

Localizing Sources of Variability in Crowded TESS Photometry

MICHAEL HIGGINS¹ AND KEATON J. BELL^{2,*}

¹*Department of Physics, Duke University, Durham, NC-27708, USA*

²*DIRAC Institute, Department of Astronomy, University of Washington, Seattle, WA-98195, USA*

(Received; Revised; Accepted March 4, 2022)

Submitted to AJ

ABSTRACT

The Transiting Exoplanet Survey Satellite (TESS) has an exceptionally large plate scale of $21'' \text{ px}^{-1}$, causing most TESS light curves to record the blended light of multiple stars. This creates a danger of misattributing variability observed by TESS to the wrong source, which would invalidate any analysis. We develop a method that can localize the origin of variability on the sky to better than one fifth of a pixel. Given measured frequencies of observed variability (e.g., from periodogram analysis), we show that the corresponding best-fit sinusoid amplitudes to raw light curves extracted from each pixel are distributed the same as light from the variable source. The primary assumption of this method is that other nearby stars are not variable at the same frequencies. Essentially, we are using the high frequency resolution of TESS to overcome limitations from its low spatial resolution. We have implemented our method in an open source Python package, `TESS_localize` (github.com/Higgins00/TESS-Localize), that determines the location of a variable source on the sky given TESS pixel data and a set of observed frequencies of variability. Our method utilizes the TESS Pixel Response Function models, and we characterize systematics in the residuals of fitting these models to data. Given the ubiquity of source blending in TESS light curves, verifying the source of observed variability should be a standard step in TESS analyses.

Keywords: astronomical object identification — CCD photometry — time series analysis — variable stars

1. INTRODUCTION

The Transiting Exoplanet Survey Satellite (TESS) is executing a thorough census of photometric variability over 85% of the sky (Ricker et al. 2015). It collects continuous images in a series of pointings (sectors), visiting each of these overlapping fields for approximately 27 days. During its first two years of operations, TESS obtained full-frame images (FFIs) every 30 min, as well as 2-min cadence subframe images around $\approx 15,000$ bright stars and other targets of specific interest per sector. The current extended mission has increased the FFI rate to 10 min, added a 20-sec cadence, and increased the number of short-cadence targets in each sector. The

records of stellar variability obtained during these extensive photometric campaigns achieve frequency resolutions of $\approx 0.43 \mu\text{Hz}$ or finer, and they are not confused by aliasing from temporal gaps in the data that affect ground-based surveys.

What TESS achieves in frequency resolution, it lacks in spatial resolution. TESS has an exceptionally large plate scale of $21'' \text{ px}^{-1}$. As a result, multiple stars often contribute light to the same detector pixels, and it can be difficult to identify which star is the source of recorded variability. The source blending problem varies with direction, with upwards of dozens of stars brighter than magnitude 18 contributing light to most pixels near the galactic plane. When analysing TESS light curves, there is considerable risk of attributing detected variability to the wrong source, leading to erroneous conclusions.

Corresponding author: Michael Higgins
michael.higgins@duke.edu

* NSF Astronomy and Astrophysics Postdoctoral Fellow

Even with $4''$ pixels, blending was a concern for analyses of data from the *Kepler* spacecraft. *Kepler* (Borucki et al. 2010) can be considered a predecessor of TESS that also obtained continuous light curves of stars primarily to detect exoplanet transits, though in a field of view 400 times smaller. *Kepler* observed over 500,000 stars from which a list of *Kepler* objects of interest (KOIs) that had transit-like signals was compiled. Nearly half of these KOIs turned out to be false positives caused by contamination from eclipsing binaries (Bryson et al. 2013). Colman et al. (2017) demonstrated another example of blended starlight complicating analyses of *Kepler* light curves, showing that anomalous peaks in the periodograms of red giant stars could indicate a companion orbiting within the convective giant envelope, while in many cases such signals likely originate from chance alignments with other variable sources in the field.

Since *Kepler* and TESS are motivated by the search for transiting exoplanets, many tools have been developed to vet candidate transit signals. Shallow planet transits can be mimicked by contaminating light from eclipsing binary systems. Giacalone et al. (2020) review the history of software tools for vetting candidate transits that test the data against transit and blended-binary models, and introduce the **Triceratops** package that utilizes the *Gaia* DR2 star catalog to model the possible contamination from many stars that is especially relevant for TESS. The software tool **LATTE** provides an interactive interface for investigating potential contaminant or systematic origins of candidate transit signals in TESS (Eisner et al. 2020). One such diagnostic is to test whether the centroid of light from a target source moves during transits or eclipses, which could indicate that the transited star is spatially offset from the target (Bryson et al. 2013; Hedges 2021).

Time series photometry from TESS is invaluable for studying many types of brightness variability of astronomical objects. Less effort has gone into developing methods to address the challenges of source blending for generic variability besides just transit-like signals. The method used in Colman et al. (2017) to identify whether signals originate from their target stars or nearby contaminating sources was to compute a periodogram for light curves extracted from each pixel in a *Kepler* “postage stamp” and identify where the corresponding peaks appear to be centered in the pixels. This process was done by visual inspection to get a general idea of the location of the source of the anomalous peaks in the periodograms and spatially resolve them. Hedges et al. (2021) developed a method that can extract individual, deblended light curves for sources that are separated by at least one pixel and differ in bright-

ness by more than a magnitude. In many cases, potential source confusion can be resolved by comparing to additional archival or follow-up time series photometry that is spatially resolved for the various sources in the field (e.g., Collins et al. 2018). To aid in disentangling variability of blended sources in northern TESS Cycle 2 fields, the Zwicky Transient Facility conducted a nightly, contemporaneous photometric survey at high angular resolution that can be compared to TESS data to match variability in many cases (van Roestel et al. 2019).

This paper presents a new method to solve the persisting problem of spatially resolving signal sources in low-resolution time series survey photometry. Our strategy for localizing where variability is originating from in the pixel data is to fit the spatial distribution of signal amplitudes for frequencies of interest. We recommend verifying the source of variability for all analyses of TESS data where significant frequencies of variation can be measured. We are releasing a Python package, **TESS_localize** at github.com/Higgins00/TESS-Localize, that can fit the sky location of observed variability in the TESS pixel data. We detail the method in Section 2, validate its performance with simulations in Section 3, explore some case studies with real TESS data in Section 4, and provide practical guidance for users in Section 5.

2. METHODS

In general, TESS pixels record the combined light contributions from many sources. If we detect variability in an extracted light curve, it is not immediately clear which source the variability originates from. We have developed a method for localizing the source of detected variability in the TESS pixel data based on the fact that the (unnormalized) amplitude of variation in each pixel is proportional to the flux contribution from the variable source. Essentially, we are able to resolve the variable source in frequency space for each pixel. We develop the concepts and assumptions behind the method before detailing its implementation in the Python package that we are releasing as an open-source research tool.

