An Introduction to LINCC Frameworks

Jeremy Kubica (CMU)
The LINCC Frameworks team

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

The LINCC Frameworks Project

LSST Interdisciplinary Network For Collaboration And Computing

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

PIs: Andy Connolly (UW), Rachel Mandelbaum (CMU)

Director of Engineering: Jeremy Kubica (CMU)

LSST Science Pipelines

Science Platform Research

Inference

Algorithms

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

New LINCC Frameworks Team Members

Software Engineering Team

- Jeremy Kubica
- Carl Christofferson (TL: UW)
- Melissa DeLucchi (TL: CMU)
- Max West
- Doug Branton
- Drew Oldag
- Olivia Lynn
- **more to come**

Project Scientists

- Colin Chandler
- Neven Caplar
- Samuel Wyatt
- Alex Malz
- **1 more at CMU, to be hired**
- **2 more to come from the University of Pittsburgh**

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

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)
- Hackweeks - Provide tutorials and training in any new tools / frameworks.
- Incubators - Scientists work with team to get their science applications working (open proposal process).
- LINCC Frameworks members joining science collaborations.

Incubators

Incubators will provide support for researchers at LSSTC member institutions and the broader science community community 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.

Incubators - Process

- Starting early 2023
- Open proposal process and ~6 projects selected each year.
  - Selection will be an open process managed by LSSTC.
  - At least 3 of the slots each year will be awarded to teams led by researchers at LSSTC member institutions
- ~3 month direct collaborations
- ~20k funding provided to offset travel costs, provide graduate student salaries, pay summer salaries for faculty, etc.

Incubators

<u>One example implementation:</u>

- A research group proposes a novel research investigation that requires using/developing new computational tools. The proposal gets selected by LSSTC.
- The engineering and research teams collaborate on a research project.
- A graduate student spends 3 months with one of the LINCC Frameworks teams (in-person or collaborating virtually).

<u>Example outcomes:</u>

- The graduate student is first author on a joint scientific paper.
- New computational tools are created for the Rubin community and/or existing tools are improved.
- The research team PI has pilot results for their next NSF proposal.
- Knowledge is shared between the two groups.
- A foundation is created for future collaborations.

Workshop: From Data to Software to Science with the Rubin Observatory LSST

Workshop goals:

1. Enabling *interactive development* of exciting scientific use cases for early LSST data, and identifying the common computational/technical challenges and enabling technologies associated with them.
2. Promoting the development of a broad and inclusive community of researchers engaged with LINCC Frameworks.

Paper: https://arxiv.org/abs/2208.02781

Science use cases

Divided the science into 7 research areas (not a 1:1 mapping to the LSST Science Collaborations)

- Solar System Science: 6 cases (active asteroids, TNOs)
- Local Universe Static Science: 5 cases (IMF, accreted stellar pops, dwarf gals)
- Local Universe Variable and Transient Science: 9 cases (YSO, microlensing)
- Extragalactic Static Science: 7 cases (morphologies, extinction, LSB dwarfs)
- Extragalactic Variable Science: 8 cases (AGN, lensing)
- Extragalactic Transient Science: 7 cases (SNe, TDEs, classification)
- Cosmology: 6 cases (weak lensing, SNe classification, spectroscopic followup)

	Cross-matching	Photo-z	Selection functions	Time series	Image reprocessing	Image analysis
Cosmology	✓✓	✓✓	✓✓	✓✓	✓	✓
Extragalactic static	✓✓	✓✓	✓✓		✓✓	✓
Extragalactic transient	✓✓	✓✓	✓	✓✓	✓	✓
Extragalactic variable	✓✓	✓	✓	✓✓	✓	✓
Local Universe transient & variable	✓✓		✓	✓✓		
Local Universe static	✓✓		✓✓		✓	✓
Solar system	✓		✓✓	✓✓	✓	✓✓

Table 1. Table highlighting the connection between scientific and technical areas discussed at the workshop. Rows are science areas while columns are for infrastructure capabilities. A double checkmark (✓✓) signifies that some infrastructure capability is essential to enable a particular scientific area, while a single checkmark (✓) signifies that the infrastructure capability would enhance or expand scientific discovery within that area but is not necessary to enable all of it.

Common technical areas identified at the meeting

1. Scalable Cross-matching: real-time (low-latency) positional matching of ~10k sources to ~10 catalogs of ~1Bn sources; offline/batch match and join of ~1Bn sources to catalogs of ~1Bn sources.

2. Photometric redshifts: run and update photo-z's tailored to specific science cases; outputting PDFs for error estimates (~10TB for LSST data); run in parallel

3. Selection function determination: build on DM selection function capabilities; extend to broad science cases (scalar and vector selection functions)

4. Scalable job execution system: run time series, image analysis, classification, model fitting at an LSST scale ~1Bn sources in parallel

Common technical areas identified at the meeting

5. Sky image access and reprocessing at scale: reprocessing of subsets of images (cutouts and full-focal plane data); requires scalable data access services, processing infrastructure, and processing software (built from DM software)

6. Object image access and analysis at scale: processing individual (object-level) images (e.g. deblending, classification); requires scalable image cutout service of arbitrary size; ability to link results to archival data; run in parallel

7. Time series analysis support infrastructure: extract features and classify the captured time-series; enable parametric and model fitting; enable anomaly detection; run in parallel; store, link, and update outputs

Current Project Overviews

Team started with a series of short term starter projects to ramp up and is transitioning to larger, longer term contributions now.

- Cross Matching
- Timeseries (AstroPy*)
- Photo-z (RAIL* and QP*)
- Moving objects (KBMOD*)
- Infrastructure (LINCC-HUB)

* Contributions to existing efforts where LINCC Frameworks plays a supporting role.

