Report on background estimation methods for TESS FFIs

Oliver J. Hall

January 25, 2018

Contents

1 Introduction

This quick report breaks down the individual background estimators for TESS Full Frame Image s(FFI) on the TASOC/SkyBackgrounds repository and gives a visual guide to how they fare against increasingly complex backgrounds.

1.1 CvE esitmate

The CvE estimate (method author: Carolina van Essen) attempts to find the background continuum along each row in the FFI and consequently each column, after which these are averaged together.

This is done by binning each column/row of data, finding the value of a user-chosen percentile (default: 10) in each bin, and smoothing across these values to find the background continuum. See Figure 1 for an example on a single row in Y.

Pros:

- Very fast

- Works well when approriate percentile is chosen

Cons:

- Percentile must be chosen by the user (supervised)

- Background has artificial small scale structure due to seperate binning in X and Y

1.2 ML estimate

The ML method (method author: Mikkel Lund) attempts to fit to the background continuum by first finding the mode of subsections of the full FFI and fitting to these.

First, the data is subdivided into squares of a given width and height (default: 128 pixels). Then we run RANSAC to fit to the continuum of the data in each of these squares, producing an inlier mask. The mode is taken as the location of the maximum value of a Kernel Density Estimate (KDE) on the data considered by RANSAC to be an inlier to the continuum (this eliminates high flux pixels containing stars). These modes are then again fit to using RASAC to obtain inlier masks, which are then used as weighting factors for a subsequent polynomial fit of a user-selected order.

Pros:

- Achieves good results when appropriate polynomial order is chosen.

- Mode calculation is very accurate

- Running on multiple FFI frames allows for re-use of RANSAC inlier masks.

Cons:

- Require input of the correct polynomial order (supervised)

- Runs very slowly for a single FFI

- Polynomial fitting eliminates small scale structure
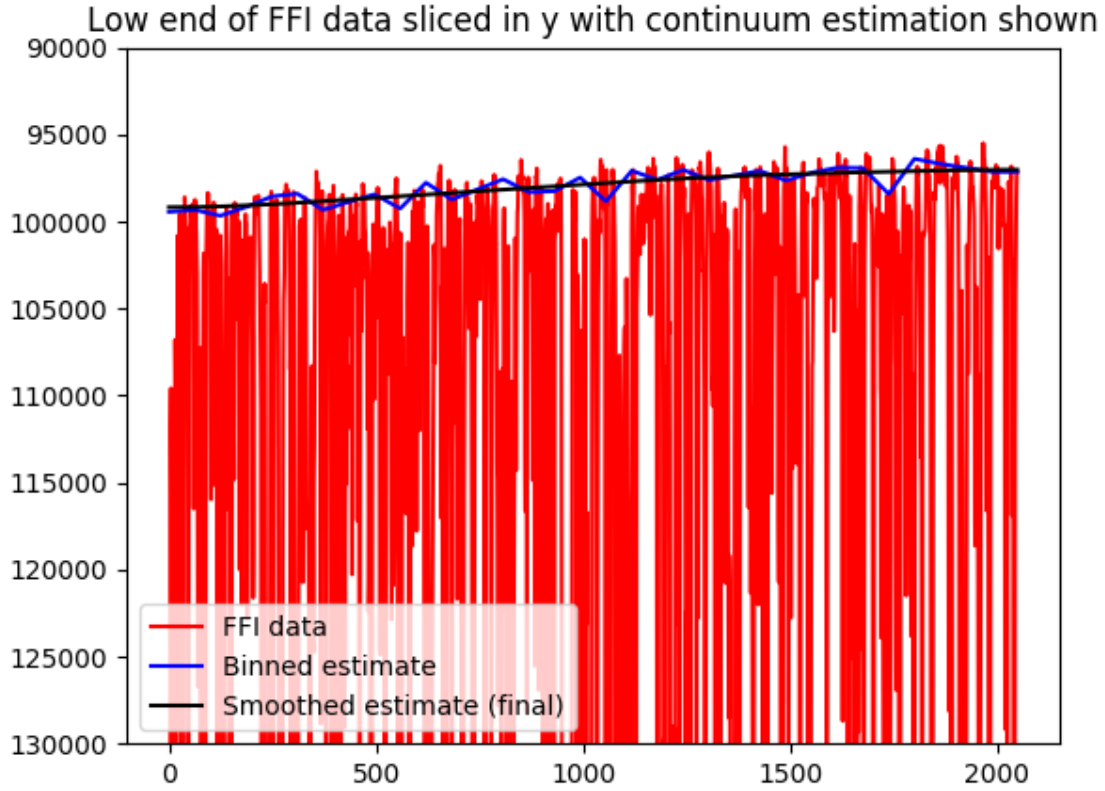
Figure 1: Example of CvE run on the first row in Y for a TESS FFI.

