A PSF photometry tool for NASA's Kepler, $\mathcal{K}2$, and TESS missions

José Vinícius de Miranda Cardoso[1,2,4], **Geert Barentsen**[1,2], **Benjamin T. Montet**[3,⋆], **Ann Marie Cody**[1,2], **Christina Hedges**[1,2], **Michael Gully-Santiago**[1,2], **Martin Still**[4], and **Thomas Barclay**[5]

[1]NASA Ames Research Center, Moutain View, CA, USA

[2]Bay Area Environmental Research Institute, Petaluma, CA, USA

[3]Department of Astrophysics, University of Chicago, IL, USA

[4]NASA Headquarters, Washington DC, USA

[5]NASA Goddard Space Flight Center, Greenbelt, MD, USA
⋆NASA Sagan Fellow

Introduction

- NASA's Kepler and $\mathcal{K}2$ missions have been delivering high-precision time series data for a wide range of stellar types.
- However, both the official and community developed pipelines [1–3] tend to focus on studying isolated stars using simple aperture photometry, which tends to perform sub-optimally in crowded fields and on faint stars. [4, 5].
- Although Point Spread Function (PSF) photometry methods are well known [4, 5], open source tools to perform PSF photometry specifically on Kepler and $\mathcal{K}2$ data are scarce. To address this issue, we present an open source PSF photometry toolkit for Kepler and $\mathcal{K}2$, and extensible to TESS, as part of the $\mathcal{K}2$ Guest Observer Office data analysis tool, PyKE.

Methods

Fitting multiple PSFs jointly

Given an image with n pixels and m stars, we treat the pixel values Y_i as independent *non-identically* distributed Poisson random variables $\boldsymbol{Y} \triangleq \{Y_i\}_{i=1}^n$ such that $\mathbb{E}[Y_i] = \sum_{j=1}^m \lambda_i(\boldsymbol{\Theta}_j)$, where λ_i is the PSF model at the i-th pixel and $\boldsymbol{\Theta}_j$ is the vector of random variables which encode the flux and center of each star and the background.

The likelihood function of the pixel values given the model parameters can then be written as

$$P\left(\boldsymbol{Y}=\boldsymbol{y}\,\middle|\,\{\boldsymbol{\Theta}_j\}_{j=1}^m=\{\boldsymbol{\theta}_j\}_{j=1}^m\right)=\exp\left(-\sum_{i=1}^n\sum_{j=1}^m\lambda_i(\boldsymbol{\theta}_j)\right)\prod_{i=1}^n\frac{\left(\sum_{j=1}^m\lambda_i(\boldsymbol{\theta}_j)\right)^{y_i}}{y_i!}. \tag{1}$$

And the log likelihood function is

$$\log P\left(\boldsymbol{Y}=\boldsymbol{y}\,\middle|\,\{\boldsymbol{\Theta}_j\}_{j=1}^m=\{\boldsymbol{\theta}_j\}_{j=1}^m\right)=\sum_{i=1}^n\left(-\sum_{j=1}^m\lambda_i(\boldsymbol{\theta}_j)+y_i\log\sum_{j=1}^m\lambda_i(\boldsymbol{\theta}_j)\right). \tag{2}$$

Hence, the Maximum Likelihood Estimator can be formulated as the following optimization problem

$$\boldsymbol{\theta}^*(\boldsymbol{y})=\arg\min_{\boldsymbol{\theta}\in\Lambda}\sum_{i=1}^n\left(\sum_{j=1}^m\lambda_i(\boldsymbol{\theta}_j)-y_i\log\sum_{j=1}^m\lambda_i(\boldsymbol{\theta}_j)\right). \tag{3}$$

Note that it is often necessary to impose prior probability densities on the model parameters to aid the optimization.

