ATLAS forced photometry - combining the detections into weighted averages with sigma clipping (`AtlasForceCombine.py`).

Code developer : Matthew Dobson (QUB)
Supervisor : S. Smartt (QUB).
Date : 4 Sept 2019
Version : 1.0

Code at QUB : `/home/sne/soft/atlas/`
Code on github: https://github.com/MatthewDobsonOnGitHub/AtlasForceCombine

Summary

Photometry in ATLAS is performed with `tphot` and forced photometry is done with the same code, forcing `tphot` measurements, with two different wrapper scripts. At QUB we run `tphorce` (written by D. Young) and in Hawaii, `force.sh` (J. Tonry) runs. Both are shell scripts, that force a PSF fit at a given position on an individual image, and return a flux in microJanskys. Comparison tests between the two (at QUB) show them to give identical results

The QUB ATLAS Transient Server displays the results from `tphorce`, and the results from all individual images are available as a CSV for download. The script `force.sh` is set to run within the Hawaii directory structures but can be tweaked to run at QUB.

Each of these measure the forced flux on an individual 30 second image, and on any one night the (typically) 4 measurements can be combined into a nightly average. This code carries out that task. It provides a variance weighted average for the night and rejects individual rogue points (3 sigma outliers).

Most science users will be downloading the CSV files from the QUB webpages but the code will run with either the output of `tphorce` or `force.sh`. The calculations, and returned results, are all in flux (microJanskys) and not magnitudes, although conversion for the user is trivial.

Input files and calculation of weighted mean

The `tphorce` output files on the QUB ATLAS Transient Server have the following columns for each 30s sec exposure :

```
#expname,ra,dec,mag,dm,snr,filter,zp,mjd,x,y,peakval,skyval,peak
fit,dpeak,skyfit,flux,dflux,chin,major,minor,snrdet,snrlimit,apf
it
```

The definition of the AB magnitude system is (where *f* is in erg/s/cm^2/Hz)

$$m_{AB} = -2.5 log f_{v} - 48.60$$

Or for f_{Jy} in Janksys (1 Jy = 10^{-26} J/s/m^2/Hz^{-1})

$$m_{AB} = -2.5 log f_{Jy} - 8.9$$

Simple rearranging for f_{Jy} , and from the standard definition of a zeropoint (1 ADU per second) gives the flux zeropoint of an image in Janksys as

$$f_{ZP} = 10^{\left(\frac{-ZP}{2.5}+3.56\right)}\text{Jy}$$

Where ZP = MAGZPT= AB mag zeropoint in the ATLAS image header. Hence the flux of any source, f_{Jy}, in Janskys is

$$f_{Jy} = \frac{(COUNTS \pm ERR)}{EXPTIME} 10^{\left(\frac{-ZP}{2.5}+3.56\right)}\text{Jy}$$

Where *COUNTS* gives the total ADU in the PSF, and *ERR* is the uncertainty on that.

The `tphot` measurement of flux, and error in a PSF (in image ADUs) is

$$m_{AB} \pm err = -2.5 log\left(\frac{peakfit \times major \times minor \pm dpeak \times major \times minor}{EXPTIME}\right) + \text{ZP} + \text{APFIT}$$

Which means flux in microJanksys is

$$f_{\mu Jy} = \frac{(peakfit \times major \times minor) \pm (dpeak \times major \times minor)}{EXPTIME} 10^{\left(\frac{ZP+APFIT}{-2.5}\right)+9.56}\mu\text{Jy}$$

And EXPTIME is (always) 30 sec.

The output file from `tphorce` includes all forced photometry measurements on the individual 30s exposure. The `mag` and `dm` columns in the `tphorce` output are populated with real values if the forced photometry returns a significance greater than 1 sigma and the significance is given as the signal-to-noise value `snr` in the subsequent column. If the measurement is less than 1 sigma significance then a 3-sigma upper limit is returned, denoted with a >, and `dm = 0,` and `snr = 3.0.`

This code, `AtlasForceComb.py`, converts all forced `tphot` measurements into flux (respecting the errors) and calculates a nightly weighted mean. The values are weighted by their inverse variance and a single binned value for each MJD is returned. The MJD returned is the simple mean of the individual exposure `mjd` values.

