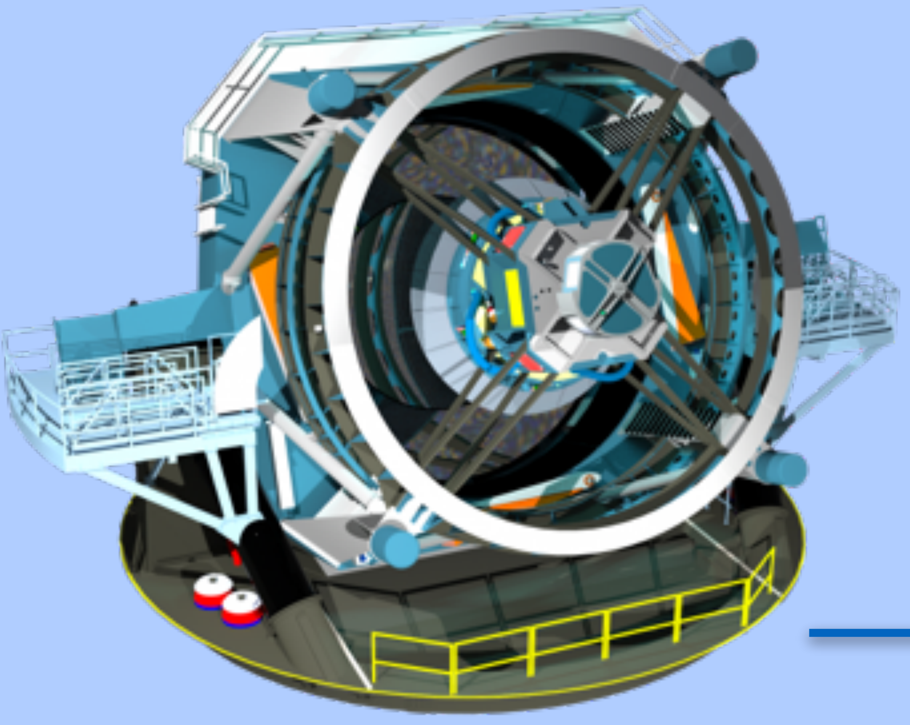
LSST Simulation Software Tools

Scott Daniel, University of Washington, Seattle
scottvalscott@gmail.com



LSST Simulations team

UW: Andy Conolly, Lynne Jones, Peter Yoachim, Rahul Biswas, Bryce Kalmbach

NOAO, Tucson: Kem Cook, Cathy Petry, Michael Reuter, Stephen Ridgway, Francisco Delgado

Purdue: John Peterson

LSST Simulation Software

- Summary website

  http://lsst.org/scientists/simulations/

- Installation instructions here

  https://confluence.lsstcorp.org/display/SIM/Catalogs+and+MAF

- Advantages:
  - Open source; everything is on github
  - Inter-dependent with Data Management software
  - Developed in close concert with the engineering teams

LSST Simulation Software

Two major goals of LSST Simulations

Question	**Tools to answer the question**
"What is the best way to schedule the survey to meet our science goals?"	OpSim (the operations simlator) MAF (the Metrics Analysis Framework)
"How well do our data analysis tools work?"	CatSim (the catalog simulator) ImSim (the image simulator)

