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], **Ben Montet**[3,⋆], **Ann Marie Cody**[1,2], **Christina Hedges**[1,2] and **Michael Gully-Santiago**[1,2]

[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]Federal University of Campina Grande, Campina Grande, Brazil

⋆NASA Sagan Fellow

Introduction

- NASA's Kepler and K2 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, hence performing sub-optimally in crowded fields
- Although Point Spread Function (PSF) photometry methods are well known [4, 5], as of now, there exist no open source tool that takes into account the caveats inherent in Kepler and $\mathcal{K}2$ data
- To address this issue, we present an open source PSF photometry toolkit for Kepler and K2 and extensible to TESS, as part of the $\mathcal{K}2$ Guest Obverser Office data analysis tool, PyKE

Methods

Fitting multiple PSFs jointly

Consider an experiment that outputs an image with m (m known) stellar objects as a collection of n independent *non-identically* distributed random variables $\boldsymbol{Y} \triangleq \{Y_i\}_{i=1}^n$ (pixels), each of which has expected value $\mathbb{E}[Y_i] = \sum_{j=1}^m \lambda_i(\boldsymbol{\Theta}_j)$, where λ_i is the PSF model at the i-th pixel, $\boldsymbol{\Theta}_j$ is a random vector that enconodes the information about flux and center position of the j-th star. Hence, the likelihood function can be written as

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

Perhaps of more practical interest is the log likelihood function

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

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

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

Furthermore, prior probability densities (often taken to be uniform on Λ) on the stars positions, fluxes, and sky background, are used.

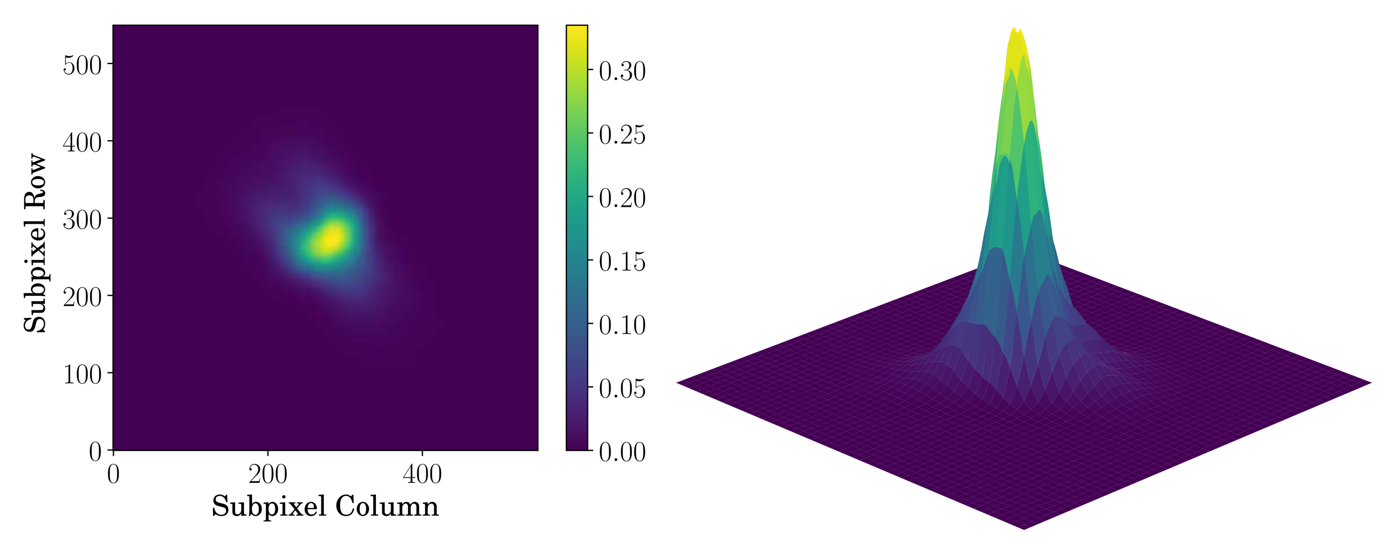
Estimating uncertainties using the Cramér-Rao Lower Bound

Uncertainties on the fitted values are computed using the Cramér-Rao Lower Bound. Mathematically,

