

Fast Detection and Classification of Planetary Anomalies in Roman Microlensing Light Curves

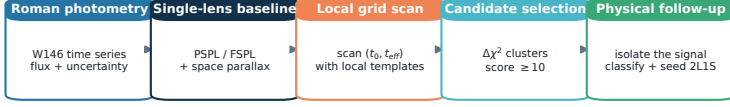
Kansuke Nunota (University of Osaka) and the RGS PIT WG5 Anomaly Search Team



1. Motivation

- Roman will deliver survey-scale, high-cadence microlensing light curves.
- Planetary perturbations are short-lived, diverse, and difficult to identify manually.
- We develop a fast front-end detector to find local residual anomalies before computationally expensive 2L1S modelling.

Anomaly-search pipeline



2. Local template scan

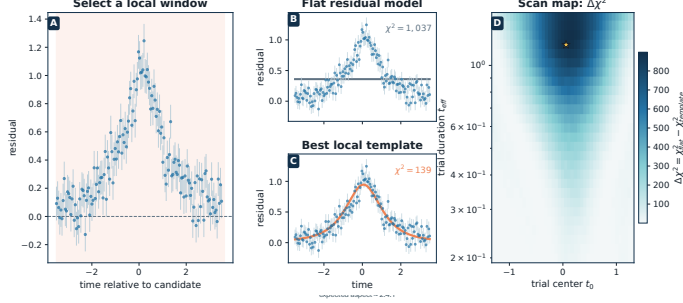
$$Q = 1 + \left(\frac{t-t_0}{t_{\text{eff}}}\right) \quad A(t) = \frac{1}{\sqrt{Q}} \text{ or } \frac{Q+2}{\sqrt{Q(Q+4)}}$$

window: $t_0 - 3 \times t_{\text{eff}} < t < t_0 + 3 \times t_{\text{eff}}$

$$\Delta\chi^2 = \chi^2_{\text{flat}} - \chi^2_{\text{template}}$$

$$\text{score} = \frac{\Delta\chi^2_{\text{best}} - \text{median}(\Delta\chi^2)}{\text{std}(\Delta\chi^2)}$$

Local template scan



3. Candidate quality diagnostics

Large $\Delta\chi^2$ values can be driven by a single high-weight point. The detector therefore stores per-candidate support metrics and grid-level diagnostics for filtering and interpretation.

- n_{eff} : effective number of contributing points
- peak_frac : strongest-point contribution fraction
- ρ_1 : time-correlation structure
- longest_run : Longest consecutive run of above-threshold contributing points.

Implementation

The computational core of **jacscanomaly** is implemented in compiled C++ for low-memory, high-throughput processing of Roman-scale light curves. A lightweight Python interface is used for data input, configuration, candidate extraction, and visualization.

Input: time, flux or magnitude, and uncertainty

Output: location, duration, strength and quality diagnostics

```
from jacscanomaly import CandidateCriteria, Finder, FinderConfig

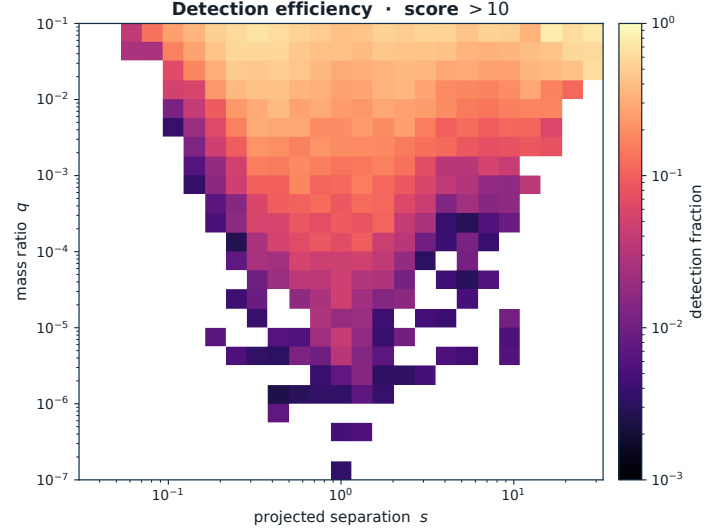
config = FinderConfig(
    fitter_kind="pspl",
    candidate_criteria=CandidateCriteria(min_n_eff=2.0),
)

finder = Finder(config)
result = finder.run(time, flux, flux_err)

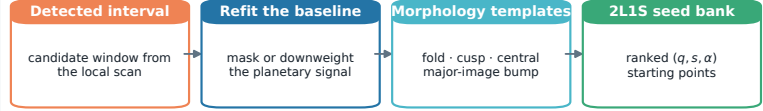
result.print_summary()
finder.plot_result()
```

4. Detection efficiency

- 100,000 Roman light curves were simulated with GULLS.
- q , s , and α were sampled uniformly over the adopted ranges.
- Each light curve was processed with the same anomaly-search pipeline.
- A planet was counted as recovered when score > 10 .
- The detection efficiency in each q, s bin was calculated.



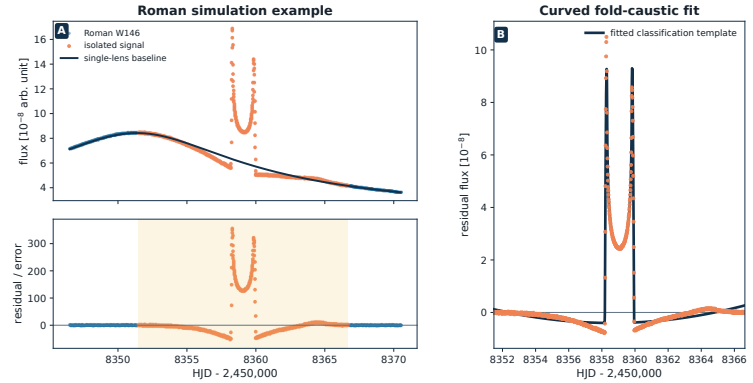
5. From detection to 2L1S seeds



- Planetary anomalies are local perturbations to a smooth single-lens event.
- Fold, cusp, and image-perturbation geometries produce characteristic residual shapes.
- A dictionary of physically motivated templates provides morphology, timing, and approximate physical constraints without a full binary-lens fit.
- The resulting parameters are used to generate several plausible 2L1S seeds —not final physical measurements.

6. Roman simulation example

We apply the pipeline to a simulated Roman W146 microlensing event containing a planetary caustic crossing.



Conclusions

Fast candidate detection

A C++-accelerated local residual scan identifies short planetary anomalies in Roman-scale light curves without performing a full 2L1S fit on every event.

Quantified survey sensitivity

Using 100,000 GULLS simulations with uniformly sampled (q, s, α) , we measure the recovery fraction across the (q, s) plane for a fixed threshold of score > 10 .

Physics-informed follow-up

Local morphology templates encode characteristic fold, cusp, and image-perturbation shapes, providing ranked—and potentially degenerate— (q, s, α) seeds for targeted 2L1S modelling.