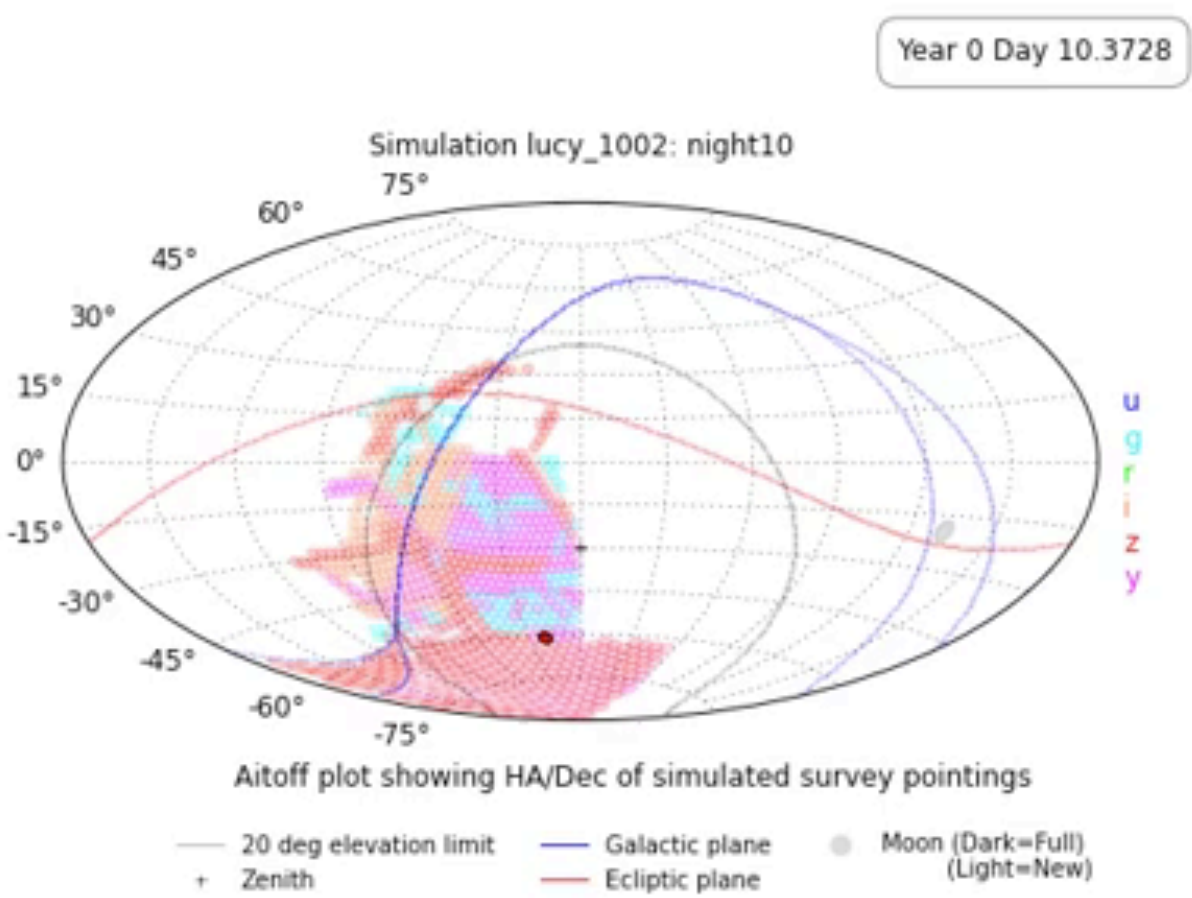
OpSim and MAF

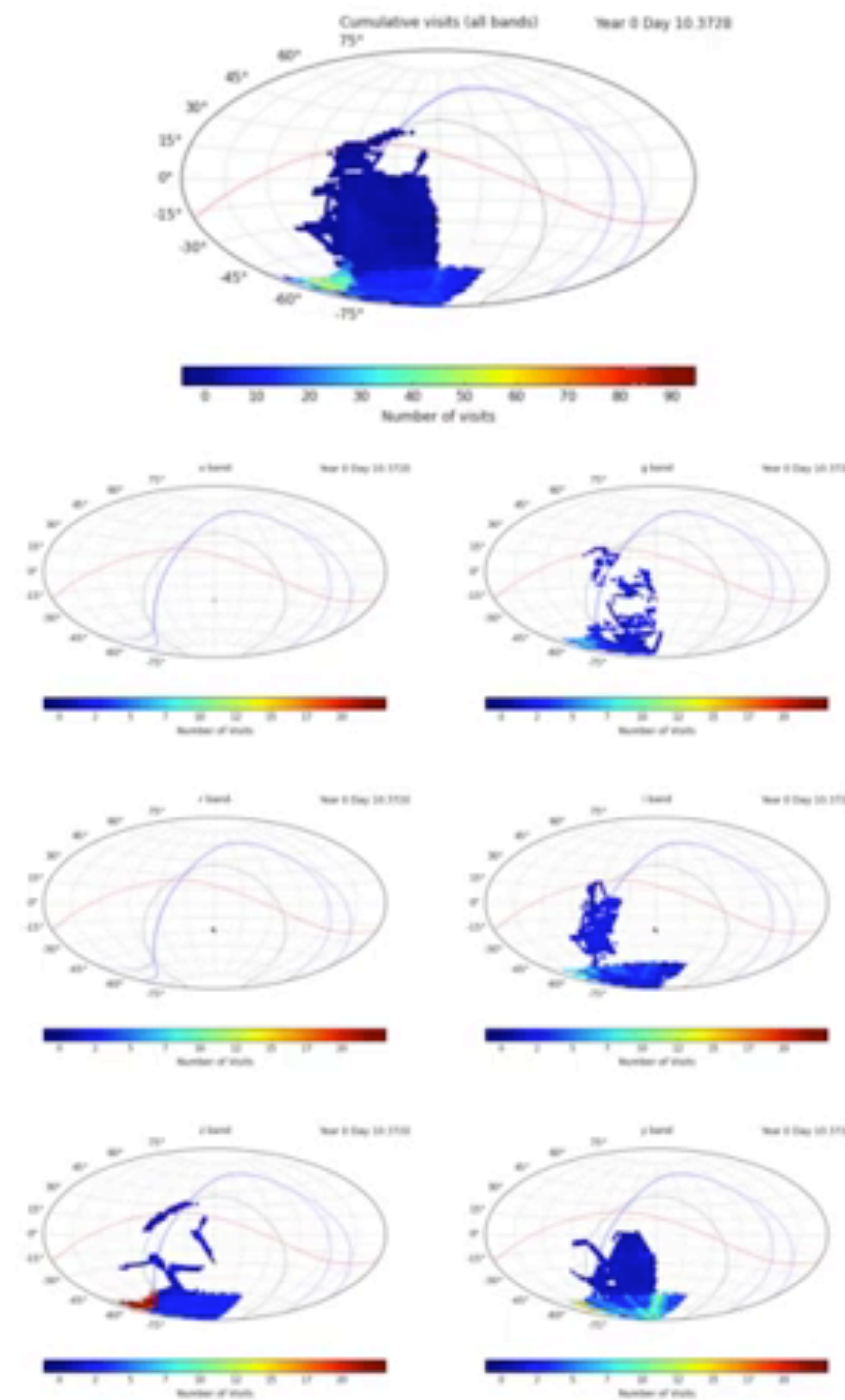
OpSim — The operations simulator

- Models the telescope-plus-scheduler system
- Includes all of the physical limitations of the telescope structure (cable wrap, dome orientation, maximum rotator speeds)
- Next version will use code that can actually serve as the LSST scheduling software.

Delgado et al (2014), SPIE 9150, doi:10.1117/12.2056898

OpSim and MAF





movie credit: Lynne Jones, https://www.youtube.com/watch?v=PqKsafmxNWA

OpSim and MAF

OpSim — The operations simulator

- Produces a SQLite database of all of the visits in a simulated ten-year survey.
- Benchmark runs can be downloaded from http://lsst.org/scientists/simulations/opsim/opsim-surveys-data-directory
- Suggestions for new runs to be made at https://github.com/LSSTScienceCollaborations/ObservingStrategy/blob/master/opsim/README.md

Delgado et al (2014), SPIE 9150, doi:10.1117/12.2056898

OpSim and MAF

- http://lsst.org/scientists/simulations/opsim/opsim-surveys-data-directory

OpSim Surveys Data Directory

This is an index of the sets of simulated surveys which are available to the LSST Project and to the scientific community. The "Description" column includes an explanation of the purpose of each dataset and links to relevant information.

A white paper is being written by the community outlining various science cases and the impacts that observing strategy ("cadence") will have on them, quantified using the Metric Analysis Framework. Check out the Observing Strategy github repository for more information on how to get involved.

