_bad_training_df.sample(frac=[0.87, 0.13], 42)

# transform to labeled feature vectors (R=0 = bad, 1.0 = good, as conveniently already defined in previous
# projections above)

# denoise and transform appropriate to the input required by the classifier's API.
# if need a dataframe with labels and features: use vector assembler.
from gaiautils.feature import VectorAssembler
ignore = ["label"]
assembler = VectorAssembler(inputCols=[c for c in good_87pc.columns if a not in ignore], outputCol="features")

# training sets
good_training_df = assembler.transform(good_87pc).drop("astrometric_features")
bad_training_df = assembler.transform(bad_87pc).drop("astrometric_features")
# ... R.S. the original individual feature columns are dropped to save memory (since they are duplicated into the
# resulting feature vector).

# testing sets
good_testing_df = assembler.transform(good_13pc).drop("astrometric_features")
bad_testing_df = assembler.transform(bad_13pc).drop("astrometric_features")

# concatenate the training sets into a single dataframe
training_df = good_training_df.union(bad_training_df)
testing_df = bad_testing_df
```

Classification confusion matrix

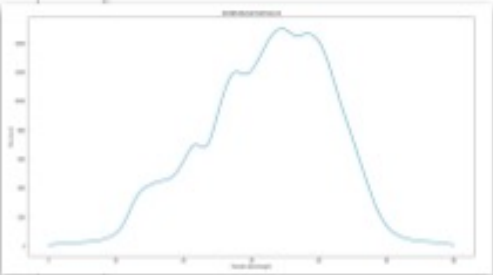
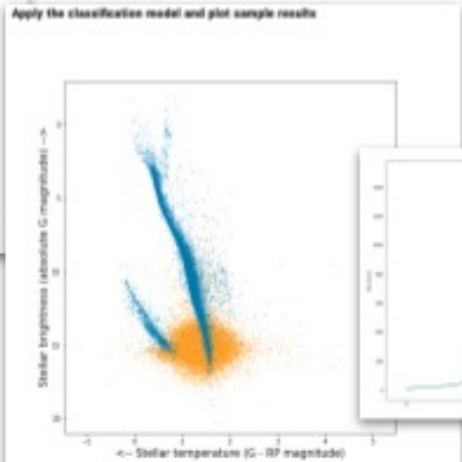
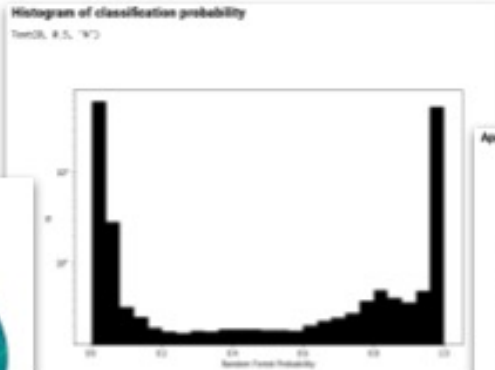
	1	2
1	88190	100
2	23	88174

Size: Classification for the test set: 8.33 %

Total: 111164

Relative importance of the selected features

parallel_error	0.261284
astrometric_sigma_max	0.186011
parallax_error	0.161482
parallax_over_error	0.148853
parallax_error	0.175266
astrometric_excess_noise	0.066267
ipd_gof_harmonic_amplitude	0.049801
ipd_frac_multi_peak	0.016112
ra_m	0.011676
visibility_periods_used	0.007354
pm_m	0.005126
astrometric_gof_all	0.004188
parallax	0.003384
ipd_frac_bad_align	0.003141
astrometric_excess_noise_sig	0.001889

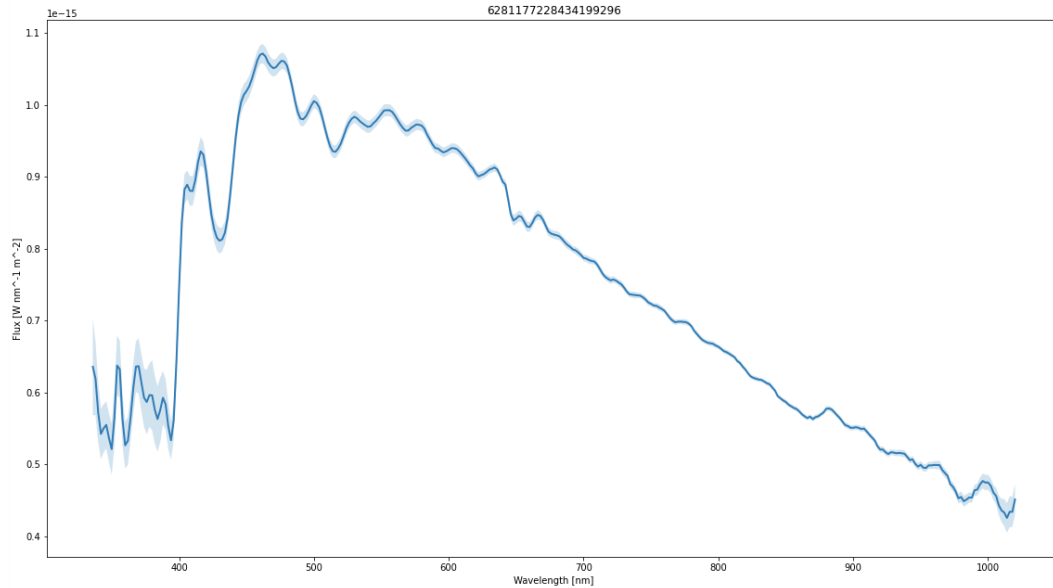


A detailed example: searching 2×10^8 spectra