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

The Kepler Pixel Response Function Model

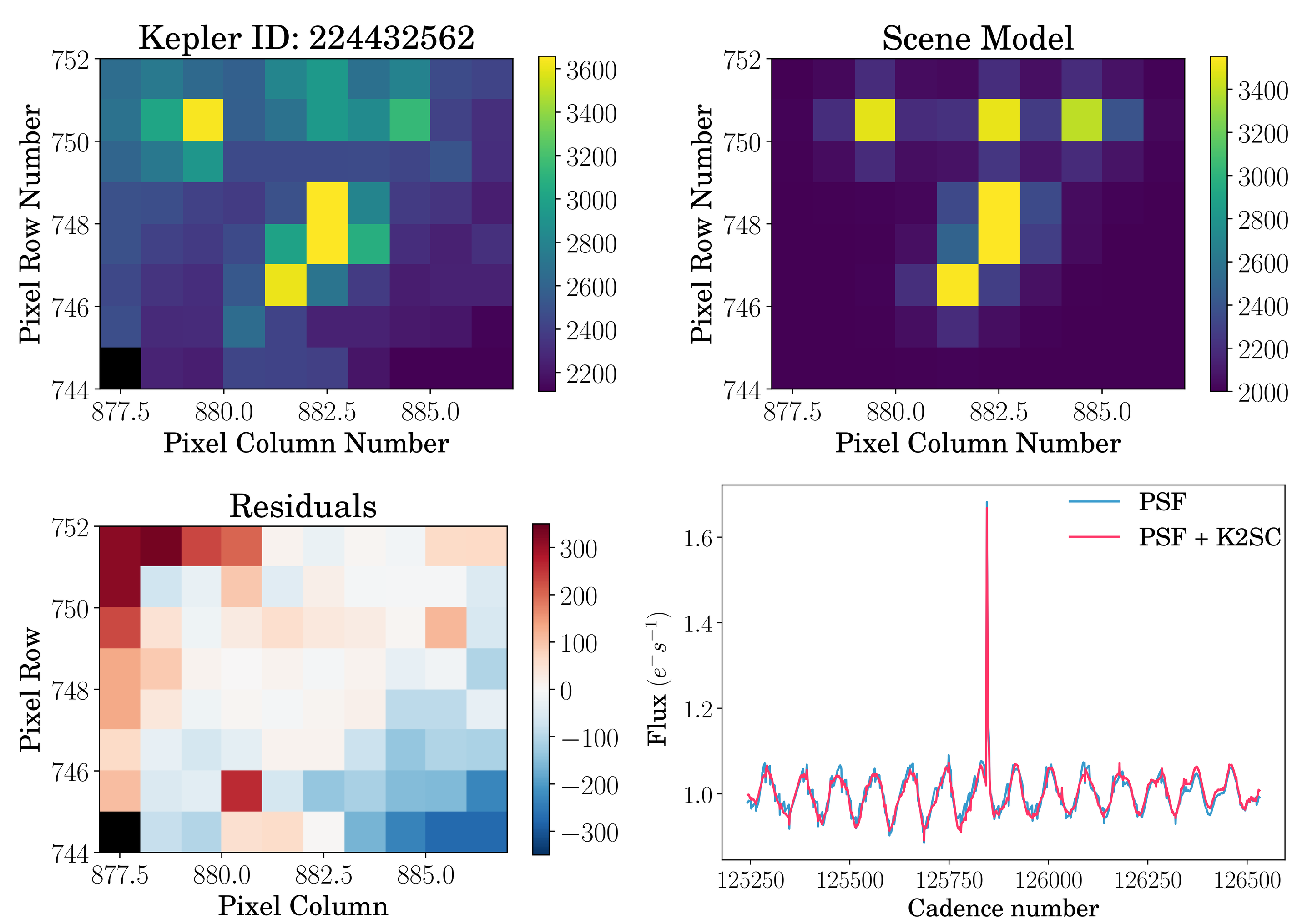
- Kepler's pixel response function (PRF) has been shown to be nonsymmetric and spatially variable accross the detector [6]
- The PRF model used in PyKE is constructed following a similar procedure as stated by Bryson *et al.* [6]



