

PyMovie und PyOTE

pymovie 4.1.5

```
pip install pymovie
```

pyote 5.7.11

```
pip install pyote
```



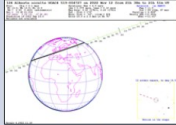
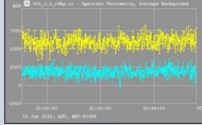
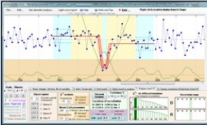
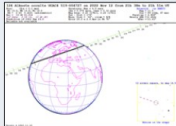
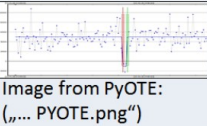
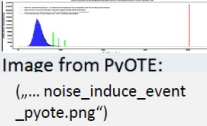
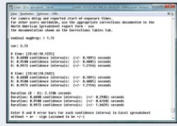
<https://pypi.org/project/pymovie/>
<https://github.com/bob-anderson-ok/pymovie>

<https://pypi.org/project/pyote/>
<https://github.com/bob-anderson-ok/py-ote>

PyMovie & PyOTE eine Alternative zu Tangra & AOTA

Files to be sent to SODIS: positive report

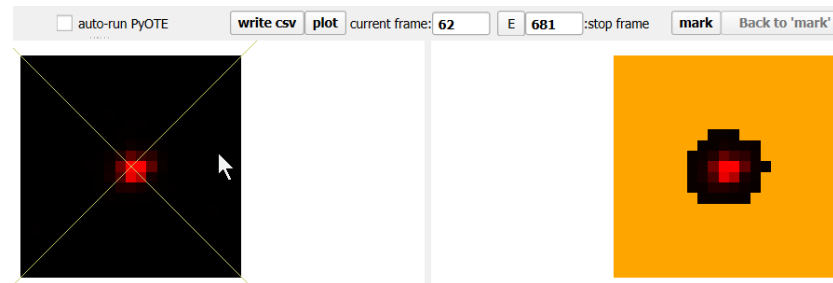


Pipeline	Event	Overview	Reduction	Log	New: csv File
Tangra, AOTA	 PNG-Image from Occult-Watcher: „Open Event in Occult“	 PNG-Image from Tangra: „Export lc / Save as Image File“	 PNG-Image from AOTA: „tab 5“	 Textfile („... AOTA_Report.txt“) from AOTA „tab 6“: „Save Report“	csv file from Tangra: „Export lc / Save as csv File“
Py-Movie, PyOTE	 PNG-Image from Occult-Watcher: „Open Event in Occult“	 PNG-Image from PyMovie: „Plot“ („Composite Lightcurve Plot“)	 Image from PyOTE: („... PYOTE.png“)  Image from PvOTE: („... noise_induce_event_pyote.png“)	 Textf. („... PYOTE.log“) from PyOTE	PyMovie csv file (Result of PyMovie photometry)
Other (SORA, Li-movie, ...)	Please provide similar information as described above.				

PyMovie Motivation und Merkmale

Für die Extraktion von Lichtkurven (Diagramme der Sternintensität im Verhältnis zur Zeit) aus Videos von Sternbedeckungsereignissen verfügt die Astronomie-Community bereits über Tangra und Limovie. Was sind also die Gründe für eine weitere Anwendung?

1. Der Hauptgrund für die Entwicklung dieses Programms war die Überzeugung, dass ein Aperturphotometrieprogramm ein nützliches Werkzeug wäre, das Videos, die durch **Windverwacklungen** beeinträchtigt sind, **robust und effektiv verarbeiten** kann. PyMovie nutzt die Idee, eine Abtastmaske aus dem Bild selbst zu bilden, um eine Maske zu erhalten, die sich an die Form eines verzerrten oder verwackelten Sternbildes anpasst. Eine solche Maske **minimiert die Einbeziehung von Hintergrundpixeln**. Der erwartete Vorteil dieses Verfahrens ist ein **erhöhtes SNR**, insbesondere während der Bedeckung, wenn die Intensität abnimmt und das Bild kleiner wird. Dieser Effekt ist sogar für Videos nützlich, die keine Verwackelungen durch Windstöße aufweisen.
2. **Betriebssystem-Interoperabilität**: PyMovie wurde von Anfang an so konzipiert, dass es sowohl auf **MacOS**- als auch auf **Windows**-Plattformen läuft. Es wurde **mit Python 3.7 und PyQt5** (für die grafische Benutzeroberfläche oder kurz GUI) implementiert und erbt somit die Plattformunabhängigkeit dieser Pakete. Das Programm läuft wahrscheinlich auch unter **Linux**, wurde jedoch auf dieser Plattform nicht getestet.