1.3 OH estimate

The OJH method (method author: Oliver Hall) builds on the *Kepler* background estimation, where small 2x2 pixel squares would be measured across a full CCD and then interpolated across to form the background.

This method attempts to do the same by taking squares of 9x9 pixels spread across the CCD and calculating the mode of the pixels inside these squares. To ensure accuracy in the corners and sides of the CCD, the density of the points is doubled. The points are taken as the intersections of an irregular cartesian grid, as was the original method for the *Kepler* background (see: Kepler Handbook). The modes are then fit to using RANSAC to obtain inlier masks, which are then used as weighting factors for a subsequent polynomial of user-selected order. An example of the spread of the measured points can be seen in Figure 2.

Pros:

- Builds upon existing framework for accurate background measurement

- Works quickly

- Mode calculation is very accurate

Cons:

- Requires input of correct polynomial order (supervised)

- Polynomial fitting eliminates small scale structure
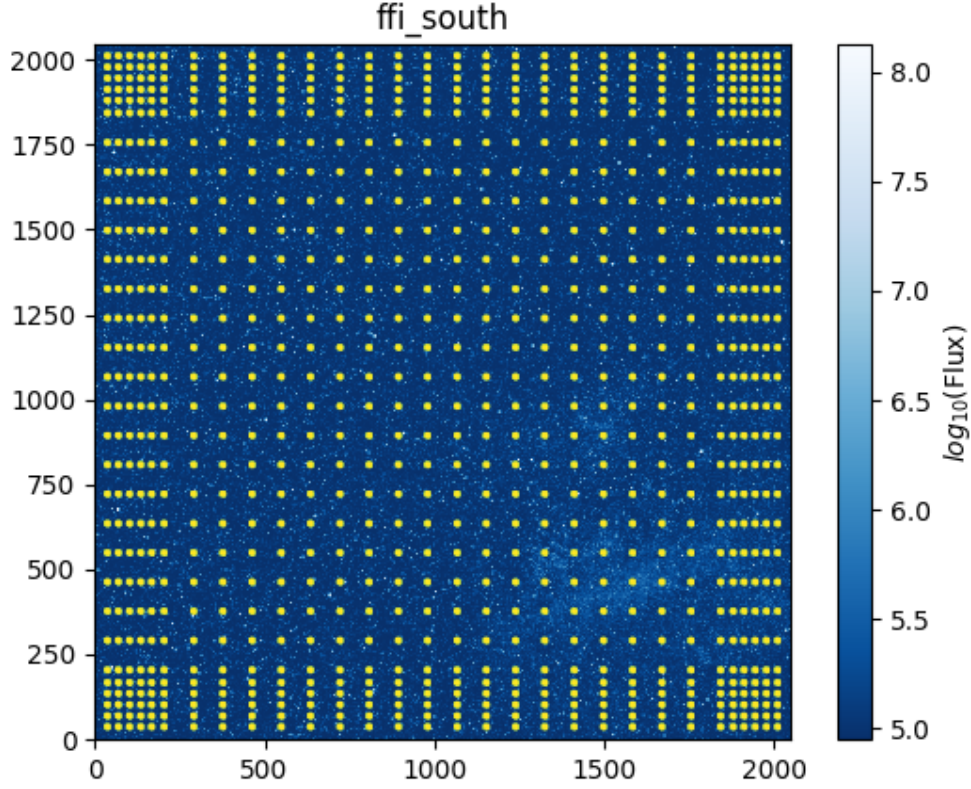
Figure 2: Example of the spread of measurement boxes across a TESS FFI.

1.4 RH estimate

This method (method author: Rasmus Handberg) uses the `photutils` Background2d and SExtractor-Background packages. It first masks out any high flux values, then performs a 5-iteration sigma clip to 3 σ before running the Background2D package using the SExtractorbackground as the background estimator.

