Functionality of TOM Systems
TOM's role in a wider system

"Here's an event"

LSST → Other surveys

ANTARES
Aggregates alerts with catalog data
Classifier
Catalog Databases

Alert Database
Non-alert Database

"Send me everything matching these criteria"
"Here's what I learned"

"Observe X with parameters Y"
"Tell me status of X"
"Send me data of X"

Scheduler
Observation Interface

Tel 1 → Tel 2 → Tel 3

Data Interface
Archive 1
Archive 2
Archive 3
Following a use-case to motivate a discussion of functionality

**Microlensing events**
- unpredictable, rapidly evolving transients
- high target volume
Gathering info on targets

- Where do my targets come from?
- What format?
- How do I receive/fetch these targets?
- How frequently do I receive updates?

Build on existing work
### Example TOM Database

Some parameters are always needed:
- Name
- RA
- Dec
- $A_0$
- $t_0$
- $t_e$
- $I_{[\text{mag]}}$
- $\Delta t_{[\text{d]}}$
- Model $[\text{mag}]$
- Latest data
- Cadence $[\text{hrs}]$
- Priority
- Survey
- Observers

Other information is science-specific:
- Recommended
- Spitzer
- Ground
- Visits/night
- KMT

#### TOM for 2015 Spitzer Microlensing Campaign

<table>
<thead>
<tr>
<th>Name</th>
<th>RA</th>
<th>Dec</th>
<th>$A_0$</th>
<th>$t_0$</th>
<th>$t_e$</th>
<th>Latest data</th>
<th>Cadence [hrs]</th>
<th>Priority</th>
<th>Survey</th>
<th>Observers</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB150079</td>
<td>17:52:20.48</td>
<td>-31:17:44.39</td>
<td>1.0000</td>
<td>2457219.83998</td>
<td>127.00</td>
<td>None</td>
<td>None</td>
<td>8</td>
<td>1</td>
<td>None</td>
</tr>
<tr>
<td>MB150117</td>
<td>17:55:10.24</td>
<td>-35:04:48.89</td>
<td>1.4117</td>
<td>2457165.23555</td>
<td>29.73</td>
<td>14.927</td>
<td>3.97</td>
<td>15.307</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>MB150237</td>
<td>17:57:52.27</td>
<td>-31:28:20.47</td>
<td>2.4084</td>
<td>2457224.60371</td>
<td>51.18</td>
<td>17.25</td>
<td>58.21</td>
<td>16.902</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>
A Common View of TOM Database

Important target info

Display graphics useful to observers

Team can post advice

Allow observers to indicate when they select the target
A less common View

RoboNet-II Pipeline Status


For help interpreting this page, please look here

Status of pipeline codes running under the crontab:

- pipemonitor.py Operational. Running every 30min
- dcontrol.py Operational. Running every 5min

Pipeline Config

Instructions on how to modify the selection of event data to be processed can be found under the Pipeline Operations Help

Currently processing data for RECENT events with data taken in the last 2 days

Ongoing Reduction Processes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MOA-2014-BLG-0425 cpt-domin-1m0-12-kb75_ip</td>
<td>0.006111111111111</td>
<td>1</td>
<td>65584</td>
<td>Locked</td>
<td>Running</td>
<td>OK</td>
</tr>
<tr>
<td>OGLE-2014-BLG-1586 cpt-domin-1m0-12-kb75_ip</td>
<td>0.003333333333333</td>
<td>1</td>
<td>65694</td>
<td>Locked</td>
<td>Running</td>
<td>OK</td>
</tr>
<tr>
<td>OGLE-2050-BLG-5121 1sc-domb-1m0-09-fl03_ip</td>
<td>0.014444444444444</td>
<td>1</td>
<td>65428</td>
<td>Locked</td>
<td>Running</td>
<td>OK</td>
</tr>
</tbody>
</table>

Reduction Status for all events
Target Selection

Need to interface with science-specific code

- How do I prioritize my targets?
  → *May be algorithm, manual, algo-human hybrid*
  → *May change over project lifetime*

- What information does this process need?
  What information does it return?
  → *Define interface*
  → *Enable scientists to use own selection code*

- What information do I need to display to users?