2.1. Overview

We will consider our source of interest to be some theoretical variable star with average total flux F measured by the detector. This flux is distributed on the detector following the TESS pixel response function (PRF¹;

¹ TESS uses this term instead of “point spread function”: <https://heasarc.gsfc.nasa.gov/docs/tess/observing-technical.html#point-spread-function>

Bryson et al. 2010). We indicate the fraction of the source flux measured by the pixel at column i and row j as $P_{i,j}$. This light will be blended with a potentially complicated distribution of background light from other nearby sources, and we represent the average fractional ratio of the flux from the source star to the total flux in each pixel as $C_{i,j}$.² The average total flux measured from all sources in each pixel is $F * P_{i,j}/C_{i,j}$.

The primary assumption underlying our method is that the other nearby stars do not exhibit significant variations at frequencies that are unresolved from those of our source. Here, nearby means contributing light to the target pixel files (TPFs) that are read out around every short-cadence target, or to the sub-frame cutouts acquired from the full frame images with TESSCut (Brasseur et al. 2019). For simplicity, we will refer to both as TPFs in this work. Putting this assumption another way, the only appreciable effect that other stars have on the periodograms of light curves extracted from a TPF is on the measurement noise at the frequencies specified. This assumption will be satisfied in most cases, owing to the high frequency resolution achieved by the TESS data of $\approx 0.43 \mu\text{Hz}$, as well as the minimal aliasing. For 2-minute cadence data, for instance, this resolution corresponds to over 10,000 effectively independent frequency bins below the Nyquist frequency of $4467 \mu\text{Hz}$. It is unlikely that the frequencies of variation of blended background stars will happen to fall within the same frequency bins as our target source variability. We provide practical guidance for validating this assumption in Section 5.

Suppose the target star of interest varies sinusoidally with frequency ν and relative (fractional) amplitude a . Since we assume that only the source exhibits significant power at frequency ν , we can fit a model for how the flux varies *at this frequency only* in each pixel

$$\begin{aligned} F_{\nu(i,j)}(t) &= F \times P_{i,j}(a \sin(2\pi(\nu t + \phi)) + 1/C_{i,j}) \\ &= A_{i,j} \sin(2\pi(\nu t + \phi)) + \langle \text{Flux}_{i,j} \rangle, \end{aligned} \quad (1)$$

where ϕ is a phase term shared across pixels. This is essentially the Fourier component at frequency ν , plus an offset for the average flux in the pixel. Since F and a are constant values, the fitted amplitude $A_{i,j}$ is proportional to $P_{i,j}$, the fraction of light from the variable source in pixel (i,j) . Therefore, if we fit this model for every pixel in the TPF, we can empirically recover how the light from the variable source is distributed on the detector. We can then fit a PRF model to this distribu-

tion to estimate the location of the variable source on the sky.

This localization procedure can be improved by fitting a common location for multiple frequencies of variation, assuming that they all arise from the same target. General non-sinusoidal variations can be represented as a sum of sine waves (harmonics for periodic variability).

2.2. Implementation

In practice, we anticipate that users will have identified frequencies of variability from a periodogram analysis of a light curve, and they then wish to verify where on the detector those variations arise from. Our implementation of this signal localization method in the Python package `TESS_localize` requires a user to provide a list of frequencies, TPF data, and optionally the aperture used to extract the light curve that the frequencies were measured from. If an aperture is not provided, the TPF pipeline aperture will be used by default. There is also an option that tries to automatically choose an appropriate aperture containing the highest Lomb-Scargle periodogram power at the provided frequencies. The flow of how the software proceeds based upon user input is presented in Figure 1.

Intrinsic variability from any individual source will arrive with the same phase across the image. `TESS_localize` performs an initial fit to the light curve extracted from the aperture, which should provide a high signal-to-noise measurement of signal phases. The first fit fixes amplitudes based on the Lomb-Scargle periodogram amplitudes evaluated at the provided frequencies, fixes the average flux to the mean measured flux, and brute-force samples phase in 20 evenly sampled steps to obtain good starting phase values for further optimization. Then a second, non-linear least-squares fit is performed where phases, amplitudes, and mean flux are all free to vary. These fits are all performed using the implementation of the Levenberg–Marquardt algorithm via the Python package `lmfit` (Newville et al. 2018).

With the signal phases measured, `TESS_localize` moves on to fit the signal amplitudes from the unnormalized light curves extracted from each pixel in the TPF. Before fitting, `TESS_localize` optionally removes an automatically determined number of trends most common among pixels outside the provided aperture with principal component analysis using the `RegressionCorrector` tools from the `lightkurve` package (Lightkurve Collaboration et al. 2018). The `auto_pca` function computes a periodogram of the first five (by default) principal components and returns the number the last periodogram to not show significant

² Analogous to the CROWDSAP header value that approximates crowding in the Science Processing Operations Center pipeline apertures.

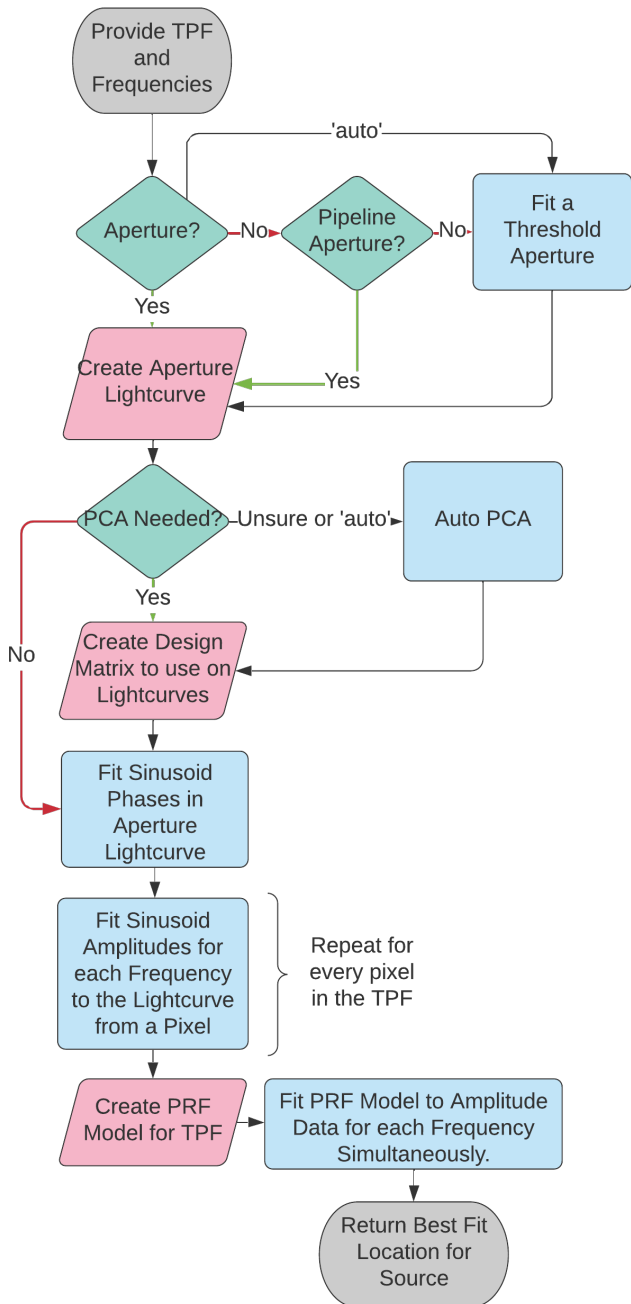


Figure 1. Flow chart of the `TESS_localize` method described in this work. The first grey box and the green diamonds are the only parts of the software that require user input.