Publish Date	Name	Description
Jan 2016	OpSim v3.3.5 Benchmark runs	A set of 14 10-year surveys designed to explore various combinations of science programs and executed with OpSim v3.3.5. This "benchmark" set is created for every release of OpSim. This dataset is being vetted and when approved will replace the OpSim v3.3.2 Benchmark run set. Please note that the OpSim v3.3.5 Summary table schema has changed since OpSim v3.3.2.
Aug 2015	OpSim v3.3.2 Benchmark runs	A set of 14 10-year surveys designed to explore various combinations of science programs and executed with OpSim v3.3.2. This "benchmark" set is created for every release of OpSim. This dataset was presented at the LSST2015 Project and Community Workshop held in Bremerton, WA, Aug 2015, and contains the candidate reference run enigma_1189. This is the description of the OpSim v3.3.2 Summary table schema.

Print

OpSim and MAF

OpSim — The operations simulator

- Produces a SQLite database of all of the visits in a simulated ten-year survey.
- Benchmark runs can be downloaded from http://lsst.org/scientists/simulations/opsim/opsim-surveys-data-directory
- Suggestions for new runs to be made at https://github.com/LSSTScienceCollaborations/ObservingStrategy/blob/master/opsim/README.md

Delgado et al (2014), SPIE 9150, doi:10.1117/12.2056898

OpSim and MAF

https://community.lsst.org/c/sims/sims-announce

LSST Community

New Opsim Run: Galactic Plane observed at WFD cadence

Simulations   Sims Announcements   opsim

ljones Lynne Jones LSST   2d

In response to a request from the Observing Cadences working group (https://github.com/LSSTScienceCollaborations/ObservingStrategy/blob/master/opsim/Proposal_GP.md)
a simulated LSST survey where the galactic plane was observed at a normal WFD cadence has been completed.

It is currently available from links at
http://astro-lsst-01.astro.washington.edu:8080/ 2

The galactic plane run is called "astro_lsst_01_1004" (sorry for the terrible names).
The link to download the sqlite file is:
http://astro-lsst-01.astro.washington.edu:8080/db_gzip/astro_lsst_01_1004_sqlite.db.gz

The standard MAF analyses for this run are also available at:
http://astro-lsst-01.astro.washington.edu:8080/metricSelect?runId=3 1
and
http://astro-lsst-01.astro.washington.edu:8080/metricSelect?runId=4

OpSim and MAF

MAF — The Metrics Analysis Framework

- Framework for analyzing simulated surveys generated by OpSim
- Meant to answer the question "what does a survey need to do to be 'good enough' for X science goal?"
- Needs community interaction and contribution
  http://github.com/LSST-nonproject/sims_maf_contrib

Jones et al (2014), SPIE 9149, doi:10.1117/12.2056835

OpSim and MAF

MAF — The Metrics Analysis Framework

MAF

OpSim

Database of simulated observations

→

group OpSim observations according to specified “slicer” (position on the sky, MJD, etc.)

→

calculate user-provided metric at each of the “slice points”

Jones et al (2014), SPIE 9149, doi:10.1117/12.2056835

OpSim and MAF

Large Synoptic Survey Telescope

```
import numpy as np
import lsst.sims.maf.db as db
import lsst.sims.maf.metrics as metrics
import lsst.sims.maf.slicers as slicers
import lsst.sims.maf.metricBundles as metricBundles
import time


class MeanRevisitTimeMetric(metrics.BaseMetric):
    """
    Calculate the mean time between consecutive visits at this grid point
    """
    def __init__(self, dateCol='expMJD', metricName='MeanRevisitTime',
                 **kwargs):

        self.dateCol = dateCol
        super(MeanRevisitTimeMetric, self).__init__(col=dateCol,
                                                     metricName=metricName,
                                                     **kwargs)

    def run(self, dataSlice, slicePoint=None):
        obs_time = np.sort(dataSlice, order=self.dateCol)
        obs_time_diff = np.sort(np.diff(obs_time[self.dateCol]))
        return obs_time_diff.mean()


if __name__ == "__main__":

    opsdb = db.OpsimDatabase('data/kraken_1042_sqlite.db')

    slicer= slicers.OpsimFieldSlicer()

    sqlconstraint = 'night<1095'

    start = time.clock()

    meanMetric = MeanRevisitTimeMetric()
    meanBundle = metricBundles.MetricBundle(meanMetric, slicer, sqlconstraint)

    group = metricBundles.MetricBundleGroup({0:meanBundle}, opsdb,
                                            outDir='test_output_mean', resultsDb=None)
    group.runAll()
    meanBundle.setPlotDict({'colorMin':0, 'colorMax':8.0})
    group.plotAll()
    print 'that took ',time.clock()-start
```

OpSim and MAF

```
import numpy as np
import lsst.sims.maf.db as db
import lsst.sims.maf.metrics as metrics
import lsst.sims.maf.slicers as slicers
import lsst.sims.maf.metricBundles as metricBundles
import time
```

