

Enabling science from the Rubin alert stream with Lasair

Roy D. Williams ¹★, Gareth P. Francis ¹, Andy Lawrence ¹, Terence M. Sloan ¹,
 Stephen J. Smartt ², Ken W. Smith ³ and David R. Young ³

¹Royal Observatory, University of Edinburgh, Blackford Hill, Edinburgh EH9 3HJ, UK

²Department of Physics, University of Oxford, Keble Road, Oxford OX1 3RH, UK

³Astrophysics Research Centre, School of Mathematics and Physics, Queen's University Belfast, Belfast, BT7 1NN, UK

Accepted 2024 July 1. Received 2024 June 21; in original form 2024 April 12

ABSTRACT

Lasair is the UK Community Broker for transient alerts from the Legacy Survey of Space and Time from the Vera C. Rubin Observatory. We explain the system's capabilities, how users can achieve their scientific goals, and how Lasair is implemented. Lasair offers users a kit of parts that they can use to build filters to concentrate their desired alerts. The kit has novel light-curve features, sky context, watchlists of special sky objects and regions of the sky, dynamic cross-matching with catalogues of known astronomical sources, and classifications and annotations from other users and partner projects. These resources can be shared with other users, copied, and modified. Lasair offers real-time machine-to-machine notifications of filtered transient alerts. Even though the Rubin Observatory is not yet complete, Lasair is a mature system: it has been processing and serving data from the similarly formatted stream of the Zwicky Transient Facility alerts.

Key words: Software – Rubin – Lasair – Transient.

1. INTRODUCTION

The Legacy Survey of Space and Time (LSST, Ivezić et al. 2019), using the *Charles Simonyi Telescope* at the Vera C. Rubin Observatory, will be a 10-yr survey of the changing sky expected to begin operations in early 2025. Of the ~ 10 million alerts expected each night, most will be the flickering of active and variable objects and wandering of Solar-system objects. There will also be a few thousand nightly detections of supernovae (SNe) and active galaxies.

Lasair is designed to filter the massive stream of transients in different ways for different kinds of science, add value to the LSST alerts, and store alerts with their added value. It is available to users as a website¹ and an API, as well as real-time push notifications that can communicate machine to machine.

The broker is built to process transient alerts rapidly and to make the key decision: *is this an object I want to follow up?* LSST alerts will come at a very high rate, and Lasair takes advantage of the design of the distribution system, which issues events in rich alert packets to enable standalone classification. Incoming alerts are judged only on that rich alert packet, without data base interaction, leading to a fast and scalable ingestion system. Lasair has been available for some years (Smith & Williams 2019), as an LSST prototype, delivering alerts from the Zwicky Transient Facility (ZTF, Bellm et al. 2018).

2. LASAIR CONCEPTS

The philosophy of Lasair is to be a *platform for science* rather than producing the science itself. Lasair does not focus exclusively on

classifying; rather, it allows scientists to make their own filters and classifiers from numerical features and attributes. Suppose, e.g. a classifier is built to pick out a specific type of SN. The advantage of Lasair's approach, with user-made features and filters, is that it may be possible to differentiate SN subtypes, based on a user-defined filter, whereas pre-built classifiers typically lump them together.

Lasair users combine attributes from several rich data tables to build a filter, or they can copy a public filter built by another user and modify it. These rich tables are described below. Filters can be run on-demand on the data base of past alerts or in real time, with the results fed rapidly into a machine-readable notification so that a user's own machine can take further action.

Lasair adds to the light-curve features already computed by Rubin for periodic/stochastic sources, with Lasair features being directed to finding new explosive transients.

Lasair filters can utilize the powerful *Sherlock* system (Young 2023b): a contextual classifier for astronomical transients that cross-matches a transient's sky-location against an extensive library of astronomical catalogues and, based on matched data, attempts to categorize the transient into one of seven classes; variable star – VS, cataclysmic variable – CV, bright star – BS, nuclear transient – NT, supernova – SN, active galactic nucleus – AGN, and ORPHAN (if the transient fails to be matched against any catalogued source). Lasair also has a continuously updated copy of the Transient Name Server (TNS)² data base (Gal-Yam 2021); the IAU reporting system for SNe and other transient sources.

Users can upload their own selection of interesting objects as a *watchlist*, which can become a filter of those alerts which are

* E-mail: roy@roe.ac.uk

¹<https://lasair.lsst.ac.uk>

²<https://www.wis-tns.org/>

cross-matched with the watchlist. A related resource is the *watchmap* to find all alerts that fall into a specific region of the sky.

Lasair encourages users to build *annotator* systems as a way to contribute further value to the information portfolio of an event. Lasair has agreements with other brokers to contribute their classifications as annotations. A user can set up a machine to receive, compute, and contribute information about alerts: an example is NEEDLE (Sheng et al. 2024), which collects pixels and light curves to find rare SNe and tidal disruption events (TDEs).

All these resources can be used together in Lasair filters: light-curve features, Sherlock, TNS cross-matching, watchlists, watchmaps, and annotations. User-contributed resources can be shared with others, copied and modified, or kept private. Lasair has extensive documentation as text, Python notebooks, and how-to videos.³

Lasair filters return an initial selection of objects, based on the rich value-added data content, and users can then manually scan the results with the Lasair Marshall Notebook (Lasair Team 2023a), or run their own local code on the light curve and all the other data about each object. SQL (Structured Query Language) filters can be escalated from static (run on command) to streaming filters that run whenever new alerts arrive and push the results via email or Kafka. Another approach could use the Lasair API to download results into a user's own marshall system.

2.1 Scientific goals of Lasair

Lasair and its partners (Section 2.3) will provide combined access to the alerts, annual LSST data releases, and external data sources. The joint system will provide a flexible platform that creative users can adapt to their own ends. The science themes are below.

2.1.1 Extragalactic transients

Luminous transients outside our own galaxy include SNe, kilonovae, TDEs, active galactic nucleus (AGN) flare activity, nuclear transients of unknown origin, gamma-ray burst afterglows, stellar mergers, and compact object mergers. All this science requires light curves, links to galaxy and redshift catalogues, precise astrometric cross-matching, correlation with high energy information, and multi-wavelength cross-matching.

2.1.2 Multi-messenger astronomy

Lasair responds rapidly to alerts from NASA-GCN (Barthelmy & Racusin 2023) about gravitational waves, and is able to set up a watchmap (see below) for the sky area. Lasair will collect LSST alerts in that area and allow users to build filters to find the possible optical counterpart. In the future this capability will be extended to gamma-ray bursts, neutrinos, etc.

2.1.3 Massive samples of SNe

Lasair will link all transients to a list of likely host galaxies together with their photometric and spectroscopic redshifts. See Time Domain Extragalactic Survey (TiDES), 4MOST, and SoXS in the 'Partners' Section 2.3 below.

