# D3.11.2: Demonstration Software for One Example Catalogue

## WP3.11: Cross-Matching and Astrometry at LSST Depths

| | |
|---|---|
| **Project Acronym** | LUSC-B |
| **Project Title** | UK Involvement in the Legacy Survey of Space and Time |
| **Document Number** | LUSC-B-11 |

| | |
|---|---|
| **Submission date** | 7/Jan/2021 |
| **Version** | 2.0 |
| **Status** | Final |
| **Author(s) inc. institutional affiliation** | Tom J. Wilson (Exeter)<br>Tim Naylor (Exeter)<br>George Beckett (Edinburgh)<br>Mike Read (Edinburgh) |
| **Reviewer(s)** | Bob Mann (UEDIN),<br>Raphael Shirley (SOTON) |

| **Dissemination level** |
|---|
| Public |

## Version History

| Version | Date | Comments, Changes, Status | Authors, Contributors, Reviewers |
|---|---|---|---|
| 1.0 | 07/01/21 | First draft for review | Tom J. Wilson |
| 2.0 | 22/02/21 | Update with reviewer comments | Tom J. Wilson |
| | | | |
| | | | |
| | | | |
| | | | |

# Table of Contents

**Version History** 2

**1 Introduction** 4

**2 Demonstration Software** 4
2.1 Deliverable Name . . . . . 4
2.2 Preliminary Software . . . . . 5

**3 Deliverables** 5

**4 Future Work** 7
4.1 Data Access Centre Interactions . . . . . 7

# List of Figures

# List of Tables

# 1 Introduction

WP3.11 is investigating the astrometry of LSST. An unprecented parameter space in optical photometric surveys, the depths to which the Rubin Observatory will probe during its 10-year initial survey will challenge data reduction and analysis tools. We are focusing on the problems surrounding the creation of value-added catalogues: those combining the LSST dataset with external, ancillary catalogues, to improve the science that can be done. To achieve this, we must identify those sources in common between the two catalogues which are truly one physical object on the sky, for which we have two detections. These must be, in turn, differentiated from sources which simply happen to appear very close in the sky to one another, but are two independent sources in the sky. This process of creating a single, composite dataset is the "cross-matching" of the two catalogues.

The challenge for LSST is primarily one of sheer source counts. Never before has an optical survey reached the expected depths of LSST – 27th magnitude at its full completeness limit – and thus never had to cope with the increase in source detection rate that goes along with such sensitivity. And yet the survey will still be limited to ground-based resolutions, with the atmospheric distortions that "spread" out otherwise point source objects into a larger pool of light on the CCD of the telescope. When crowding of sources, as we expect for LSST, becomes significant, these "point spread functions" (PSFs) can and will start to overlap one another. Thus a very bright source can "hide" within its PSF a faint object, which the detection algorithms will fail to include in the final catalogue of objects produced by the "LSST stack". Unfortunately, the faint source can influence the measured position of the bright source, even if it was not detected, by subtly tugging on the centroid of the bright object during its position determination.

Usually a photometric catalogue records the measured position of a source, and a corresponding uncertainty: the precision with which the algorithms and pipelines were able to pinpoint the determined position. For LSST, these precisions will be very good, and thus the positions will likely be quoted to high confidence. Crucially, however, these "tugs" from faint sources, buried within the light of the brighter source, can be larger than the precisions with which sources are pinpointed, and thus the simple questions asked by traditional cross-match algorithms "given the positions and corresponding precisions of these two sources, how likely are they to have the same sky position?" will break down. It is overcoming this limiting factor, expected to be significant for LSST, that WP3.11 is most interested in.

# 2 Demonstration Software

To enable more robust cross-matches of LSST and other catalogues, including the effects of position perturbation from blended objects, WP3.11 is mainly tasked with the creation of new software to allow for such cross-matches to occur. These matches will then be hosted on the UKDAC, accessible to users. Thus, after we had established it was possible to overcome or sidestep a few implementation challenges (see the D3.11.1 report for more details), we began the establishment of the codebase core, on to which the more advanced, and WP-focused, algorithms could be added.

## 2.1 Deliverable Name

The formal deliverable for D3.11.2 is titled "Demonstration Software for One Example Catalogue". This is because, during the proposal phase for this WP, it was envisaged that demonstration code would be written to specifically match an example catalogue pair, the code then

moved to run on the DAC and subsequently generalised to allow for any catalogue combinations in D3.11.3. In practice we thought it would save time to go directly to writing the software for the DAC, skipping the demonstration software. Thus D3.11.2 should really be "Demonstration of Preliminary Software for DAC Integration". The software that forms D3.11.2 is an end-to-end complete cross-match code, installable by the end user, featuring documentation, but does not yet have the sophisticated algorithms which are WP3.11's main task and will be the focus of D3.11.3.

