LINCC - developing software for large-scale analysis of time domain data

LSST

SCHMIDT FUTURES

DiRAC

The LINCC Frameworks Project

LSST Interdisciplinary Network For Collaboration And Computing

- A collaboration between UW, CMU, LSSTC, U Pitt, and NOIRLab to build software systems for key LSST science

- PIs: Andy Connolly (UW), Rachel Mandelbaum (CMU)
- Director of Engineering: Jeremy Kubica (CMU)

- **Science** software infrastructure: combining user algorithms & code, astro packages, and industry tools to build scalable science analysis packages

https://www.lsstcorporation.org/lincc/frameworks

LINCC Frameworks Mission

LINCC Frameworks mission is to enable scientists by developing scalable and productionised software/algorithms in collaboration with broader community.

We want to:

- be engineering and algorithmically focused,
- collaborate with other software efforts (projects may be contributions to existing code bases),
- leverage existing tools (build on top of the Rubin Science Platform and standard community tools/libraries), and
- coordinate with community to avoid unnecessary duplication of effort.

LINCC Frameworks Collaboration

Additional venues for collaboration:

- **Workshops** - Work with LSST Science Collaborations to identify areas of need.
    - Data to Software to Science Workshop (March 2022)
- **Incubators** - Scientists work with team to get their science applications working (open proposal process).
- **Tech talks** - showcase work done by the broad Rubin software and archives community that's designed to enable LSST science
- LINCC Frameworks members joining LSST science collaborations.

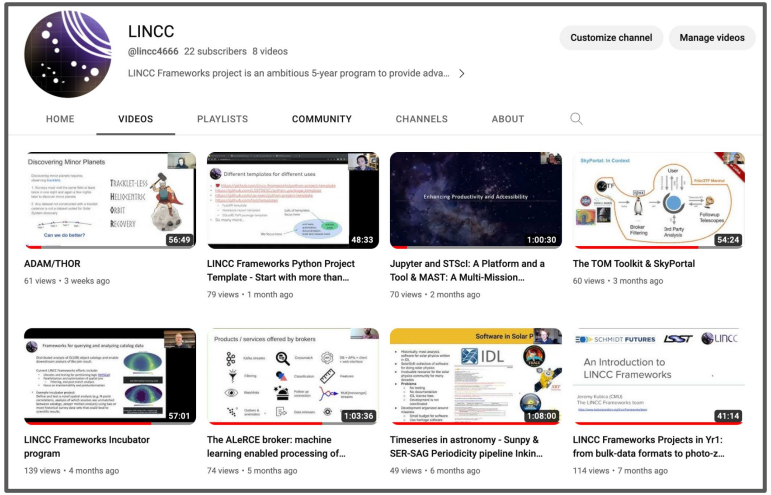
Incubators

- Incubators provide support for researchers to work directly with LINCC Frameworks team to apply new tools to research problems.

- Goal: Establish long-term software development collaborations that serve both the selected teams and LINCC Frameworks.
- 2nd proposal deadline was June 15, 2023, next one is expected October

- **Incubator #1 - Solar System Simulation**
  - Meg Schwamb at Queen's University Belfast
- **Incubator #2 - Supernova Classification**
  - Kaylee De Soto at Penn State (Session 3A)

Tech talks



- Talks that showcase the work done by the broad Rubin software and archives community.

- So far: Brokers, in-kind contributors, data centers, LINCC engineers…
- Future: Addition of talks by Roman software group & results from incubators

2nd Thursday of the month, 10 am Pacific at

https://ls.st/lincc-talks

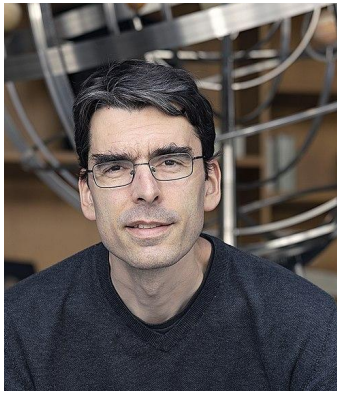
Overview of the current time domain effort in LINCC

