specBaselineEstimator()
&
specFlatFieldRange2()

Roland Vavrek (HSC)

23 April 2013

Outline

- Architecture of new long-range flat-field (FF) task
- The fitter engine: **`specBaselineEstmator()`**
- Optimization of FF algorithm on a chopped dataset, performance numbers for **`specFlatFieldRange2()`**

Limitations on the current FF task

- **Polynomial (5th order) model does not fit well to broad continuum coverage, especially for full SED scans**. This implies some limitations on method, particularly for ranges with strong gradients (such as full R1 scan with 2nd order leak beyond ~190 micron)
- The continuum is estimated as the median spectrum of individual sub-populations of the L1 dot-cloud. **The median flux is calculated in one step, i.e. for all populations, all segments and all pixels**. This way RSRF residuals – which are attributes of a given pixel – are smeared out in the flat-fielded product. RSRF inversion from data is not possible on this product.

Motivation for a new FF task

- **Find more adaptative model than polynomial fit**, this is particularly important for unchopped mode where response drifts modulate the signal at various frequencies in individual spectral segments
- **Split up the FF for two steps**:
  - 1st step: apply response correction between segments (and populations) for a given pixel
  - 2nd step: apply FF between the 16 pixels
- Take advantage of two-step approach to **extract and characterize (and possibly correct for) RSRF residuals**, this could be done after completion of step 1.

L1 sliced cubes

FF flow chart

Step1: flat between segments per pixel

- apply low-pass filter per segment (highest freq. cutoff @ 3-4 data points)
- divide segments with their filter function
- multiply each segment with median of filter functions

Correct for RSRF residuals

Optional

- apply low-pass filter per pixel (high freq. cutoff @ ~0.5 micron)
- divide pixel data with their filter function*
- subtract normalized RSRF residual product**+ 1
- multiply each pixel with its filter function

Step2: flat between 16 pixels in a module

- apply low-pass filter*** per pixel (high freq. cutoff @ ~0.5 micron)
- divide pixel data with their filter function
- multiply each pixel with median of filter functions

L1 sliced cubes

* point to extract data for the purpose of normalized RSRF residual product generation

** residual product is being derived using the same cut-off frequency as the task does on the actual observation

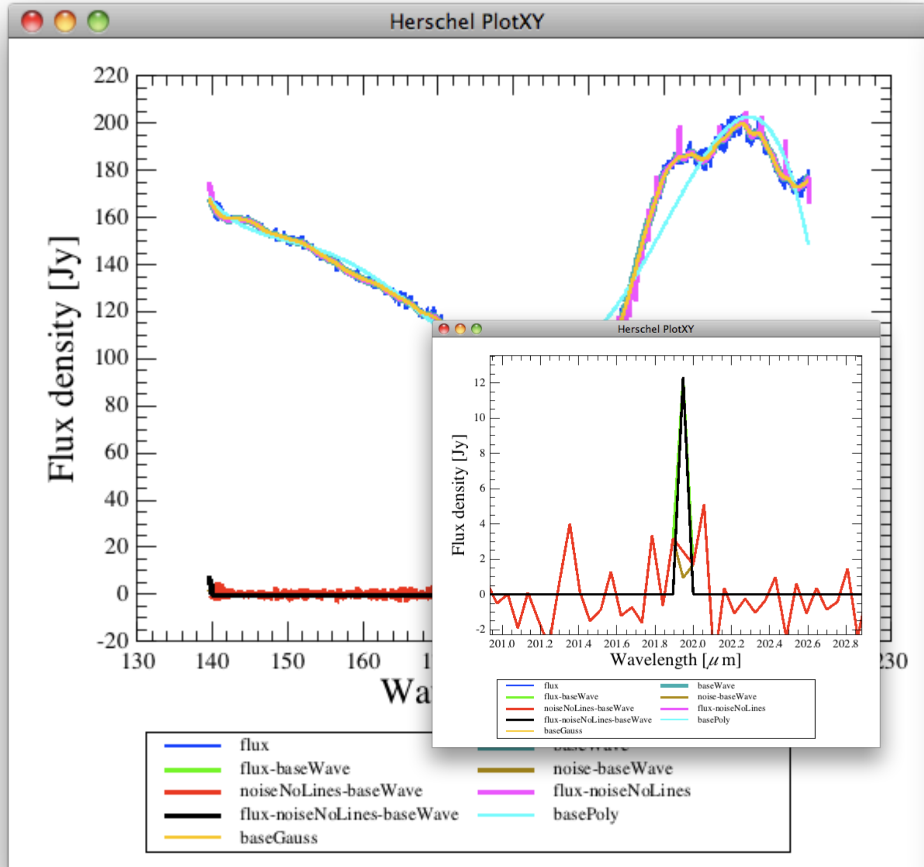
*** low-pass filter in step 2 may not be identical to the filter applied in RSRF correction

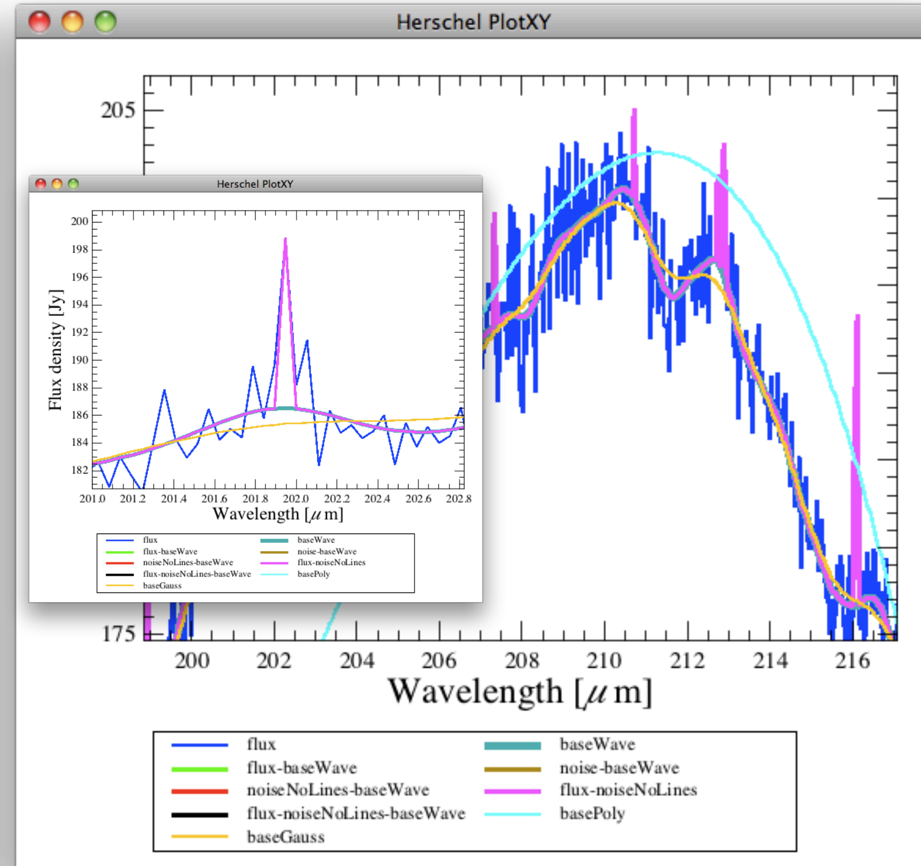
FF2 in the pipeline

- Runs on **SlicedPacsCubes**
- Continuum modulation due to pointing jitter should not be 'corrected' by FF on the physical spaxel!
- Taking advantage of improved pointing reconstruction:
  - **If point source:** first correct flux against pointing jitter (reconstruct point source)
  - **If oversampled raster:** a new SlicedPacsCubeSampled product could be necessary: dot-cloud at projected pixel resolution with WCS. FF is three-stage per projected pixel:
    - Per detector pixel
    - Per detector module
    - Per projected pixel