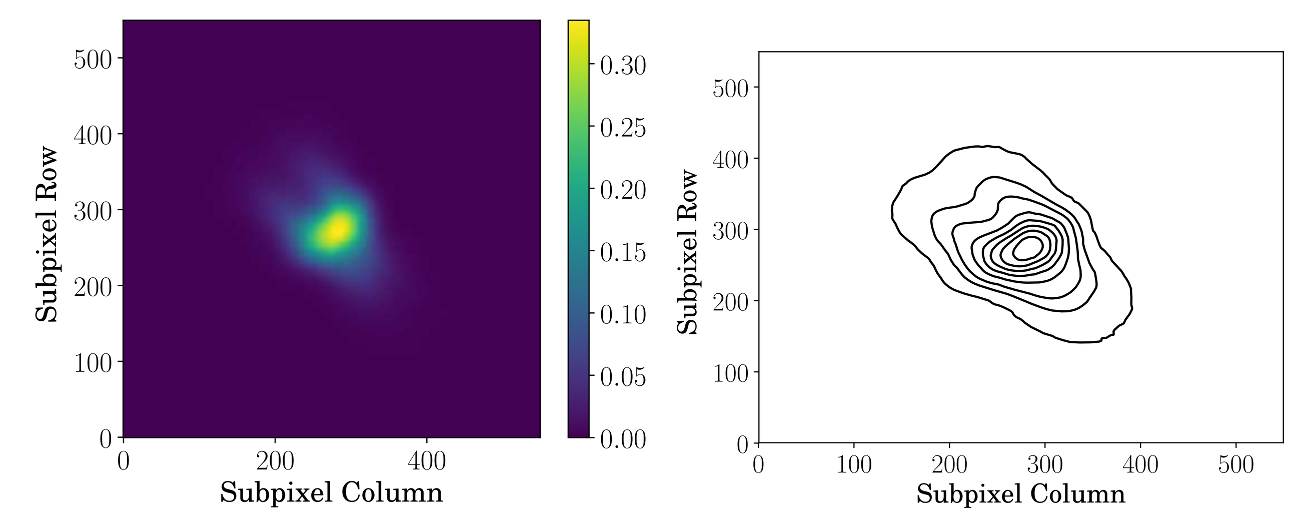
Estimating uncertainties using the Cramér-Rao Lower Bound

Uncertainties on the fitted parameters are approximated using the Cramér-Rao Lower Bound. Mathematically,

$$cov(\boldsymbol{\theta}^*(\boldsymbol{Y}))\geq\left(\mathbb{E}_{\boldsymbol{\theta}}\left[\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{Y}|\boldsymbol{\theta})\left[\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{Y}|\boldsymbol{\theta})\right]^T\right]\right)^{-1}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*(\boldsymbol{y})} \tag{4}$$

The Kepler Pixel Response Function Model

- Kepler's pixel response function (PRF) has been shown to be nonsymmetric and spatially variant accross the detector [6].
- The PRF model used in PyKE is based on dithered data acquired during Kepler's comissioning phase [6].