Development Approach

Like LSST/RSP we are using a development approach based on industry best practices for projects we are leading:

- Product requirements gathering from users
- Design documents and reviews for major changes
- Code reviews for all changes
- Productionisation, testing, and coding best-practices
- Using LSST style guide: https://developer.lsst.io/
- All software will be interoperable with RSP

Using established processes and styles for existing projects / code bases.

All new software will be open source and available to the community.

Cross Matching

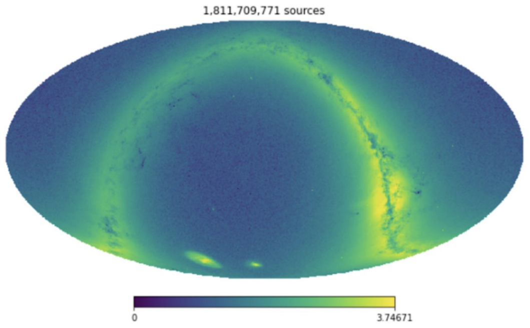
(Samuel Wyatt)

Spatial join of O(10B) x O(1B) object catalogs and enable downstream analysis of the join result. Prototype on GAIA DR3 data uses HIPS style partitioning in healpix space to make the spatial join tractable.

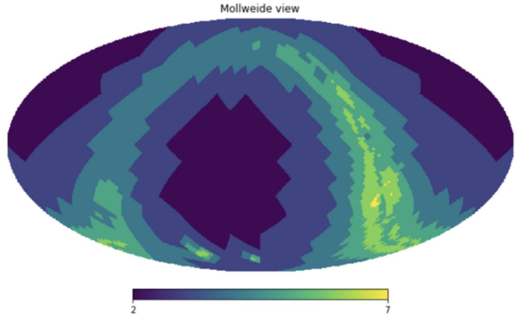
LINCC Frameworks efforts will include:

- Libraries and testing for partitioning logic (*HIPSCat*)
- Parallelization and optimization of spatial join
  - Filtering, and post-match analysis
- Focus on maintainability and productionisation

Interested?: contact sdwyatt@uw.edu or join #lincc-xmatch



Gaia DR2 Catalog Counts (log scale)



Visualization of file storage (color = healpix level)
3933 partitions of similar size (128-256 MB)

Time Series

(Neven Caplar)

Flexible framework for analysis of time-series

Follow the example set forth by the Lightkurve package - established package for analysis of Kepler & TESS lightcurves, with a high uptake in the community. Replicate the user-friendly API of Lightkurve, while at the same time provide powerful tools for LSST science analysis

Initial effort - we aim to develop framework that would be focused on two science cases:

1. Period finding in RR Lyrae stars -> multiband Lomb-Scargle
2. Structure function calculation for AGN + CARMA modelling

Photo-z

(Drew Oldag)

RAIL is an openly developed, public, flexible photo-z toolkit (initiated by DESC)

- Forward models mock photometry with realistically complex imperfection
- Estimates and compresses redshift PDFs by many methods and formats
- Evaluates comprehensive metrics of photo-z performance

Early LINCC activities aim to build up functionality to enable cross Science Collaboration use cases

- There's lot of room for expansion of the evaluation portion of RAIL, new metrics, metric organization, and utilization.
- Interested in integration with the Rubin Science Platform

Interested? Contact aimalz@cmu.edu or join #desc-pz-rail

Moving Objects

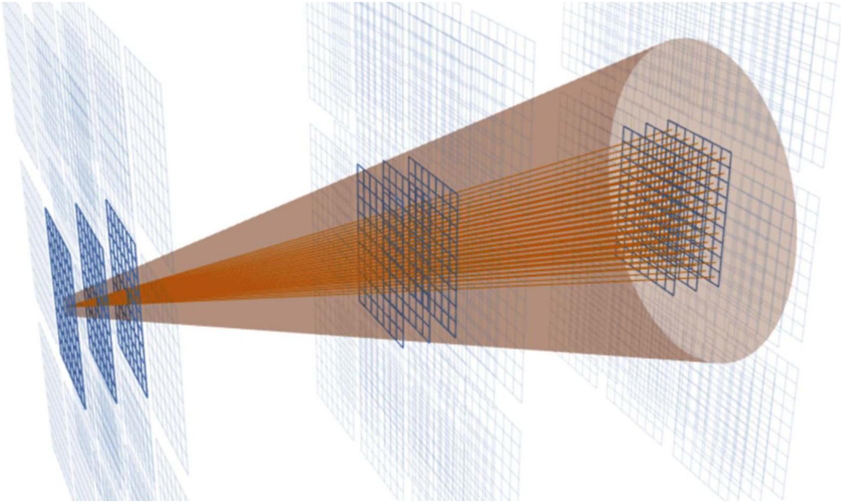
(Jeremy Kubica)

kbmod

KBMOD (Kernel Based Moving Object Detection) uses a shift and stack approach to detect faint objects (e.g. TNOs). [https://github.com/dirac-institute/kbmod]

LINCC Frameworks Efforts:

- **Scale software to LSST volumes**
  - Improve speed
  - Reduce computational cost
  - Reduce memory usage
- Productionize software
- Improve maintainability



(From Whidden et. al. 2019)

Moving Objects

kbmod

Example work (so far):

- New features:
  - Known object lookup
  - Image encoding for reduced memory usage
  - Spatial and temporal varying PSFs
- Rigorous testing added (unit tests, regression test, diff test)
- Productionisation, acceleration, clean ups, and bug fixes

Aiming toward a version 1.0 release in the near future.

Collaborators, reviewers, and/or users welcome. We’d like to ensure work is aligned with communities needs. Contact: jkubica@andrew.cmu.edu

Feedback and collaboration

Collaborators welcome!

Feedback on focus areas or projects is also welcome. How can we most help the community?