The multiresolution wavelet engine for new flat-field: specBaselineEstimator()

Fitter engine: baseline estimator tool

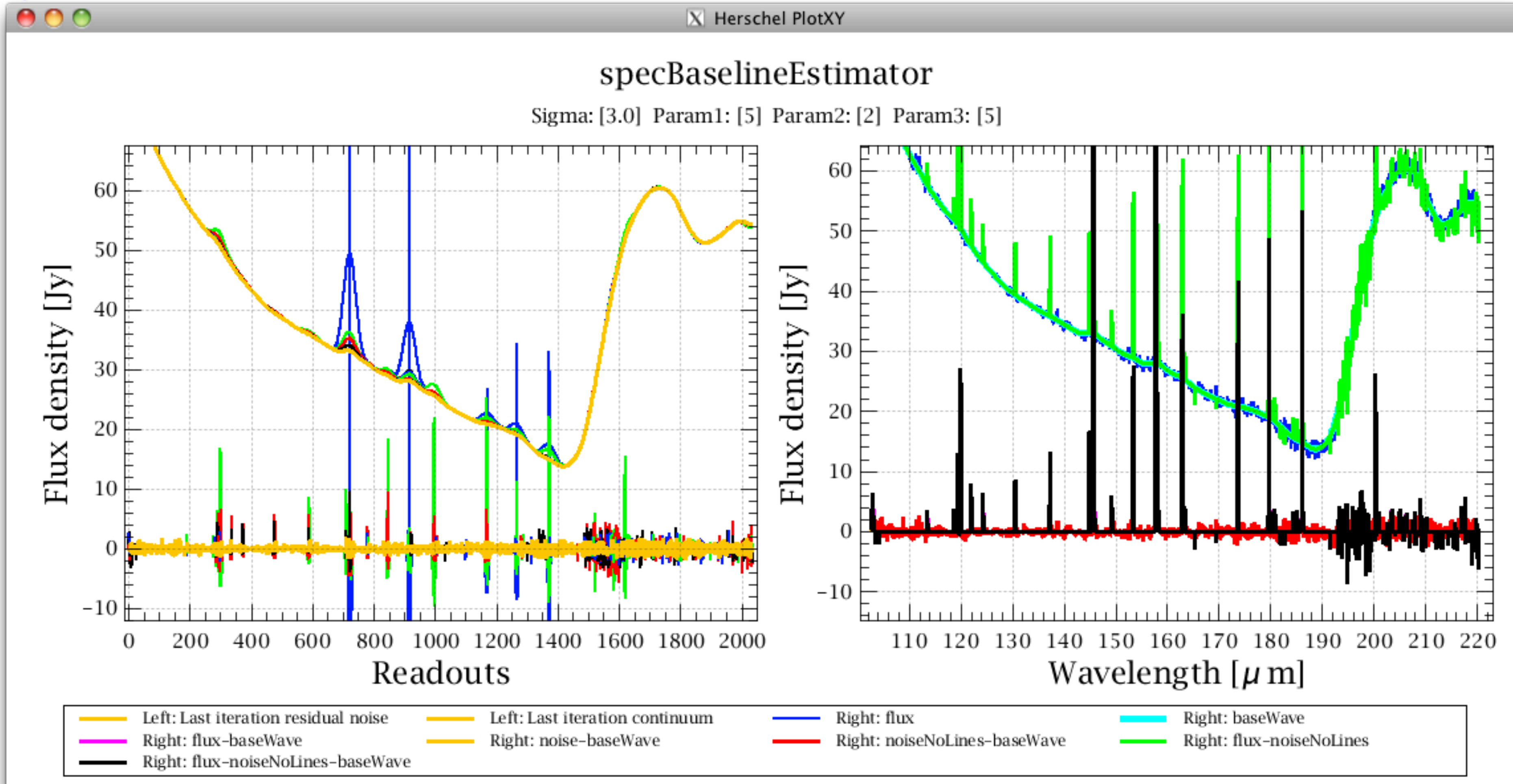




Blue: rebinned data (scale invariant resolution)
Magenta: baseline + spectral lines
Red: noise cube
Black: outliers (lines) cube

Neptune, chopped R1 scan

Fitter engine: baseline estimator tool



Blue: rebinned data (scale invariant resolution)

