D3.11.3: Demonstration of Full DAC Integrable Software

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-16

Submission date	28/4/2021
Version	2.0
Status	Final
Author(s) inc. institutional affiliation	Tom J. Wilson (Exeter) Tim Naylor (Exeter)
Reviewer(s)	Bob Mann (UEDIN), Raphael Shirley (SOTON)

Dissemination level
Public

Version History

Version	Date	Comments, Changes, Status	Authors, Contributors, Reviewers
1.0	28/04/21	First draft for review	Tom J. Wilson
2.0	24/05/21	Updated with reviewer comments	Tom J. Wilson

Table of Contents

List of Figures

List of Tables

1 Executive Summary

WP3.11's primary objective is to create a cross-match service, to identify sources in common between two photometric catalogues and allow for their detections in each dataset to be combined. In D3.11.2 we outlined the main infrastructure of this codebase, allowing for a full "many-to-many" match between two catalogues, for a simple match setup, purely considering the positions and centroid uncertainties of each source. D3.11.3 extends the options available in the cross-matches to include two important aspects for the matching of LSST to ancillary datasets, mostly infrared catalogues such as *WISE*.

First, we now include the photometric information in the matching process, using the ensemble common coevality of sources in the same area of the sky to establish "in situ" magnitude-magnitude relationships between the two datasets. These colours can then be used to break ties in matches where two sources are similar distances from a source in another catalogue. Here, the "better" colour object should be chosen over the object unlikely to be a counterpart given the brightness of both it and the other source.

Second, we include a more complete description of the Astrometric Uncertainty Function – the description of the probability that a source has some "true" position given where we measured it to be. An important aspect for matching both lower angular resolution datasets in the infrared, such as *WISE*, and for crowded LSST fields, we include as a component of the AUF astrometric perturbations to the measured positions of objects by hidden, contaminating sources within the PSF of the brighter object. These unresolved objects can nudge the center-of-light of the centroid of the (composite) source, sometimes by several times the precision to which the bright object was centroided (the key metric in the "simple" AUF, assumed to be Gaussian).

With both of these components in place, we now have a robust and accurate cross-match tool, and will be able to provide LSST:UK users with robust matches between LSST and other photometric catalogues.

Where relevant, we will include reference to the Science Requirement Document requirements for WP3.11, R11.X.

2 Introduction

WP3.11 is tasked with creating a codebase that performs cross-matches – identification of sources across two datasets that are the same on-sky object observed twice – between LSST releases and a set of ancillary catalogues, mostly infrared datasets such as *WISE* (R11.1, R11.3, R11.5-7). We began with the task of sidestepping the "completeness depth crisis", in which simulations of the Galaxy would not be faint enough to allow us to model faint LSST objects correctly, which was dealt with in D3.11.1 (R11.1). Then, in D3.11.2, we implemented a "simple Bayes" cross-match service (R11.3, R11.6), a "many-to-many" match service which made a few key assumptions about the nature of the detections in each photometric catalogue, but allowed us to lay down the framework of the codebase. We refer the reader to the D3.11.2 report for an overview of the cross-match process, and our previous implementation.

2.1 Algorithm Extensions

Here we have extended the codebase with additional algorithms which relax the limiting assumptions made in D3.11.2. We have made two significant improvements to the algorithms used, or available to be used, in the cross-matching of photometric catalogues (R11.5).

First, we have extended the matches to consider both the photometry and astrometry of the sources when assigning probability of (non-)match, where previously we only considered the astrometric positions. This allows for the rejection of serendipitous false matches, randomly placed objects which may have no counterpart in the other catalogue (due to the faintness limit of each catalogue, for example), based on the ensemble magnitude-magnitude relationship between sources in the area of the sky in question. This is crucial for LSST, where chances will be very high at its unprecedented faintness limit that a random object will appear close enough, just by the sheer number of sources observed, to be considered a potential counterpart to a source.

Second, we have improved the "Astrometric Uncertainty Function", the function describing the probability of a source truly having originated from a particular sky location given where we measured it to be, to include the effect of hidden contaminants. Here we have included the center-of-light tugs that these blended objects impart on the brighter source, which can in certain scenarios overwhelm the "centroiding" uncertainty of the bright source. This is also important for LSST – and its successful matching to IR catalogues – as both LSST and external catalogues such as *WISE* will suffer non-negligible levels of crowding in large parts of the sky, meaning this component cannot be ignored.

3 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. Crucially, however, we needed to include the additional considerations detailed above to the cross-match process to allow for robust matching at faint brightnesses and in crowded fields. Thus, after we implemented the main code infrastructure in D3.11.2, we began the inclusion of the more complex algorithmic code to perform the improved cross-matches for crowded fields.