- **Astropy contributions** (multiband periodograms, implemented in the latest Astropy 5.13)

- Tape (Timeseries Analysis & Processing Engine)
  - **Enable scaling** of external functions (via dask)
  - Internal functions - **filling the gaps** that currently exist
    - E.g., structure function calculations
  - **Widely useful algorithms**
    - E.g., color correction, custom ``ubercal`` of individual objects
  - Aggregating **brokers** results

- LSDB development (how to crossmatch, work with large catalogs, integrate with time domain)

LSDB - Quick Project Recap



- This decade is marked by many projects producing large datasets. **Rubin's Year #1 dataset will be O(100TB) in size.**

- A major use case for these data is **whole-dataset science (statistics, searching, mapping)**. Examples: variable star classification, time-domain analyses, dust distribution, Milky Way, large-scale structure, etc.

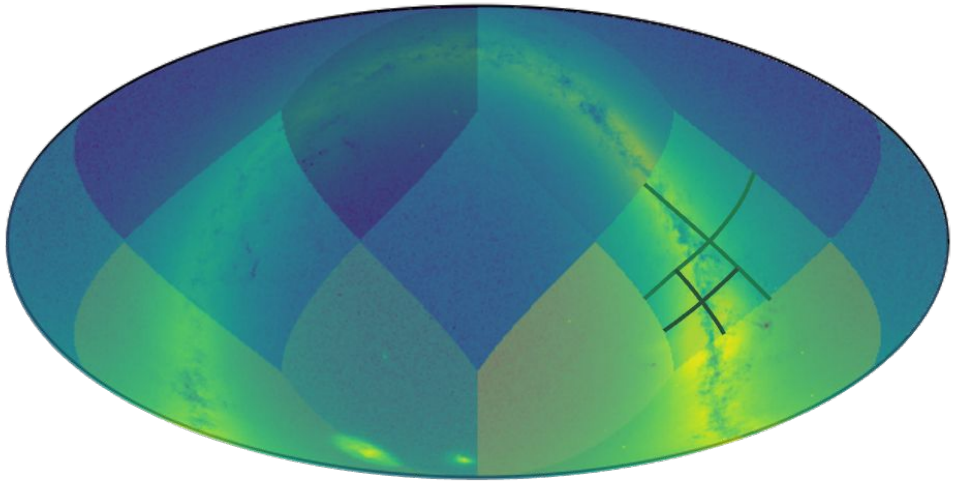
- **Few ready-to-use tools exist today for such work, and the way data are distributed hinder it.** Barrier to entry is very high.

- Objectives: **Develop tools that enable scientists to conduct large-scale analytics on multiple datasets (LSDB). Develop formats that enable dataset providers to expose their dataset to such analyses (HiPSCat).**

The Format: HiPSCat

Details: https://dirac.us/teq

- A hierarchical data storage scheme, where the sky is hierarchically split into HEALPix tiles until each tile has roughly a similar number of objects (rows).
- These tiles are stored as Parquet files within a directory tree that encodes their location on the sky.



