Faro: computation of scientific performance metrics

Faro (lighthouse) - shining light on the situation

- Not originally conceived as an acronym

- But can stand for: **F**ast (or **F**lexible) **A**nalysis of **R**ubin **O**bservatory performance

Collaboration between DM Science & Commissioning:

- Jeff Carlin, Simon Krughoff, Keith Bechtol, Leanne Guy, Colin Slater and the DM System Science Team

- Thanks also to Lauren MacArthur and the middleware team
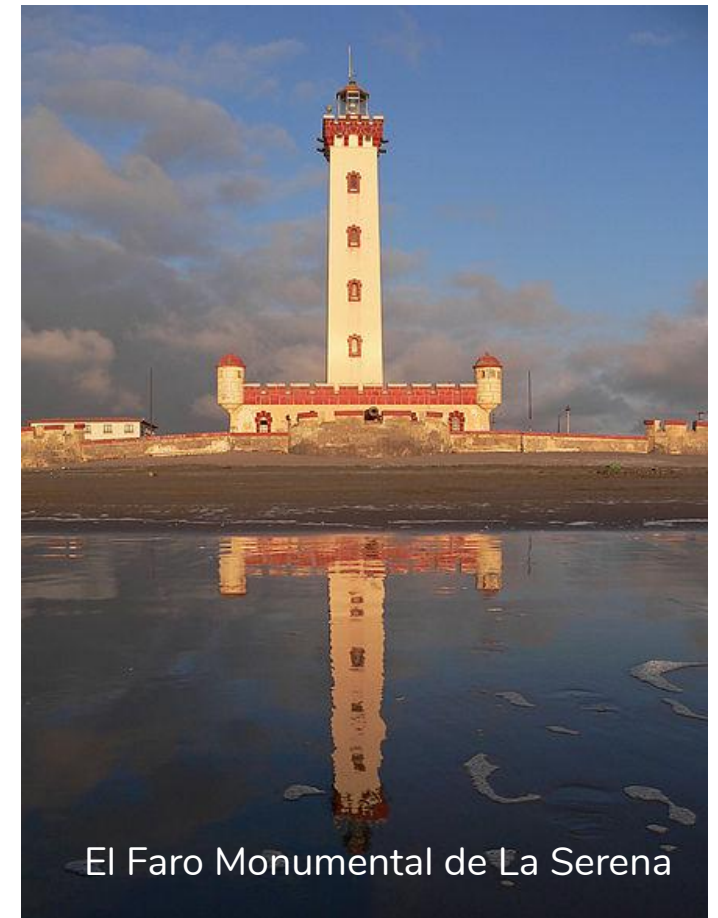
El Faro Monumental de La Serena

Scientific performance metrics - Requirements

Rubin
Observatory

DMSR:

- The DMS shall include software to enable the calculation of the scientific performance metrics defined in the OSS + LSR.

- Target values for LSST but are not verified as part of the DM requirement.

OSS + LSR:

- Verify scientific performance of as-built system with on-sky data.

- Commissioning science V&V team during commissioning.

Why a new package for metric computation?

- Computation of KPMs thus far has been part of the validate_drp package. Only a subset of all KPMs were implemented in validate_drp.

- When DM Science took over validate_drp the first goal was to complete implementation of the remaining metrics. At the same time Gen 3 was coming up. We saw an opportunity to redesign the metric computation package to be based on Gen3 rather than continue with implementation in validate_drp and port to Gen3 later.

- Redesign focused on calculating performance metrics at LSST scale that is suitable for both DM development and commissioning needs.

Key features

Rubin
Observatory

- Ability to work on additional dataset types besides matched visits, e.g. coadd catalogs, DiaSources, metadata, etc.

- Scalable to larger datasets (easily parallelizable)

- Simple / straightforward to add new metrics

- Ability to build pipelines via configuration rather than code

- Persistence of the configuration with the measurements in the Butler repo with the dataset

- Enable multiple calculations of the same metric, but with different config parameters

- Enable selection of which metrics to calculate via configuration (via pipelines; validate_drp calculates "all" of its metrics each time)

Design concepts

- Gen 3 middleware

- Based on the MetricTask design described in DMTN-098: Metric Measurement Framework

- Each metric has an associated lsst.pipe.base.Task class that measures a metric based on data previously written to a Butler repository.

- Separate aggregation tasks combine data and/or intermediate results from individual datasets, e.g. across bands, or from individual patches into a tract, or detectors into a visit.

- Aggregations and MetricTasks are composed together in a pipeline, makes it easy to add new metrics and apply them to the Science Pipelines processing.