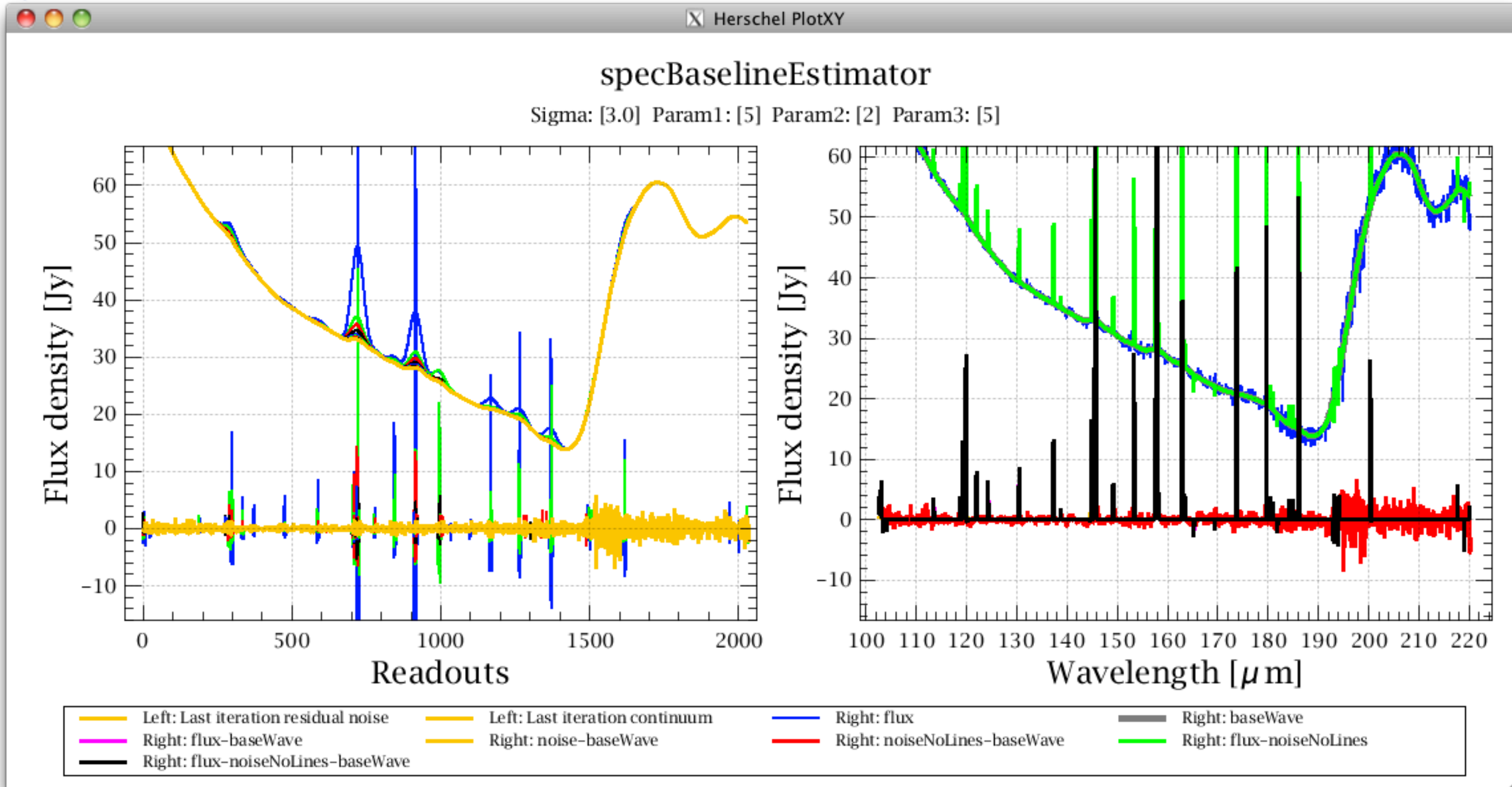
Green: baseline + spectral lines

Red: noise cube

Black: outliers (lines) cube

NGC 6543, chopped R1 scan

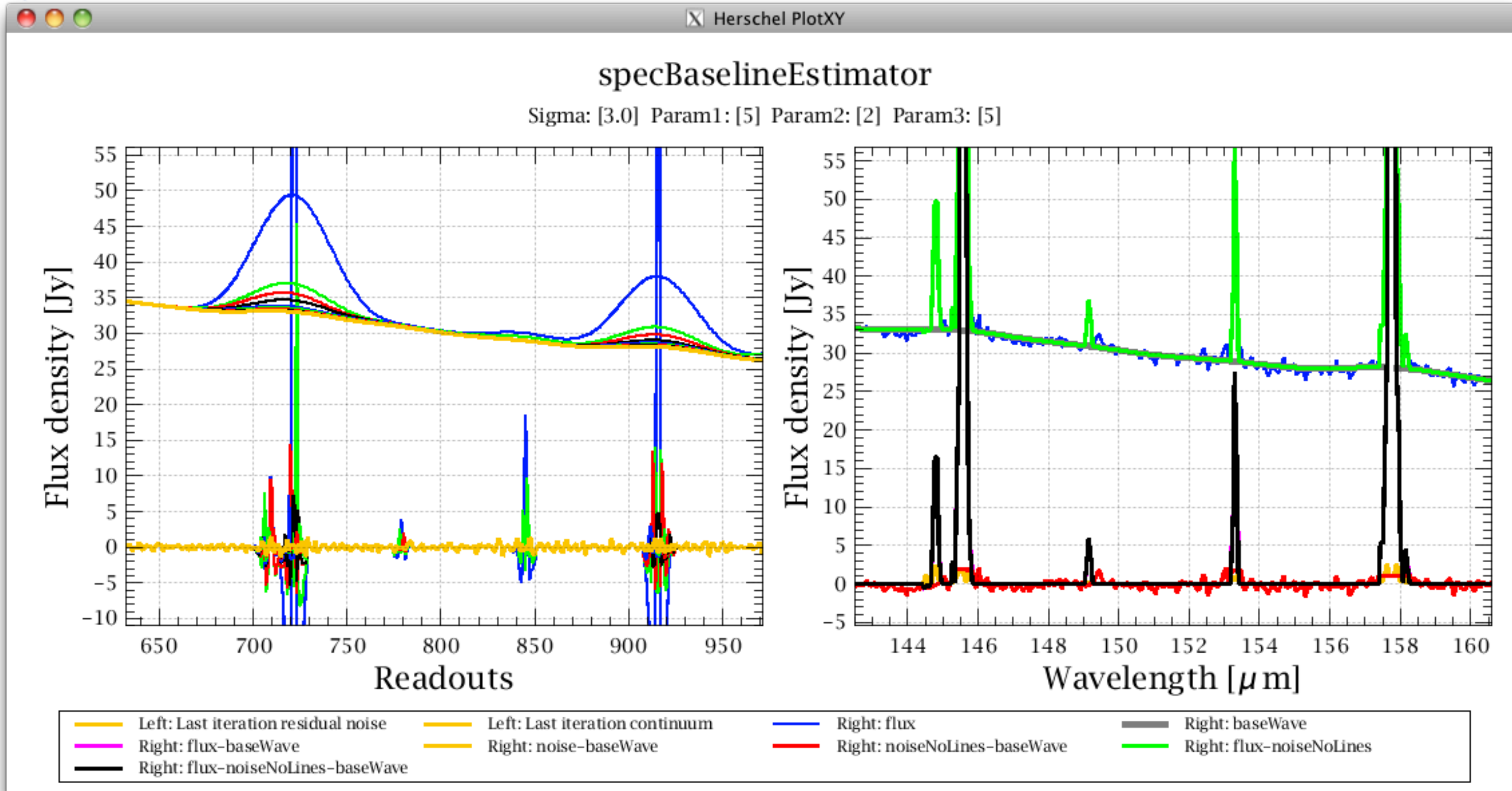
Fitter engine: baseline estimator tool



Blue: rebinned data (scale invariant resolution)
Green: baseline + spectral lines
Red: noise cube
Black: outliers (lines) cube