Pros:

- Works quickly

- The result is directly sampled from the data and not a featureless polynomial

Cons:

- Not certain how much of the small scale structure is reliable

- Relies on some hard-coded parameters

- Relies heavily on a single external module

2 Test 1: Flat Gaussian Noise Background

To test the background estimators we first applied them to the most simple case: a FFI-shaped flat background with Gaussian noise. The noise was generated in a 2048x2048 Python meshgrid with a mean of 1000 and a standard deviation of 100. The resulting model can be seen in Figure 3.
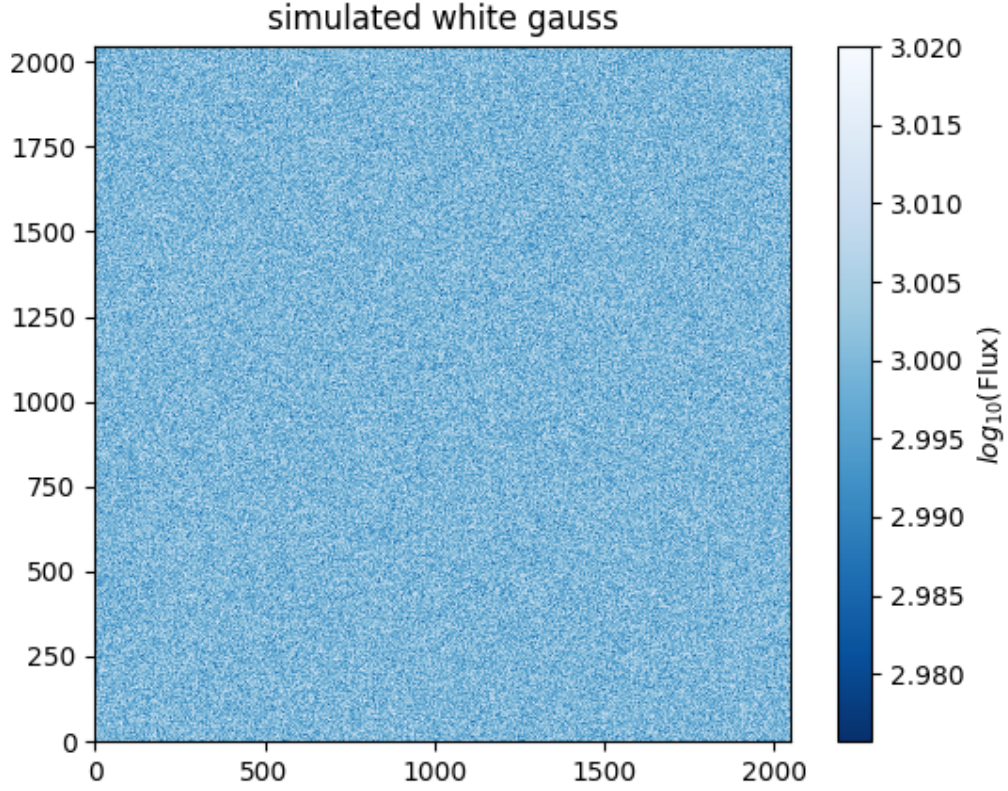
Figure 3: Flat background of gaussian white noise

The resulting distribution of estimated background points can be seen in Figure 5. The residuals on the estimated background can be seen for every 562 data point in Figure 4.

2.1 CvE

The CvE method does not fare very well against this test unsupervised— by default it measures the 10th percentile of the data, and when there are no high-flux stars contaminating each bin it will underestimate the background left unsupervised. However when setting the percentile to 50 it recovers the background to within 1σ and thus performs well.

Changes made during this test:

- Now takes the x-th percentile in each bin where x is a default of 10, as opposed to the minimum.

2.2 ML

The ML method fares well against this test provided it is run with a low-order polynomial fit. Using a default first-order polynomial, the method comes within ¡1% of the background level, although it does find a slope that is not present in the data. The residuals center around 0, and thus the median of the background does yield the correct value.