power at any of the frequencies in the frequency list input by the user. `TESS_localize` then performs least-squares fits of our (multi-)sinusoid model to the light curve from each pixel with `lmfit`, with frequencies fixed to the input values, and phases fixed to the best-fit values to the aperture light curve from the previous step. The uncertainty on the fitted amplitudes are expected

to be $\sigma_a = \sqrt{2/N}\sigma_F$, where σ_F is the typical light-curve flux error, and N is the number of points in the light curve (Montgomery & O’Donoghue 1999). Since $\sigma_F \approx \sqrt{F}$, pixels with more flux contamination will yield noisier measurements of the signal amplitudes. This is accounted for when least-squares fitting sinusoids to the light curves by weighting by the reciprocal of the flux uncertainties.

With amplitudes measured for each frequency in each pixel, `TESS_localize` finally fits the common location where this variability arises from. We created a Python package called `TESS_PRF`³ that interpolates the TESS PRFs available on the Mikulski Archive for Space Telescopes (MAST) for a given location on the detector, which we describe in Appendix B. `TESS_localize` simultaneously fits a PRF model to the location in the TPF from which all provided frequencies are most consistent with originating, scaled to different amplitudes for each signal. With `lmfit`, the program minimizes the absolute difference between the pixel-by-pixel amplitudes from the previous step and the PRF model, divided by the amplitude fit uncertainties.

The `TESS_localize` program returns the optimized column and row location within the TPF where the observed variability is most likely located, with intrinsic uncertainties. The code adopts the FITS WCS standard that integer pixel locations represent the center of pixels (Greisen et al. 2006). For each signal in the input frequency list, a “heatmap” of amplitude, signal to noise, error, or the fit model can be displayed with the `plot` function. The program can optionally query the Gaia Archive for the locations of stars in the field, ranking them by the relative probability that their locations are consistent with the source of variability (Brown et al. 2018). Relative probabilities are calculated by using the extrinsic error model discussed in Section 4.2.1. The extrinsic error accounts for systematic errors that arise when fitting a PRF model to real TESS data. The user can use these outputs to verify which stars are likely to be the source of variability they are interested in, so that misattributions of variability to incorrect sources may be avoided.

3. SIMULATIONS

To assess the performance of our method under ideal conditions, we simulated TESS-like pixel data for three scenarios. These tests provide empirical evidence that our implementation of this method recovers accurate variable source positions, and that the intrinsic uncer-

³ https://github.com/keatonb/TESS_PRF

tainties given by the fitting procedure closely match the residuals. The three situations we simulated are as follows:

1. An isolated star that is variable with different amplitudes relative to noise.
2. A star that is at varying distances to the edge of a TPF.
3. Two blended stars at varying separations from each other.

For these simulations, we approximate TESS TPF data by generating sequences of 11x11 pixel images every 2 min for a duration of 27 days. Light from each star is distributed as a Gaussian across the pixels with a scale factor (standard deviation) of 0.7 pix. These Gaussians are sampled nine times finer than the plate scale in each dimension and integrated over the pixel locations to approximate a TESS PRF. In all simulations, the star that is variable is only variable at one signal that has a period of 10 days. Our method will perform better for sources that vary with multiple frequencies, which boosts the effective signal-to-noise for our fitting procedure. A 5-pixel cross-shaped aperture centered at the simulated star is used to determine the signal-to-noise. The noise is simulated by adding normally distributed independent random values scaled to the square root of the flux in each pixel of each image.

3.1. Case 1: Star Variability Signal-to-Noise

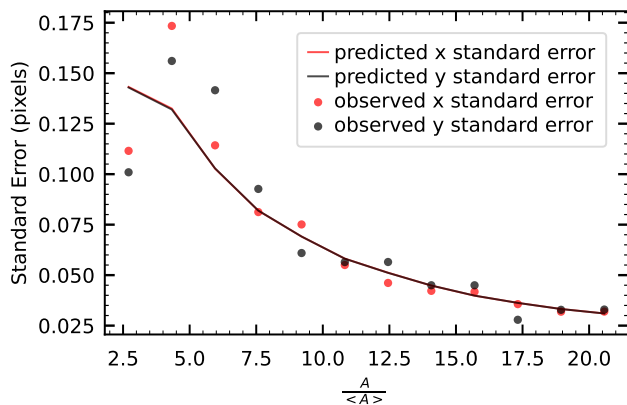


Figure 2. Predicted standard errors from the fitting procedure (solid lines) and the observed standard deviation of the fit residuals (points) as a function of signal-to-noise in the aperture. The signal-to-noise is computed as the signal amplitude divided by mean amplitude in the periodogram of the light curve. We ran 1000 simulations for each signal-to-noise value tested.

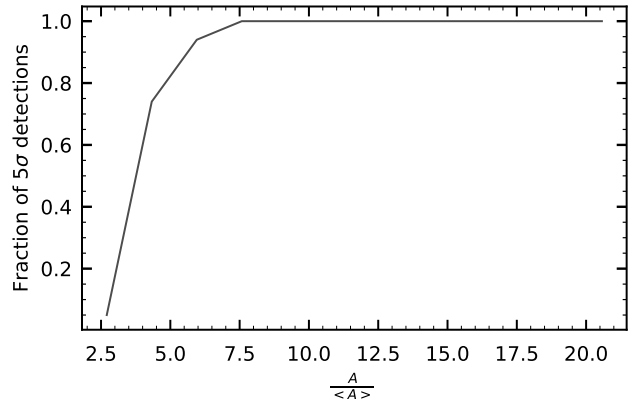


Figure 3. Fraction of the source star fits with a relative amplitude error less than 20% for the same simulations at different signal-to-noise levels displayed in Figure 2.

We simulated a star 1000 times in the center of the TPF for multiple signal-to-noise ratios. As a metric for signal-to-noise, we define $A/\langle A \rangle$ as the amplitude of our signal divided by the mean amplitude in the periodogram of the light curve extracted from a 5 pixel cross-shaped aperture centered about the true location of the simulated star. For context, simulating 10,000 light curves shows that a signal with $A/\langle A \rangle$ greater than 4.4 has less than a 0.1% probability of being caused by pure noise (false alarm probability; see also Baran & Koen 2021). We applied our source localization method to all 1000 simulated stars per $A/\langle A \rangle$ to obtain the best-fit location with estimated uncertainties.