NGC 6543, chopped R1 scan, adaptive noise filter

Fitter engine: baseline estimator tool



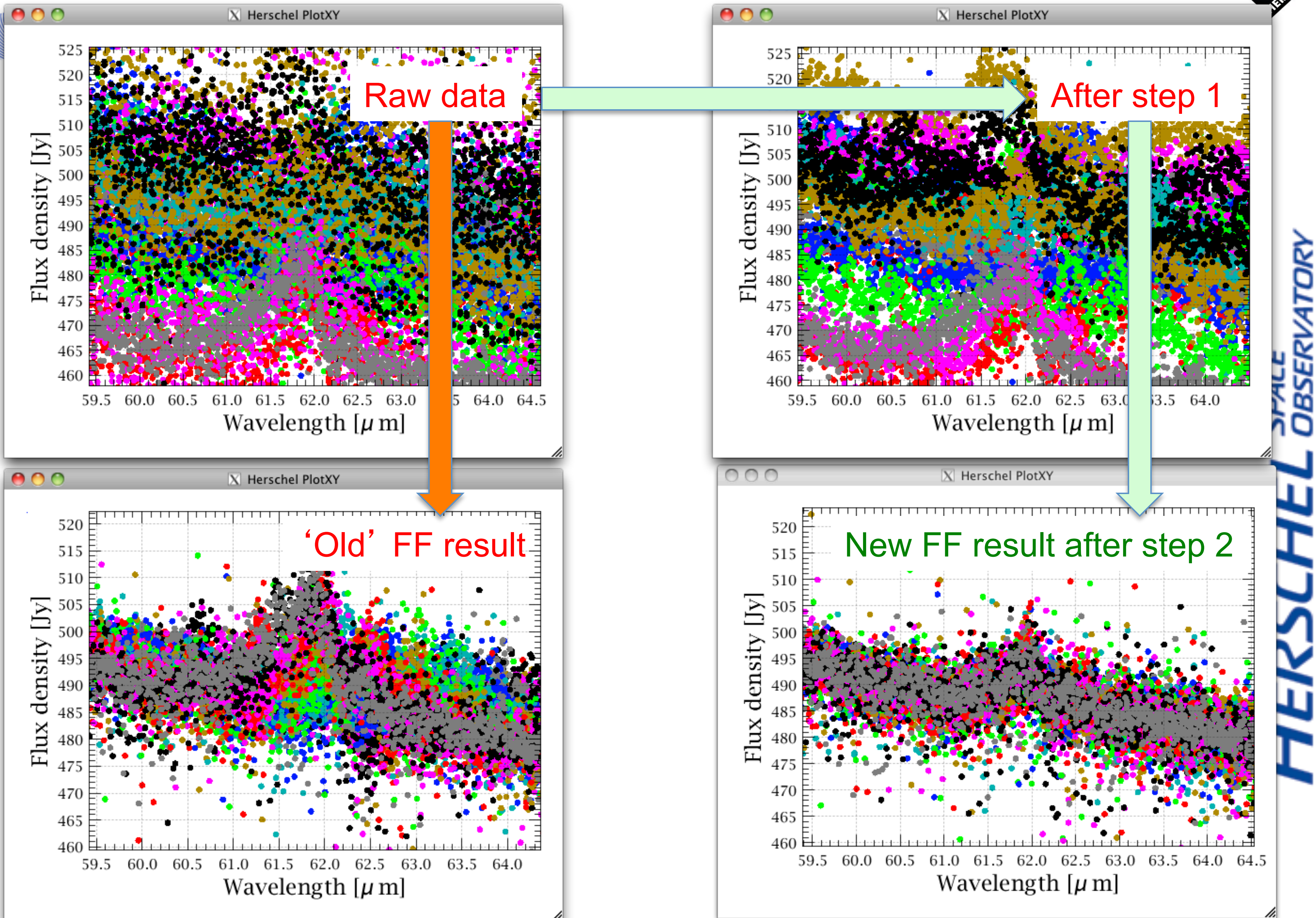
Blue: rebinned data (scale invariant resolution)
Green: baseline + spectral lines
Red: noise cube
Black: outliers (lines) cube

NGC 6543, chopped R1 scan, adaptive noise filter

specFlatFieldRange2()