- Runs as an afterburner (as does valiate_drp)

Current status

- All metrics originally in validate_drp have been ported and implemented as MetricTasks in faro – PA1, PA2, PF1, AD1-3, AF1-3, AM1-3, and TE1-2

- Additionally more complex metrics, AB1 (requires multi-band data) and the stellar locus width metric (requires multi-band coadd catalogs) implemented

- Added the capability to calculate metrics at different scales (patch, tract)

- Several runs on all three tracts of RC2 (tract-by-tract rather than all at once). Lots of support from the middleware team - thank you!!

- Demonstrated near parity with validate_drp. Some remaining differences need to be understood.

- Package: https://github.com/lsst-dmsst/metric-pipeline-tasks
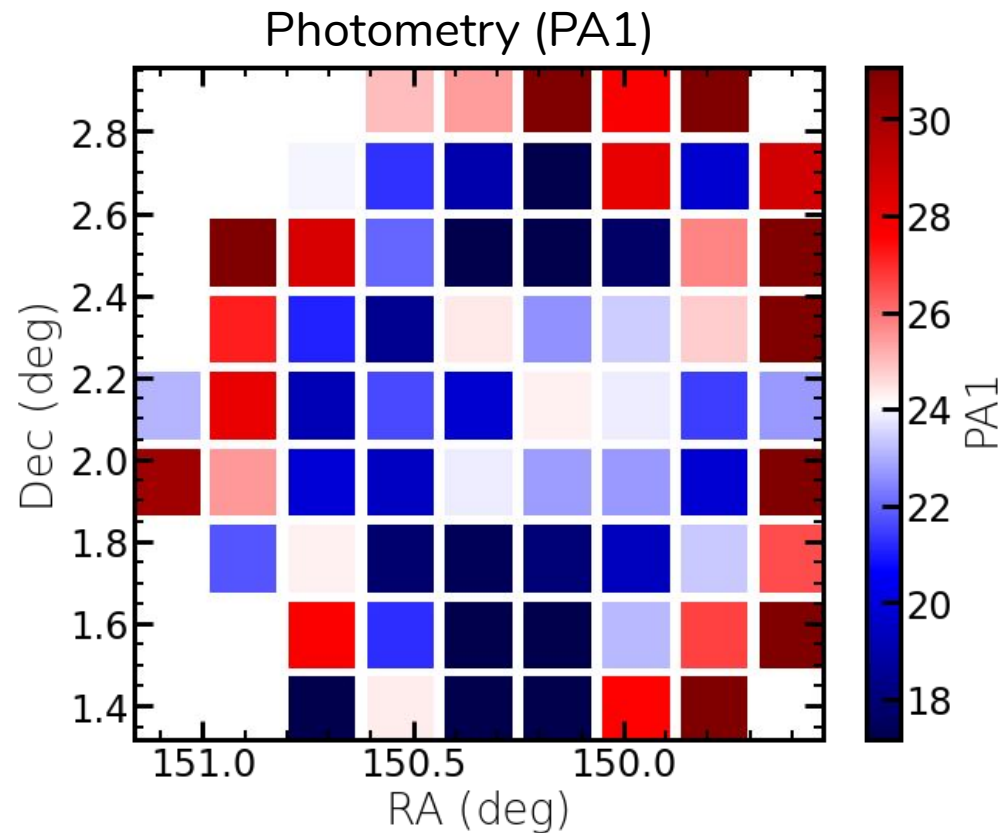
Comparison with validate_drp