3.1 Full Software Implementation

The software that delivers D3.11.3 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" of sources across the two catalogues. It accepts the astrometric uncertainty – the precision with which sources were centroided – and uses the on-sky separation of two sources to calculate their relative match likelihood. This aspect of the codebase is unchanged from D3.11.2; for further details please see the D3.11.2 report.

Importantly for matching faint LSST data, and matching to lower angular resolution datasets such as *WISE*, the code now includes a fuller description of the AUF, incorporating simulated perturbations due to blended objects. This is combined with the "centroiding" uncertainty description to provide the likelihood of two objects being two detections of a single on-sky source given their separation. We now take into account both their respective astrometric uncertainties but also the local density of sources around them and their quoted magnitudes to derive a statistical description of the level of additional astrometric offset between the two objects in question.

It also includes the photometric likelihood, deriving from the "in situ" data the overall probability of two sources with respective measured magnitudes being coeval on the sky. This also crucially takes into account the effect of bright objects blocking out detections of faint objects near them on the sky, and avoids the common issue in which faint "field" sources are over-subtracted from the observed distribution of sources surrounding bright objects to calculate the "counterpart" distribution.

4 Deliverables

The main deliverables for this work, as with the D3.11.2 report, 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*: Python and Fortran code to handle the creation of the perturbation component of the Astrometric Uncertainty Function.
  - photometric_likelihood*: Python and Fortran code to handle the creation of source match likelihoods on photometric grounds.
  - 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.
    * test_full_match_process.py: An additional test script, which also includes functionality to generate a "dummy" dataset for testing the end-to-end capability of the 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.

4.1 Comparisons Between Match Algorithms

Now that we have a complete cross-match algorithm, it is useful to compare the impact each new algorithmic aspect has on the matches found. To do so we turn to our most well understood matches, those of *Gaia* DR2 and *WISE* in the Galactic plane, around $130 \leq l \leq 132$, $0 \leq b \leq 1$. The distribution of separations between returned matches is shown in Figure 1, for four combinations of "including the perturbation component of the AUF" and "including the photometric likelihood information". In addition, we show the official *Gaia* DR2-*WISE* cross-matches (Marrese et al., 2019), the so-called "best neighbour" matches, for the same region of the Galactic plane.

Here we see the main difference in match distance is whether we include the perturbations in the description of the positions of the sources – the *WISE* sources mostly, but we model *Gaia*'s crowding as well technically – with the photometric information a second-order effect. Without the knowledge of the extent that *WISE* object positions are tugged by hidden contaminants, we would miss a significant number of matches – crucial for crowded LSST fields as well. These perturbations extend the "allowed" separations by a factor ~ 3 compared with the non-inclusion of this AUF component (cf. the black and red solid lines in Figure 1). While *WISE* is the "poster child" for this effect, we expect LSST to suffer similar order of magnitude levels of crowding, certainly compared with its centroiding precision, towards the faint end of the catalogue.

However, as *Gaia* and *WISE* are well-matched in terms of dynamic range and completeness, the photometric information is not an obvious effect. We see it in the comparison between "sb_run" and "phot_run", and "perturb_run" vs "full_run", where some small fraction of the time there is a nearby, faint source in the opposing catalogue with large positional uncertainty. This source therefore has a smaller normalised separation to the source in the second catalogue, and a purely astrometric match would select it with higher probability; using the colours of the sources as an ensemble over large areas of the sky allows for these to be revealed as false matches, removing them from consideration. Alternatively, it can simply just be the case that two sources are near one another, but neither has a counterpart in the other catalogue (which would be the closer astrometric match), in which case the colour check tells the true story, where both of the objects should return no match, instead of being matched with one another.

We also show, as the black dashed line, the distribution of matches from the official *Gaia* cross-match service. This service uses a purely astrometric match, in a one-to-one (likelihood ratio) configuration. The matches provided by *Gaia* should, in theory, follow roughly the same distribution as our "sb_run" setup – with some small effect from our many-to-many "island" probability vs. their likelihood ratio – but as can be seen in Figure 1 the *Gaia* matches extend to slightly larger separations. This is primarily due to the fact that they broaden the uncertainties of objects with no known proper motion to attempt to account for the five-year baseline of object drift between the two catalogues. We do not include any additional terms, however, and hence faint objects with high proper motion – which we assume to be a negligible number – would fall