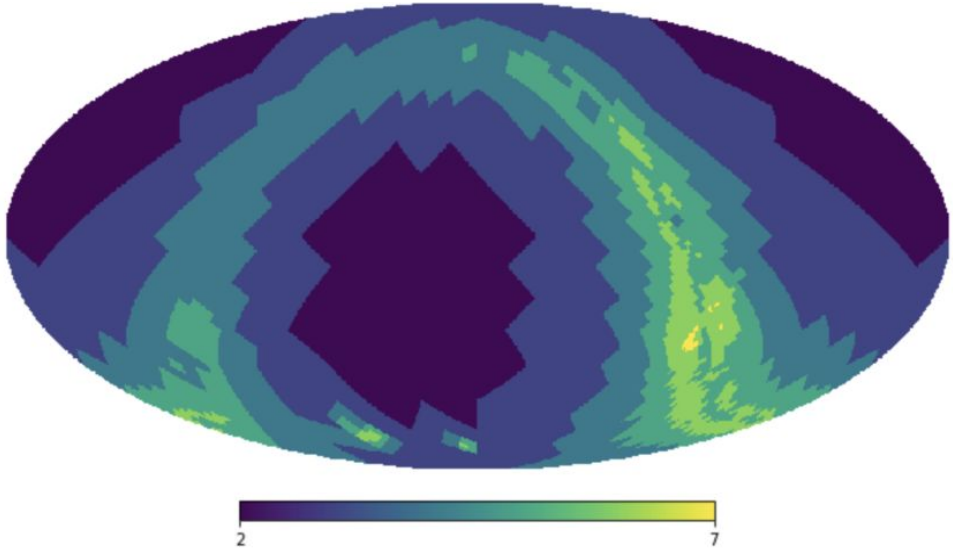
```
Norder=0/Dir=0/Npix=0/catalog.parquet
…
Norder=1/Dir=0/Npix=28/catalog.parquet
Norder=1/Dir=0/Npix=29/catalog.parquet
Norder=1/Dir=0/Npix=30/catalog.parquet
Norder=2/Dir=0/Npix=112/catalog.parquet
Norder=2/Dir=0/Npix=113/catalog.parquet
Norder=2/Dir=0/Npix=114/catalog.parquet
Norder=2/Dir=0/Npix=115/catalog.parquet
…
Norder=0/Dir=0/Npix=11/catalog.parquet
```

Enables: fast spatial lookup, distributed analytics, distributed joining and cross-matching. Based on 10+ yrs of thinking/experience/experimentation (LSD + AXS).

The Format: HiPSCat

Details: https://dirac.us/teq

- A hierarchical data storage scheme, where the sky is hierarchically split into HEALPix tiles until each tile has roughly a similar number of objects (rows).
- These tiles are stored as Parquet files within a directory tree that encodes their location on the sky.

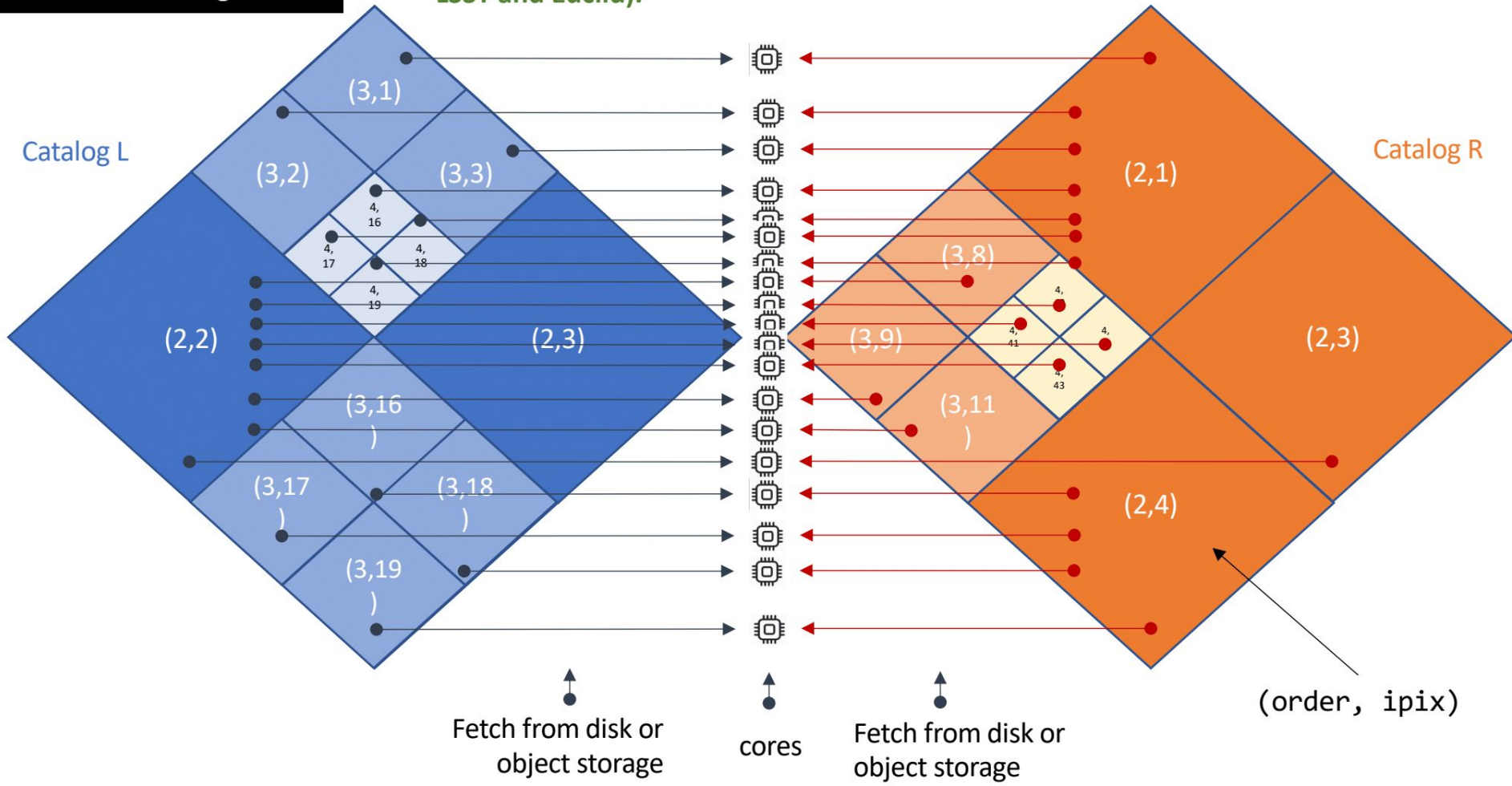



```
Norder=1/Dir=0/Npix=28/catalog.parquet
Norder=1/Dir=0/Npix=29/catalog.parquet
Norder=1/Dir=0/Npix=30/catalog.parquet
Norder=2/Dir=0/Npix=112/catalog.parquet
Norder=2/Dir=0/Npix=113/catalog.parquet
Norder=2/Dir=0/Npix=114/catalog.parquet
Norder=2/Dir=0/Npix=115/catalog.parquet
…
Norder=0/Dir=0/Npix=11/catalog.parquet
```