- What controls do users need?
Example Target Selection

RoboNet TArget Priority Generator

<table>
<thead>
<tr>
<th>Event</th>
<th>RA</th>
<th>DEC</th>
<th>Cadence OGLE, KMTNet</th>
<th>N.Exp</th>
<th>Texp [s]</th>
<th>Priority</th>
<th>Tsamp [h]</th>
<th>Imag</th>
<th>$\Omega_S$</th>
<th>$\sigma_{\Omega_S}$</th>
<th>$\Omega_S@\text{peak}$</th>
<th>AKA</th>
</tr>
</thead>
<tbody>
<tr>
<td>OGLE-2016-BLG-1446</td>
<td>17:45:18.38</td>
<td>-34:30:30.60</td>
<td>LO.1.0 MOA: yes</td>
<td>2</td>
<td>93</td>
<td>anomaly</td>
<td>2.00</td>
<td>15.0</td>
<td>7.70</td>
<td>0.42</td>
<td>8.57</td>
<td>OB161446,KB160470</td>
</tr>
<tr>
<td>OGLE-2016-BLG-0799</td>
<td>18:05:59.41</td>
<td>-28:06:13.20</td>
<td>LO.1.0</td>
<td>2</td>
<td>100</td>
<td>medium</td>
<td>4.00</td>
<td>15.0</td>
<td>8.57</td>
<td>0.08</td>
<td>18.90</td>
<td>OB160799</td>
</tr>
<tr>
<td>OGLE-2016-BLG-1291</td>
<td>18:11:35.21</td>
<td>-27:56:50.80</td>
<td>VLO.0.5</td>
<td>2</td>
<td>300</td>
<td>medium</td>
<td>4.00</td>
<td>17.1</td>
<td>7.91</td>
<td>0.08</td>
<td>14.31</td>
<td>OB161291</td>
</tr>
<tr>
<td>OGLE-2016-BLG-1664</td>
<td>17:56:33.60</td>
<td>-21:59:46.10</td>
<td>VLO.0.0</td>
<td>2</td>
<td>144</td>
<td>medium</td>
<td>1.75</td>
<td>15.4</td>
<td>7.02</td>
<td>755.61</td>
<td>8.66</td>
<td>OB161664</td>
</tr>
<tr>
<td>OGLE-2016-BLG-1066</td>
<td>18:11:35.20</td>
<td>-27:56:50.40</td>
<td>VLO.0.5 MOA: yes</td>
<td>2</td>
<td>300</td>
<td>medium</td>
<td>4.00</td>
<td>16.7</td>
<td>6.90</td>
<td>0.19</td>
<td>12.71</td>
<td>OB161066,KB160455</td>
</tr>
<tr>
<td>OGLE-2016-BLG-0985</td>
<td>18:01:47.20</td>
<td>-26:31:59.20</td>
<td>VLO.0.5</td>
<td>2</td>
<td>101</td>
<td>low</td>
<td>4.00</td>
<td>15.0</td>
<td>5.84</td>
<td>0.02</td>
<td>6.89</td>
<td>OB160985</td>
</tr>
<tr>
<td>OGLE-2016-BLG-1828</td>
<td>18:13:51.63</td>
<td>-26:26:24.30</td>
<td>VLO.0.5 MOA: yes</td>
<td>2</td>
<td>300</td>
<td>low</td>
<td>1.36</td>
<td>17.1</td>
<td>5.81</td>
<td>0.14</td>
<td>7.38</td>
<td>KB160569,OB161828</td>
</tr>
</tbody>
</table>

→ Draws current targets model parameters, returns priority, recommended observations
→ Pure algorithm selection, but users can override. Can be a problem!
Observation Control

For each selected target:

- What observations do I need right now?
- Which facilities do I need?
- How do I describe these observations?

→ Build on existing work

Need to interface with science-specific code
Observation Control

- Submit observations to telescopes
- Interfaces, response will vary
- Programs may need several interfaces:
  Manual  Remote  Robotic
Observation Control

- How can I specify my observations?
- What information do I need to include?
- What format is the information transmitted in? → build on existing work
- What information do I need back from the observatory?
Multiple Observing Interfaces

Spitzer Microlensing Program

MINERVA robotic telescope

Interactive interface for human observer

Different robotic facilities using different protocols

[Originally based on eSTAR agent]
Getting data back

- What data products do you need from the observatories?
- When do you need to get them?
- Drives archive & interface design

We can develop interfaces to common-user facilities which would benefit everyone
Data Reduction & Analysis

- What information do you need from your data?
- Does the observatory provide the data product(s) needed?
- When do you need to have the information?
- Do you need this information to decide your next observing strategy?
- How do you input the information to the TOM?
  → Define interfaces

Highly science-specific Facility dependant

Archive 1

TOM DB

Archive 2

Custom analysis
Example Data Reduction & Analysis

Instrument-signature corrected image from LCO

Photometry via Difference Image Analysis

Model fit to combined data-to-date

Timescale: mins-hours

Parameters used to
  - prioritize targets
  - describe next obs

TOM DB
**Coordination**

**Benefits**
- More efficient use of resources
- More efficient coverage of targets
- More observers bring more resources

**(Possible) Cons**
- Need to share target lists
- May need to share data
- Introduces complexity in prioritization
- Introduces complexity in interfaces
Coordination between TOMS

- Do users want to share information?
- What information do they need to share?
- When do they need to share it?
- How widely do they need to share it?
Coordinating Observations