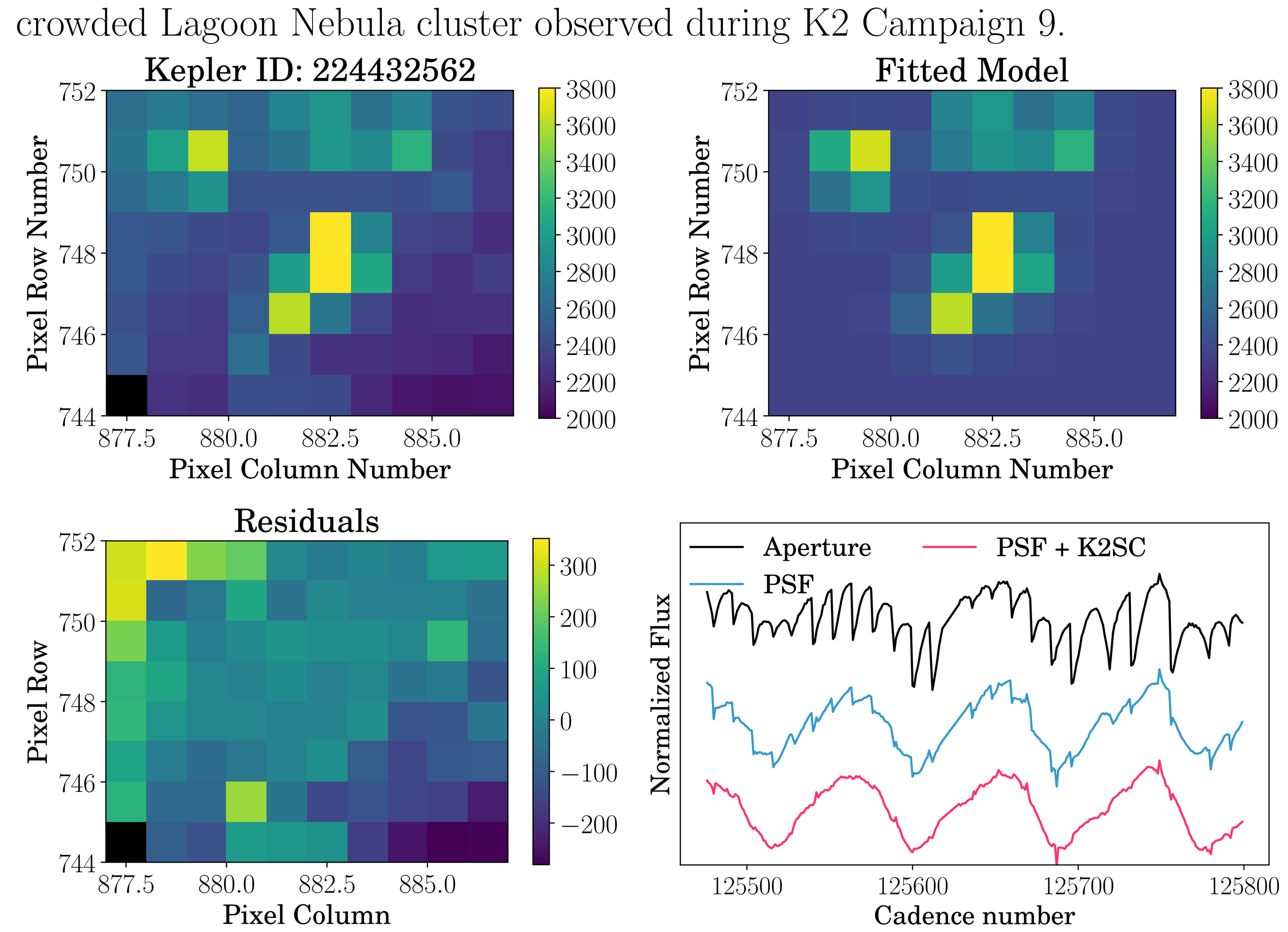


Using the PyKE package in Python, the PSF model for a given detector position can be instantiated using the information contained in Kepler Target Pixel File (TPF) header:

```
>>> from pyke import KeplerTargetPixelFile
>>> tpf = KeplerTargetPixelFile("PATH")
>>> prf = tpf.get_prf_model()
```