As a criterion for a significant detection, we keep only the result that have a fractional error on the amplitude fit of less than 20%, corresponding to a 5σ significance threshold. The distribution of the locations fit for these stars is approximately Gaussian. Figure 2 displays the average standard error reported by our fitting procedure as a function of $A/\langle A \rangle$ as solid lines. The standard deviations of the residuals between simulated and fitted position are plotted as points for each signal-to-noise level, and they follow the reported uncertainties from the fits to within a few percent. Figure 3 shows the fraction of simulations that yielded a 5σ detection for different signal-to-noise ratios. Our method successfully fits signals detected to a significance of $A/\langle A \rangle > 7.5$ every time, with intrinsic errors less than 0.1 pix in each direction.

3.2. Case 2: Star Location

To test whether the precision of our method degrades when light from the variable star is not entirely contained within the TPF, we simulated a star 1000 times each at locations with various distances from the edge of

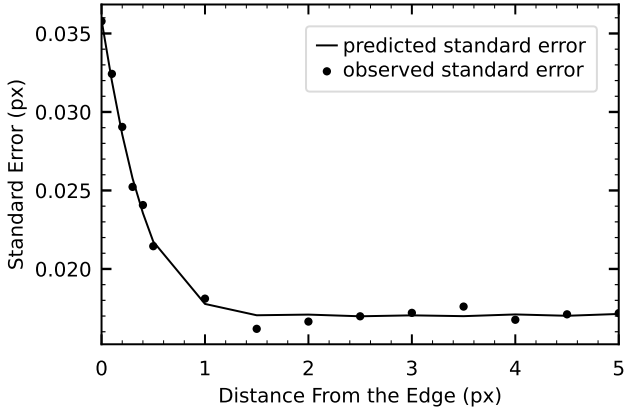


Figure 4. Errors on the position of the fit star as a function of the distance from the edge of the TPF.

the image in the Y-axis, but centered in the X-axis. We found that a star with a $A/\langle A \rangle > 7$ could be successfully fit to the edge of a TPF with fit residuals that agree to within a tenth of the reported errors. Figure 4 shows the predicted and observed uncertainties for varying source distances from the edge of the TPF. Fit locations become less precise as light from the source falls off the edge of the TPF, but this is accurately reflected in the reported intrinsic error. The `TESS_localize` code raises a warning if the fit indicates that most of the flux is located outside the TPF. You may achieve a more precise localization using `TESSCut` (Brasseur et al. 2019) to retrieve pixel data encompassing more light from the source.

3.3. Case 3: Two Blended Stars

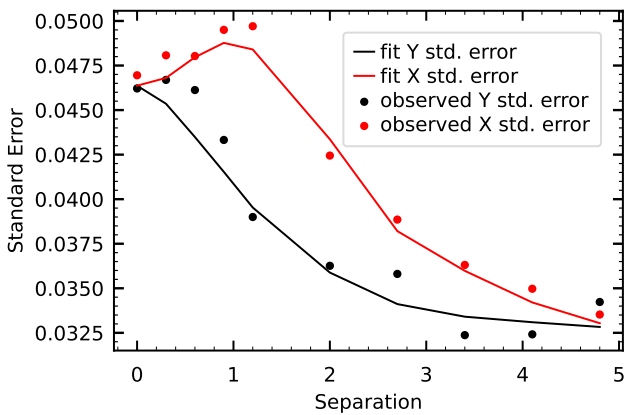


Figure 5. Predicted and observed X and Y standard errors as a function of the separation between the source star and the non-source star.

To determine the effectiveness of our method where light from multiple sources is blended in the same pixels,

we introduced a second star with constant brightness at different distances in the X direction from our target star. This second star was set to have the same average flux as the variable star. The variable star by itself has a flux of 1000 and a variability of 3% resulting in a $A/\langle A \rangle$ by itself of approximately 16, which is sufficient for our code to obtain a $> 5\sigma$ detection for all trials. For each separation from the central variable star, we simulated 1000 data sets.

The plot of the typical reported and observed uncertainties in the best-fit column and row for different source separations in Figure 5 shows three distinct features. First, column and row uncertainties match when the stars are simulated at the same position or are completely separated. Second, the error in column position initially increases with separation as the combined starlight is elongated along this dimension, and the additional contaminant flux increases the fitting uncertainty on signal amplitudes. Finally, the overall precision improves as separation increases since the $A/\langle A \rangle$ in the target aperture increases as contamination decreases. Despite these effects, the uncertainties reported by our fitting procedure closely match the typical residuals for these simulations.

4. DEMONSTRATIONS WITH TESS DATA

The simulations in the preceding section support that our methodology can faithfully recover a variable source location in pixel data in idealized cases (e.g., Gaussian point spread function, no data systematics), with reported errors matching the intrinsic uncertainty of the fitting procedure. Here we apply our localization tools to actual TESS data, starting with a practical analysis that we contributed to Córscico et al. (2021). We then apply our tools to an ensemble of eclipsing binary systems that enables us to define a model for extrinsic uncertainty and pointing systematics that we incorporate into the `TESS_localize` software.

4.1. Pulsations and Eclipses in the Light Curve of the GW Vir Star RX J2117.1+3412

Córscico et al. (2021) carried out an analysis of 2-min cadence TESS observations of six GW Vir type pulsating white dwarfs. The light curve of one of these targets, RX J2117.1+3412 (TIC 117070953, Sector 15), showed evidence of both eclipses and pulsations. This implied an exciting interpretation that this hot pulsating white dwarf in a young planetary nebula could also be part of an eclipsing binary system. Due to the prevalence of source blending in TESS data, this hypothesis must be rigorously tested, and Córscico et al. (2021) were able to reject this interpretation with the use of `TESS_localize`.