Spitzer Microlensing Program

Coordinate with humans

Python harvester

MINERVA
Coordinate with robot TOM

Ground-based Follow-up Program
List of current observers: RoboNet

Add observer
Remove observer
Submit

RoboNet ObsControl

LCO API
Liverpool Telescope RTML
Coordinating between TOM systems

TOM for LCO Microlensing Key Project

TOM for Spitzer Microlensing Project

Sharing details on target selection
Surveys, follow-up teams send real-time data to broker

Real-time analysis system returns model output → guides future observations
Possible TOM Design

Core Modules:
- Alert subscribers
- Target Database
- Project portal
- Target Selection & Prioritization
- Observation Control
- Data reduction
- Science analysis

User-developed interacting software:
- Observing Facility Interface
- Data Archive Interface
- Telescope Facilities

External alert brokers/Survey Facilities

External TOMS and other brokers
Getting data back

Microlensing event OGLE-2015-BLG-0966

Real-time
- Survey data received via external broker (ARTEMiS)
- LCO data harvested from own archive
- Other follow-up team data from broker

Offline
- Spitzer data processed, later bulk data release

From Street et al. (2016)
Target Feed

- Harvest alerts of new targets of interest as well as information on existing targets.
- Information feeds come both from surveys and from other teams performing follow-up or added-value services such as cross-matching with existing data.
- Projects may subscribe to multiple feeds

Potential issues:
- Lack of standardization of alert and data formats often means building a new subscriber code for each data source
- Potentially high traffic
- Multiple projects (each with their own TOM) often want to subscribe to many of the same feeds.
Database

Stores and compiles all information on the targets themselves, observations and data products.

May also store information on analysis software status so teams can verify that their automated system is running.

Design likely to have many commonalities between projects (e.g. tables for target position on sky or ephemerides) but also elements which need to be customized for the project (e.g. tables containing the parameters output by software modeling the physical phenomena in question, which may factor into decisions on target priority or observing strategy).

Potential issues:

• Science users inexperience with database design or development. They can describe the observations and analysis they need to do, but often struggle to map this into an efficient table structure. A user-friendly way of easily building a suitable database is needed, perhaps one which asks users targeted questions to lead them to an appropriate design for their needs.
User Portal

Views onto the database through which users access the database content and control the program.
- Some 'standard' views can be imagined which will be needed by all projects (e.g. display a list of all current targets) while others will need to be easily customizable by the project.
- Often need to be accessible by geographically distributed team members, so online-accessibility is important
- Each project is likely to need multiple Views, sometimes for different user groups (e.g. all members of a project may be allowed to view new alerts, but a smaller group may need access to a view which enables them to edit configurations or monitor the status of analysis software).
Target selection/prioritization

Science specific – each project will want to develop their own.

- Could be an algorithm, human-assessment or a combination (e.g. algorithm selects candidates which are presented for human vetting and approval before observation).

- Could be implemented as a separate piece of software which outputs to the TOM database – if so, an importable library for interacting with the TOM would be needed.
Observation Control

Composes and submits observation requests, handles all interactions with observatory interface(s)

• Needs to interpret scientific strategy to formulate observation requests

• Some highly dynamic and may depend on (time-variable) parameters of each specific target,

so needs to be science-user written but capable of interaction with TOM interface layers
Observing Facility Interface

Provides functionality to request, edit and cancel observations and to monitor their progress

• Provides functionality to monitor the status of the observing facility as a whole and the conditions on-site

• Needs to support fully robotic, remotely-operated and manually-operated facilities. Options include developing APIs to interact with robotic facilities and dedicated Views which enable human Telescope Operators to log onto a project TOM and see a display of the selected targets and observation requirements.

• Many projects will need to use the same facilities, so multi-purpose interfaces can be developed for them.

• Many projects will need to use multiple facilities simultaneously. Potential issues:

  • Developing a user-friendly observation request language capable of describing all necessary parameters for observations from a diverse (and evolving) range of telescopes and instruments.

  • Similarly, developing a standardized way of returning information from diverse facilities without infinite degrees of customization.
Data Archive Interface

Provides access to publicly-accessible and project specific data products from a range of sources including the observing facilities being used by the project as well as major national and international archive such as MAST, IPAC, CDS.

- Highly dynamic or high volume project will need programmatic access to the archives. Potential issues:

- Rapid response programs will need rapid access to new data, possibly with a high data rate

- Ensuring propriety data access rights are respected.
Data Reduction and Analysis

Projects may use the data product direct from the observatory if it is processed by the facility, and/or they may conduct additional reduction of their own.

• Data reduction products will feed into science-specific analysis programs. Output parameters and products may be recorded in DB; actual data content may not need to. E.g. DB may store path to reduced spectrum or lightcurve, not necessarily the datapoints themselves.

• Software may operate outside TOM system itself provided it can communicate with the DB.