Crowded K2 Clusters

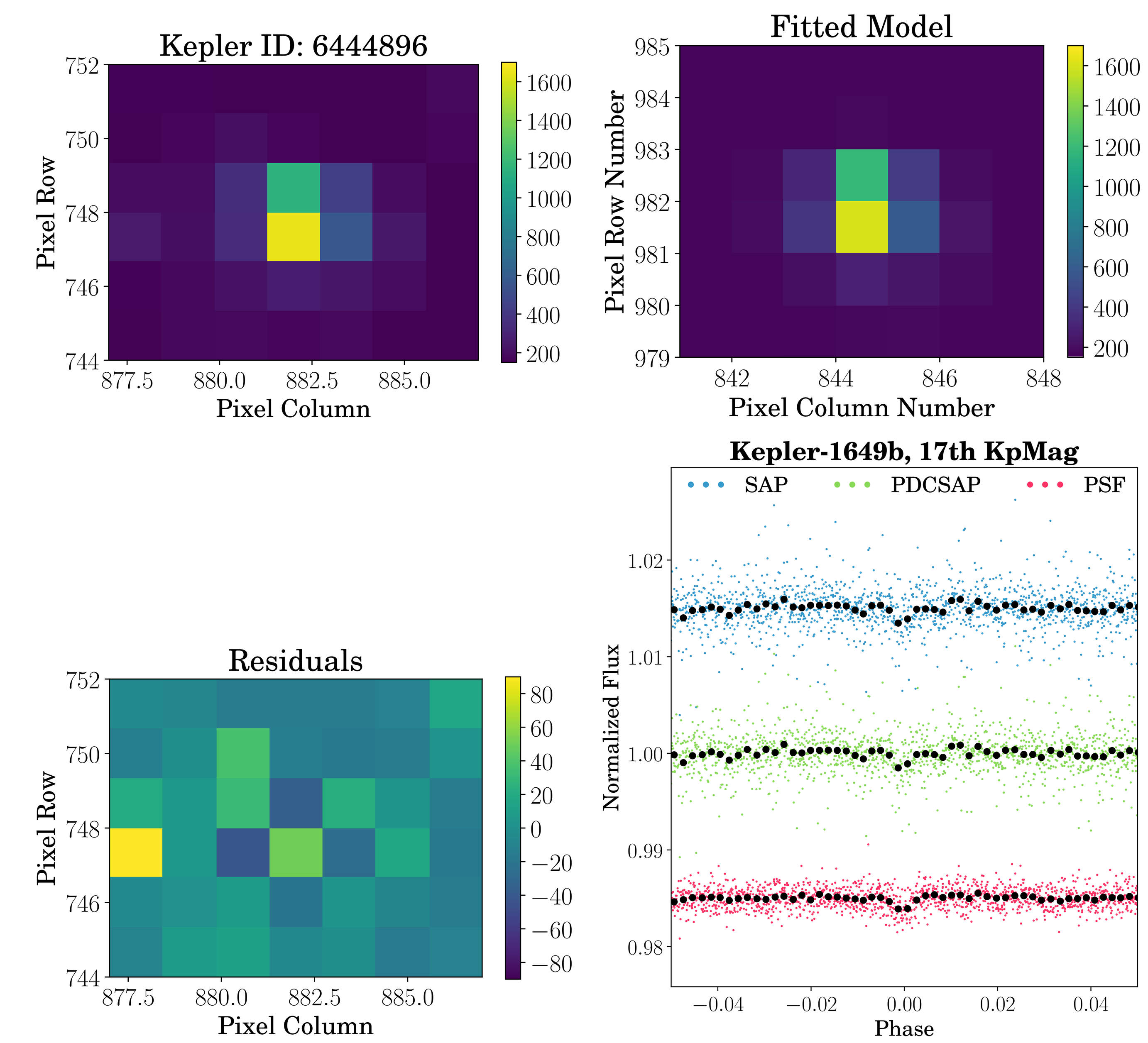
Having instantiated suitable PSF models for a set of stars, we can use PyKE to fit the joint model to the data. In the example shown here, we fit data from the crowded Lagoon Nebula cluster observed during K2 Campaign 9.




```
>>> from pyke.prf import SceneModel, PRFPhotometry
>>> scene = SceneModel(prfs=5*[prf])
>>> phot = PRFPhotometry(prior=prior, scene_model=scene)
>>> results = phot.fit(tpf.flux)
```

Kepler Faint Stars

In addition to being necessary in crowded fields, we find that PSF photometry also benefits the light curves of faint stars. For example, star `KIC 6444896` is known to contain a small planet, Kepler-1649b [7]. Using PSF photometry, we recover the planet signal at a higher SNR. Unlike aperture photometry, PSF photometry is not sensitive to the choice of an aperture mask.



Conclusions

- We have presented an open-source tool to perform PSF photometry on Kepler and $\mathcal{K}2$ data.
- For crowded clusters and faint stars, we find that PSF photometry can outperform aperture photometry.
- In future work, we intend to infer the PSF model from nearby stars to improve the accuracy of the fit. We will also study the performance as a function of stellar brightness and crowding.

References

[1] Luger R. *et al.* *ArXiv e-prints:1702.05488*, February 2017.

[2] Vanderburg A. *et al.* *PASP*, *arXiv:1408.3853*, 126:948, October 2014.

[3] Aigrain S. *et al.* *MNRAS*, *arXiv:1603.09167*, 459:2408–2419, July 2016.

[4] Anderson J. *et al.* *2000PASP..112.1360A*, 112:1360–1382, October 2000.

[5] Libralato M. *et al.* *MNRAS*, *arXiv:1510.09180*, 456:1137–1162, February 2016.

[6] Bryson S. T. *et al.* *APJL*, *arXiv:1001.0331*, 713:L97–L102, April 2010.

[7] Angelo I. *et al.* *AJ*, *arXiv:1704.03136*, 153:162, April 2017.