2.1.4 AGN, TDE, and long-lived transients

Lasair will allow users to select known AGN, upload their own AGN catalogues, and select flaring events in both active and passive galaxies. This will support the science of TDEs, changing-look quasars, AGN flares, microlensing of background AGNs by foreground galaxies, and unusual long-lived nuclear transients.

2.1.5 Stellar transients

Most science for variables (typically recurrent and periodic signals) will be achieved with the annual data releases. However, there is a great opportunity in combining alerts with the data releases. Users can discover outbursts or large amplitude variability through the alerts, which link to the data releases and full multi-year light curves. Lasair will provide streams of objects matched to known stars and trigger on a particular magnitude variability index.

2.2 Objects and sources

Lasair handles objects and sources delivered by the Rubin Observatory. A source is a detection by the telescope of an object, a collection of pixels on the telescope's light-collection device, which is significantly different than it was in the reference imagery taken at the beginning of the survey. A source is detected with a specific narrow-band optical filter: LSST uses filters *u,g,r,i,z,y* and ZTF uses *g,r*.

The brightness of a source in a transient survey is different from that source in a reference sky acquired in the past. Difference flux can be positive or negative, but when expressed as magnitudes, the measurement has two parts: an absolute value converted to magnitudes and a flag to indicate a positive or negative difference. Note that if nothing was detected in the reference sky, then the difference magnitude is the same as the apparent magnitude.

To detect transients, the LSST will trigger on significant (5σ) sources which are a positive excess with a point-spread-function shape measured on the difference image.⁴ The 5σ detection limit is with respect to the noise in the difference image after the reference image has been subtracted. When such a detection is made, a data packet called `diaSource` captures the result of the source detection, where `dia` means 'difference image analysis'. The Rubin Observatory searches for previous transients close to that sky location, and from these, it builds a collective data packet called a `diaObject`. If the transient is associated with a fixed object in the sky, the `diaObject` and its `diaSources` define the light curve of that object. The Lasair project is primarily devoted to these fixed sources. For more information on LSST data products, see Jurić et al. (2023).

However, if the source detection is from a moving Solar-system object, there will be no previous detection at that sky position. In a further processing step at the Rubin Observatory, the locations of sources are cross-matched with all known Solar-system objects. If a match is found, additional data packets are added to the alert, including `ssObject` with properties of the Solar-system object and its orbit.

Thus the alerts are organized as a central data packet and a number of associated packets. For astronomical transients fixed in the sky, the central packet is the `diaObject` and the light curve is defined by the

³<https://lasair.readthedocs.io/>

⁴Negative sources with point-spread-function shapes will also be catalogued for variable object selection.

`diaSources`, as well as forced photometry and non-detections. For moving objects, there is primarily the `ssObject` and `ssSources`. Lasair is a platform for science involving fixed objects but it also collaborates with the ‘Adler’ project, a platform for Solar-system objects (more information in the section ‘Partners’ given next).

2.3 Partners

2.3.1 UK Data Access Centre

The UK will build and operate an Independent Data Access Centre (IDAC, Beckett & Mann 2023), sized to serve 20 per cent of the Rubin community. Like the two Rubin-operated DACs, in the US and Chile, this will be a ‘Full IDAC’, serving all the Rubin data release products, and it will be available to anyone with Rubin data rights. Lasair will work hand-in-hand with the UK IDAC, giving longer term light curves and images to back up discovery from the alert stream, facilitating joint data-mining projects, and allowing Lasair to serve proprietary data to those of its users who are Rubin Data Rights Holders (Rubin Observatory 2022).

2.3.2 TiDES and 4MOST

Lasair works closely with the two major ESO projects that will provide tens of thousands of spectra for LSST SNe. We will coordinate SN discoveries in Lasair with spectra from the 4MOST multi-fibre spectrometer on the *ESO VISTA* telescope. We expect to select 35 000 live transients for spectra and obtain spectra of 70 000 host galaxies in the TiDES (Swann et al. 2019). This will provide the largest cosmological sample of Type Ia SN, together with a massive statistical sample to understand SN explosion physics across a range of redshifts and host galaxy masses and metallicities. Lasair will provide both (reproducible) selection and extract the scientific content (type, phase, redshift, etc.) to re-ingest into the broker for further user exploitation.

2.3.3 SoXS

Lasair also works closely with the UK team responsible for building the science software infrastructure for SoXS (Schipani et al. 2018); a single-shot 0.35–2 μm spectrometer on ESO’s *New Technology Telescope (NTT)*. ESO is fully dedicating the *NTT* to time domain science, with the schedule being run by the SoXS consortium. We will enable the SoXS marshall to interface with Lasair to select LSST transients for classification and then re-ingest the information and public data for all users to access.

2.3.4 Public European Southern Observatory Spectroscopic Survey of Transient Objects

Lasair works with the Public European Southern Observatory Spectroscopic Survey of Transient Objects (PESSTO, Smartt et al. 2015). It is a public spectroscopic survey that began in 2012, classifying transients from publicly available sources and wide-field surveys, and selecting science targets for detailed spectroscopic and photometric follow-up.

2.3.5 Citizen science

Citizen science builds unique and authentic research experiences for the public that directly engage individuals with little or no scientific

training or background, thus lowering the barrier for the public to contribute directly to scientific investigations. Lasair already has a strong collaboration with Zooniverse.org, having established a project (Smith et al. 2022) to hunt for rare superluminous SNe. This will be broadened, and other projects added, in the LSST era.

2.3.6 Solar-system science with Adler

The LSST will be the largest catalogue of Solar-system objects to date (Schwamb et al. 2018). With each planetesimal receiving hundreds of observations across six filters, LSST will radically transform the view of our planetary system and usher in a revolution for time-domain Solar-system science. The Adler system will work closely with Lasair, ingesting the Rubin Solar-system alerts and identifying potentially active sources and unexpected photometric behaviour.

2.3.7 Other Rubin alert brokers

The LSST alert stream is expected to be high bandwidth (0.2–5 Gb s^{-1} , Graham et al. 2020), so it would be impractical to host large numbers of readers, each taking the whole stream. The Rubin Observatory instead selected seven ‘Community Brokers’ through a proposal process, each providing its own vision of how such a broker should work. In addition to Lasair, they are Alerce (Förster et al. 2023), ANTARES (Matheson et al. 2021), AMPEL (Nordin et al. 2019), Babamul (Graham et al. 2023), FINK (Ishida et al. 2023), and Pitt-Google (Raen et al. 2023).