PyMovie Motivation und Merkmale

3. Eine **Reihe von Hilfsmitteln**, die den Benutzer bei der Platzierung einer Messöffnung an der Position des verdeckten Sterns unterstützen, sind integriert:

- Es besteht eine Online-**Verbindung zu nova.astrometry.net**. Über diese Verbindung kann ein Bild an nova.astrometry.net zur WCS-Kalibrierung (World Coordinate System) übermittelt werden. Mit einer verfügbaren **WCS-Kalibrierung** (die Pixel-x-y-Koordinaten auf RA- (Rektaszension) und Dec- (Deklination) Koordinaten abbildet) kann der Benutzer eine Mess-Blende an der Position des bedeckten Sterns platzieren, indem er dessen RA-Dec-Position angibt. Der Stern muss nicht auf dem Videobild zu sehen sein.
- Um die Ermittlung der RA-Dec-Informationen zu erleichtern, gibt es eine Online-**Verbindung zu VizierR**, die eine UCAC4-Stern-ID akzeptiert und **ICRS-Koordinaten** (International Celestial Reference System) zurückgibt, die sofort verwendet werden können – sie liegt im Format hms dms vor (Stunden Minuten Sekunden Grad Minuten Sekunden).
- Es gibt ein einfach zu bedienendes Tool, das hundert oder mehr Bilder registriert und summiert (ein vereinfachtes RegiStax) und so ein bildverstärktes „**Finder**“-Bild erzeugt, das alle Objekte sichtbar macht, für die eine Lichtkurve extrahiert werden kann.
- Es ist auch möglich, zwei Sterne in einem Bild auszuwählen, ihre RA-Dec-Koordinaten anzugeben und eine verwendbare WCS-Kalibrierung für dieses Bild zu erhalten.



PyMovie Motivation und Merkmale

4. Bei der Entwicklung wurde stets Wert auf eine **einfache Bedienung** gelegt, insbesondere für Gelegenheitsnutzer. Ob dieses Ziel erreicht wurde, entscheiden natürlich die Nutzer.
5. In der Kategorie „seltsam, aber interessant“ ist für die Verfolgung eines Zielsterns neben einem oder zwei Referenzsternen (im Falle einer **Bildrotation**) auch eine **Rückwärtsanalyse** möglich. Sinnvoll um z.B. ein Video zu bearbeiten, das anfangs keine brauchbaren Referenzsterne enthält (Wolkendurchzug).

– Bob Anderson

Zur Umgebung

- Da PyMovie und PyOTE beides plattformunabhängige Python Skripts sind, muss zuerst Python installiert werden. Anschliessend können dann PyMovie und PyOTE installiert werden. Bei der Skriptsprache Python ist zu beachten, dass die neuste Version (3.14, Stand Okt. 2025) nicht mit diesen beiden Tools läuft. Die höchste lauffähige ist weiterhin die Version 3.10.xx. Python kann unter <https://www.python.org/downloads/> heruntergeladen werden. Die Version 3.10.xx ist etwas weiter unten bei «Looking for a specific release?» zu finden.
- Falls bereits eine neuere Python Version auf dem PC installiert ist, dann ist es empfehlenswert, Python 3.10.xx in einer virtuellen Umgebung zu installieren. Auch sonst macht es Sinn eine virtuelle Umgebung zu verwenden, da für PyMovie und PyOTE einige weitere Python Module nötig sind, die evtl. Konflikte zu einer bestehenden Installation verursachen könnten. Falls Python aber sonst auf dem Rechner nicht verwendet wird, dann kann eine neuere Version auch deinstalliert und anschliessend einzig 3.10.xx installiert werden (in diesem Fall direkt weiter zur Seite «Installation der beiden Tools».
- Die nachfolgenden Schritte beziehen sich auf MS Windows. Unter anderen Betriebssystemen wie Mac oder Linux ist die Installation ebenfalls möglich, aber das Vorgehen und einzelne Befehle können leicht abweichen. Prüfen, ob Python bereits installiert ist, kann man wie folgt:

Kommandozeile öffnen (z.B. über Windows-R, cmd). Anschliessend Eingabe von «python» oder «py» auf der Kommandozeile. Falls Python bereits installiert ist, wird dies mit der entsprechenden Versionsnummer quittiert:

```
Eingabeaufforderung - python
Microsoft Windows [Version 10.0.19045.6396]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\sternwarte>python
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> █
```



- Den Python Prompt verlassen, kann man mit der Eingabe von `>>> exit()`. Bei der Eingabe von `python --version` auf der Kommandozeile wird Python gar nicht erst gestartet, sondern nur die Version angezeigt.