An object of the PSF model can be instantied as follows

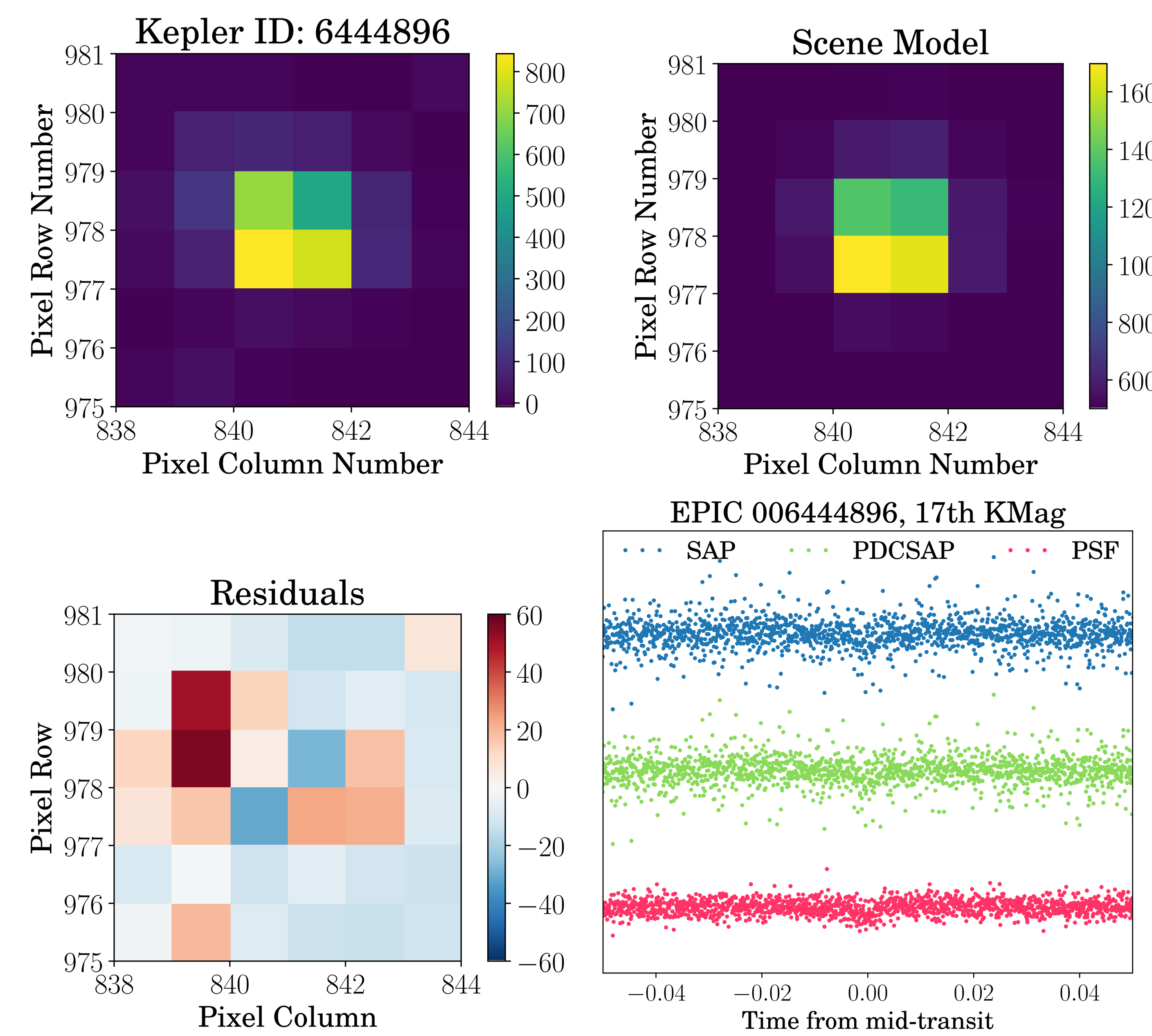
```
>>> from pyke import KeplerTargetPixelFile
>>> tpf = KeplerTargetPixelFile("ADDRESS_TO_TPF")
>>> prf = tpf.get_prf_model()
```

Crowded K2 Clusters




```
>>> 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



Conclusions

- We have presented an open-source tool to perform PSF photometry on Kepler and $\mathcal{K}2$ data which precisely and accurately estimates stellar positions and fluxes on the CCD
- Motion-dependent noise, primarily caused by the subpixel flat-field variations, which are not taken into account in the PSF model, is still present. Nevertheless, it can be easily removed by the procedures developed by detrending pipelines
- As future works, we intend to study novel ways to build the PSF model, especially by using data-driven approaches

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.