2.4 What Lasair is not

The ‘rich data packet’ that comes with each alert means that light-curve features and subsequent filtering are made only on that packet, which has only a year of past data about each object (only a month for the ZTF prototype). Light-curve features and thus decisions are *not* based on the full multi-year light curve, even though the Lasair data bases include these. Rather, decisions are based on these shorter ‘packet’ light curves. Even though the light-curve features are based on the packet light curve, the web page, and API deliver all of it. In the case of ZTF, Lasair light curves go back 6 yr, and these can be effectively mined – see Section 4.2 ahead.

The fact that a feature is not built into Lasair does not mean it cannot be done; rather, the user has to do some of the work themselves through annotation – and we try to provide a good set of example notebooks to act as starting points for some of these things. It comes down to the distinction between doing science and providing a toolkit for people to do science with.

3. LASAIR FUNCTIONALITY

Lasair provides a portfolio of data for each object (`diaObject`), and a means to build a filter that selects which objects the user finds interesting. In addition to that provided by the Rubin Observatory pipeline, Lasair adds novel light-curve features, rich sky context via Sherlock, watchlists, and watchmaps, TNS cross-match, and annotations from other users and brokers. All this can be combined into the WHERE clause of an SQL statement that is the heart of a Lasair filter.

The components described below are like LEGO bricks that can be used to build a bespoke filter that concentrates the alerts a scientist is searching for. Each component is expressed as a data base table whose attributes can be used to create the filter. Filters can be public

and, therefore, shared, copied and modified; the same is true of the other resources described below.

In addition to making filters, users can create watchlists of their special objects and watchmaps of sky areas or build their own classifier running on their own machine. They can then push the results back to Lasair as annotations.

3.1 Sky context from Sherlock

Sherlock finds known astronomical objects at the sky location of an alert. Published astronomical catalogues are carefully curated collections of stars, galaxies, variable stars, cataclysmic variables (CVs), AGNs, etc. When a transient object is detected, an astronomer will want to know if the transient is associated with a previously known source and, if so, what kind of source it is. If astronomers are searching for extragalactic explosive events such as SNe, they are typically interested in transients associated with a galaxy.

Sherlock consists of a Python package and a curated library of astronomical catalogues (in a MariaDB data base) and provides a rapid and reliable spatial cross-match service for any astrophysical variable or transient. It associates a transient's position against its library of astronomical catalogues and uses an intelligent ranking algorithm to assign a classification. At its most basic, Sherlock can triage transients into stars, AGN, and SN-like objects. It has been used for many years as part of the ATLAS, Pan-STARRS, and Lasair-ZTF decision-making.

One of the main purposes of Sherlock within Lasair is to identify known variable stars, since they will make up a majority of LSST alerts, and to associate candidate extragalactic sources with potential host galaxies. Sherlock's library of catalogues contains data sets from many all-sky surveys such as

- (i) *Gaia* DR2 (Gaia Collaboration 2023);
- (ii) Pan-STARRS1 Science Consortium surveys (Chambers et al. 2019) and the catalogue of probabilistic classifications of unresolved point sources by Tachibana & Miller (2018) based on the Pan-STARRS1 survey data;
- (iii) The SDSS DR12 `PhotoObjAll` and SDSS DR12 `SpecObjAll` Tables (Alam et al. 2015). The `PhotoObjAll` table contains a photometry-based measurement of the star-galaxy separation, and the `SpecObjAll` table contains a spectroscopic redshift for each source;
- (iv) Guide Star Catalogue v2.3 (Lasker et al. 2008); and
- (v) 2MASS point- and extended-source catalogues (Skrutskie et al. 2006).

Sherlock also employs many smaller source-specific catalogues such as

- (i) Million Quasars Catalogue v5.2 (Flesch 2023);
- (ii) Veron-Cetty AGN Catalogue v13 (Véron-Cetty & Véron 2010);
- (iii) Downes Catalogue of CVs (Downes et al. 2001); and
- (iv) Ritter Cataclysmic Binaries Catalogue v7.21 (Ritter & Kolb 2003).

For spectroscopic redshifts and/or non-redshift based distance measurements Sherlock uses

- (i) LASr-GPS which is a 100-Mpc volume-limited galaxy catalogue (Asmus et al. 2020);
- (ii) NED-D Galaxy Catalogue v17.1.2. A Master List of Redshift-Independent Extragalactic Distances (Steer et al. 2017);

- (iii) A dynamic query of the NASA/IPAC Extragalactic Database via the NEDDY software package (Young 2023a); and
- (iv) SDSS DR12 `SpecObjAll` table (Alam et al. 2015).

At a base level of matching, Sherlock distinguishes between transient objects synonymous with (the same as, or very closely linked to) and those it deems as merely associated with the catalogued source. The resulting classifications are tagged as synonyms and associations, with synonyms providing intrinsically more secure transient nature predictions than associations. Depending on the underpinning characteristics of the source, there are seven types of predicted-nature classifications that Sherlock will assign to a transient:

- (i) VS: If the transient lies within the synonym radius of a catalogued point-source,
- (ii) CV: If the transient lies within the synonym radius of a catalogued CV,
- (iii) BS: If the transient is not matched against the synonym radius of a star but is associated within the magnitude-dependent association radius,
- (iv) AGN: If the transient falls within the synonym radius of catalogued AGN or QSO,
- (v) NT: If the transient falls within the synonym radius of the core of a resolved galaxy,
- (vi) SN: If the transient is not classified as an NT but is found within the magnitude-, morphology- or distance-dependent association radius of a galaxy, or
- (vii) ORPHAN: If the transient fails to be matched against any catalogued source.

For Lasair-ZTF the synonym radius is set at 1.5 arcsec, which we find is a good match to the PSF (Point Spread Function) and astrometric accuracy of that survey. This is the cross-match-radius used to assign predictions of VS, CV, AGN, and NT. The process of attempting to associate a transient with a catalogued galaxy is relatively nuanced compared with other cross-matches as there are often a variety of data assigned to the galaxy that help to more reliably inform the decision to associate the transient with the galaxy or not.

Once each transient has a set of independently cross-matched synonyms and associations, Sherlock selects what it thinks is the most likely classification. The last step is to calculate value-added parameters for the transients, such as absolute peak magnitude if a redshift or distance can be assigned from a matched catalogued source. The predicted nature of each transient is presented to the user along with the light curve and other information.

As part of the Lasair project, there is public access to the integrated Sherlock code and data base information through the Lasair API (Lasair Team 2023a). We plan further additions to the Sherlock data base with the Legacy Survey catalogues and eventually LSST data releases.