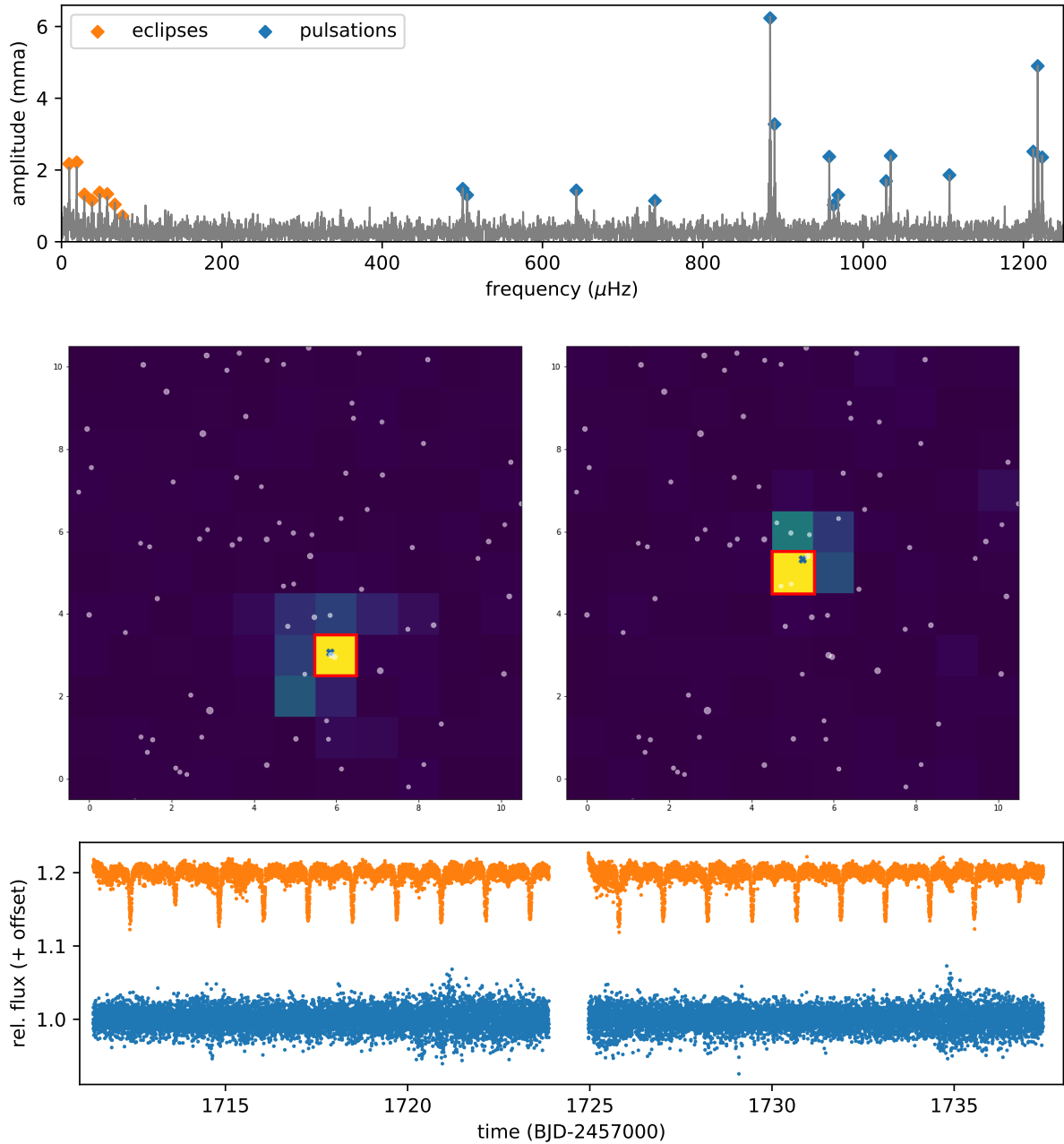


Figure 6. TOP: Periodogram of RX J2117.1+3412. Low-frequency signals of an eclipsing binary (orange) and high-frequency signals from stellar pulsations (blue) fit with `Pyriod` are marked. MIDDLE: Heatmaps for low- (left) and high-frequency (right) signals. These two different types of signal clearly originate from different positions on the sky. The gray points indicate the positions of *Gaia* DR2 sources brighter than $G = 18$ mag in the field. BOTTOM: Light curves extracted from the single hottest pixels in each heatmap (marked with a red border in heatmaps). The binary eclipse signature is confirmed to be much more pronounced in the light curve extracted away from the white dwarf target (orange), compared to the light curve extracted at the position of the white dwarf (blue).

The Lomb-Scargle periodogram for this light curve in the region of significant variability signals is displayed in the top panel of Figure 6. In addition to the fifteen pulsation signatures in the frequency range 500–1225 μHz , there appears a sequence of low-frequency harmonics at multiples of 9.511 μHz , indicative of binary eclipses. Because the high- and low-frequency signals appear to be associated with different physical processes, we test each set of signals independently for consistency between the concentration of power on the sky and the location of the GW Vir target. The two sets of signals that we fit with the pre-whitening frequency analysis software `Pyriod`⁴ are marked with different colors in the top panel of Figure 6, and the heatmaps for each are displayed in the middle panels. These heatmaps are a result of summing the individual amplitude heatmaps for each frequency and dividing by the root sum squared of the error heatmaps. While the power associated with pulsations is concentrated in the pixels surrounding the white dwarf RX J2117.1+3412, the signals from binary eclipses are significantly offset from our target source. To confirm this analysis, we extract light curves from the single “hottest” pixel associated with each set of signals, normalize and remove long-term trends with `lightkurve`, and display these in the bottom panel of Figure 6. The binary eclipses are far more pronounced in the light curve extracted farther from the white dwarf target where the binary signal amplitudes appear largest, confirming that the eclipsing binary is a different contaminating source in the field. RX J2117.1+3412 was dismissed as an eclipsing binary by [Córscico et al. \(2021\)](#) on the basis of this analysis.

The full code needed to localize the eclipse and pulsation signals in the TIC 117070953 TPF is provided in Appendix A. Besides measuring the frequencies of variability from a periodogram, which we assume has been done already, it only takes one line of code to download the TPF data with `lightkurve`, and one line of code to localize the source with `TESS_localize`. The localization step takes roughly one minute on a typical consumer laptop.

4.2. *Eclipsing binary systems*

[Prša et al. \(2022\)](#) derived a catalog of 4584 eclipsing binaries in the first two years of short-cadence TESS data.⁵ We run our code on the 120-s cadence light curves for these targets to test its performance. Starting with the orbital periods tabulated by [Prša et al.](#), we use `Pyriod` to automatically determine the frequen-

cies of greatest power in the periodograms of the 3470 eclipsing binaries with orbital periods shorter than five days. The periodogram of an eclipsing binary light curve typically consists of many harmonics of the orbital frequency to reproduce the eclipse profiles, as seen in the previous example. We iteratively determine the highest periodogram value at a multiple of the orbital frequency, find a corresponding best-fit sinusoid to the light curve, update the orbital frequency estimate, and then look for the next highest harmonic. The first signal is required to exceed seven times the median value across the periodogram. We repeat this process on the residuals of the fit to include up to the 20th harmonic (or up to the Nyquist frequency), or until no additional harmonics exceed five times the median periodogram value. Many of these targets were observed in multiple TESS sectors, and we end up with a total of 9774 binary frequency solutions through Sector 42 (allowing for a couple hundred failed downloads), from which we can compare our fitted source location to the location of the targeted star.