### 2.2 Preliminary Software

The software that delivers D3.11.2 is a full, end-to-end, "many-to-many" cross-match code, capable of accepting two catalogues and producing posterior probabilities of likely matches and non-matches between "islands" – sources potentially astrometrically correlated with one another and definintely not correlated with any other source – of sources across the two catalogues. This limit to two catalogues, but allowing for inclusion of multiple matches simultaneously, is what makes it a "many-to-many" match software. It accepts the astrometric uncertainty – the precision with which sources were centroided – and uses the normalised on-sky separation of two sources to calculate their relative match likelihood. This simple match assumption – the natural assumption of most match algorithms, ignoring any other systematic effects which might cause separation between sources, such as proper motions or blending – and the non-inclusion of any other discriminating information that might be used to distinguish between two competing counterparts for a source mean that this implementation is currently a "naive Bayes" cross-match. However, these assumptions hold for a large number of previous generation photometric catalogues, and even in its half-finished state the cross-match codebase provides useful functionality.

## 3 Deliverables

The main deliverables for this work, being software development, are available online. As per the Project Management Plan, software development should be maintained in a version control repository. The main codebase is therefore located at https://github.com/Onoddil/macauff. It features a full test suite for validation, as well as functionality to generate test data to input into the cross-match algorithm for end-to-end verification. The folder structure is as follows.

- Top-level files.
  - CHANGES.rst: changelog file, itemising the updates to the codebase.
  - LICENSE and README.md: details of licensing of codebase, and a top-level overview description of the software.
  - MANIFEST.in, pyproject.toml, and setup.cfg: minor files that aid with the setup and installation of the Python package.
  - python-package.yaml: within the `.github/workflow` folder, this details the `GitHub` virtual environment setup and the running of the test suite.
  - setup.py: main installation file for the Python package, which describes the various dependencies for installing the package, and controls the compiling of `fortran` code.
  - tox.ini: configuration file describing setup of the `tox` environment for test purposes.

- Documentation files, `docs/`.
  - conf.py: configuration file, for Python automated creation of documentation.
  - *.rst: files containing the raw text that makes up the documentation of the codebase.
- Code files, `macauff/`.
  - __init__.py: File used during the installation process, indicating which files and functions should be importable.
  - counterpart_pairing*: Python and Fortran code (`.py` and `_fortran.f90` respectively) to run the cross-match assignment functionality and calculate cross-match probabilities.
  - group_sources*: Python and Fortran code used in the generation of "islands" of potential counterparts, independent of other sources in the respective catalogues.
  - make_set_list.py: Python script used in group_sources.py to derive "sets" of source overlaps on astrometric considerations, based on lists of sources near to each object.
  - matching.py: Python code handling the overall cross-match process.
  - misc_functions*: Python and Fortran code that is used in multiple places throughout the cross-match process, and hence cannot be kept within any single script.
  - perturbation_auf.py: Python code to handle the creation of "dummy" arrays for the perturbation component of the Astrometric Uncertainty Function. Main algorithms currently not implemented, and included for future backwards compatibility.
  - photometric_likelihood.py: Python code to handle the creation of source match likelihoods on photometric grounds. Also currently not implemented for the most part, and featuring backwards compatible code.
  - shared_library.f90: Similar to misc_functions, contains subroutines that are needed across other Fortran modules, and are therefore accessible from those other files.
  - Test files, `tests/`.
    - __init__.py: same as in the parent folder.
    - test_*.py: Unit test scripts, one per Python script in the folder above, to ensure consistency and accuracy of the main codebase.
    - Data files, `data/`.
      - *.txt: example files of the configuration files used in the cross-match process.

In addition, to aid in the review, preliminary documentation – guides to installation and getting started with the codebase – is available at https://onoddil.github.io/macauff/. This documentation, formed from the raw files in `macauff/docs` in the repository, is structured as follows.

- Homepage: brief description of module and links to various starting pages.
- Installation: details the installation process of the module, its dependencies, and how to run the test suite to ensure successful installation.

- Quick Start: description of how to begin working with the package, describing the necessary files to run a cross-match, and giving examples of how to run matches between two catalogues.
- Inputs: more in-depth descriptions of each parameter that should be specified in the input configuration files used in the cross-match process.
- Documentation: collection of available functions within both the Python and Fortran code in the codebase, describing the inputs and outputs from each function or subroutine in detail.