3.2 Transient Name Server

An astronomer who is interested in an explosive event will want to know if it has already been seen and registered by a different survey. The TNS (Gal-Yam 2021) is the official IAU mechanism for reporting new astronomical transients, with a focus on extragalactic explosive transients. Once spectroscopically confirmed, new SN discoveries are officially designated an SN name. Lasair keeps a cache of the data base, updated every few hours, which can be used in Lasair filters.

3.3 Watchlists and watchmaps

Many astronomers study a specific set of existing objects and are interested in transients from those – e.g. active galaxies, galaxy clusters, or star formation regions. A watchlist is a set of named points in the sky, together with a radius in arcseconds – which can be the same for all sources, or different for each – i.e. a set of named cones. It is assumed to be a list of ‘interesting’ sources so that any transient that falls within the radius of one of the sources might indicate the activity of that source. Each user of the Lasair system can have one or more watchlists, and can make a filter to be alerted when a transient is coincident with a watchlist source. An ‘active’ watchlist is one that is run every day so that it is up to date with the latest objects. Watchlists are restricted to at most a million sources.

To be specific, suppose we are interested in the 42 objects in the catalogue of BL Lac candidates for TeV observations (Massaro et al. 2013), which can be found in the VizieR library of catalogues. The user can make their watchlist ‘public’ so other Lasair users can see it and use it in queries, and can make the watchlist ‘active’, meaning that the cross-match is kept up to date with incoming alerts, and active filters respond immediately. Full details are in Lasair Team (2023a).

A watchmap is a specification of an area of the sky that can be used as part of a Lasair filter. The crucial purpose here is for gravitational wave events and other multi-messenger transients, where the sky location is actually a probability distribution. However, the watchmap does not embody the idea of a probability distribution – it is just an area of the sky – so it might cover the 90 percentile. A watchmap might also be the footprint of another survey or a large sky area like the Orion Nebula. If the watchmap has been set to ‘active’, then all alerts ingested to Lasair are tested against it, and tagged if inside the watchmap. A filter can then be built that selects only alerts inside the watchmap. Watchmaps are created by building and uploading an MOC (Multi-Order Coverage) file (Fernique et al. 2014).

3.4 Annotations

Lasair allows users to add information to the data base that can then be used as part of a query by another user. An annotation is a structured packet of extra information about a given object that is stored in the annotations table in the SQL data base. This could be the result of running a machine-learning algorithm on the light curve, the classification created by another broker, or data from a follow-up observation on the object, e.g. a link to a spectrum. Users that put annotations into the Lasair data base are validated, and administrators then make it possible. That user will run a method in the Lasair API – from their own machine – that pushes the annotation: all this can be automated, meaning the annotation may arrive within minutes of the observation that triggers it.

Each annotation is associated with a specific Lasair object, and may contain

- (i) `objectId`: the Lasair object being annotated;
- (ii) `topic`: the name of the annotator that produced this annotation;
- (iii) `classification`: a short string drawn from a fixed vocabulary, e.g. ‘kilonova’;
- (iv) `explanation`: a natural language explanation of the classification, e.g. ‘probable kilonova but could also be SN’;
- (v) `classjson`: the annotation information expressed as a JSON dictionary; and
- (vi) `url`: a URL where more information can be obtained, e.g. a spectrum of the object obtained by follow-up.

The NEEDLE project (Sheng et al. 2024) is an example of an annotator, designed to find superluminous SNe in dwarf galaxies, and TDEs occurring in the centres of nucleated galaxies. There is a ‘pre-filter’ that selects based on the Sherlock classification and the current magnitude versus mean magnitude; results of this filter are pushed via Kafka to a separate analysis machine, which pulls the light curve and cutout images for analysis by a neural-net system. Promising candidates are pushed back to Lasair as an annotation, with the `classjson` attribute set to something like:

```
{'SN': '0.799', 'SLSN-I': '0.103', 'TDE': '0.099'}
```

These returned attributes can then be used as part of a filter in the usual way; there is special syntax to refer to the parts of the `classjson` in the SQL language.

Another annotator is *Fastfinder* that models light curves of new transients to find rapid brightening from kilonovae. It classifies alerts as SLOW, FAST, and SN so that the latter could be picked out with a clause `fastfinder.classification='SN'` in the WHERE part of the filter. As explained above, the `classjson` can hold complex information, and querying is more sophisticated; for more information see the Lasair documentation (Lasair Team 2023a). Annotations can be pushed to the Lasair data base using the Lasair client; however, the user must be authenticated to do so. Lasair staff are happy to receive a request to create an annotator, and the successful user’s topic name and API key will allow them to upload annotations.

Because annotation is a process of inserting data into the Lasair data base, the Lasair Team needs to know who is doing it. Therefore, running an annotator starts with asking the team to set it up, and upload of annotations can only be done by the person responsible.

3.4.1 Fast annotations

Some special annotations can be upgraded to ‘fast’ annotations. This means that as soon as the annotation is uploaded, all user filters that involve that annotator are re-run immediately. Suppose, e.g. a user builds a filter based on the NEEDLE annotator, the criterion being that the SLSN-I probability for that object is more than 0.7. If NEEDLE is a fast annotator, then that user’s filter will run as soon as the annotation arrives, with an immediate email or Kafka result. Otherwise, there will be a delay, until another detection of that object occurs, which may be days later.

3.4.2 Alerce and Fink

Lasair is one of seven Community Brokers for the Rubin alerts and has arranged with Alerce (Förster et al. 2023) and Fink (Ishida et al. 2023) to consume their outgoing Kafka streams and ingest the results of their classifiers as annotations. Currently, these include the Alerce Stamp Classifier and Light Curve Classifier, as well as some of the many Fink classifiers, which include Kilonova, TDE, Microlensing, and subtypes of SNe.

3.5 Real-time output

Alerts can be visualized on the Lasair web interface, or the underlying data fetched using the Lasair API. But there is another way to get Lasair data, through *push* notification, where the Lasair system pushes data as soon as these are available. The push can be a human-readable email, or via a machine-readable Kafka stream. Such a stream is designed to be consumed by machines, not people;

SELECT

```
objects.objectId, objects.gmag, jdnow()-objects.jdmax as 'since'
```

WHERE

```
objects.gmag < 12
```

Watchlist Watchmaps Annotators

Select a Watchlist Select a Watchmap Select an Annotator

Figure 1. Screenshot of the filter builder showing the user inputting the SELECT and WHERE parts of the filter.

it may be that an analysis program sits waiting, and springs into action as soon as the Kafka alert arrives. It could compute a light-curve classification, it could drive a follow-up telescope, perhaps followed by an annotation. The consumer of a Kafka stream need not run continuously as alerts can build up until they are summoned, perhaps by the Lasair Marshall Notebook (see Section 4.6). For more information about Lasair’s Kafka output, see Section 4.5.