OpSim and MAF

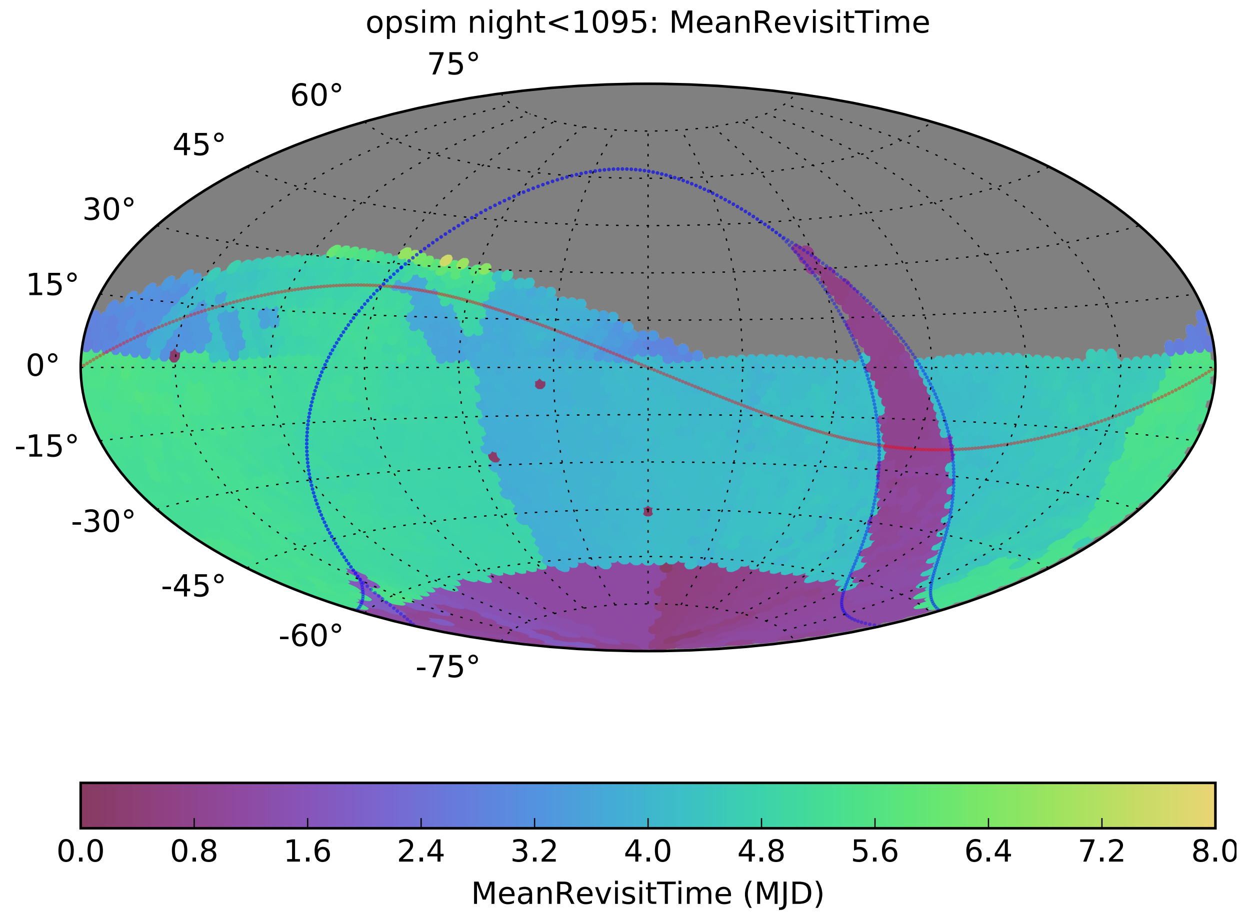
```
if __name__ == "__main__":

    opsdb = db.OpsimDatabase('data/kraken_1042_sqlite.db')

    slicer= slicers.OpsimFieldSlicer()

    sqlconstraint = 'night<1095'

    start = time.clock()

    meanMetric = MeanRevisitTimeMetric()
    meanBundle = metricBundles.MetricBundle(meanMetric, slicer, sqlconstraint)

    group = metricBundles.MetricBundleGroup({0:meanBundle}, opsdb,
                                            outDir='test_output_mean', resultsDb=None)
    group.runAll()
    meanBundle.setPlotDict({'colorMin':0, 'colorMax':8.0})
    group.plotAll()
    print 'that took ',time.clock()-start
```

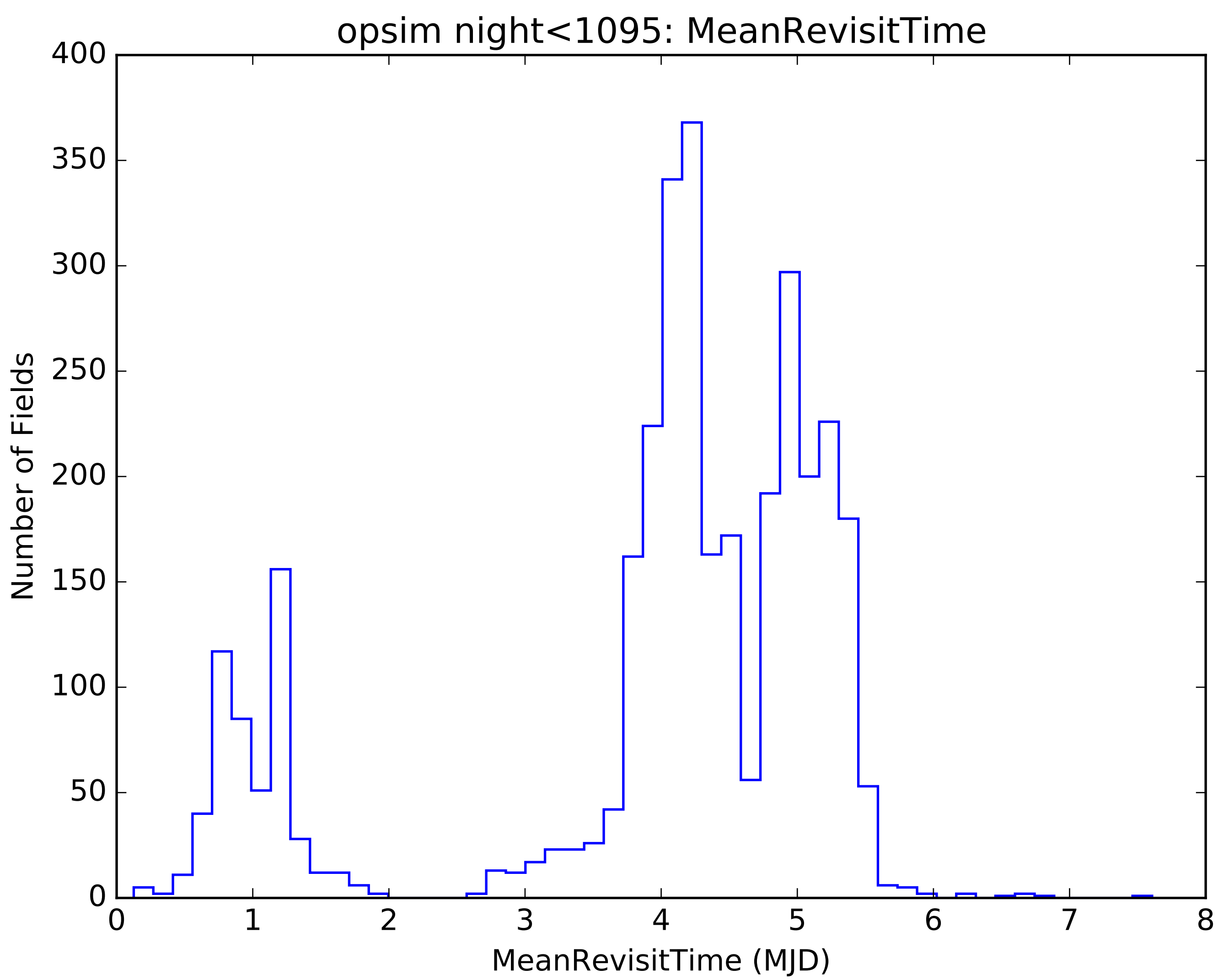
OpSim and MAF

```python
class MeanRevisitTimeMetric(metrics.BaseMetric):
    """
    Calculate the mean time between consecutive visits at this grid point
    """
    def __init__(self, dateCol='expMJD', metricName='MeanRevisitTime',
                 **kwargs):

        self.dateCol = dateCol
        super(MeanRevisitTimeMetric, self).__init__(col=dateCol,
                                                     metricName=metricName,
                                                     **kwargs)

    def run(self, dataSlice, slicePoint=None):
        obs_time = np.sort(dataSlice, order=self.dateCol)
        obs_time_diff = np.sort(np.diff(obs_time[self.dateCol]))
        return obs_time_diff.mean()
```