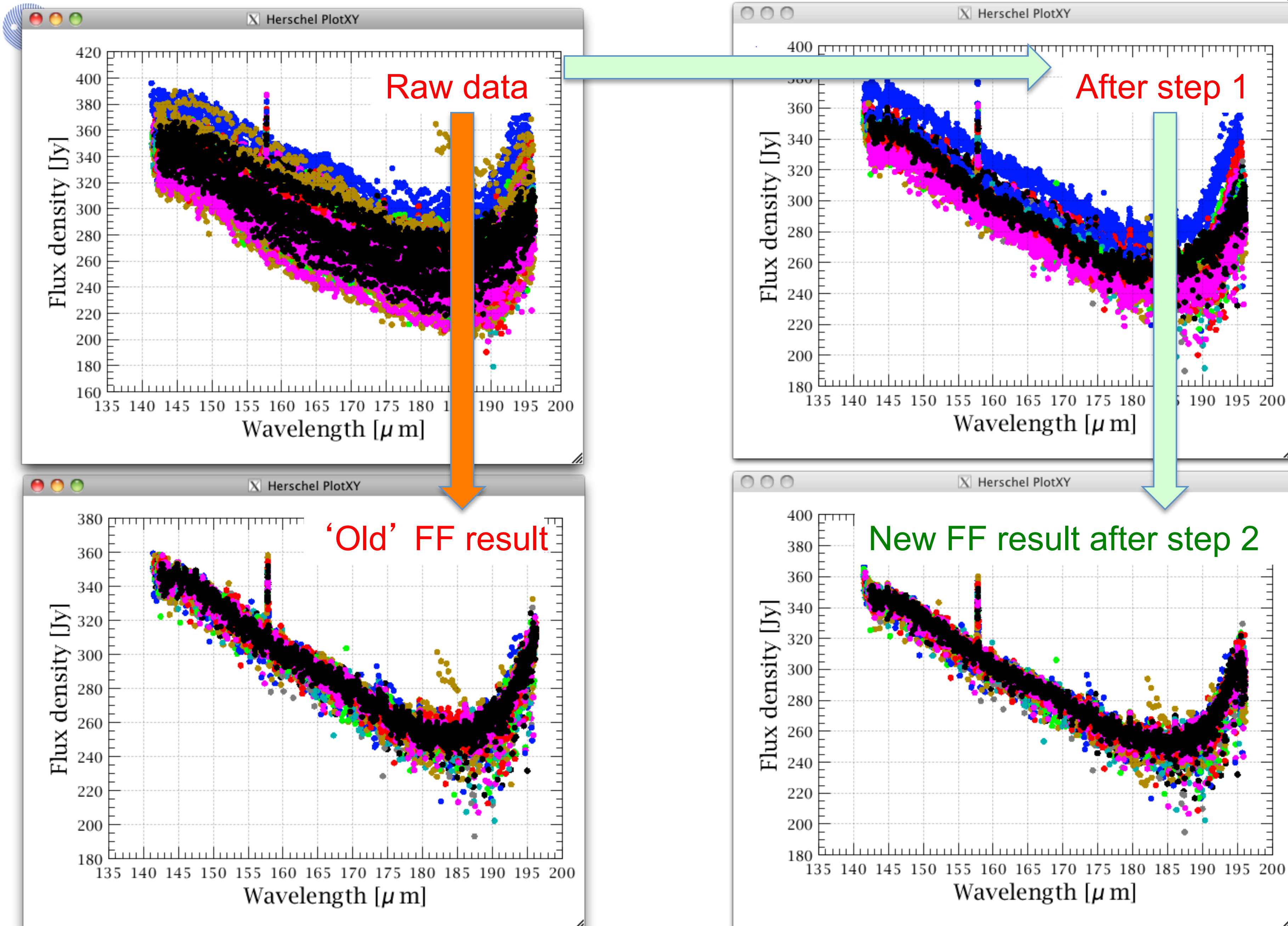
B3A, L1 PacsCube

OFF scan, colours for the 16 pixels, module 12



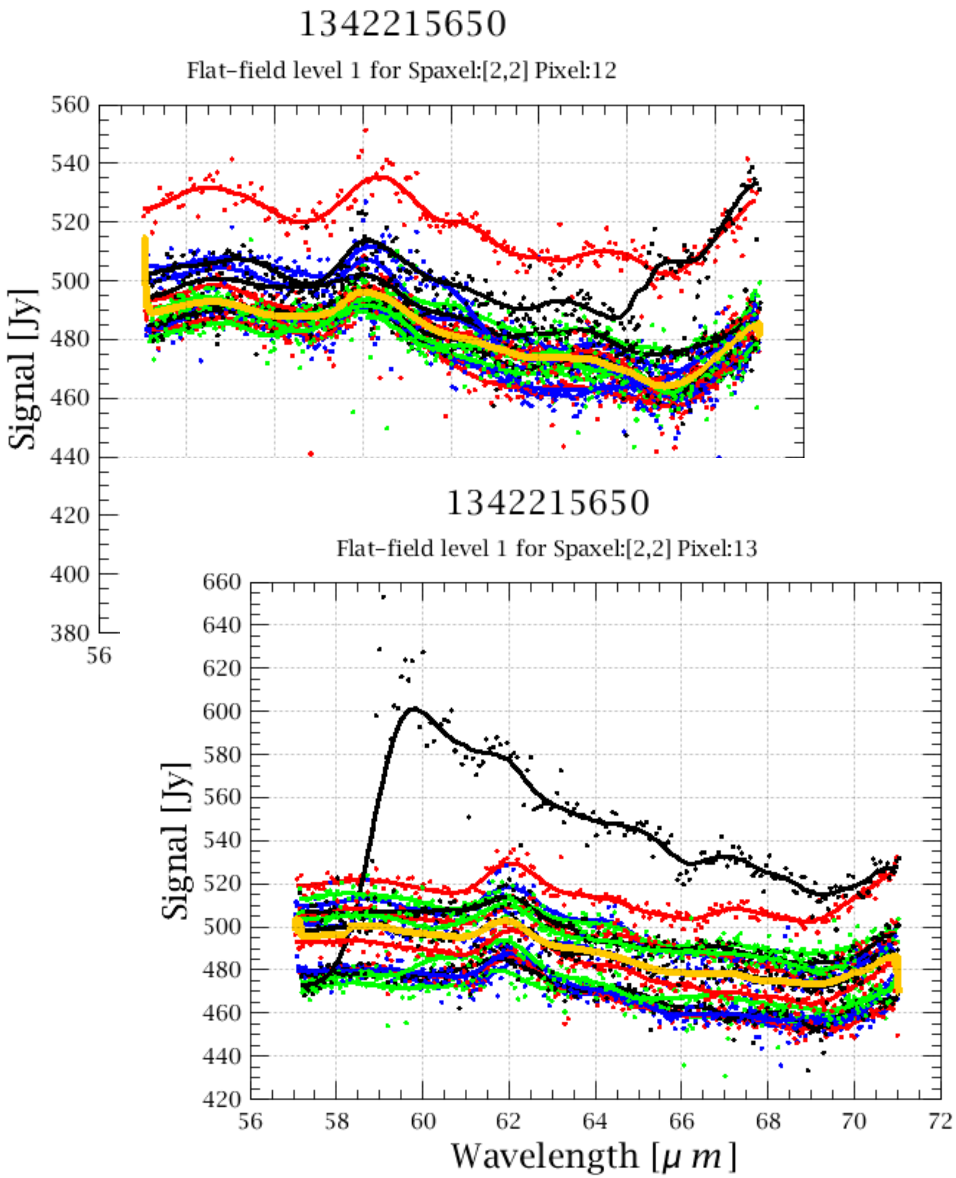
R1, L1 PacsCube

OFF scan, colours for the 16 pixels, module 12

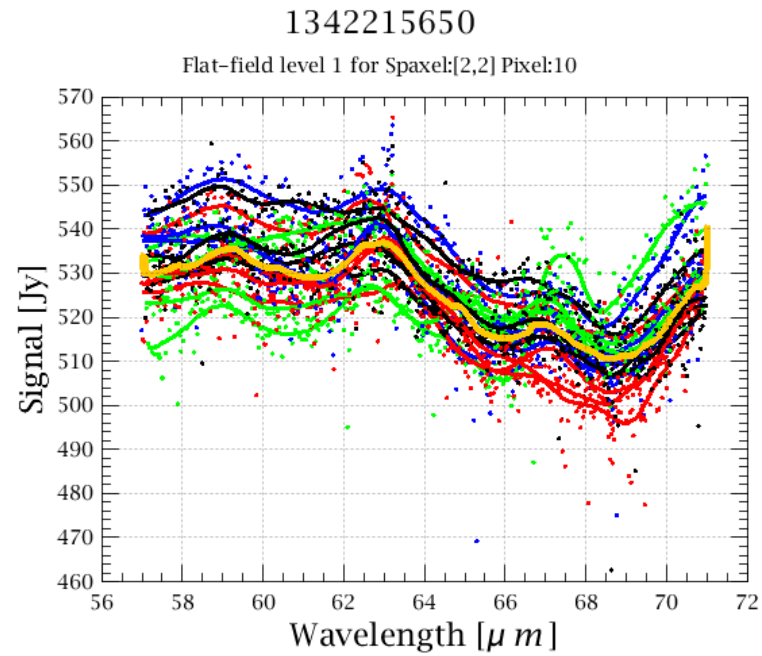


B3A, L1 PacsCube

OFF scan, colours for spectral segments, module 12, single pixels, unchopped mode



Flat-field: level-1 segment fitting (1 up- or down scan)

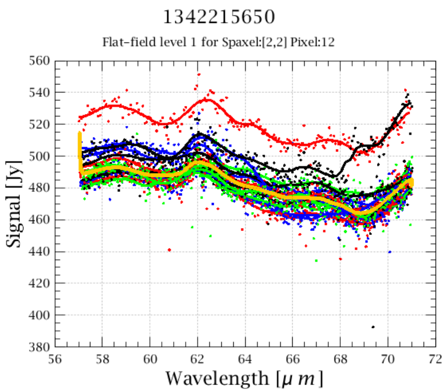


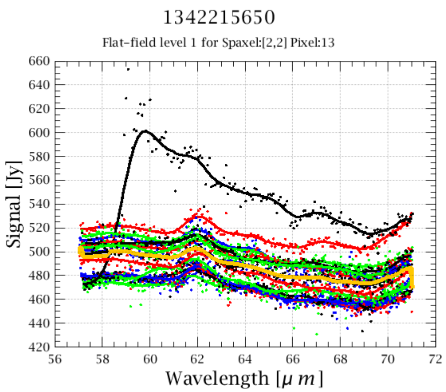
Wavelet cutoff frequency: ~0.5 µm

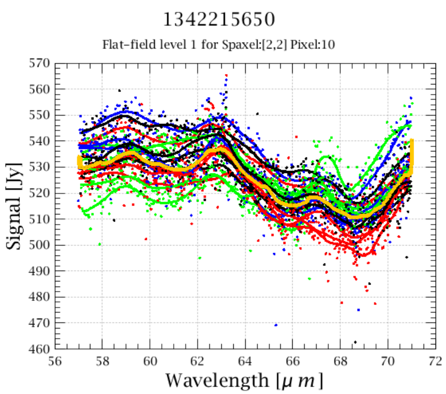
Drift correction prior this step (see Dario's presentation) may improve broad-feature detectability)