4. LASAIR IN PRACTICE

4.1 Making a filter

Lasair is built around ‘filters’ of the alert stream. Users create a filter with SQL clauses, based on the attributes of the object associated with the alert; its light curve, sky context, etc. First, the user makes a filter and runs it on previous alerts, and then they can save the filter. The user can convert a filter to an ‘active’ filter, so that results are sent to email or to their own machine as soon as they are available.

Fig. 1 is a screenshot of an example filter for bright stars that have had recent alerts.

The SELECT clause can be typed (with the assistance of auto-complete and schema browser) as

```
objects.objectId, objects.gmag,
jdnow()-objects.jdmax AS since
```

and the WHERE clause similarly as

```
objects.gmag < 19 ORDER BY objects.gmag
```

In English, the filter is: select the object ID, *g*-magnitude, and time in days – call it *since* – since the latest detection, and we only want to see those objects brighter than 19th magnitude, sorted with the brightest first. There is also a FROM section listing the tables being joined to make the query; this list is made automatically from the content of the SELECT and WHERE clauses.

Clicking ‘Run Filter’ gives the brightest from years of stored alerts, as expected. If, however, the filter is saved, there is a popup form asking for name, description, etc., as well as if it should be public and if it should be ‘active’. The active filter is one of Lasair’s crucial and powerful ideas: it means that all incoming alerts are passed through the filter automatically, in near-real time, and what passes the filter is pushed to the user, by email or by Kafka, which can also drive a Slack workspace or other alert mechanism. Filters run the same way whether the filter operation is on-demand by click/API, or whether it runs in near-real-time. However there is one difference: the above filter run on-demand returns the brightest alerts first (because of the ORDER BY clause), but by the nature of a streaming filter, it can only return all its results in time order.

To get alerts coincident with a watchlist, Fig. 2 shows how all that is needed is choosing the watchlist in the pull-down list. The

SELECT

```
objects.objectId, objects.gmag,
arcsec, name
```

WHERE

```
objects.gmag < 20
order by objects.jdmax DESC
```

Watchlist Watchmaps Annotators

BLLac_TeV (roy) Select a Watchmap Select an Annotator

Figure 2. Screenshot of the filter builder showing the user inputting the SELECT and WHERE parts of the filter.

attributes *arcsec* and *name* are then available for the name of the coincident object from the watchlist, and its distance in *arcsec* from the alert.

There will be many light-curve features from the LSST data pipeline, packaged with each alert (Bellm 2023). Some ideas for adding value to these, now in development, are described in Appendix A.

4.2 Mining the Lasair data base – an example

Here, we provide a science example of how the Lasair data base can be mined. There is a ‘pre-filter’, then all the resulting light curves are fetched by API and cuts applied, until there are few enough to look by eye. Our example is based on the transient known as AT2021lwx, which released 1.5×10^{53} erg of energy over 3 yr, more than any other optical transient. It was a transient from the ZTF survey but sat in the Lasair data base until it was found (Wiseman et al. 2023). AT2021lwx was extragalactic, and had a long, smooth decline in brightness over 3 yr. Below we sketch a way to find other such events in the Lasair data base.

For this search, there were two Lasair filters: one for transients in a catalogued galaxy, and another for those where the host galaxy is too faint to have been catalogued. The first query finds alerts with Sherlock classifications SN, NT, or AGN; the second finds Sherlock ORPHANs away from the Milky Way [*abs*(galactic latitude) < 10]. Each of the resulting 59 000 ZTF light curves is fetched from Lasair and features computed for both *g* and *r* filters: being a linear fit to the light curve after the peak brightness. If the variance is low about this line, and the rate of fading is slow, then we have a candidate. The result of this specific case produced 58 candidates, still under investigation (Wiseman, in preparation).

A data-mining project of this kind requires a lot of API calls, therefore requesting that the Lasair team remove the default API throttling – which is only 20 API calls allowed in any given hour.

4.3 Learning

Lasair has full documentation (Lasair Team 2023a), explaining the basic concepts of Lasair, as well as detailed how-to tutorials. There are also videos showing how to get a user account, how to make watchlist and watchmap, and other topics. On the Lasair web page itself are numerous tooltips – short popup explanations when the mouse hovers – each of which has a link to more detail in the main documentation. There are Jupyter notebooks in the documentation, showing simple and complex queries through the API, the API interface to Sherlock, how to utilize Kafka streams and other topics.

4.4 The Lasair client

There is a Python software package called `lasair` (Lasair Team 2021) which can be installed with `pip`. It can be used to query the data base and Sherlock, to create an annotation, or to consume a Kafka stream. Full details are in the Lasair documentation (Lasair Team 2023a).

When a filter has been set to ‘active’ and Kafka output, a topic name will be displayed on the web page of that specific filter. It is this that is used to select the output stream from that filter. Users need to understand the concepts of `topic` and `groupId`, perhaps by watching the relevant videos in the documentation. The code then creates a `consumer` object and calls the `poll` method to get the messages. Full details are in the Lasair documentation (Lasair Team 2023a).

4.5 Real-time notification

As described above, the Lasair broker sends notifications when an active filter sees an interesting alert. This can be done by email, which is suitable for low-volume filters and human attention. Notifications can also be sent via Kafka, an open-source distributed event streaming platform used throughout industry for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications (see kafka.apache.org). Indeed, the data ingestion from Rubin and the Lasair processing pipeline are both based on Kafka. Other methods of notification are then easily implemented (e.g. Slack).

4.6 Marshall notebook

The purpose of a transient survey is to find the needle in the haystack: the scientifically interesting alert among millions of others. Lasair users can do this in two parts: first an automated filter that runs as the alerts are ingested, then a human looking at the results, with an application known as a ‘marshall’. The scientist looks at a batch, flagging some as ‘favourites’, and some as ‘do not show me this object again’ (veto), and checking full objects information for some. Then the next batch, until all have been seen. The next night there will be a fresh batch to be checked. The Lasair project has built a ‘marshall’, implemented as a Jupyter notebook. The alert in Fig. 3 is an SN in the centre of the Science image – near the centre of the spiral NGC 1086.

At the top are the attributes selected by the filter, then light curves, the ZTF reference and latest image, and the colour image from Pan-STARRS. The SN II is already registered in TNS, and has a link. There are two checkboxes and a place for a comment. Ticking ‘veto’ means the object will not be seen again (like the object at the bottom of the image), and ticking ‘fave’ means it will be emphasized next time. If the notebook is run again, a new set of results is shown, until all have been seen. Using the notebook for eyeballing is much easier than having Lasair send email notifications. Further instructions can be found in the Lasair documentation (Lasair Team 2023a). This Jupyter notebook is meant for individual users, but we expect that other projects (e.g. SoXS) will build their own marshall system, as the PTF, ZTF, and PESTO projects have done, connecting through Kafka and the Lasair API.