Changes made during this test:

- The polynomial model now defaults to an order of 1.
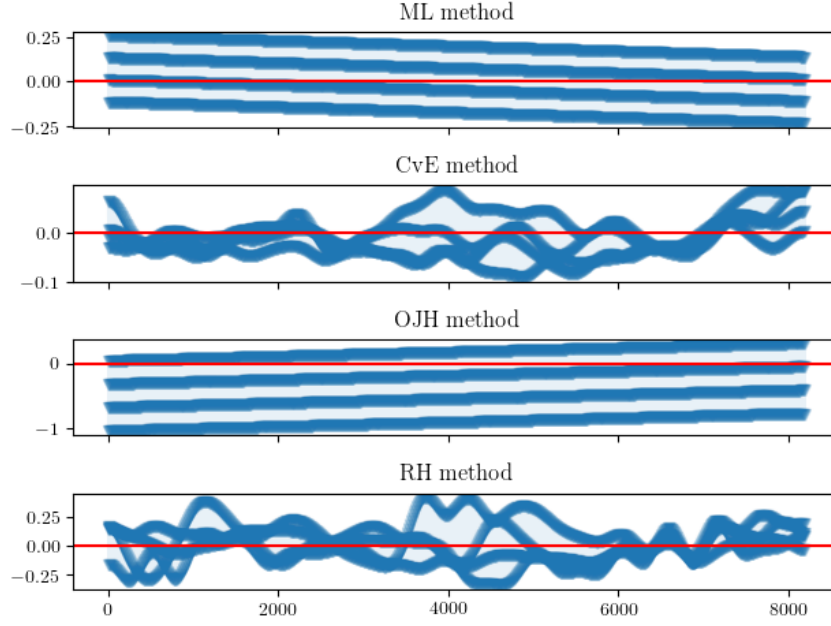
Figure 4: Every 562nd residual for the four different background estimators applied to a white noise background. The y-axis is residuals as a percentage of the background level. Note that only the residuals in one direction are shown, as it takes every 562nd data point
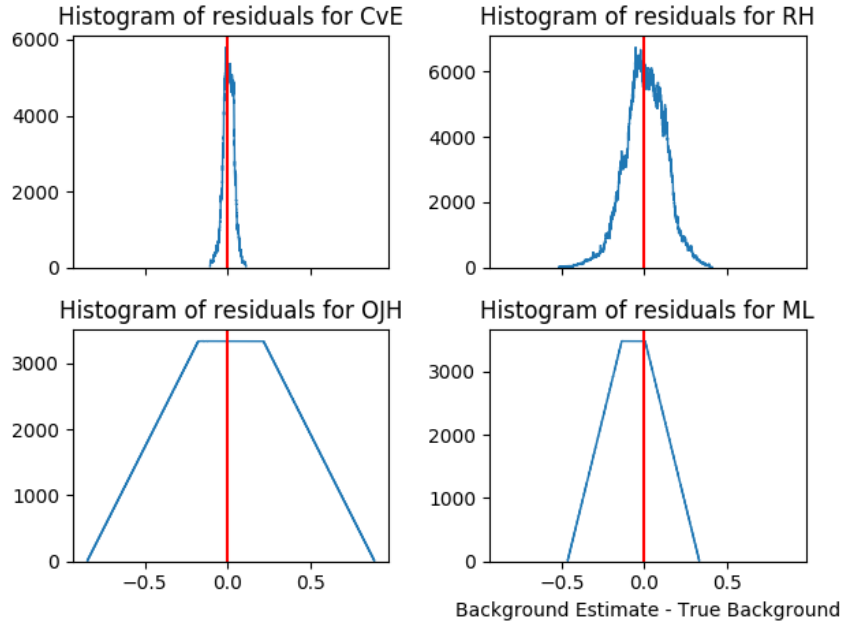
Figure 5: Histograms of the residuals for the four different background estimators applied to a white noise background.

2.3   OJH

The OJH method fares similarly to the MH method, where it finds a non-existent slope in its polynomial fit. In general, as can be seen from the histograms, it centers around the correct value.

Changes made during this test:

- The polynomial model now defaults to an order of 1.

6

- The OJH method now also fits 500 iterations of RANSAC to the modes and uses the resulting inlier mask as weights on the polynomial fit in the same manner as the MH method. Without this addition, the background estimate was biased away from the true value.

2.4   RH

The RH method fares well on this test. No major changes had to be made.

Changes made during this test:

- The background estimation is now smoothed using a median filter before returning from the function.

2.5   Conclusions

In conclusion, all four methods are able to recover a distribution of residuals centered around 0., as would be expected for a simple background with no stars. All four backgrounds progress to the next stage of testing.