B3A, L1 PacsCube

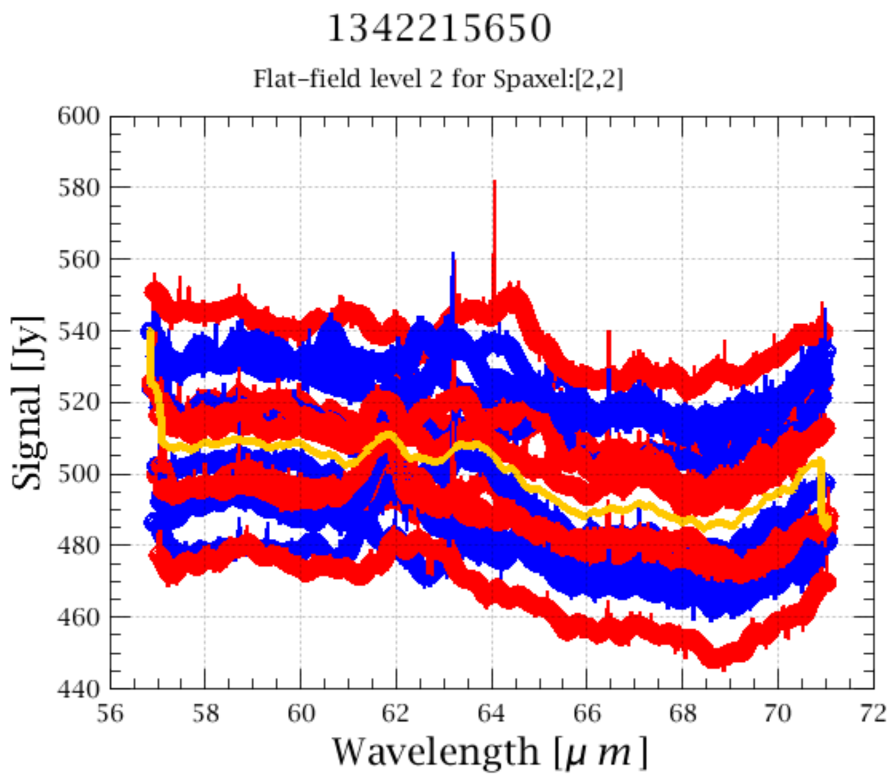
OFF scan, colours for spectral segments, module 12, 16 pixels, unchopped mode