4.7 Kinetic display

This display shows the status of the Lasair data flow, as shown in Fig. 4. The total cost is about £100. A 32 × 32 RGB matrix is

powered by a Raspberry Pi, and it can be mounted in a box frame to make a wall-mounted display. The RPi needs to be on a Wi-Fi network. The code for the display and full instructions can be found in Williams (2022).

Each alert that arrives has RA and Dec in the sky: RA is 0–360 from left to right, and Dec is –25–90 from bottom to top. Underlying the colours of the display is a matrix of bits. When a batch of alerts has come in, the matrix is replaced with the new matrix with bits switched on based on the positions of the alerts of the batch. Simultaneously, Conway’s Game of Life (Conway 1970) cellular automata runs continuously on the matrix, leading to patterns that evolve and change rapidly after the batch of alerts is received. The actual colour display is an RGB matrix derived from the history of the binary matrix, with R, G, and B fading at different rates.

The display can also be shown on a computer monitor, without buying extra hardware, by installing the `OPenCV` (OpenCV Team 2022) Python package.

5. TECHNICAL IMPLEMENTATION

In this section, we give a brief overview of the technical implementation of the Lasair system. The overall architecture is illustrated in Fig. 5. Given that Rubin will produce 10 million alerts per night, the processing system must be carefully architected and well-resourced to keep up. We do not want Lasair falling hours behind, still processing a night’s alerts well into the next day, or even worse, not being finished when the firehose starts up the next night.

Lasair runs on an Openstack cloud ‘Somerville’ at the University of Edinburgh Advanced Computing Facility, which is part of ‘IRIS’ – the UK academic cloud.⁵ It shares that cloud with the DAC described in Section 2.3.1.

Lasair ingests data with a pipeline of scalable clusters, as shown in Fig. 5: Kafka, ingest, Sherlock, filter. Each cluster does a different job, some more compute/data intensive than others. It is difficult to know a priori how much resource should be allocated to each cluster, so our design gives flexibility: each cluster can be grown or reduced according to need. Also, there are persistent data stores (Cassandra, MariaDB via a Galera cluster); again, each is a resilient cluster architecture that can be grown or reduced according to need. The diagram shows the concept: data enters the Kafka system on the left and progresses to the right. The Kafka cluster (grey) consumes and caches data from Rubin and from the other pipeline clusters; the ingest cluster (yellow) splits and redirects, then puts data back into the Kafka communication bus; the Sherlock cluster (red) consumes and produces Kafka; and the filter cluster (blue) consumes the result and writes them to the Galera data base cluster and public Kafka. We also include the web and annotator nodes in this picture (bottom and right), as well as the mining nodes, although they are not part of the data ingestion pipeline.

The webserver supports users by delivering web pages and responding to API requests. The annotator nodes may be far from the Lasair computing centre and controlled by Lasair users, but they are in this picture because just like the others, they push data into the data storage and may consume from public Kafka. The Kafka system is represented by the grey nodes in Fig. 5 as well as the grey arrow at the top. It is responsible for reading and caching the alert packets from Rubin, as well as sending data to the compute clusters and receiving their resulting packets.

⁵IRIS: Digital Research Infrastructure for UK Science, <https://www.iris.ac.uk/>.

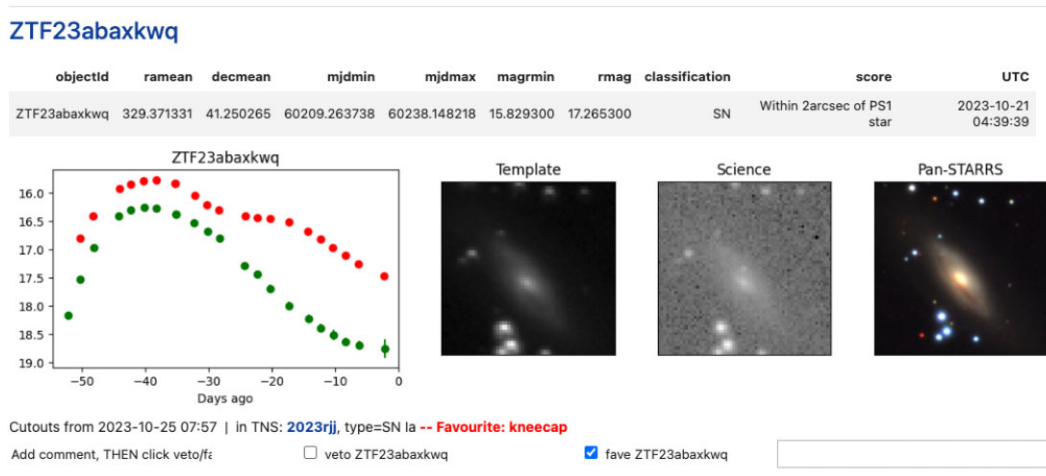


Figure 3. Screenshot from the Lasair Marshall Notebook.



Figure 4. The do-it-yourself real-time data display for Lasair’s public Kafka.

The Ingest nodes read the original alerts from the Kafka system and put the cutout images in a dedicated data base. The recent light curve is then sent to the NoSQL (Cassandra) data base. The alert without cutouts is reformatted as JSON since there is no binary content, and then it is pushed into the Kafka system.

Each Sherlock node has its own data base of 5 TB of astronomical catalogues. If necessary, this can be replicated to provide a higher throughput.

Each filter node computes features of the 12-month LSST light curve that comes with the alert, as well as matching the alert against user-made watchlists and watchmaps. Records are written to a local SQL data base on-board the node for the object and features, the Sherlock data, the watchlist and watchmaps tags. Other tables have already been copied into the local data base from the main SQL data base. After a batch of perhaps 40 000 alerts is ingested into the local data base, it can now execute the user-made queries and push out results via the public Kafka system – or via email if the user has chosen this option.

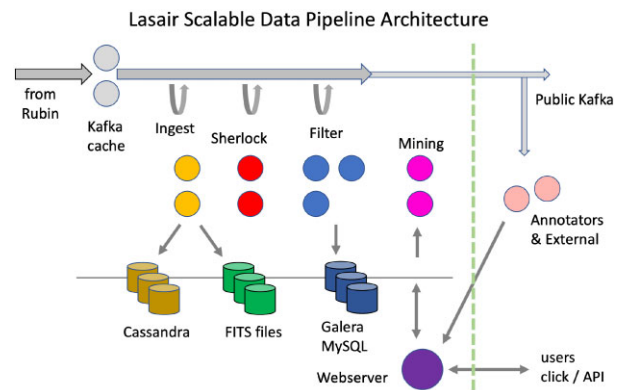


Figure 5. The architecture of the Lasair system. The nodes and pipes at the top are the Kafka communication system, and scalable homogeneous clusters read, process, and write to it, as well as writing to the three data bases, two Cassandras and one Galera. Public Kafka is emitted top right. The vertical dashed line separates the Lasair system (left panel) from user control (right panel). Annotators run by users can receive Kafka and push data back into Galera via the webserver.