The weighted mean is

$$\acute{f} = \frac{\sum w_i f_i}{\sum w_i}$$

where the weight w_i of each i^{th} measurement, f_i is equal to the inverse square of its associated uncertainty value. And the error σ_f is

$$\sigma_f = \sqrt{\frac{\sum_{i=1}^{N} w_i \left(f_i - \acute{f}\right)^2}{\frac{N-1}{N}\sum_{i=1}^{N} w_i}}$$

where $\boldsymbol{f_i}$ is the **i**$^{\mathbf{th}}$ point that went into the weighted mean value $\acute{f}$, and w_i is that point's weight, and **N** is the total number of points that were averaged over for a given weighted mean value.

For days where there is only one measurement available from an individual 30 sec exposure, that value's error is adopted as the intrinsic flux uncertainty.

Sigma clipping

There are often rogue points that come from poor subtractions, or residuals on the subtracted frames, or otherwise unconstrained PSF fits. These are rejected with sigma clipping. The standard deviation and median of all the individual flux measurements in a given day are calculated. Each of the points are then compared with the median, $\boldsymbol{m}$, and if they lie further than $\boldsymbol{n\sigma}$ from the median then they are rejected. The calculation is repeated for a given MJD until all flux values in the range $\boldsymbol{m \pm n\sigma}$

The weighted mean is calculated as above on these remaining points. In the code, $\boldsymbol{\sigma}$=3 is set, but it can be changed :

```
number_sigma = 3
```

Instructions for running the code and outputs

The code is python3, and requires the following libraries (from top of the file) :

```
import numpy as np
import matplotlib.pyplot as plt
import math
import pylab
import csv
import sys
import statistics as stat
import os
import sys
```

`statistics` is one that is not often standard in a conda installation and may have to be installed manually by the user.

Download a forced photometry file from the QUB webpages, and rename it to `SN2019xxx.csv` or `SN2019xxx.txt`.

With the code and input file in your pwd :

```
bash$> python3 AtlasForceCombine.py SN2019clq.csv
```

Which creates 2 files :

```
SN2019clq_flux_weighted_mean_clipped.txt
SN2019clq_flux_vs_time_raw_weighted_clipped.pdf
```

The pdf is an illustrative plot for quick look.t Three plots of flux against time are produced:

- fluxes from the individual 30s exposures,
- the weighted mean fluxes per day with no sigma clipping,
- the weighted mean fluxes per day with sigma clipping,

The measured data points from the 30s exposures are in translucent, faded colours, with the weighted mean values of the flux plotted against the average time of measurement for a given day in stronger colours

The data file contains 6 tables of time and flux (plain text)

```
# Time (MJD), Flux and Flux Error (microjanskys) for Orange Filter (30s
exposures)

# Time (MJD), Flux and Flux Error (microjanskys) for Cyan Filter (30s
exposures)

# Time (MJD), Flux, Flux Error (microjanskys) for Orange Filter (weighted
means, number of points included in the weighted mean

# Time (MJD), Flux and Flux Error (microjanskys) for Cyan Filter weighted
means), number of points included in the weighted mean

# Time (MJD), Flux and Flux Error (microjanskys) for Orange Filter clipped
weighted means), number of points included in the clipped weighted mean

# Time (MJD), Flux and Flux Error (microjanskys) for Cyan Filter clipped
weighted means), number of points included in the clipped weighted mean
```

Comparison of the last two shows the number of points clipped, which are more than 3σ from the median.

The code will also take the output of `force.sh` , John Tonry's forced photometry wrapper for `tphot`. No other difference in running the code is required, the code will notice the difference between the files and produce the combined daily means in the same way.