Direct comparison of validate_drp ("Gen2") and faro ("Gen3")
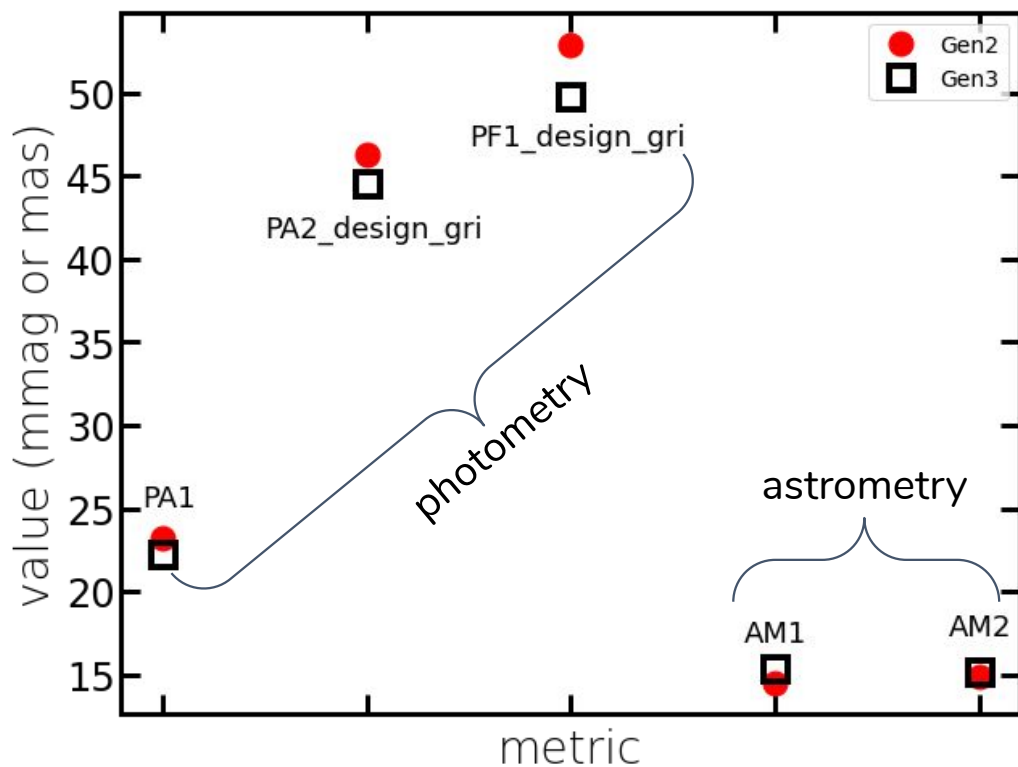
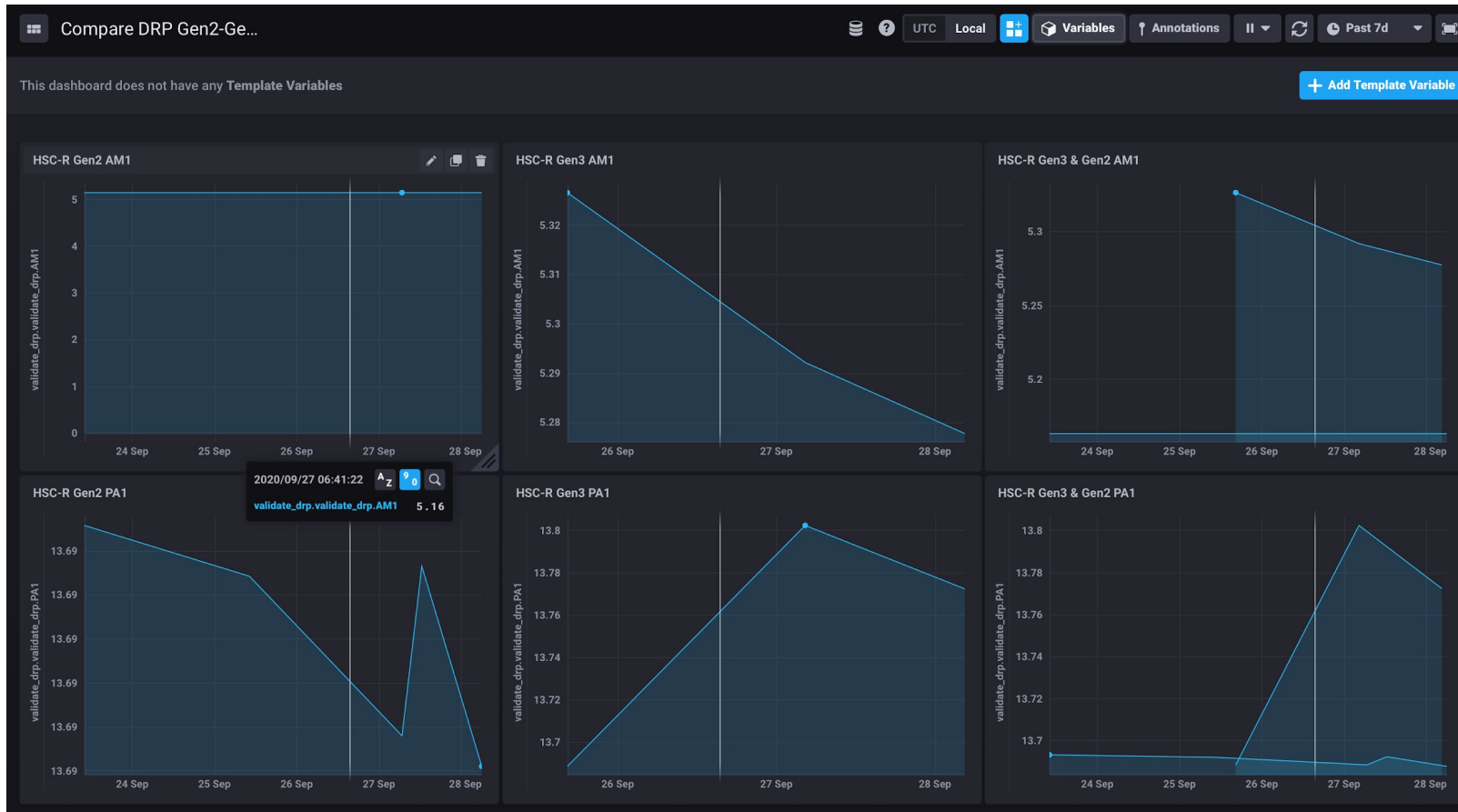Small differences probably mostly due to small differences in selected catalogs