The tables in the local data base are then pushed to the main SQL data base and replace any earlier information where an object is already known. Once a batch is finished, the local data base tables are truncated and a new batch started.

There are also ‘twilight services’ to prepare the ingestion system for the night: caching annotations, watchlists, and watchmaps for the filter nodes, the latest version of the TNS data base, and others.

5.1 Scalability and efficiency

The Lasair project will store all the alert information from Rubin in a Cassandra data base (although the more recent cutouts may only be stored there). Each Rubin alert is issued because of a single detection that is 5σ significance above the noise in the difference image (target image minus the reference sky), but it is accompanied by a host of additional data to allow judgement to be made. In principle, this extra data is not needed since it is already available in the Cassandra data base; however, this redundancy allows the alert processing to be handled by many independent nodes, with no need to communicate with each other or read from the data bases. Thus

during data ingestion, there is no reading from the data bases at all. When writing data, these systems can work asynchronously without blocking, an ‘eventual consistency’ paradigm that allows maximum write speed. Any processes that require consultation with the data base are then left to the next day, when the only usage of the data bases is much lighter: only the web/API customers.

There is a multi-node relational data base behind the webserver and API, providing users with the richness of the SQL language. The much large Cassandra data base is indexed on `diaObjectId` and sky position, so it is very much a key-value retrieval system rather than a query-based relational data base.

5.2 Webserver

The Lasair webserver is built in Python with Django to separate computation from presentation. The Twitter Bootstrap v5.2 framework is used alongside Plotly for light-curve display. AladinLite (Boch & Baumann 2023) shows sky context in numerous wavebands, and the JS9 package is used to display image stamps. A Grafana dashboard shows the status of all the nodes, with live traces of the pipeline’s status in terms of alerts per minute.

5.3 Self-protection

The Lasair Team has worked to ensure flexible and useful filtering of alerts for users and has chosen the SQL language for this, as it is well-known now to astronomers who are data specialists. However, Lasair guards against mistakes and attacks through SQL. All queries are run in a user account with read-only access, and query strings are carefully vetted before being run against the data base. User-made queries have a time-out of 10 s when running in streaming mode against a batch of alerts, so they cannot hold up the entire system if they are too resource-intensive.

The filter nodes compare each alert against active watchlists and watchmaps and run each batch against active user queries. Each of these active resources slows the ingestion system. Therefore, Lasair has a 6-month expiry for each, and the user must then go to the website and renew its activity.

5.4 Deployment

Lasair has a flexible system for deploying an entire system on the Openstack cloud, using Ansible and Terraform, with a specification of the numbers of nodes to be used for each functional cluster, their memory and local disc, spinning disc or SSD, how they are to be located on hypervisors, and other hardware considerations. This makes it easy to have extra systems running the entire Lasair stack: e.g. a development system or a system for measuring performance.

6. SUMMARY AND CONCLUSION

Lasair has been operating for years with the ZTF alert stream, has several hundred user accounts, and is ready for the greater challenge of LSST, with about 30 times the alert rate. A user of Lasair does not simply take a ready-made sub-stream of classified alerts; rather, they combine different elements from a kit. The kit has the Sherlock association engine to find what is already published about a specific sky location; it has watchlists, so a user can ask for alerts associated with a set of astrophysical sources; it has watchmaps to restrict attention to arbitrary sky regions; it has sophisticated features to characterize the light curve. Lasair allows users to do deep analysis of objects on their own machine and annotate Lasair with their

computed results and classifications. These different resources are then combined, using the SQL language, to create a filter, and the filter can run from a web-click, by API, or run in real-time streaming mode as alerts come in, so no time is lost, with machine-readable output. A user can publicly share (or not) their watchlists, watchmaps, annotations, and filters. Lasair is being upgraded to tackle gravitational-wave events, thus allowing users to pick the LSST alert most likely to be associated with the counterpart.

DATA AVAILABILITY

The Lasair software is available at Lasair Team (2023b). ZTF data are available at ZTF Collaboration (2018). LSST data rights are described at Rubin Observatory (2022).

ACKNOWLEDGEMENTS

Lasair is currently supported by the UKRI Science and Technology Facilities Council and is a collaboration among the University of Edinburgh, Queen’s University Belfast, and the University of Oxford (grants ST/X001334/1 and ST/X001253/1) within the LSST:UK Science Consortium. Lasair is hosted by the STFC IRIS academic cloud. Lasair relies on the ZTF survey, supported by the National Science Foundation under Grant No. AST-2034437 and a collaboration including Caltech, IPAC, the Weizmann Institute for Science, the Oskar Klein Center at Stockholm University, the University of Maryland, Deutsches Elektronen-Synchrotron and Humboldt University, the TANGO Consortium of Taiwan, the University of Wisconsin at Milwaukee, Trinity College Dublin, Lawrence Livermore National Laboratories, and IN2P3, France. Operations are conducted by COO, IPAC, and UW. The ZTF forced-photometry service was funded under the Heising-Simons Foundation grant 12540303. This research has made use of the NASA/IPAC Extragalactic Database (NED), which is funded by the National Aeronautics and Space Administration and operated by the California Institute of Technology.

REFERENCES

- Alam S. et al., 2015, *ApJS*, 219, 12
 Asmus D. et al., 2020, *MNRAS*, 494, 1784
 Barthelmy S., Racusin J., 2023, <https://gcn.nasa.gov/missions/lvk> (Accessed January 2024)
 Bazin G. et al., 2011, *A&A*, 534, A43
 Beckett G., Mann R., 2023, https://lsst-uk.atlassian.net/wiki/download/attachments/1146928/LUSC-B-46-D2.1.3_lusc_operational_plan.pdf (Accessed January 2024)
 Bellm E. C., 2023, <https://dmtn-118.lsst.io/> (Accessed January 2024)
 Bellm E. et al., 2018, *PASP*, 131, 018002
 Boch T., Baumann M., 2023, <https://aladin.cds.unistra.fr/AladinLite/doc/> (Accessed January 2024)
 Chambers K. C. et al., 2019, preprint (arXiv:1612.05560)
 Conway J. H., 1970, https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life (Accessed January 2024)
 Downes R. A. et al., 2001, *PASP*, 113, 764
 Fernique P., Boch T., Donaldson T., Durand D., O’Mullane W., Reinecke M., Taylor M., 2014, MOC - HEALPix Multi-Order Coverage Map Version 1.0, IVOA Recommendation 2 June 2014
 Flesch E. W., 2023, *Open J. Astrophys.*, 6, 49
 Förster F., Cabrera-Vives G., 2023, <https://alerce.science/> (Accessed January 2024)
 Gaia Collaboration, 2023, <https://www.cosmos.esa.int/web/gaia> (Accessed January 2024)
 Gal-Yam A., 2021, American Astronomical Society Meeting Abstracts #237, p. 423.05