OpSim and MAF

MAF — The Metrics Analysis Framework





OpSim and MAF

MAF — The Metrics Analysis Framework

- Precomputed metrics available for OpSim runs at

  http://astro-lsst-01.astro.washington.edu:8080
  http://ops2.tuc.noao.edu:8888/

- Tutorials for running MAF on your own (and contributing metrics to the community) at

  http://github.com/LSST-nonproject/sims_maf_contrib

Jones et al 2014, SPIE, doi:10.1117/12.2056835

OpSim and MAF

http://astro-lsst-01.astro.washington.edu:8080

List of all Opsim Runs

Run List | Opsim Configuration | Metrics List | All Results | Multi Color | Summary Stats

OpsimRun	MafComment	OpsimComment	SQLite File	MafDir	ResultsDb	OpsimDate
astro_lsst_01_1000	Scheduler	Baseline 10 year run	astro_lsst_01_1000_sqlite.db.gz	/local/lsst/opsim /astro_lsst_01_1000 /scheduler	ResultsDb	04/08/16
astro_lsst_01_1000	Science	Baseline 10 year run	astro_lsst_01_1000_sqlite.db.gz	/local/lsst/opsim /astro_lsst_01_1000 /science	ResultsDb	04/08/16
astro_lsst_01_1004	Scheduler	Tier2: Normal galactic plane	astro_lsst_01_1004_sqlite.db.gz	/local/lsst/opsim /astro_lsst_01_1004 /scheduler	ResultsDb	04/16/16
astro_lsst_01_1004	Science	Tier2: Normal galactic plane	astro_lsst_01_1004_sqlite.db.gz	/local/lsst/opsim /astro_lsst_01_1004 /science	ResultsDb	04/16/16
astro_lsst_01_1006	Scheduler	15 year NEO baseline run with NES==WFD (neo/run01)	astro_lsst_01_1006_sqlite.db.gz	/local/lsst/opsim /astro_lsst_01_1006 /scheduler	ResultsDb	04/18/16
astro_lsst_01_1006	Science	15 year NEO baseline run with NES==WFD (neo/run01)	astro_lsst_01_1006_sqlite.db.gz	/local/lsst/opsim /astro_lsst_01_1006 /science	ResultsDb	04/18/16

OpSim and MAF

http://astro-lsst-01.astro.washington.edu:8080

OpSim Run: astro_lsst_01_1000

Run List	Opsim Configuration	Metrics List	All Results	Multi Color	Summary Stats

Plot Metrics

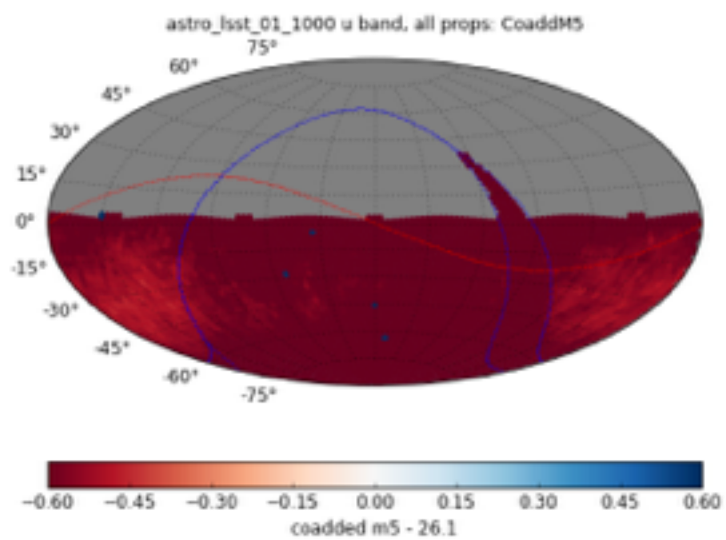
1. A: Summary
   - 1: NVisits
   - 2: On-sky Time
   - 3: Obs Per Night
2. B: Completeness
   - All Props
   - WFD
3. C: NVisits
   - All Props
   - All Props, ratio
   - DD
   - WFD
   - WFD, ratio
4. D: NVisits (per prop)
   - DDcosmology