Installation von Python 3.10 (1)

1. Python 3.10.xx herunterladen und installieren (s.a. letzte Seite). Für 3.10.11 gibt es noch einen Windows Installer. Die späteren 3.10er Versionen sind sogenannte "security fixes only" für die es nur noch Source Files gibt. Bei der Installation ein Häkchen bei «Add python.exe to PATH» setzen.
2. Anschliessend den Installationspfad ermitteln, wo die 3.10er Version installiert wurde. Falls das nicht offensichtlich ist, kann dies mit folgendem Python Befehl selber herausgefunden werden (dazu aber noch prüfen, ob mit «Python» auch wirklich die 3.10er Version aufgerufen wird):

```
C:\Users\sternwarte>python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr  5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import os
>>> import sys
>>> os.path.dirname(sys.executable)
'C:\Users\sternwarte\AppData\Local\Programs\Python\Python310'
>>> █
```

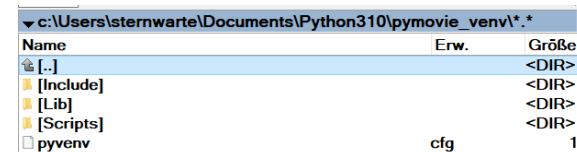


3. Erstelle eine virtuelle Umgebung mit der Python-Version 3.10:
Navigiere dazu zum Projektverzeichnis im Terminal. Dieses Verzeichnis kann z.B. im User-Verzeichnis unter Documents angelegt sein. Im Beispiel oben habe ich dazu c:\Users\sternwarte\Documents\Python310\ erstellt. Verwende dann im Terminal den Pfad zum gewünschten Python-Interpreter, um eine virtuelle Umgebung zu erstellen. Der Befehl dazu mit einem konkreten Beispiel wäre:

C:\path\to\python310\python.exe -m venv <Umgebungsname>

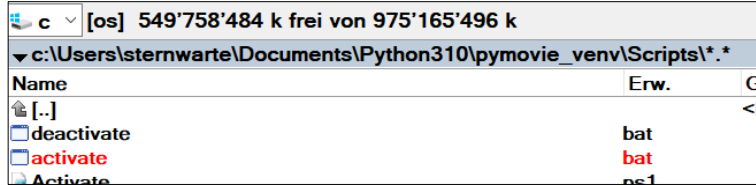
```
C:\Users\sternwarte\Documents\Python310>
C:\Users\sternwarte\Documents\Python310>C:\Users\sternwarte\AppData\Local\Programs\Python\Python310\python.exe -m venv pymovie_venv █
```

Nach einigen Sekunden sollte die virtuelle Umgebung erstellt worden sein und das Projektverzeichnis sieht bereits etwas gefüllter aus als vorher:



Installation von Python 3.10 (2)

4. Nun kann die virtuelle Umgebung aktiviert werden. Dazu gibt es ein kleines «activate.bat» File unter Scripts, das in der Konsole gestartet wird:



```
C:\Users\sternwarte\Documents\Python310\pymovie_venv\Scripts>activate.bat  
(pymovie_venv) C:\Users\sternwarte\Documents\Python310\pymovie_venv\Scripts>
```

Nach der Aktivierung steht (wie oben im schwarzen Feld) der Name der Umgebung in Klammern vor der Eingabeaufforderung.

5. Installieren der Pakete:
Nachdem die virtuelle Umgebung aktiviert wurde, kann normal «pip» zum Installieren von Paketen verwendet werden. Diese werden dann ausschließlich in der virtuellen Umgebung installiert und sind vom System und anderen Umgebungen isoliert.

Beispiel: `pip install <Paketname>` oder `pip install <Paketname>==<Version>`

Auf der nächsten Seite kommen wir zur eigentlichen Installation von PyMovie und PyOTE, die dann genau in dieser virtuellen Umgebung abläuft.

6. Deaktivieren der virtuellen Umgebung:
Wenn die Arbeit in der virtuellen Umgebung abgeschlossen ist, dann kann diese einfach mit dem Befehl deactivate (siehe deactivate.bat im ersten Screenshot von oben) beendet werden. Das Vorgehen von Pkt. 4 und 6 gilt später auch, wenn PyMovie/PyOTE nach der vollständigen Installation in der virtuellen Umgebung gestartet und wieder beendet wird.