(a)

(b)

(c)

(d)
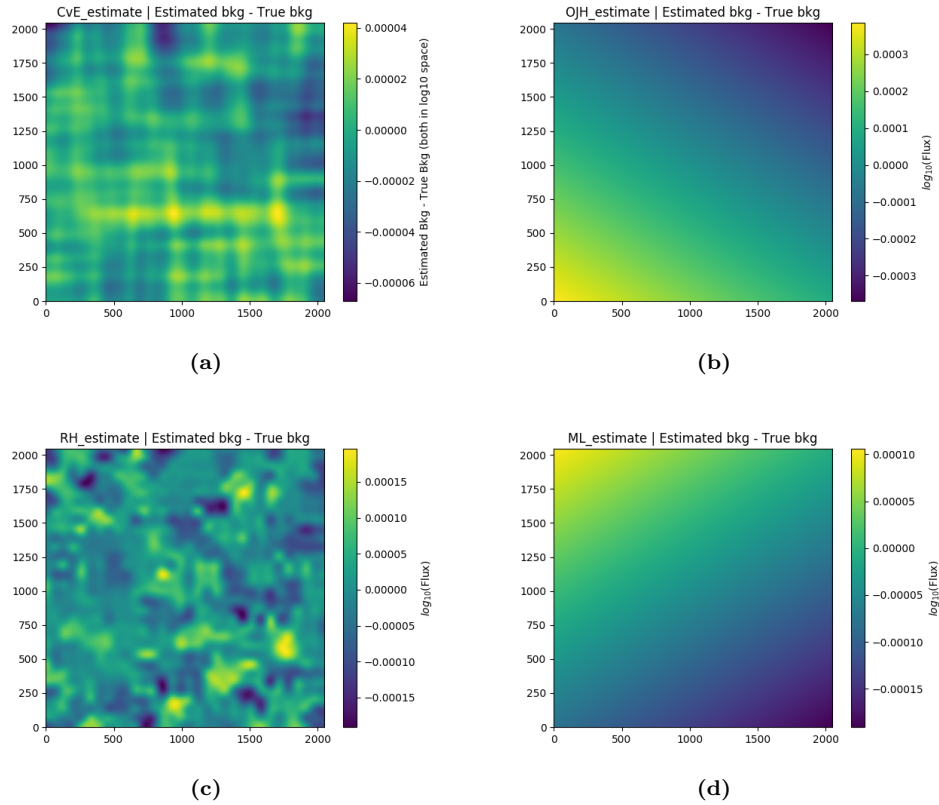
**Figure 6:** Residuals on background fits to a flat gaussian noise background.

3 Test 2: Noise Background + slope and crowding

The next stage of background estimator testing involves a more compliated background function. The background now consists of three components:

- A sloped background

- A large area of increased background level in the shape of 2D Gaussian (to function as a crowded local area)

- 2048 faint stars smoothed over before being added to the background (to function as unfocused stars)

The individual components, as well as the combined background and the simulated (star-less) FFI, can be seen in Figures 7 and 8.
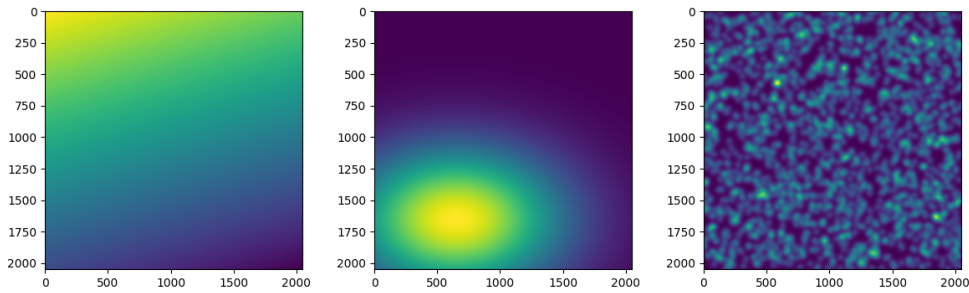
**Figure 7:** The various components used in the new complex background.
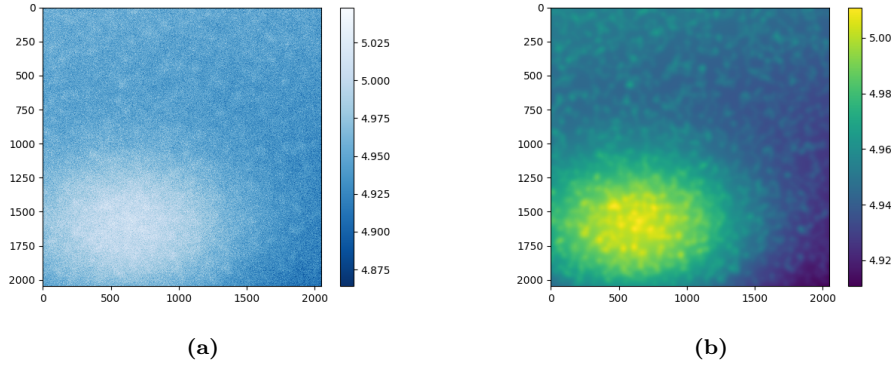
Figure 8: Background without noise (for estimation evaluation) and with noise (on which the background is measured).