We run `TESS_localize` with these targets and frequency lists, using pipeline apertures and the `auto_pca` function to estimate the best number of PCA components (up to three) that are safe to use without potentially removing the signal of interest. For 67% of these eclipsing binaries, `auto_pca` preferred to not perform any PCA correction, while it fit and removed the strongest one, two, and three principal components in 15%, 7%, and 11% of cases, respectively.

`TESS_localize` fits a separate height of the PRF model to the amplitude distribution for each provided frequency. Similar to source detection in an astronomical image, we should apply a significance test that signal is present over the background of noise. We require that the height of at least one component exceed five times its uncertainty to be considered a significant detection, discarding results for only 145 light curves. There is some structure in the resulting best-fit source locations, suggesting that there are systematics present in results from actual TESS data that did not appear in our idealized simulations from Section 3. Figure 7 displays the decimal portions of the source locations returned by `TESS_localize`. There are excesses every ninth of a pixel, which correspond to the locations with modelled PRFs that we interpolate between (see Appendix B). There is also an excess at 0.5, corresponding to pixel edges (pixel centers are defined to have integer positions; [Greisen et al. 2006](#)). It appears to be a feature of fitting with the PRF models that least-squares solutions are more likely to be found at these locations, which is an extrinsic error that will increase the fitting residuals

⁴ github.com/keatonb/Pyriod

⁵ tessebs.villanova.edu

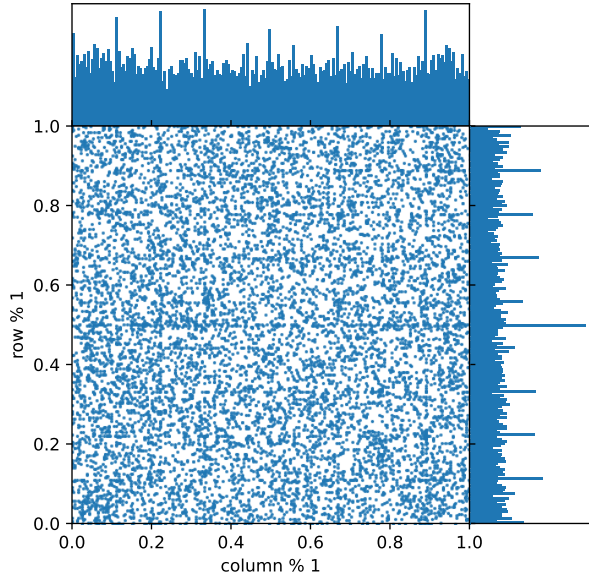


Figure 7. Sub-pixel best-fit locations for 9774 binary stars. Top and right panels display marginalized histograms. Excesses at every ninth of a pixel, as well as at the edge of pixels (0.5) indicate that the procedure of fitting a PRF model to data prefers these locations.

beyond what would be expected from the formal fitting uncertainties.

To assess the effect of extrinsic error on the results, we further restrict our analysis to the fit results that have formal intrinsic fitting errors smaller than 0.05 pix in both dimensions. The median intrinsic uncertainties are smaller than 0.01 pix in each direction, so this rejects only the most imprecise 6% of results in the tail of the fit distribution.

The residuals between the fit locations and the known locations of the targeted stars are displayed in Figure 8. The bulk of the results appear to be distributed about the targeted source locations. In some cases, the eclipsing binary is not the targeted source, and we would expect to fit an offset source location. Any bright star within a pixel of the target will contribute light to the aperture, so we expect this background of contaminant sources to be uniform within a pixel of the target. We use the `pyGMMis` package (Melchior & Goulding 2016) to fit a two-component Gaussian mixture model (GMM) to the residuals, plus a uniform background model. The GMM distributions are indicated in Figure 8, where they appear to match the marginalized histograms well. The locations, covariance matrices, and heights for the two best-fit two-dimensional Gaussians are in Table 1. The uniform background model represents 7.8% of the variable source density within one pixel of the targeted sources. Both Gaussian components are wider than ex-

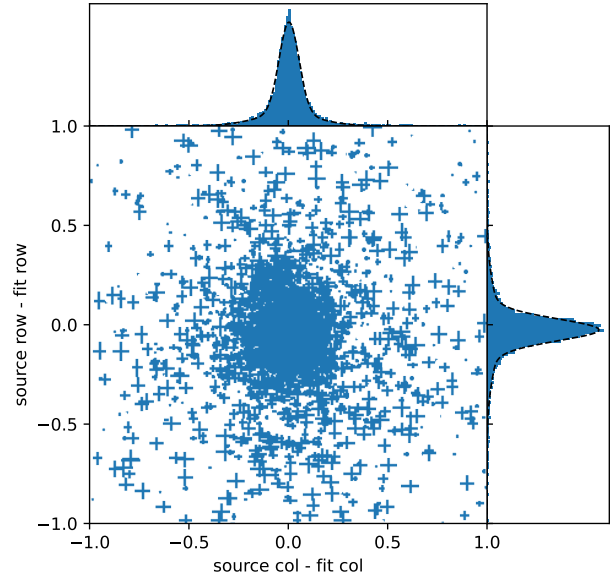


Figure 8. Residuals between the fit locations and the known locations of the targeted stars. Top and right panels display marginalized histograms, and the dashed lines show the marginalized two-component Gaussian mixture model that we adopt as an extrinsic error model in Section 4.2.1

Table 1. Extrinsic error model Gaussian mixture components (source - fit).

Location (column, row)	Covariance (pix ²)	Height
(0.0037, -0.0258)	0.00248 0.00003 0.000030 0.00296	0.653
(-0.0073, -0.0271)	0.0247 -0.0003 -0.0003 0.0383	0.269

pected from the formal fitting uncertainties, capturing the combined intrinsic and extrinsic errors. The best-fit row locations are also notably greater than the targeted source locations according to WCS header info by ≈ 0.026 pix, indicating a potential systematic error in the TESS pointing model of $\approx 0.5''$.

4.2.1. Extrinsic Error Model

The GMM fit to the eclipsing binary location residuals provides a sensible empirical model for the extrinsic error on source locations fit by `TESS_localize`. Both components have scale factors significantly larger than the median intrinsic uncertainties of < 0.01 pix in each direction, so formal fitting errors cannot account for the residuals. The small intrinsic errors in the data will slightly increase the scale of the GMM over what would be expected from purely extrinsic errors, so using this model as a proxy for the extrinsic errors is a conserva-

tive choice that errs on the side of overestimating uncertainty.

`TESS_localize` takes a `pyGMMis` object as an extrinsic error model, with the model fit to the eclipsing binary position residuals as the default. The intrinsic error from the PRF fit given by the covariance matrix is added to the covariance matrices of the GMM model, in effect convolving the extrinsic error model by a Gaussian representation of the fit uncertainties. When evaluating the relative likelihood of nearby *Gaia* sources being the origin of observed variability, the GMM is evaluated at each source location, and the total likelihood of the variability coming from any nearby source is normalized to one, assuming that the variability originates from one of the known sources. An upper magnitude limit can be provided for sources to include in this calculation (default: $G \leq 18$ mag), which is especially useful if the amplitude of variability rules out faint sources.