Enables: fast spatial lookup, distributed analytics, distributed joining and cross-matching. Based on 10+ yrs of thinking/experience/experimentation (LSD + AXS).

Efficient, parallel, joins and crossmatching

Use case #4: distributed analysis on data from two catalogs (example: LSST and Euclid).



The Tool: Large Survey Database (LSDB)

- HiPSCat is a valid Parqued (partitioned) dataset. **Parquet tools can use it out-of-the-box.**
- But a tool with full HiPSCat awareness can enable spatial queries, cross-matching, timeseries, and efficient multi-dataset joining. **Enter LSDB.**
- **LSDB: Pandas-like analysis of astronomical datasets with trillions of observations using thousands of cores.**
- **Build on existing tools and ecosystem.** Presently Dask (may look at Ray).

```
img = gaia
      .query("pm > 10")
      .crossmatch(ztf)
      .join(ztf_sources)
      .for_each(varstar_classify)
      .query("pRRLy > 0.95")
      .skymap()

hp.mollview(img)
```

TAPE - Status (May 2023)

Legend
+ New item from the last few months
= Existing item
* Work underway

Software Aspects

API

+ Object/Source Refactor
+ Loading from parquet files, HiPSCat directories*, and dictionaries
+ Filtering Operations (dropna, prune)
+ Source Binning, Insertion

Infrastructure

+ Uses Python Project Template
+ Tutorial/Documentation on readthedocs
= Unit Testing, 85% coverage (and rising!)