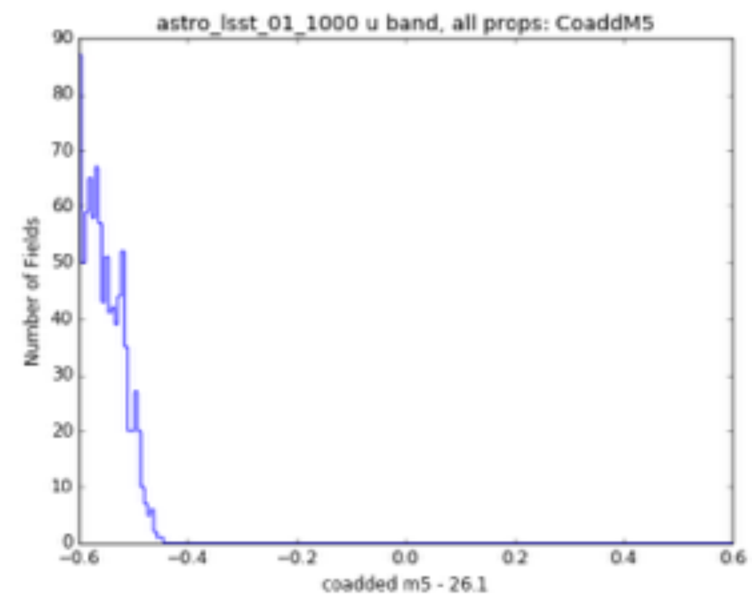
☐	NVisits Per Proposal	OpsimFieldSlicer	i band, Universal	npz JSON
☐	NVisits Per Proposal	OpsimFieldSlicer	z band, Universal	npz JSON
☐	NVisits Per Proposal	OpsimFieldSlicer	y band, Universal	npz JSON
☐	NVisits Per Proposal	OpsimFieldSlicer	u, g, r, i, z, y band Universal	npz JSON
☐	All Group: *E: Coadded depth*; Subgroup: *All Props*			
☑	CoaddM5	OpsimFieldSlicer	u band, all props	npz JSON
☐	CoaddM5	OpsimFieldSlicer	g band, all props	npz JSON
☐	CoaddM5	OpsimFieldSlicer	r band, all props	npz JSON
☐	CoaddM5	OpsimFieldSlicer	i band, all props	npz JSON
☐	CoaddM5	OpsimFieldSlicer	z band, all props	npz JSON
☐	CoaddM5	OpsimFieldSlicer	y band, all props	npz JSON
☐	CoaddM5	OpsimFieldSlicer	u, g, r, i, z, y band all props	npz JSON
☐	All Group: *E: Coadded depth*; Subgroup: *DD*			

OpSim and MAF

http://astro-lsst-01.astro.washington.edu:8080

CoaddM5 OpsimFieldSlicer u band, all props npz JSON





Coadded depth in filter u, with design value subtracted (26.1), for all proposals. More positive numbers indicate fainter limiting magnitudes.

Median:	25.44
Mean:	25.28
Rms:	0.43
N(-3Sigma):	13.00
N(+3Sigma):	5.00
Count:	2817
25th%ile:	25.38
75th%ile:	25.50
Min:	23.88
Max:	27.94

LSST Simulation Software

Two major goals of LSST Simulations

Question	**Tools to answer the question**
"What is the best way to schedule the survey to meet our science goals?"	OpSim (the operations simlator) MAF (the Metrics Analysis Framework)
"How well do our data analysis tools work?"	CatSim (the catalog simulator) ImSim (the image simulator)

CatSim and ImSim

CatSim — the catalog simulator

Databases hosted at University of Washington ("fatboy")

Individual user's system

CatSim and ImSim

CatSim — the catalog simulator

Databases hosted at University of Washington ("fatboy")

For more detail on database contents, see

http://lsst.org/scientists/simulations/catsim

Contains

- **Milky Way stars** drawn from models in Juric, et al (2008) ApJ 673,864
- **Galaxy distribution** drawn from De Lucia, et al (2006) MNRAS 366, 499
- **Distribution of Solar System objects** from Grav et al (2007) BAAS 211, 4721

Connolly et al (2010) SPIE 7738, doi:10.1117/12.857819
Connolly et al (2014) SPIE 9150, doi:10.1117/12.2054953

CatSim and ImSim

Large Synoptic Survey Telescope

CatSim — the catalog simulator

Individual user's system

Contains

- **Galaxy SEDs** from Bruzual and Charlot (2003) MNRAS 344, 1000
- **Main sequence stellar SEDs** from Kurucz (1993) http://www.stsci.edu/hst/observatory/crds/k93models.html
- **M, L, T dwaf star spectra** from https://phoenix.ens-lyon.fr/Grids/BT-Settl/CIFIST2011/SPECTRA/
- **White dwarf SEDs** from Bergeron et al (1995) http://www.astro.umontreal.ca/~bergeron/CoolingModels/
- **Solar system object SEDs** from http://sbn.psi.edu/pds/resource/busdemeotax.html
- **Dust maps** from Schelegel, Finkbeiner, and Davis (1998) ApJ 500, 525

Connolly et al (2010) SPIE 7738, doi:10.1117/12.857819
Connolly et al (2014) SPIE 9150, doi:10.1117/12.2054953

CatSim and ImSim

Large Synoptic Survey Telescope

CatSim — the catalog simulator

Individual user's system

Contains

- Official LSST bandpass throughput curves
- Code associating objects in the database with SEDs
- Variability models for RR Lyrae, AGNs, Cepheids,
  M-dwarf Flares, Type 1a Supernovae, Solar System Objects
- Representation of the LSST Camera and code to convert position on the sky to position on the LSST focal plane and detectors.

Connolly et al (2010) SPIE 7738, doi:10.1117/12.857819
Connolly et al (2014) SPIE 9150, doi:10.1117/12.2054953

CatSim and ImSim

LSST
Large Synoptic Survey Telescope

CatSim — the catalog simulator

1) User specifies catalog parameters

2) CatSim translates parameters into SQL query

Databases hosted at University of Washington (“fatboy”)

3) Database performs query and returns results

4) CatSim performs any necessary post-processing, e.g.