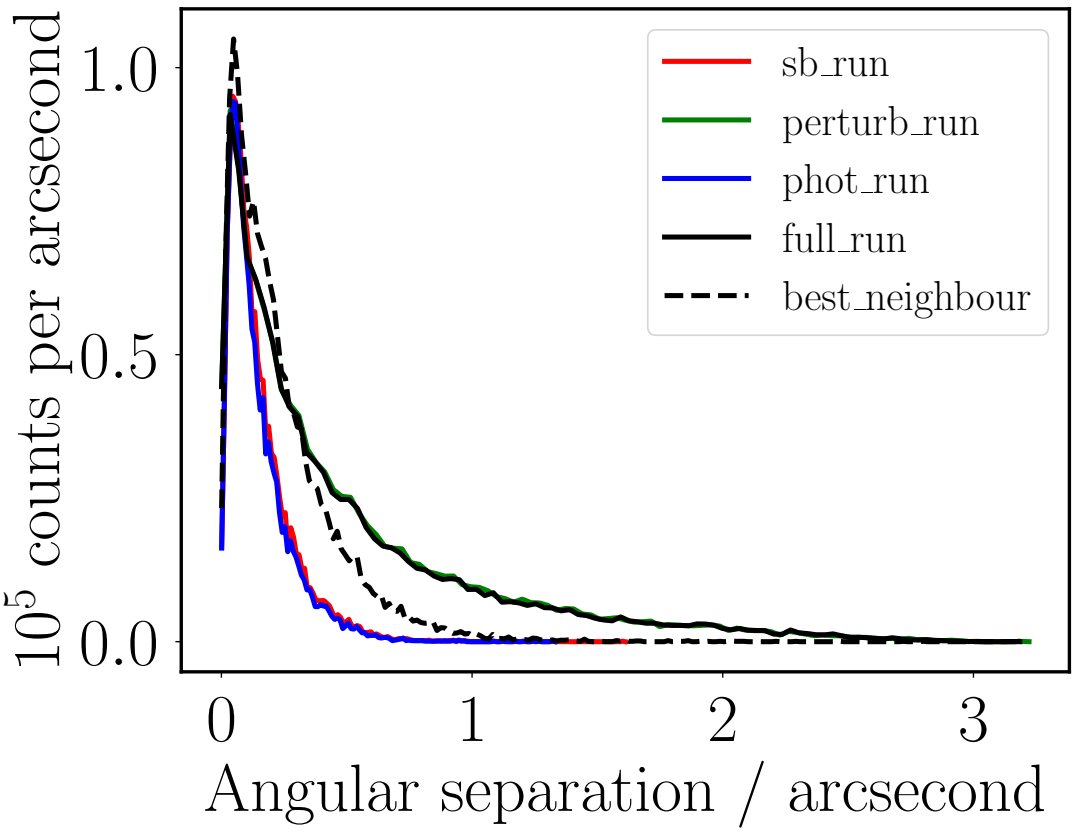

Figure 1: Match separation distribution for a series of cross-matches. The solid lines are WP3.11 matches for various setups: the "simple Bayes" cross-matches established in D3.11.2 ("sb_run", red); matches using the simple AUF (equivalent to the "simple Bayes" AUF) and including the photometric likelihood ("phot_run", blue); matches with the AUF which includes the component for perturbations due to blended sources, but no photometric likelihood information ("perturb_run", green); and matches with both the additional AUF component and photometric information ("full_run", black). Additionally, we show the *Gaia* "best neighbour" matches (Marrese et al., 2019) for the same area of the sky (dashed black line).

out of our matches, if that were the only component of the AUF that had not been accounted for. Instead, however, this additional padding of the *Gaia* uncertainties by five years' proper motion effectively increases the acceptable matching radius, when compared with the likelihood of "random" position alignment based on density. This allows for the accidental recovery of some *Gaia*-*WISE* matches; sources that are, instead, *WISE* objects significantly perturbed by contaminants. Ultimately, however, the full description of the AUF to include the dominant AUF component, in the case of *WISE*, recovers these extra sources, and of order 20-50% more than those provided by the *Gaia* cross-match service.

5 Future Work

This codebase implements a full photometric catalogue cross-match service, including the unique extensions to the algorithms necessary to match LSST to the required infrared ancillary cat-

alogues: photometric likelihood consideration, to reject false matches, and the perturbation component of the AUF, to include the effects of hidden contaminants in crowded fields on the astrometry of the recorded sources. However, it does not implement the full extent of the potential new algorithms described in D3.11.1 (R11.1), in which we describe a more accurate calculation of the center-of-light tugs from sources in the case where the sky background dominates the noise of a detection. The implementation of this algorithm, no longer being entirely forwarded modelled like the current implementation, is dependent on the efficient derivation of functionality from the data themselves, and hence requires further vetting before it can be added to the codebase.

We will include this suite of functions, used to derive the data-driven dependencies for the perturbation component of the AUF, in the codebase in the near future, ensuring that new catalogues can be easily set up to be cross-matched in the future (R11.6-7). In addition, the documentation will be expanded and improved to ensure that its functionality can be understood, improved, and fixed if necessary after the end of the current work package (R11.2).