The resulting distribution of estimated background points can be seen in Figure 10. The residuals on the estimated background can be seen for every 562nd data point in Figure 9.

3.1 CvE

Thsi method works best out of the four, with the median residual being within one standard deviation of 0. No changes to this method have been made on this iteration. The use of a percentile of 50 has been maintained.
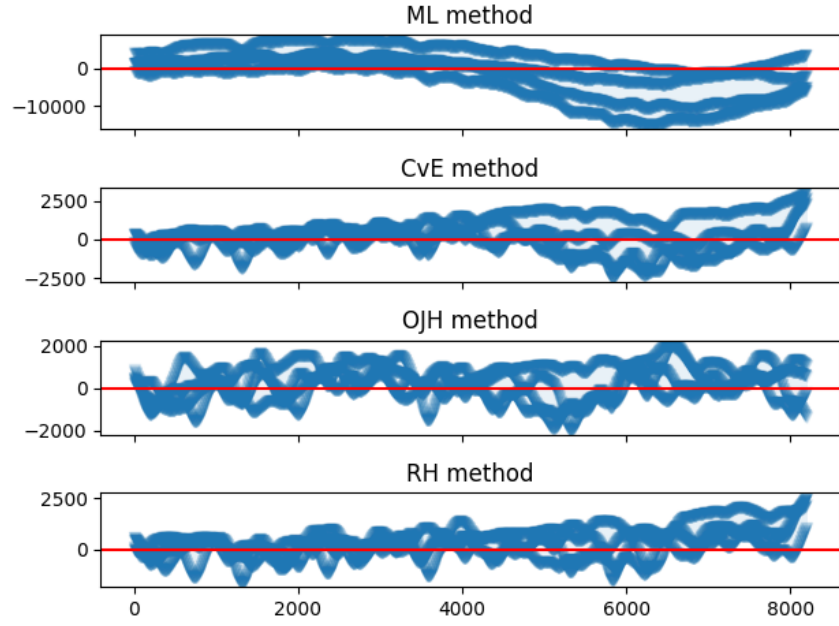
Figure 9: Every 562nd residual for the four different background estimators applied to a complex background. The y-axis is residuals as a percentage of the background level. Note that only the residuals in one direction are shown, as it takes every 562nd data point
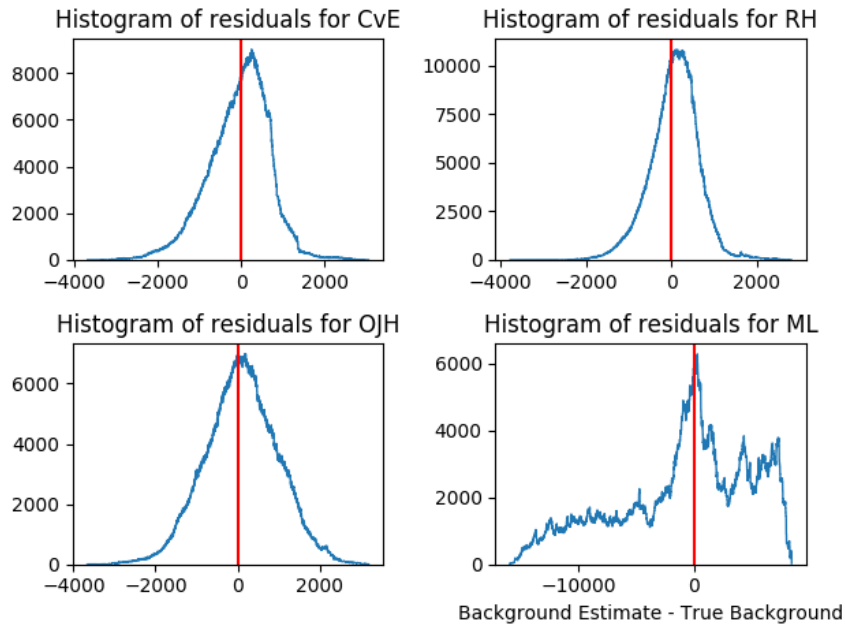
Figure 10: Histograms of the residuals for the four different background estimators applied to a white noise background.