Installation der beiden Tools

Die Schritte dazu sind in der Anleitung als PDF unter <https://occultations.org/observing/software/py movie/> beschrieben. Im PDF kann direkt auf die Seite 2 unten gesprungen werden, da wir nun ja die korrekte Version von Python bereits installiert haben. Es geht weiter mit folgenden Schritten (hier nur noch kurz zusammengefasst):

1. `pip install wheel / pip install pymovie`

Zu beachten ist, dass diese Befehle in der virtuellen Umgebung, die zuvor installiert und aktiviert wurde, ausgeführt werden. Falls die Version von pip (wie rechts im Screenshot) veraltet ist, kann diese noch aktualisiert werden. Die Installation von pymovie kann einige Minuten dauern und es werden recht viele Dateien heruntergeladen. Der Fortschritt wird in der Konsole jeweils angezeigt.

2. `pip install pyote`

Es folgt das gleiche für das zweite Tool. Falls alles geklappt hat, dann liegen beide Tools in den letzten Versionen 4.1.5 und 5.7.11 vor (Stand Okt. 2025).

3. Test und Start der beiden Installationen von der Konsole aus (je 3 Zeilen):

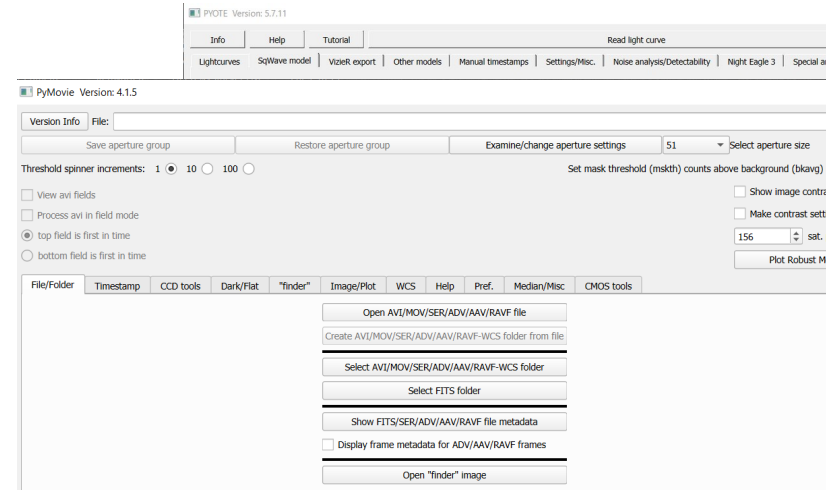
```
python
>>> from pymovie import main / from pyoteapp import pyote
>>> main.main() / pyote.main()
```

Wie in der Anleitung beschrieben, können nun noch zwei Desktop-Icons erstellt werden, um die Tools in Zukunft einfacher per Doppelklick zu starten. (dort Step 5)

```
(pymovie_venv) C:\Users\sternwarte\Documents\Python310\pymovie_venv>pip install wheel
Collecting wheel
  Downloading wheel-0.45.1-py3-none-any.whl (72 kB)
----- 72.5/72.5 kB 2.0 MB/s eta 0:00:00
Installing collected packages: wheel
Successfully installed wheel-0.45.1

[notice] A new release of pip is available: 23.0.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip

(pymovie_venv) C:\Users\sternwarte\Documents\Python310\pymovie_venv>
```



Arbeiten mit PyMovie (Demo)

Nach dem Start:

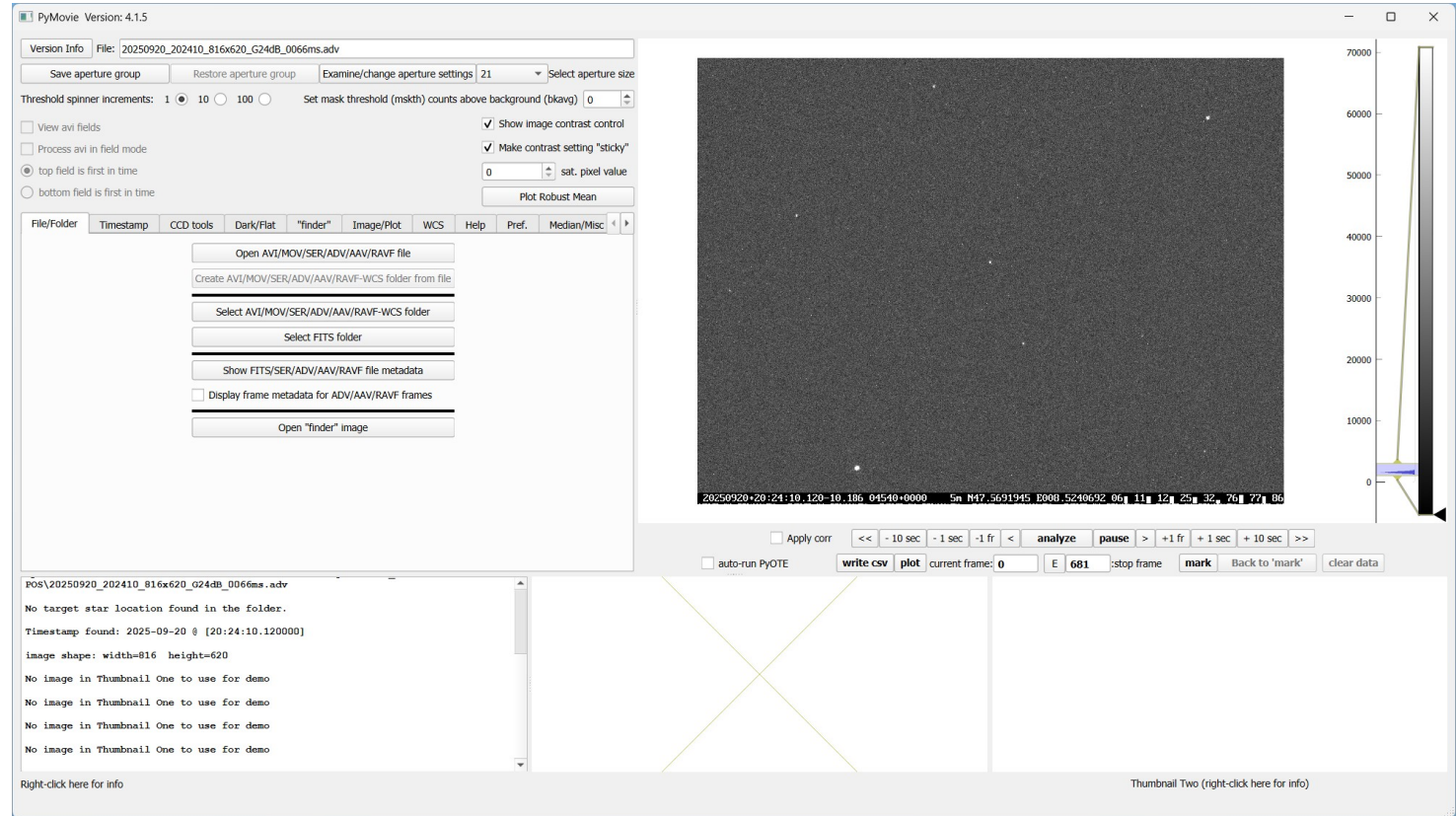
- neue Version vorhanden?
- Video laden
- «Show image contr. contr.»
- Eingrenzen des Bereiches
- Aperture festlegen, z.B. 31
- Sterne und Maske wählen (Rechtsklick im Bild)
- Namen wählen: target, track, ref.. (z.B. mit 2 letzten Zahlen der UCAC4). Erscheint später im csv File und in den Plots
- Start der Analyse mit «analyze»

Nach dem Durchlauf:

- «plot»
- Screenshot von «comp. lc plot» und evtl. anderen
- «write csv» z.B. «lc.csv»
- ev. Screenshot der verwendeten Sterne

Falls Ergebnis positiv:

- Auswertung des csv Files mit PyOTE



«Version Info» zeigt die Versions-History (PDF) und ein Rechtsklick auf einen Menüpunkt die Hilfe.

Arbeiten mit PyOTE (Demo)

Nach dem Start:

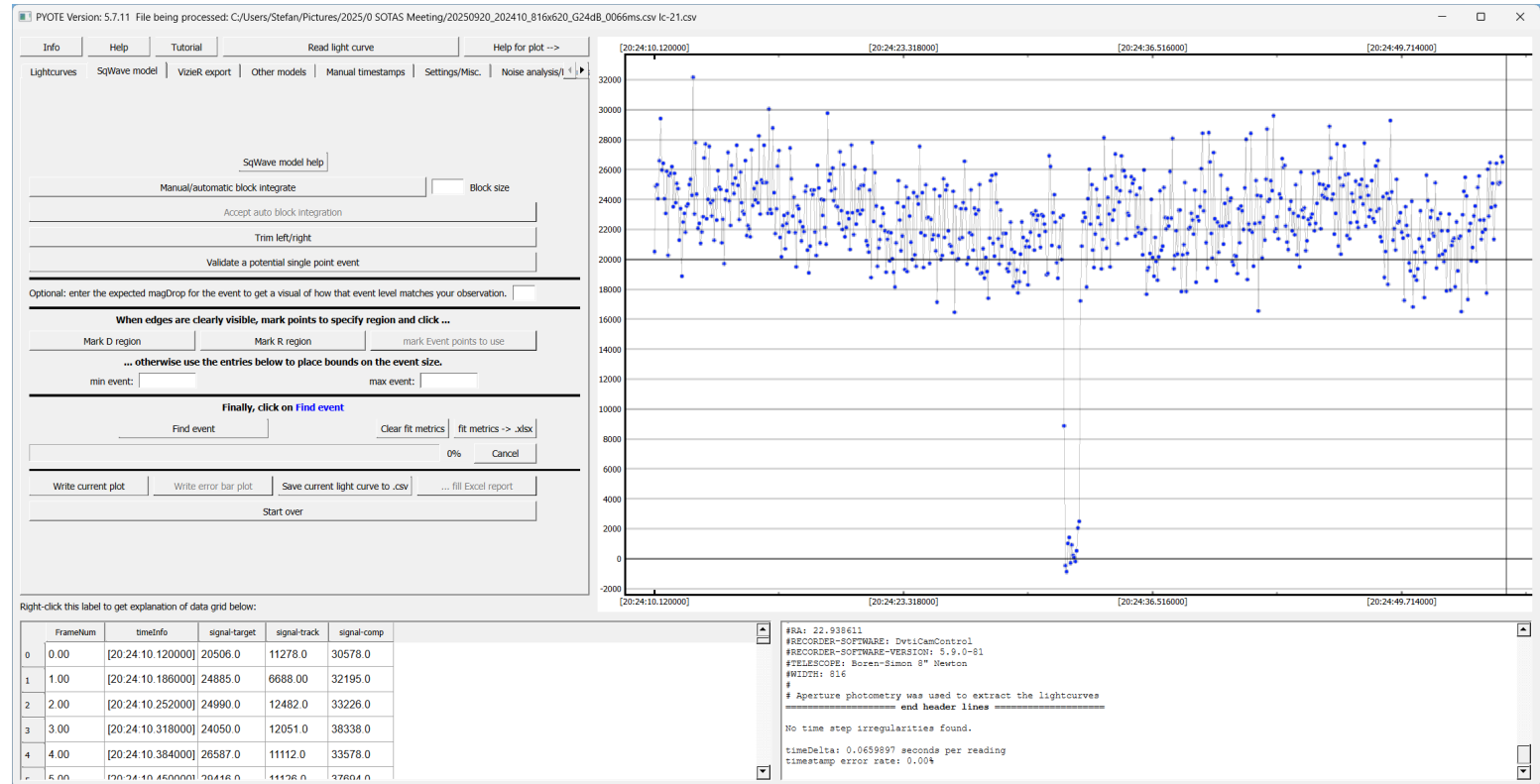
- neue Version?
- csv Laden, Read light curve
- ref / show, Offsets
- Tab «Lightcurves»
- evtl. «smoothing» → Check Flatness
- «Mark baseline region» für SNR Analyse

Ermitteln der Zeiten:

- Tab «SqWave model»
- Input «min / max event» (Datenpunkte) oder «Mark D/R region»
- «Find event»
- Screenshots von Solution und Lichtkurve
- Rechtsklick: ganze Lichtkurve
- Logfile autom. gespeichert

Erstellen des dat Files:

- Tab VizieR export
- ausfüllen, bei der DVTI Cam in der Regel schon vollständig
- Trim l/r, Add LC to V. folder



Arbeiten mit PyOTE (Demo)

Weitere Anwendungen:

- Normalization in x und y Achsen
- Single Point Events, Metrics
- Release Notes und Hilfe
- Detectability Estimate
 - Obs. Duration in s
 - Event Duration in s x2
 - Iterations
 - Mag Drop
- Andere Modelle als SqWave

Noise analysis/Detectability tab help

Mark baseline region

Calc baseline mean/sigma/corr coeffs

Clear baseline regions

ObservationDur(secs): EventDur(secs): duration step: magDrop:

Calc detectability ☐ Write example lightcurve to csv file

Validate a potential single point event