- associating objects with SEDs
- interpolating variability curves
- correcting mean positions

Individual user’s system

CatSim and ImSim

```
import numpy as np
from lsst.sims.utils import ObservationMetaData
from lsst.sims.catalogs.measures.instance import InstanceCatalog
from lsst.sims.catUtils.baseCatalogModels import GalaxyTileObj
import time

class GalaxyMagCatalog(InstanceCatalog):
    column_outputs = ['raJ2000', 'decJ2000',
                      'u_ab', 'g_ab', 'r_ab', 'i_ab', 'z_ab', 'y_ab',
                      'redshift']

    transformations = {'raJ2000':np.degrees,
                       'decJ2000':np.degrees}


if __name__ == "__main__":

    db = GalaxyTileObj()
    obs = ObservationMetaData(pointingRA=27.0, pointingDec=-45.0,
                              boundType='circle', boundLength=0.1)

    start = time.clock()
    cat = GalaxyMagCatalog(db, obs_metadata=obs)
    cat.write_catalog('test_catalog.txt', chunk_size=100000)
    print 'that took ',time.clock()-start
```

CatSim and ImSim

#raJ2000	decJ2000	u_ab	g_ab	r_ab	i_ab	z_ab	y_ab	redshift
26.9953	-45.0481	28.2687	27.9767	27.3195	26.4862	26.2489	26.0735	0.7332
26.9948	-45.0488	28.0196	27.4919	26.5578	25.7799	25.5113	25.2977	0.6565
26.9923	-45.0506	30.7064	28.2468	27.6789	27.9670	27.9089	27.4581	2.7189
26.9938	-45.0504	27.5876	27.4594	26.9958	26.0569	25.5093	24.7785	1.3273
26.9939	-45.0478	27.4205	27.0458	26.3485	25.4178	25.0090	24.8036	0.8091
26.9937	-45.0469	25.5330	25.0049	25.2653	25.0002	24.7751	24.6925	1.9867
26.9914	-45.0495	26.1839	26.1352	26.1873	26.2495	26.2809	26.2121	2.3111
26.9910	-45.0503	29.4688	28.7444	27.8895	27.1393	26.9575	26.8172	0.6640
26.9912	-45.0495	27.0218	27.1349	27.0370	26.8228	26.4035	25.9359	1.2933
26.9921	-45.0480	27.7873	27.1284	27.1812	27.0874	26.8975	26.8355	2.0428
26.9920	-45.0490	24.4159	24.4594	24.4897	24.3129	24.1638	23.7028	1.4996
26.9926	-45.0484	26.6846	26.7278	26.4549	25.7427	25.4522	25.3449	0.8233
26.9934	-45.0491	26.8667	27.1147	26.6418	26.4394	26.0376	25.8882	1.1307
26.9953	-45.0408	25.6862	25.7257	25.7702	25.5944	25.4134	24.8792	1.4340
26.9933	-45.0420	28.0681	28.0773	27.5245	27.3286	27.3158	27.3060	0.5460
26.9931	-45.0413	27.5987	27.7481	27.1087	26.5860	25.8231	25.3891	1.1820
26.9925	-45.0420	25.8358	25.3745	24.5995	24.3592	24.2541	24.2011	0.5127
26.9934	-45.0421	27.9149	27.9294	27.7620	27.3951	26.8563	26.3443	1.2728
26.9938	-45.0411	27.5440	27.5790	27.4829	27.1559	26.8320	26.1948	1.3831
26.9942	-45.0414	27.1948	25.5783	23.9754	22.8773	22.3265	22.0777	0.7831
26.9882	-45.0451	27.4685	28.0546	27.3441	27.4455	27.9248	27.5175	2.9973
26.9946	-45.0436	24.5397	24.5719	24.4329	24.0631	23.4216	23.2204	1.0728

- Detailed tutorials can be found in https://github.com/uwssg/LSST-Tutorials/tree/master/CatSim

CatSim and ImSim

ImSim — the image simulator

- PhoSim — Peterson et al (2015) ApJS 218.1, 14
  arXiv:1504.06570
    - Project-developed tool https://bitbucket.org/phosim/
    - Full photon Monte-Carlo representation of physics
    - Slow but detailed
- GalSim — Rowe et al arXiv:1407.7676
    - Community developed:
      http://github.com/GalSim-Developers/galsim
    - Noise, PSF, and other physical effects implemented by analytic models
    - Fast but approximate