Performance/Scalability

+ Have run O(million) ZTF lightcurves through analysis suite functions

Scientific Aspects

Analysis Suite

= Stetson J Function
+ Structure Function Refactor
+ Custom-User Analysis Functions

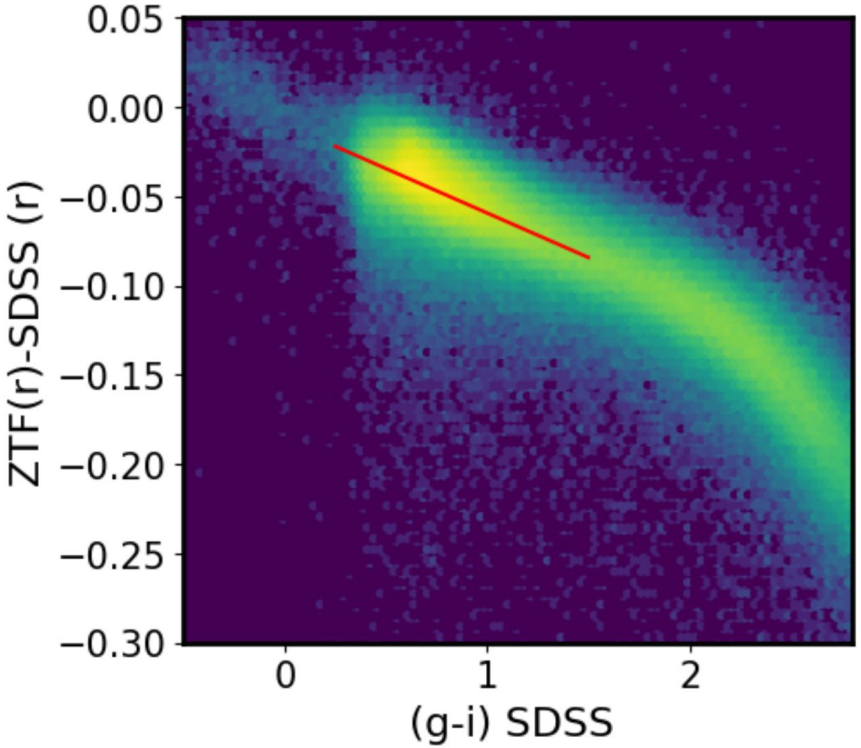
Scientific Usage/Testing

+* Time Domain Minimal Viable Product
+* TinyGP/JAX Code

Community Interaction

=* Fast Periodic Detection (Dr. Tansu Daylan)
=* light-curve (Konstantin Malanchev)