```
xp_continuous_mean_spectrum_schema = StructType([\n    StructField('source_id', LongType(), False), # Unique source identifier (unique within a particular Data Release)\n    .\n    .\n    .\n    StructField('bp_coefficients', ArrayType(DoubleType()), True), # Basis function coefficients for the BP spectrum representation\n    StructField('bp_coefficient_errors', ArrayType(FloatType()), True), # Basis function coefficient errors for the BP spectrum representation\n    StructField('bp_coefficient_correlations', ArrayType(FloatType()), True), # Correlation matrix for BP coefficients\n    .\n    .\n    .\n    StructField('rp_coefficients', ArrayType(DoubleType()), True), # Basis function coefficients for the RP spectrum representation\n    StructField('rp_coefficient_errors', ArrayType(FloatType()), True), # Basis function coefficient errors for the RP spectrum representation\n    StructField('rp_coefficient_correlations', ArrayType(FloatType()), True), # Correlation matrix for RP coefficients\n    .\n    .\n    .\n])
```

- DR3 has 200 million blue + red spectra in basis-set representation
 - N basis coefficients
 - N coefficient uncertainties
 - $N(N-1)/2$ correlation coefficients
 - $N = 55$
- 2.7TB in compact (Parquet) format
- Simple use case: given one example template, find similar spectra ...

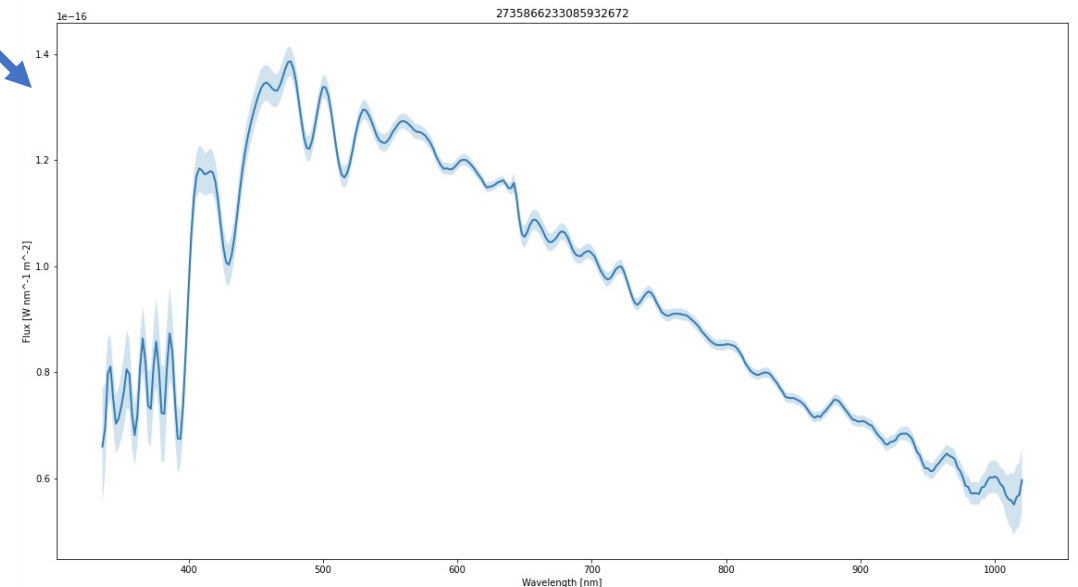
... for example, a Solar (G-) type star



- Statistical rigour: compute the *Mahalanobis* distance between the template and all others
 - In every case reconstruct the full 2d covariance matrix from the (flattened, 1d) correlation matrix and uncertainties vector
 - matrix & vector multiplications implemented as a “Pandas” (vectorized) User Defined Function for execution on Spark cluster worker nodes

Example: closest match at a given brightness limit in just over one hour

- Modest level of parallelism in (virtual) Spark cluster
- I/O bound (CPU wait time typically 50%)



Issues for discussion: when is a cloud not a cloud?!

- Currently Gaia DMP project pinned to a fixed allocation of hardware
 - Big thanks to IRIS RSAP and Cambridge HPC centre!
- Interactive analysis produces a highly variable workload ...
 - Normal day: *up to a few* active users
 - Workshop scenario: tens of users all doing similar things
 - At new data release: potentially hundreds of users wanting to take a look
- ... but that load pattern is predictable
 - We know when workshops and data releases will happen well in advance
- Solution: baseline allocation to serve daily use plus a mechanism for reserving large block of resources on specific dates, e.g.
 - Workshop: allocate 10x normal from [date] to [date]
 - New DR: allocate 100x normal from [date] to [date]
- Technical aspects (discuss)
 - OpenStack Blazar is an option (there may well be others)
 - Extra resources through another IRIS provider, or external (commercial?) cloud, or...?
 - Gaia DMP deployment is being made portable
 - RSAP (and other administrative) implications?

Commercial clouds for data science: workshop announcement



- In collaboration with colleagues at University of Barcelona / BSC “techno-week” on commercial Cloud computing is being organized
 - 29 May – 2 June 2023
 - Opportunity to compare/contrast experience with data science applications deployed in academic / commercial cloud infrastructure
 - Further details: <https://indico.icc.ub.edu/event/132/>