5. DISCUSSION AND CONCLUSIONS

We have demonstrated a method that can reliably localize the sources of observed variability in TESS data. The amplitudes of sinusoidal variability fit to the unnormalized light curves extracted from each pixel in a TPF are proportional to the fraction of light from the variable source captured by each pixel. A PRF model of how light from a point source is distributed on the detector can be fit to the pixel-by-pixel amplitudes to localize the source of variability in the image. This can be converted to a location on the sky or compared to known positions of potential sources in the images.

The localization procedure is implemented in an open-source Python package `TESS_localize`, available at github.com/Higgins00/TESS-Localize. The code requires two inputs for localization: TPF data passed as a `Lightkurve` object, and the frequencies of observed variability from periodogram analysis. The package returns a best-fit pixel position, sky coordinates, and relative probabilities of this position corresponding to known *Gaia* sources. The few lines of code needed to localize the sources of variability considered in Section 4.1 are provided in Appendix A.

Most TESS light curves represent the blended light from multiple sources due to the exceptionally large plate scale of $21'' \text{ pix}^{-1}$. This results in a significant risk of misattributing observed variability to the wrong source. The localization procedure developed here mitigates this risk, and we encourage that every analysis of TESS light curves include tests to confirm the source of variability. Not all variability in a light curve necessarily originates from one source, so using `TESS_localize` to test that subsets of signal frequencies appear to orig-

inate from a consistent location is advised. An example analysis from [Córscico et al. \(2021\)](#) was detailed in Section 4.1, where signals from pulsations and eclipses in the TPF of TIC 117070953 were shown to come from two different sources. [Mullally et al. \(2022\)](#) demonstrated a good use of `TESS_localize` when using TESS data to vet the suitability of proposed spectrophotometric standard stars for James Webb Space Telescope instrument calibration.

The main underlying assumption to our method that must be satisfied in order to obtain reliable results is that only one source in the TPF is variable at the provided frequencies. It is unlikely for multiple stars in a TPF to be variable at the same frequency by chance due to the high frequency resolution of $0.43 \mu\text{Hz}$ for a 27-day TESS sector. This is more likely to be a problem if another source in the TPF exhibits broadband variability that spans many frequency bins. If more than one source is variable at a provided frequency, the fitted amplitudes in the corresponding heatmap will not be distributed like the PRF. This issue might be circumvented by omitting the problematic frequency if it is part of a set of frequencies that must originate from the same target.

Results could also be negatively affected if systematic trends in the light curves also vary at a provided frequency. `TESS_localize` includes an option to attempt to remove systematics with principal component analysis using `Lightkurve` tools ([Lightkurve Collaboration et al. 2018](#)). It is important to check that the principal component trends determined from pixels outside the provided aperture will not remove the variability of interest, which you can do by inspecting their periodograms.

It is important that the signal you wish to localize is strong within the provided aperture (pipeline aperture by default). Signal phases are fixed by `TESS_localize` to the values that fit best to the light curve extracted from the aperture, so the quality of results are limited to the precision of this initial fit. It may be the case that most power at the given frequencies is located outside the aperture, and that a better fit could be obtained by providing a different aperture. There is an option to try to automatically choose a best aperture where the Lomb-Scargle periodogram values at the input frequencies are highest. A more precise localization may be obtained by using the results of an initial fit to choose a new aperture that better captures the signal of interest. If most of the light from the variable source is located off the edge of the TPF, a more suitable set of pixels could be obtained from the FFI with `TESSCut` ([Brasseur et al. 2019](#)), though only at the long FFI cadence which may be insensitive to high frequency variability.

If the above assumptions are not satisfied, results obtained by `TESS_localize` will be of poor quality. We recommend inspecting the following to ensure reliable localization results: that the amplitude heatmap appears to be distributed like a PRF, that the detection is significant with a fitted height exceeding five times its uncertainty, and that the signals are localized to a position consistent with an astronomical source. More guidance for troubleshooting `TESS_localize` results and up-to-date usage instructions are available on the package GitHub repository.

ACKNOWLEDGEMENTS

K.J.B. is supported by the National Science Foundation under Award AST-1903828. We thank Jon Jenkins, Rebekah Hounsell, Roland Vanderspek, Michael Fausnaugh, Scott Fleming, Susan Mullally, and Andrej Prša for helpful discussions. This paper includes data collected with the TESS mission, obtained from the MAST

data archive at the Space Telescope Science Institute (STScI). Funding for the TESS mission is provided by the NASA Explorer Program. STScI is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5–26555. This research made use of `Lightkurve`, a Python package for Kepler and TESS data analysis ([Lightkurve Collaboration et al. 2018](#)). This work has made use of data from the European Space Agency (ESA) mission *Gaia* (<https://www.cosmos.esa.int/gaia>), processed by the *Gaia* Data Processing and Analysis Consortium (DPAC, <https://www.cosmos.esa.int/web/gaia/dpac/consortium>). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement.

Software: `Astropy` ([Astropy Collaboration et al. 2013, 2018](#)), `Lightkurve` ([Lightkurve Collaboration et al. 2018](#)), `lmfit` ([Newville et al. 2018](#)), `Matplotlib` ([Hunter 2007](#)), `pyGMMis` ([Melchior & Goulding 2016](#))