Examples: color-correction

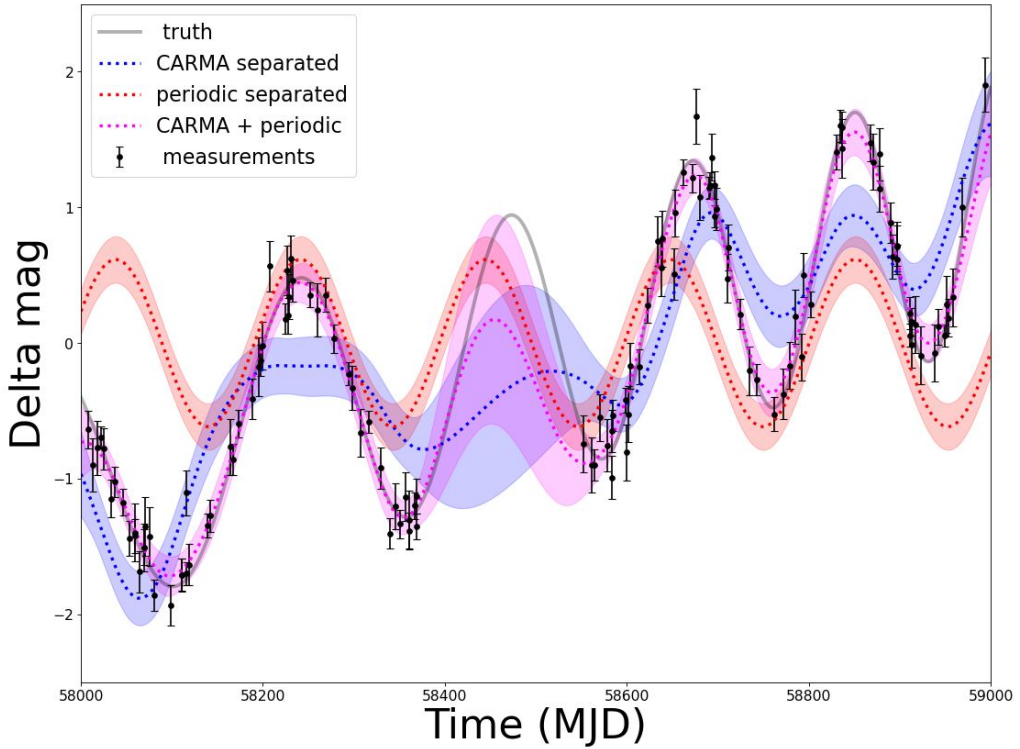


- Enable analysis of long light curves
- Primarily by cross-matching and ``re-normalizing'' PanSTARSS, ZTF + (SDSS)
- Enable estimation of custom color correction and/or apply already estimated factors from literature
- Enable estimation of errors from the spread of stars of similar magnitude/color
- **Your input appreciated**

Examples: fast JAX

- JAX - just in time compilation & autograd (e.g., plenary talk by Dan-Foreman Mackey at AAS)

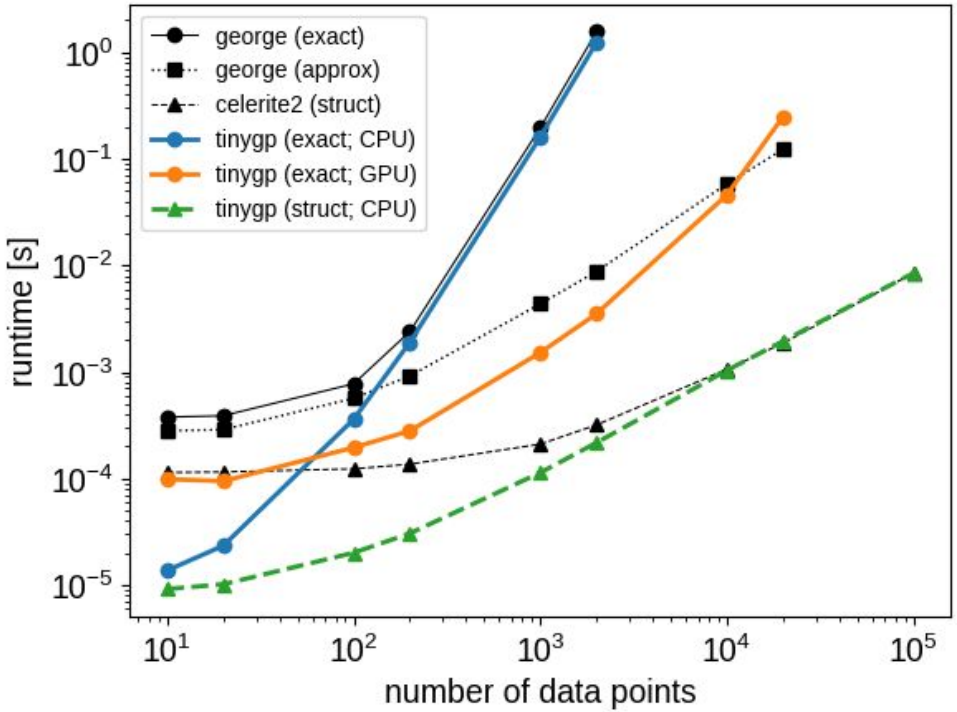
- Enables fast Gaussian process estimation
- Compilation per given length of lightcurve
  - when operating on large number of lightcurves, we want to send all objects of same length to a single dask worker to avoid recompilation



Examples: fast JAX

- JAX - just in time compilation & autograd (e.g., plenary talk by Dan-Foreman Mackey at AAS)

- Enables fast Gaussian process estimation
- Compilation per given length of lightcurve
  - when operating on large number of lightcurves, we want to send all objects of same length to a single dask worker to avoid recompilation



Summary

- LINCC - building software systems for key LSST science
- Supporting community through incubators, workshops, talks
- LSDB - crossmatching library, via HiPSCat and parquet files
- Time domain
  - Astropy contributions
  - Scaling up codes (via dask)
  - Functions useful for community

2nd Thursday of the month, 10 am Pacific at

https://ls.st/lincc-talks