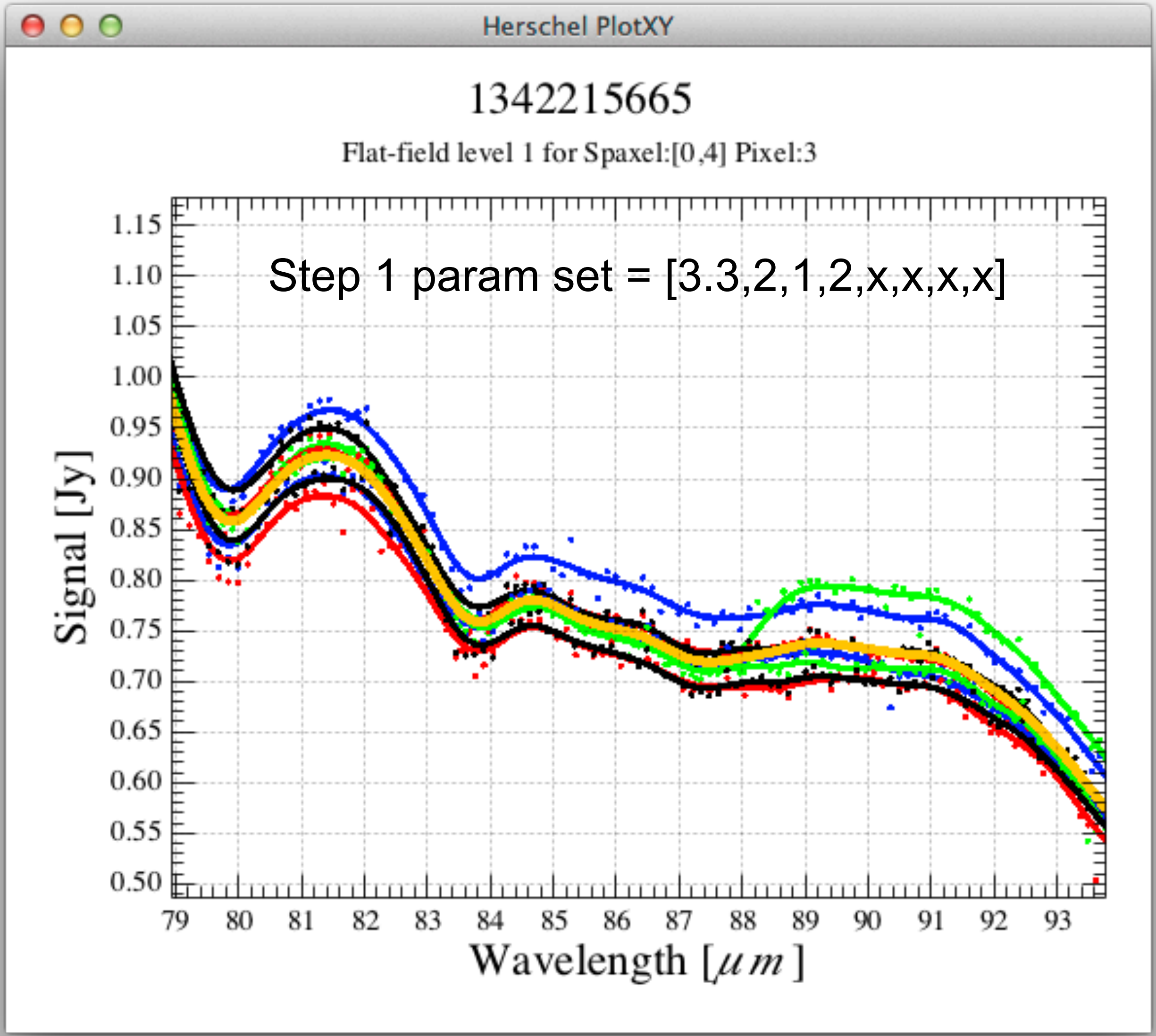


Flat-field: level-2 pixel fitting

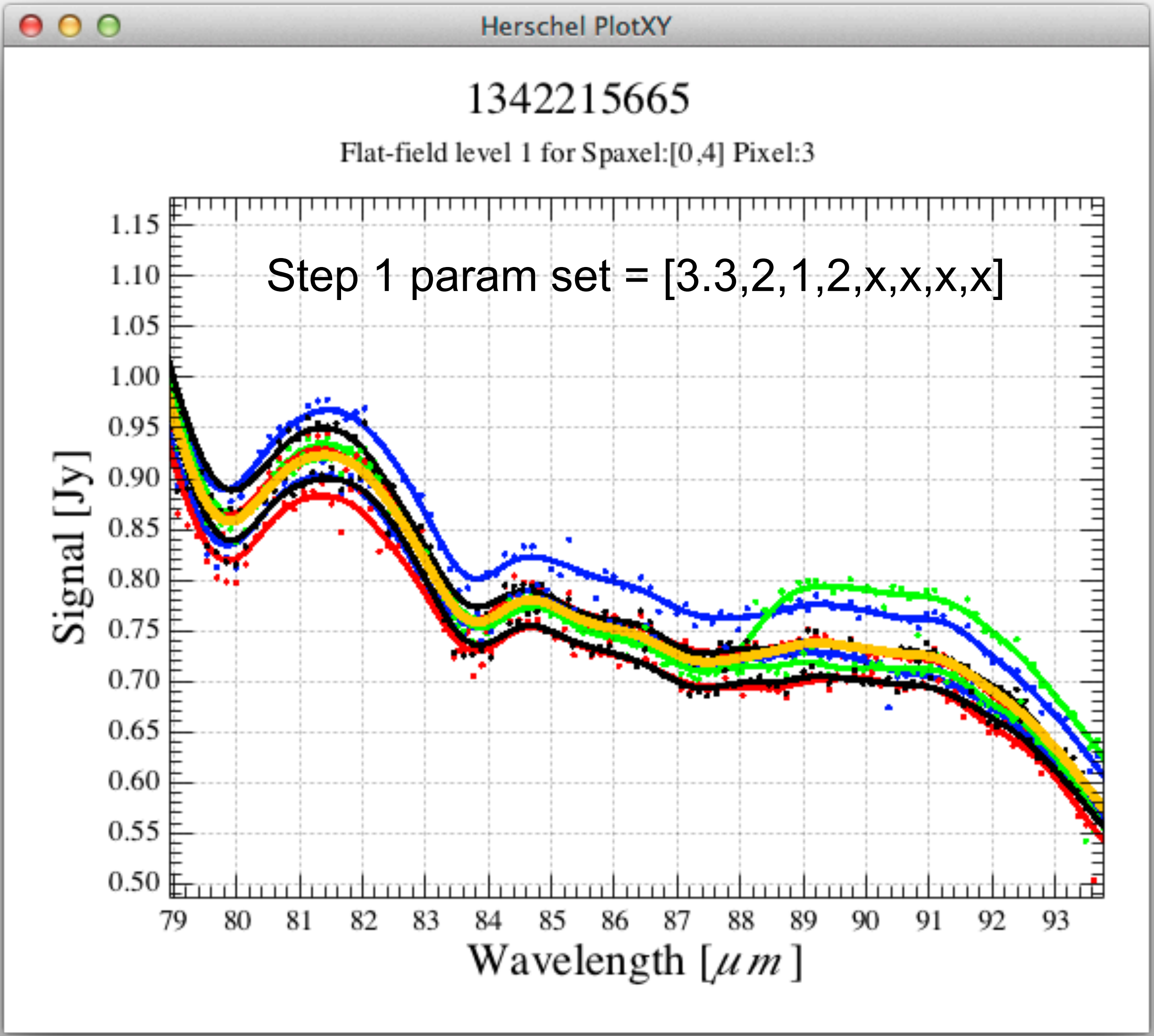


Orange is the module's continuum estimate (currently median or mean but it could be improved (e.g. estimate the mode of the distribution)

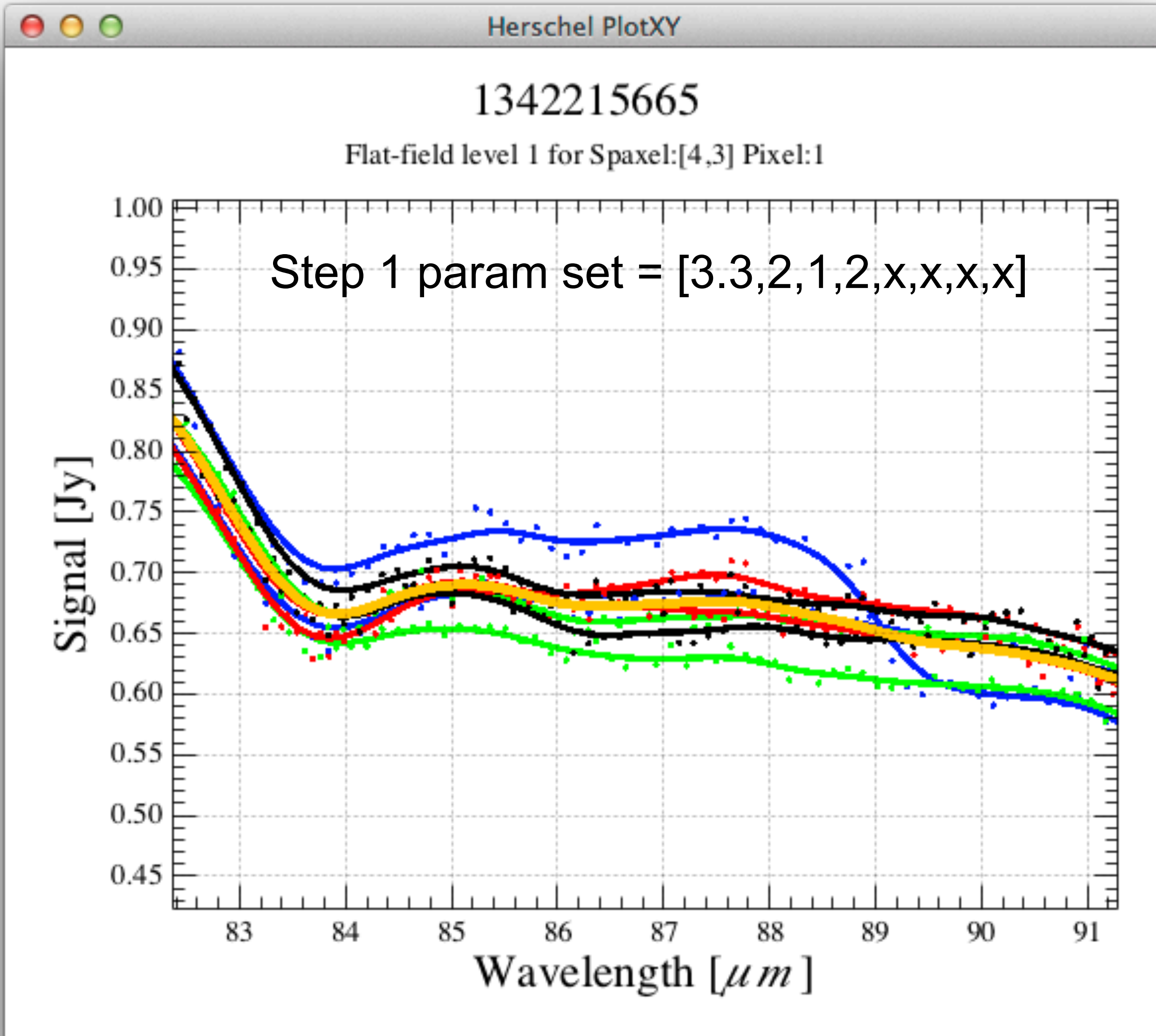
FF2 – stage 1 examples (FF per pixel)