HSC RC2, tract 9813, r-band (w36 processing).

Spatial dependence of metrics

Photometry (PA1)
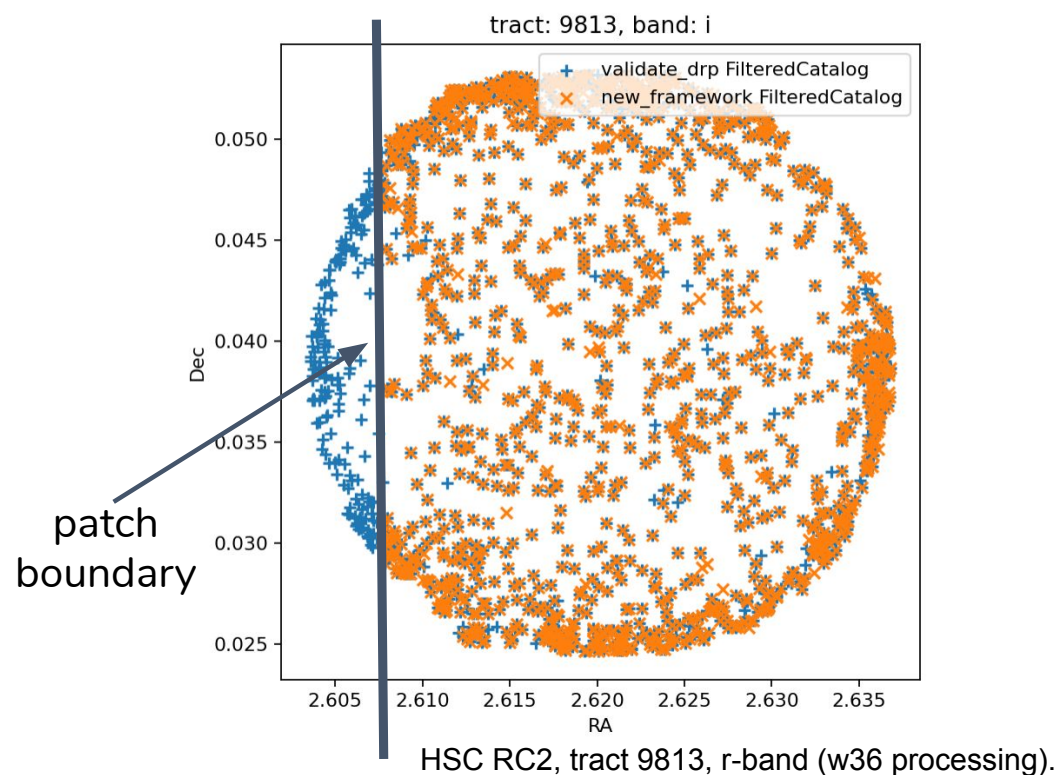
Can also measure and visualize the spatial dependence of metrics (e.g., per patch)

Integration with SQuaSH

Rubin
Observatory

Final steps before release

- Understand remaining minor differences between results from validate_drp and faro.

- Finalize the [package structure](#).

- Converge on class and metric naming conventions.

- Complete validation on the HSC RC2 dataset processed with v21.0.0.rc1.

- Improve documentation at class and pipeline level, how to run on a given repository, best practices for adding a measurement.

Future work

- Implement all remaining KPMs defined in the LSR/OSS/DMSR

- Review algorithms used to compute existing validate_drp metrics – metrics were ported as-is from validate_drp and never reviewed

- Improve the matching algorithm

- SQuaSH dashboard for all metrics à la validate_drp

- Comparison with external reference catalogs

- Consider the granularity at which we should calculate metrics – do we want to be attached to the spatial scales used for data processing (tracts, patches, etc.)?

- Near real-time analysis of data quality, e.g., to support commissioning.

- Ideas being collected at this [confluence](#) page