3.2   ML

Initial tests were not able to fit to the data properly using a polynomial fit. Replacing the polynomial with a griddata interpolation function works significantly better, recovering the large scale structure across the image.

Changes made during this test:

- Stopped using a polynomial fit to the data, instead started using `scipy.interpolate.griddata`.

3.3 OJH

Initial tests were not able to fit to the data properly using a polynomial fit. Replacing the polynomial with a griddata interpolation function works significantly better, recovering the large scale structure across the image.

Changes made during this test:

- Stopped using a polynomial fit to the data, instead started using `scipy.interpolate.griddata`.

- Added extra measurment boxes of half size along the edges of the FFI for interpolation purposes.

3.4 RH

This method works very well, with the median residual being within one standard deviation of 0. No changes to this method have been made on this iteration.

3.5 Conclusions

Following the changes to the OJH method, three out of four methods converge around median residuals within one standard deviation of 0. The ML method is the bottleneck in the testing at the moment, and takes significantly longer than other estimates, despite no clear improvement in performance over the others. As it still relies on the polynomial fitting, it falls significantly behind the others in its ability to match large scale structure.

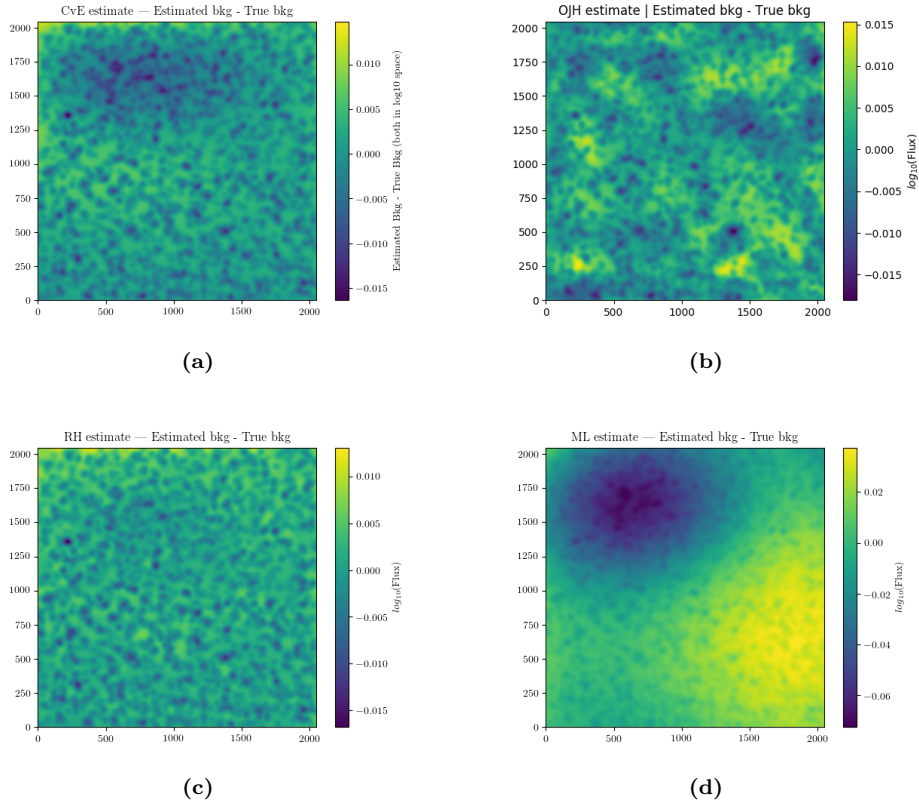All four backgrounds will progress to the next stage of testing.

(a)

(b)

(c)

(d)

Figure 11: Residuals on background fits to a complex background.

10

4 Test 3: The above + Stars

4.1 CvE

4.2 ML

4.3 OJH

4.4 RH

5 Conclusions