# 4 Future Work

The main focus of D3.11.3, now that WP3.11 has an established codebase for its software development aims, is to implement the extended algorithmic aspects of its remit. First, we must include the algorithmic components that extend the Astrometric Uncertainty Function from the simple assumption that centroiding uncertainties are the only component, and include a prescription for the uncertainty in position that derives from the perturbation of a bright source by a fainter contaminant source within its PSF. Second, we plan to include an additional improvement to better provide users confident cross-matches: the use of the photometry of the sources to distinguish between true coevality in sources and happenstance astrometry; this can be achieved using the idea that as an ensemble sources of 15th magnitude in one optical bandpass are going to have a roughly 15th magnitude source near to them in a similar optical wavelength, but unlikely to experience a 22nd magnitude source being coeval with them.

In addition, we plan to extend and improve the user documentation, both for the current features and for the additional features to be implemented. This is crucial to ensure that the codebase is useable by the LSST:UK project after the end of this current work package.

## 4.1 Data Access Centre Interactions

One major aspect of WP3.11, which sits in parallel with the deliverables, is the interactions between the "Dev" (WP3.11) and "DAC" (WP2.5) teams. Our significant output is so-called Rubin "User Generated Products" – extra products which use and extend Rubin data. We therefore require the DAC to host these products such that users can interact with them once we have generated them for a given catalogue-catalogue combination. The workflow also requires the generation of appropriate input tables for the software, which will take the two input catalogues and reduce them to a canonical format for use in the WP3.11 software.

This interaction between WP2.5 and WP3.11 necessitates the formation of workflow documentation, such that these interactions can continue beyond Phase B and WP3.11. We are in the process of formalising these interfaces between the DAC team, WP3.11, and the community. We attach here a draft overview of the in-progress interface document, as the interactions between the DAC and WP3.11 were moved forward to be part of the D3.11.2 deliverable during the rearrangement of the package (see section 2.1 for a brief outline of the change). Formally this interface document should be a part of the D3.11.3 deliverable, however, and hence the full interface document will be finalised in due course.

# 2 Work Package Scope and Implementation

Work Package 3.11 will produce three types of output:

- **Cross-match Algorithm** – the team will develop and publish a novel algorithm for more accurate identification of cross-matches between catalogues, especially targeted at the challenging case of crowded fields. Specifically, the algorithms will allow objects in the LSST object catalogue to be matched confidently to corresponding objects in other, relevant, astronomy catalogues.
- **Cross-match Software** – the team will produce a software implementation of the cross-match algorithm that is capable of handling LSST-scale catalogues in production, as part of the UK Data Access Centre.
- **User-generated Products** – the cross-match software will be applied to a number of different third-party catalogues (to be determined) to create new datasets that capture the correspondence between LSST objects and their counterparts in the specified third-party catalogues in a form that can easily be incorporated into end-user analysis tasks. To maximise their relevance, these new datasets will be produced in a timely manner, once all of the relevant input data is available.
  Because LSST data products will not be available until after the end of WP3.11, the team will only demonstrate the ability to produce these User-generated products, based on representative precursor surveys and/ or LSST data previews.

For each third-party catalogue, a four-step workflow is envisaged:

1. A survey catalogue, to be cross-matched, will be identified.
2. Both the LSST data and the third-party catalogue will be pre-processed into a canonical format suitable for ingestion into the cross-match software. In part, this will involve extracting (reducing) the essential information needed to perform the cross-match.
3. The Cross-match Software will ingest and analyse the survey inputs, creating a new dataset that describes identified correspondences between the two surveys, as well as identifying those sources for which no identified counterpart exists.
4. The cross-match dataset will be added to the UK DAC, in a way that is accessible and useful to DAC users for their science.

Each of these four steps involves one or more interfaces, which are described in more detail in the next section.

The length of the LSST survey (expected to be over 10 years, during 2023—2033) and because, at the time of writing, the WP3.11 team only have funding to develop the cross-match algorithm and software during Phase B (effectively, until 2023), consideration needs to be given to the sustainability of the software.

Over the lifetime of the survey, various changes that will affect the cross-match software (and, possibly, algorithm) should be anticipated:

- The platform on which the software is run will change. Not only will hardware be deprecated and replaced, but also the form of the hosting platform and supporting technologies will change.
- It is possible that bugs will be identified in the software (or algorithm) beyond the end of WP3.11, which need to be fixed or worked around.
- It is possible that new functionality will be required, beyond the end of WP3.11, to address unanticipated opportunities and applications.