REFERENCES

- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, *A&A*, 558, A33, doi: [10.1051/0004-6361/201322068](https://doi.org/10.1051/0004-6361/201322068)
- Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, *AJ*, 156, 123, doi: [10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f)
- Baran, A. S., & Koen, C. 2021, *AcA*, 71, 113, doi: [10.32023/0001-5237/71.2.3](https://doi.org/10.32023/0001-5237/71.2.3)
- Borucki, W. J., Koch, D., Basri, G., et al. 2010, *Science*, 327, 977, doi: [10.1126/science.1185402](https://doi.org/10.1126/science.1185402)
- Brasseur, C. E., Phillip, C., Fleming, S. W., Mullally, S. E., & White, R. L. 2019, *Astrocut: Tools for creating cutouts of TESS images*. <http://ascl.net/1905.007>
- Brown, A. G. A., Vallenari, A., Prusti, T., et al. 2018, *Astronomy Astrophysics*, 616, A1, doi: [10.1051/0004-6361/201833051](https://doi.org/10.1051/0004-6361/201833051)
- Bryson, S. T., Tenenbaum, P., Jenkins, J. M., et al. 2010, *ApJL*, 713, L97, doi: [10.1088/2041-8205/713/2/L97](https://doi.org/10.1088/2041-8205/713/2/L97)
- Bryson, S. T., Jenkins, J. M., Gilliland, R. L., et al. 2013, *PASP*, 125, 889, doi: [10.1086/671767](https://doi.org/10.1086/671767)
- Collins, K. A., Collins, K. I., Pepper, J., et al. 2018, *AJ*, 156, 234, doi: [10.3847/1538-3881/aae582](https://doi.org/10.3847/1538-3881/aae582)
- Colman, I. L., Huber, D., Bedding, T. R., et al. 2017, *MNRAS*, 469, 3802, doi: [10.1093/mnras/stx1056](https://doi.org/10.1093/mnras/stx1056)
- Córsico, A. H., Uzundag, M., Kepler, S. O., et al. 2021, *A&A*, 645, A117, doi: [10.1051/0004-6361/202039202](https://doi.org/10.1051/0004-6361/202039202)
- Eisner, N. L., Lintott, C. J., & Aigrain, S. 2020, *Journal of Open Source Software*, 5, 2101, doi: [10.21105/joss.02101](https://doi.org/10.21105/joss.02101)
- Giagalone, S., Dressing, C. D., Jensen, E. L. N., et al. 2020, *The Astronomical Journal*, 161, 24, doi: [10.3847/1538-3881/abc6af](https://doi.org/10.3847/1538-3881/abc6af)
- Greisen, E. W., Calabretta, M. R., Valdes, F. G., & Allen, S. L. 2006, *A&A*, 446, 747, doi: [10.1051/0004-6361:20053818](https://doi.org/10.1051/0004-6361:20053818)
- Hedges, C. 2021, *Research Notes of the American Astronomical Society*, 5, 262, doi: [10.3847/2515-5172/ac376a](https://doi.org/10.3847/2515-5172/ac376a)
- Hedges, C., Luger, R., Martinez-Palomera, J., Dotson, J., & Barentsen, G. 2021, *AJ*, 162, 107, doi: [10.3847/1538-3881/ac0825](https://doi.org/10.3847/1538-3881/ac0825)
- Hunter, J. D. 2007, *Computing in Science & Engineering*, 9, 90, doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
- Lightkurve Collaboration, Cardoso, J. V. d. M., Hedges, C., et al. 2018, *Lightkurve: Kepler and TESS time series analysis in Python*, *Astrophysics Source Code Library*. <http://ascl.net/1812.013>
- Melchior, P., & Goulding, A. D. 2016, *pyGMMis: Mixtures-of-Gaussians density estimation method*. <http://ascl.net/1611.013>
- Montgomery, M. H., & O'Donoghue, D. 1999, *Delta Scuti Star Newsletter*, 13, 28
- Mullally, S. E., Sloan, G. C., Hermes, J. J., et al. 2022, *arXiv e-prints*, arXiv:2201.03670. <https://arxiv.org/abs/2201.03670>
- Newville, M., Otten, R., Nelson, A., et al. 2018, *lmfit/lmfit-py 0.9.12*, 0.9.12, Zenodo, doi: [10.5281/zenodo.1699739](https://doi.org/10.5281/zenodo.1699739)

Prša, A., Kochoska, A., Conroy, K. E., et al. 2022, ApJS,
258, 16, doi: [10.3847/1538-4365/ac324a](https://doi.org/10.3847/1538-4365/ac324a)

Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2015,
Journal of Astronomical Telescopes, Instruments, and
Systems, 1, 014003, doi: [10.1117/1.JATIS.1.1.014003](https://doi.org/10.1117/1.JATIS.1.1.014003)
van Roestel, J., Bellm, E. C., Duev, D. A., et al. 2019,
Research Notes of the American Astronomical Society, 3,
136, doi: [10.3847/2515-5172/ab459c](https://doi.org/10.3847/2515-5172/ab459c)

APPENDIX

A. TESS_localize CODE EXAMPLE

The Python code below reproduces the localizations of the sources of both pulsational and eclipsing binary variability observed in the TPF for target TIC 117070953 that was presented in Section 4.1.

```

1 import TESS_Localize as tl
2 import lightkurve as lk
3 import astropy.units as u
4
5 #Binary frequencies
6 low_frequency_list = [9.51112996, 19.02225993, 28.53338989, 38.04451986,
7                       47.55564982, 57.06677979, 66.57790975, 76.08903972]
8
9 #Pulsation frequencies
10 high_frequency_list = [500.559, 506.057, 642.255, 740.266, 884.017,
11                       889.556, 957.817, 963.28, 969.013, 1028.729,
12                       1034.356, 1107.713, 1212.297, 1217.872, 1223.429]
13
14 #Download TPF data for this target with lightkurve
15 tpf = lk.search_targetpixelfile('TIC117070953', sector=15, cadence=120).download()
16
17 #Localize the low frequencies (binary eclipses)
18 low = tl.Localize(targetpixelfile=tpf, frequencies=low_frequency_list, frequit=u.uHz,
19                  principal_components='auto')
20
21 #Localize the high frequencies (stellar pulsations)
22 high = tl.Localize(targetpixelfile=tpf, frequencies=high_frequency_list, frequit=u.uHz,
23                   principal_components='auto')
```

B. SAMPLING THE TESS PIXEL RESPONSE FUNCTION WITH TESS_PRF

As an ancillary product of this work, we developed a Python package for interpolating the TESS Pixel Response Function (PRF) models called TESS_PRF, available at github.com/keatonb/TESS_PRF. The PRF is analogous to a point spread function, describing how light from a point source will be distributed at different positions on the detector. The PRF models are an important TESS data product with many potential uses, such as simulating realistic TESS data or performing PRF photometry. In this work, we used the PRF models to fit to pixel-by-pixel amplitude distributions to achieve our source localization results.

The TESS PRF model files are available on MAST at heasarc.gsfc.nasa.gov/docs/tess/observing-technical.html#point-spread-function. They were created following the methods developed for the *Kepler* spacecraft described in Bryson et al. (2010). Different sets of models are applicable to TESS Sectors 1-3 and Sectors 4 and later. For each camera and CCD, PRF model files are available for a grid of pixels separated by 512 columns or rows. Each of these files contains 81 PRF models depending on where the flux source is located within a given pixel. These are sampled every ninth of a pixel in each direction, starting from 1/18th of a pixel from the edge, and with the central model corresponding to the center of the pixel. These models are interleaved so that the PRF values at one of these subpixel locations are given by every ninth array value. The esoteric format of these PRF files motivated our creation of the TESS_PRF package.

TESS_PRF was built for local analysis of TESS pixel subregions, such as a TPF, so that the PRF can be assumed not to change appreciably across the image. As a first step, TESS_PRF interpolates between the available PRF files to a given pixel location, which should be near the center of the TPF you wish to analyze. It will access the relevant files from MAST by default, or it can be pointed to a local directory containing these files for faster or offline analysis. After this first interpolation, a second interpolation is performed to position the model at a given subpixel location within the TPF with the `locate` function. See package documentation for detailed usage instructions.