CatSim and ImSim

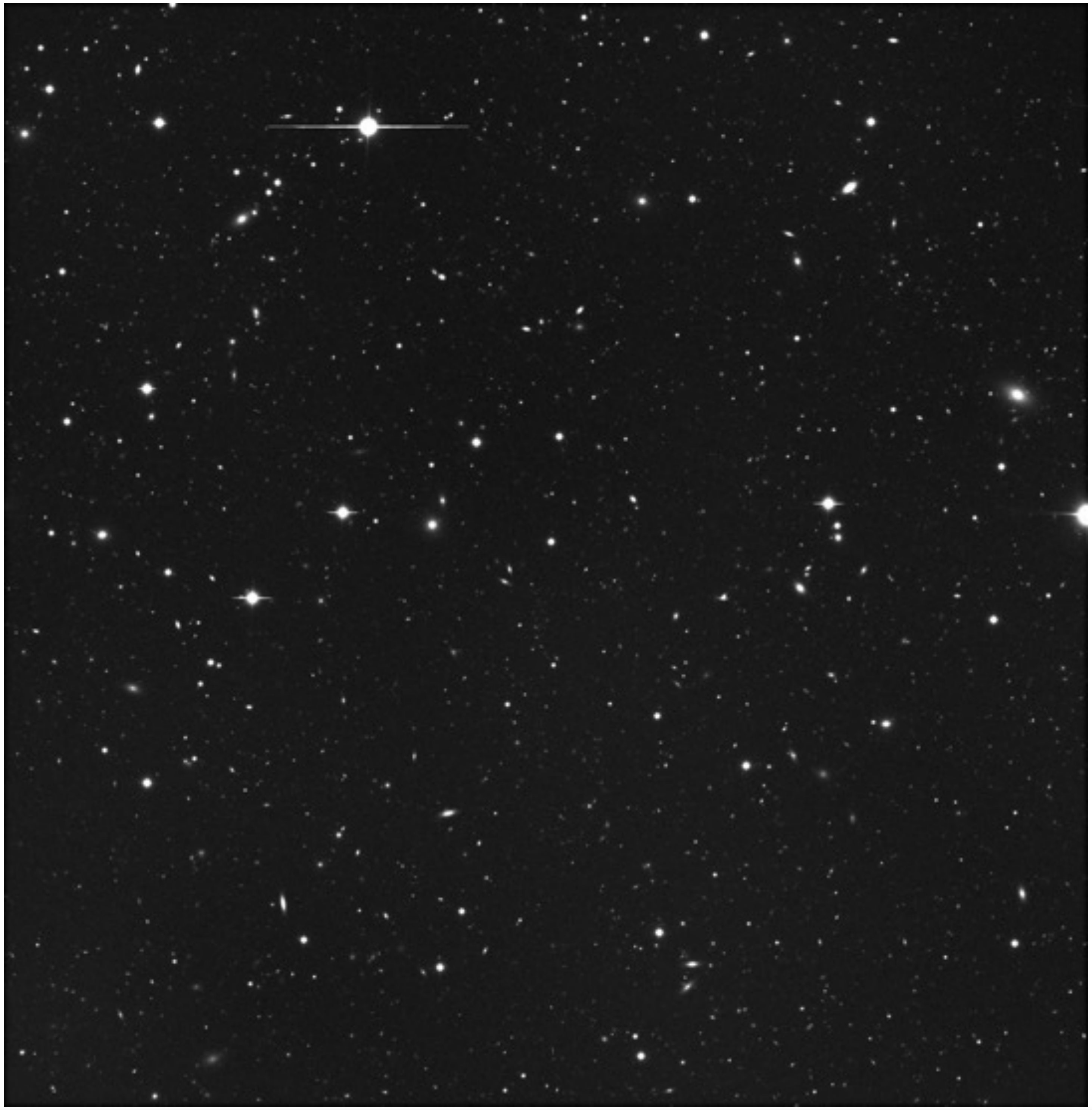
ImSim — the image simulator

- PhoSim — Peterson et al (2015) ApJS 218.1, 14
  arXiv:1504.06570
    - CatSim produces an 'Instance Catalog' (text file)
    - PhoSim reads that in and produces image as FITS file

- GalSim — Rowe et al arXiv:1407.7676
    - Seamless integration with CatSim in sims_GalSimInterface package

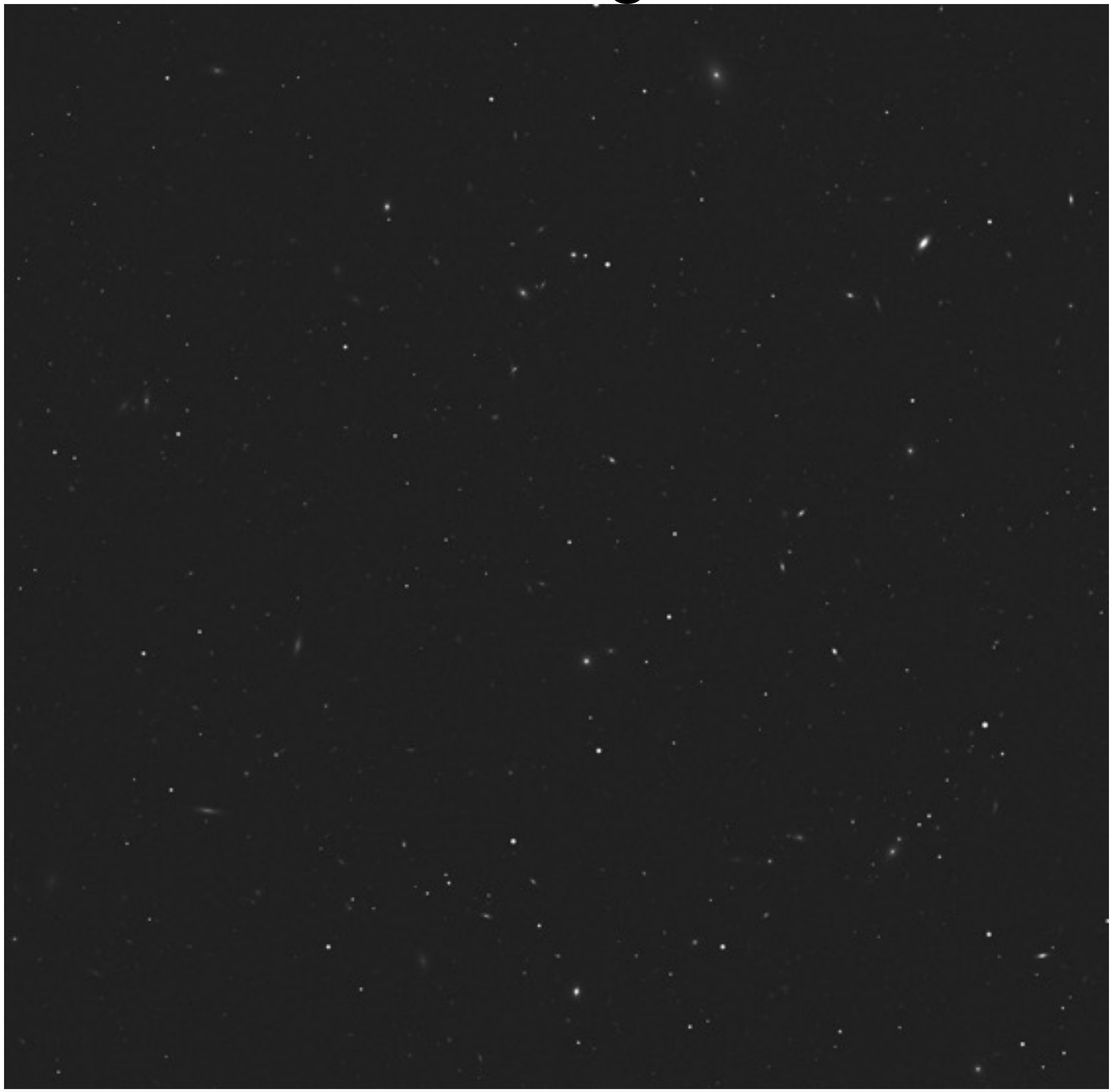
CatSim and ImSim

PhoSim image



CatSim and ImSim

GalSim Image



LSST Simulation Software

- Summary website

  http://lsst.org/scientists/simulations/

- Installation instructions here

  https://confluence.lsstcorp.org/display/SIM/Catalogs+and+MAF

- Includes tools to:
  - Simulate the LSST Survey (OpSim)
  - Evaluate the perfomance of a simulated survey (MAF)
  - Draw simulated catalogs from a simulated universe (CatSim)
  - Simulate LSST-like images (PhoSim/GalSim)