- Graham M. et al., 2020, <https://dmtn-102.lsst.io/> (Accessed January 2024)
- Graham M., 2023, <https://docs.babamul.dev/> (Accessed January 2024)
- Ishida E. O., Möller A., Peleton J., 2023, <https://fink-broker.org/> (Accessed January 2024)
- Ivezic Z. et al., 2019, *ApJ*, 873, 111
- Jurić M. et al., 2023, Data products definition document. <https://lse-163.lsst.io/>
- Lasair Team, 2021, <https://pypi.org/project/lasair/> (Accessed January 2024)
- Lasair Team, 2023a, <https://lasair.readthedocs.io/> (Accessed January 2024)
- Lasair Team, 2023b, <https://github.com/lsst-uk/lasair-lsst/> (Accessed January 2024)
- Lasker B. M. et al., 2008, *AJ*, 136, 735
- Massaro F. et al., 2013, *ApJS*, 207, 16
- Matheson T. et al., 2021, *AJ*, 161, 107
- Nordin J. et al., 2019, *A&A*, 631, A147
- OpenCV Team, 2022, <https://pypi.org/project/opencv-python/> (Accessed January 2024)
- Raen T. et al., 2023, <https://pitt-broker.readthedocs.io/en/latest/> (Accessed January 2024)
- Ritter H., Kolb U., 2003, *A&A*, 404, 301
- Rubin Observatory, 2022, <https://www.lsst.org/content/data-rights> (Accessed January 2024)
- Russeil E. et al., 2024, *A&A*, 683, A251
- Schipani P. et al., 2018, Proc. SPIE Conf. Ser. Vol. 10702, Ground-based and Airborne Instrumentation for Astronomy VII. SPIE, Bellingham, p. 107020F
- Schwamb M. et al., 2018, *Res. Notes AAS*, 3, 51
- Sheng X. et al., 2024, *MNRAS*, 531, 2474
- Skrutskie M. F. et al., 2006, *AJ*, 131, 1163
- Smartt S. J. et al., 2015, *A&A*, 579, A40
- Smith K. W., Williams R. D. o., 2019, *Res. Notes AAS*, 3, 26
- Smith K. W., O'Donnell J., 2022, <https://www.zooniverse.org/projects/mrniboc/superluminous-supernovae/> (Accessed January 2024)
- Steer I. et al., 2017, *AJ*, 153, 37
- Swann E. et al., 2019, *The Messenger*, 175, 58
- Tachibana Y., Miller A. A., 2018, *PASP*, 130, 128001
- Véron-Cetty M. P., Véron P., 2010, *A&A*, 518, A10
- Williams R. D., 2022, <https://github.com/RoyWilliams/RGBMatrix> (Accessed January 2024)
- Williams R. D., 2023, https://roywilliams.github.io/papers/Bazin_Exp_Blackbody.pdf (Accessed January 2024)
- Wiseman P. et al., 2023, *MNRAS*, 522, 3992
- Young D. R., 2023a, neddly. Zenodo. <https://zenodo.org/doi/10.5281/zenodo.8037632>
- Young D. R., 2023b, Sherlock. Contextual classification of astronomical transient sources. Zenodo. <https://zenodo.org/doi/10.5281/zenodo.8038057>
- ZTF Collaboration, 2018, <https://www.ztf.caltech.edu/ztf-alert-stream.html> (Accessed January 2024)

SUPPORTING INFORMATION

Supplementary data are available at *RASTAI* online.

lasair_final.zip

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the article.

APPENDIX A: LIGHT-CURVE FEATURES

In addition to other object properties, light-curve (photometric) features are computed on each object, and users can build filters based on these. The Rubin Observatory will pre-compute many features (Bellm 2023): a set of features that comprehensively covers long-lived periodic and stochastic light curves. But some of the added value of Lasair is extra light-curve features tailored to explosive events – emphasizing very recent behaviour – either a long-term light curve that brightens significantly, or a new transient that has appeared recently.

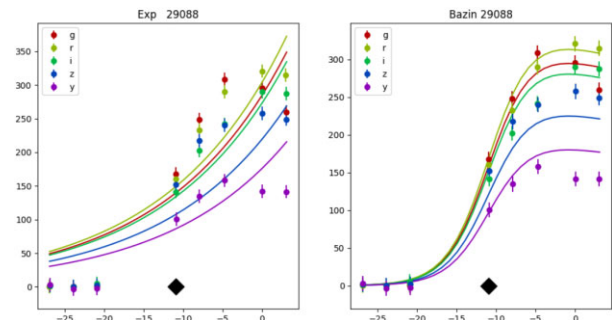


Figure A1. Early light curve simulated in six filters, fitted with both exponential (left panel) and Bazin (right panel) for the time dimension and a constant-temperature blackbody in the wavelength dimension. In this case, the best-fit criterion chooses the Bazin fit.

Non-parametric features have no underlying model; rather, they are statistics, such as mean, median, max flux, moving averages, rate of change of flux, and so on. Lasair includes these and also has a feature `fluxJump` which crudely looks for brightening: first take all fluxes together and compute the standard deviation of this early flux, then compute how many standard deviations different is the current flux. If a user wants to know about sudden brightening, this a first criterion.

Parametric features imply fitting a model to the data. In the time dimension such fits could include generic models like linear, polynomial, or exponential, as well as more physically motivated models such as microlensing, eclipsing binary, or Bazin fits (Bazin et al. 2011). Many proposed algorithms work with a monochromatic light curve, or use the word ‘colour’ to imply there are only two filters whose fluxes can be subtracted. But LSST provides six filters, so perhaps we should not simply build separate features for each filter, but think of the ‘time-wavelength surface’. Lasair utilizes this two-dimensional (2D) space to build a simultaneous blackbody fit with either a Bazin curve or a plain exponential in time, as in Williams (2023). Fig. A1 shows two fits to some simulated six-filter fluxes, on the left is a product of exponential rise in time and blackbody in wavelength, on the right is Bazin in time with blackbody. The diamond marks the discovery epoch. A goodness-of-fit criterion picks the best fit as the right panel. This 2D approach to fitting has been used later in other work (Russeil et al. 2024).

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.