Rubin Observatory

Plans for IVOA and Python interfaces to time series data

Eric Bellm & Gregory Dubois-Felsmann

DMLT | virtual | 12 May 2020

User recomputation of timeseries features highlights some open questions.

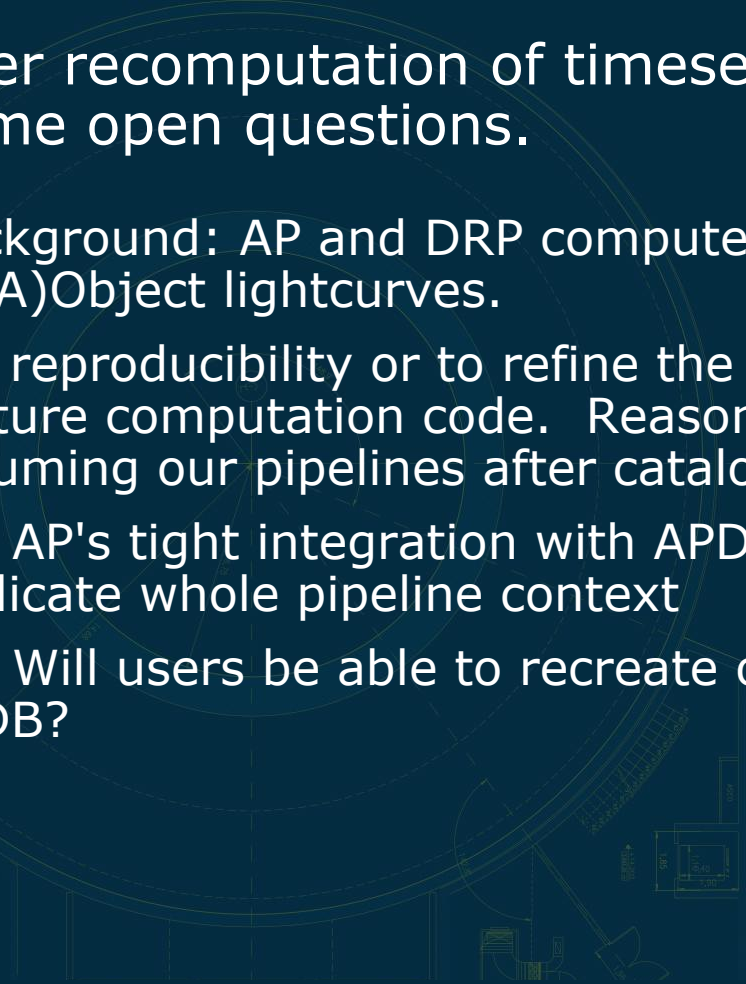
Rubin Observatory

Background: AP and DRP compute a series of timeseries statistics on (DIA)Object lightcurves.

For reproducibility or to refine the results, users will want to run our feature computation code. Reasonable from user's perspective to consider resuming our pipelines after catalog generation.

AP's tight integration with APDB makes it expensive/difficult to replicate whole pipeline context

Will users be able to recreate our data structures from queries to PPDB?



U.S. DEPARTMENT OF ENERGY | Office of Science

User story: I want to re-compute timeseries features using a subset of the data.

Rubin Observatory

I log into the Rubin Science Platform and open a notebook.

I want to query the PPDB (or QServ).

I write an ADQL query to select the data I want.

How do I send my query to the TAP server?

- PyVO?
- astroquery?
- Something custom?

What data type do I get back? VOTable?

How do I (or can I?) apply the stack timeseries feature code?

PyVO may be a natural choice for user TAP queries.

Rubin Observatory

PyVO is an astropy-affiliated python package for interfacing with VO services. SQuaRE already provides a small wrapper for auth.

astroquery may provide higher-level interfaces?

Queries return astropy Tables

But: can we run Stack code on astropy Tables? (no)

could allow users to reconstruct our internal data structures (currently pandas DataFrames)--a simple (?) convenience function

or rewrite our internal code to use astropy--not preferred

Let's make these decisions based on engineering & management judgment

Should we be returning lightcurves as some kind of Timeseries?

Rubin Observatory

Both IVOA and Astropy are developing specialized timeseries representations--basically tables with special columns and/or convenience methods.

Challenge is that there's *lots* of variation in methods & conventions among science communities (exoplanets, transients, variable stars, pulsars, solar system objects...)

DM-SST consensus is that these efforts are still in formative stages--wait for further development from the community rather than try to push from our side. Result: **our (DIA)Source/(DIA)Object tables will provide our public timeseries interface; we will not have a Timeseries abstraction.**

NSF
U.S. DEPARTMENT OF ENERGY | Office of Science

DM-SST Recommendations

Rubin Observatory

Document more clearly what users should expect to get back from ADQL queries (e.g., multiple objects that will need to be split)

Aim to develop some basic user tutorials about taking returns from ADQL back into internal representations (e.g., pandas DataFrames)

Timeseries feature code will be designed to run in the pipelines environment. We will aim to make it possible for users to run the feature code in the Science Platform on data returned from TAP queries. Don't commit to make the timeseries feature code a standalone package or generic/public "API"

Communicate these plans to the community as part of broader discussion of what timeseries features will be computed.