Software sustainability, without a funded development team, is very challenging. It is not possible to address the issue completely in this document, but there are steps that can and should be taken to reduce the potential threats, which include:

- It becomes impossible to use the software, because it becomes incompatible with available hosting platforms in the UK DAC.
- It becomes impossible to use the software, because bugs that make the outputs meaningless, cannot be resolved because the understanding of the software and algorithm has been lost.
- It becomes impossible to reproduce/ verify science outputs, because the conditions under which User-generated Products have been produced is not determinable.

To address this, the UK Data Access Centre is taking a number of steps, summarised as follows:

- Software Preservation is at the core of LSST:UK planning and provisioning.
- Software sustainability and accessibility is one of the criterion by which LSST:UK software deliverables are assessed.
- Where possible, software will be developed using an Open Source philosophy, to promote community engagement in the use and development of the software, and to avoid crucial information being inaccessible.
- Where practical to do so, open standards and/ or widely supported interfaces and technologies will be adopted in preference to proprietary or custom interfaces.
- Understanding of the software, and underpinning algorithms, is shared between multiple people, to reduce the risk due to staff movement and consequent knowledge loss.
- Algorithms and software will be supported by design and implementation documentation that is sufficient to allow the work to be reproduced (assuming effort is available to do so).

Key steps that should be taken for Work Package 11, to support these tactics is described in Section 3.

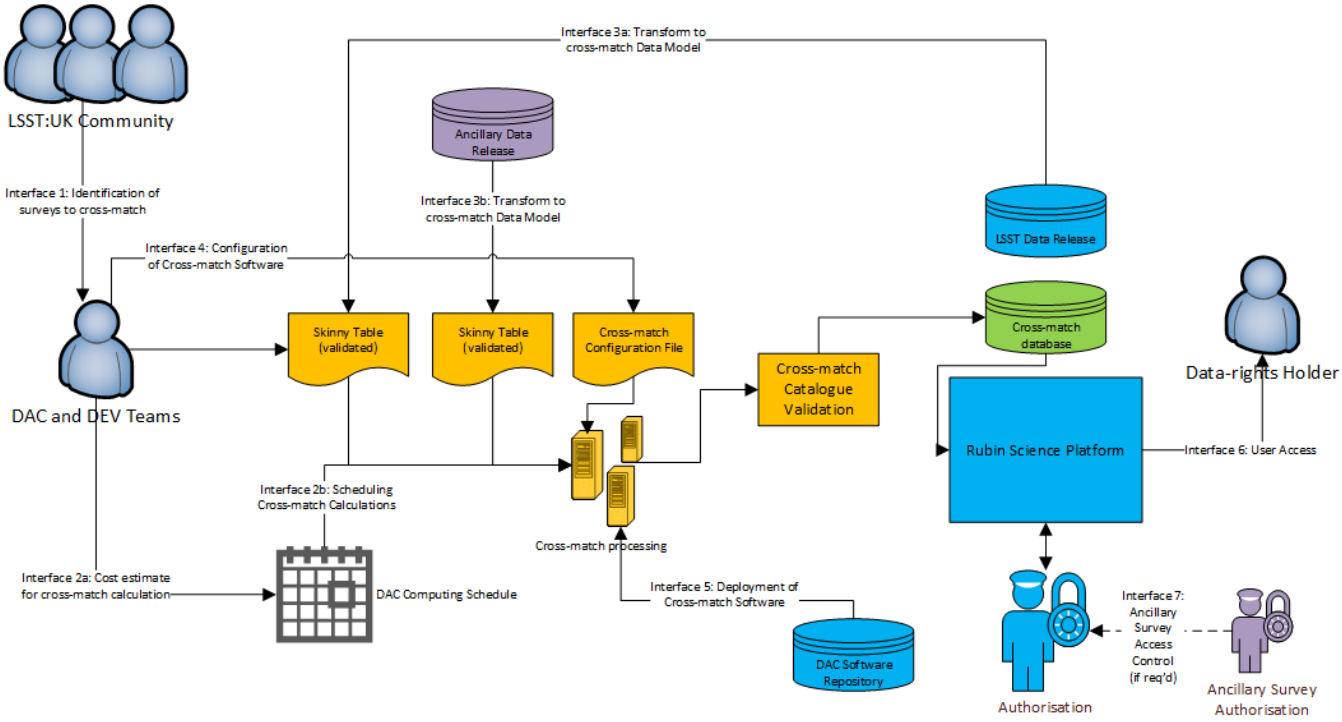


**Figure 1: Workflow for generating a Cross-match Catalogue.**