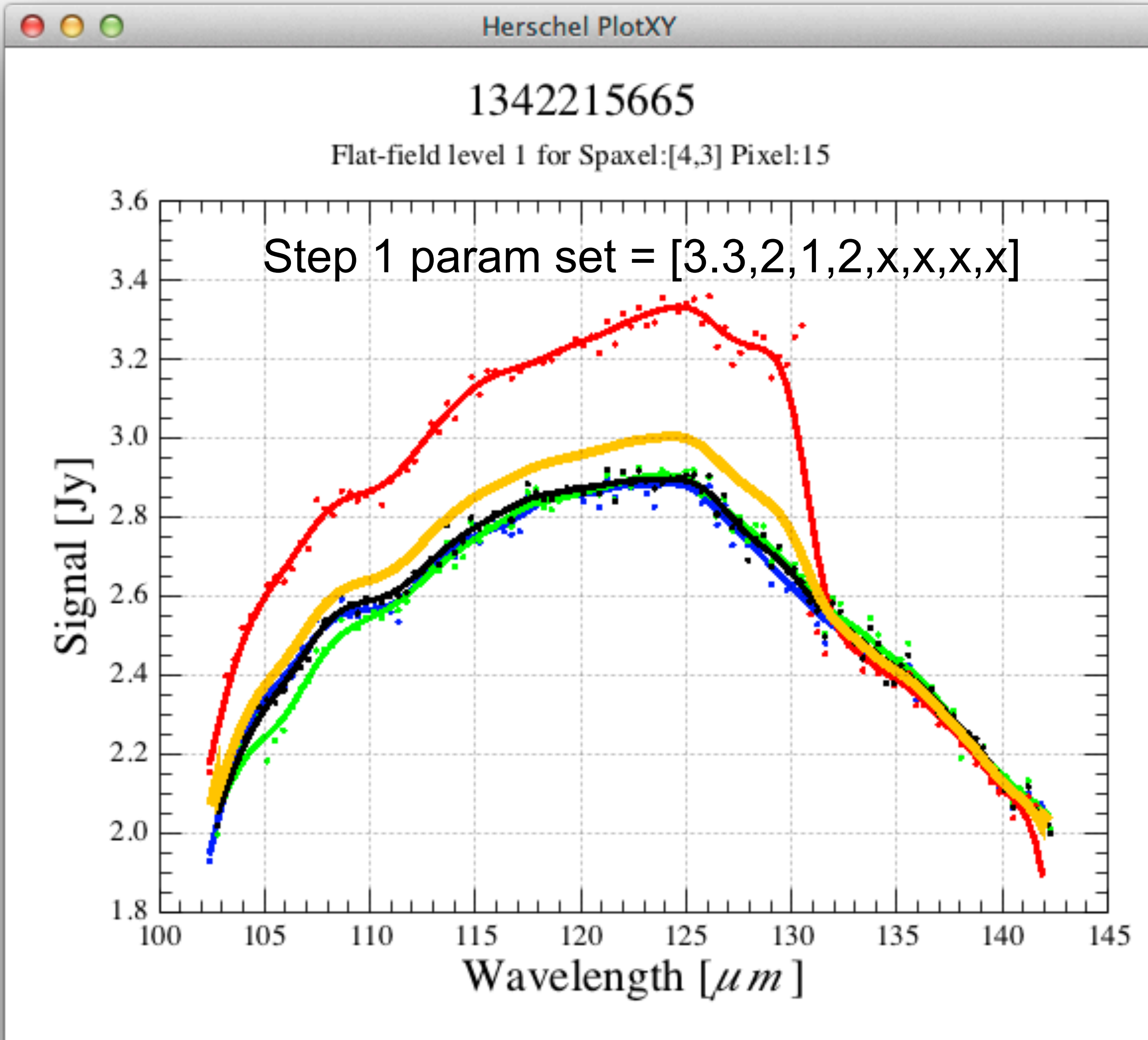
FF2 – stage 1 examples (FF per pixel)



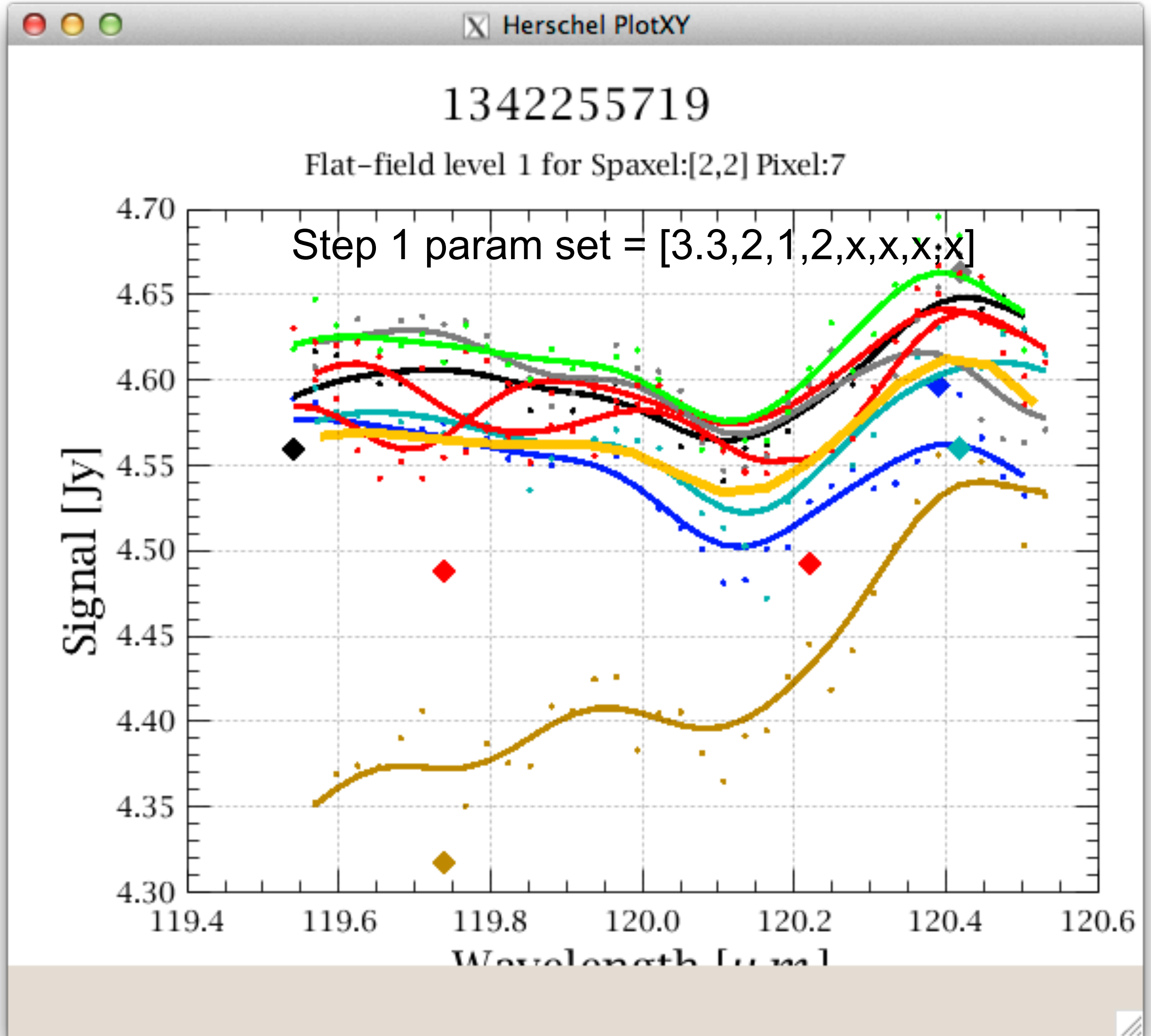
FF2 – stage 1 examples (FF per pixel)



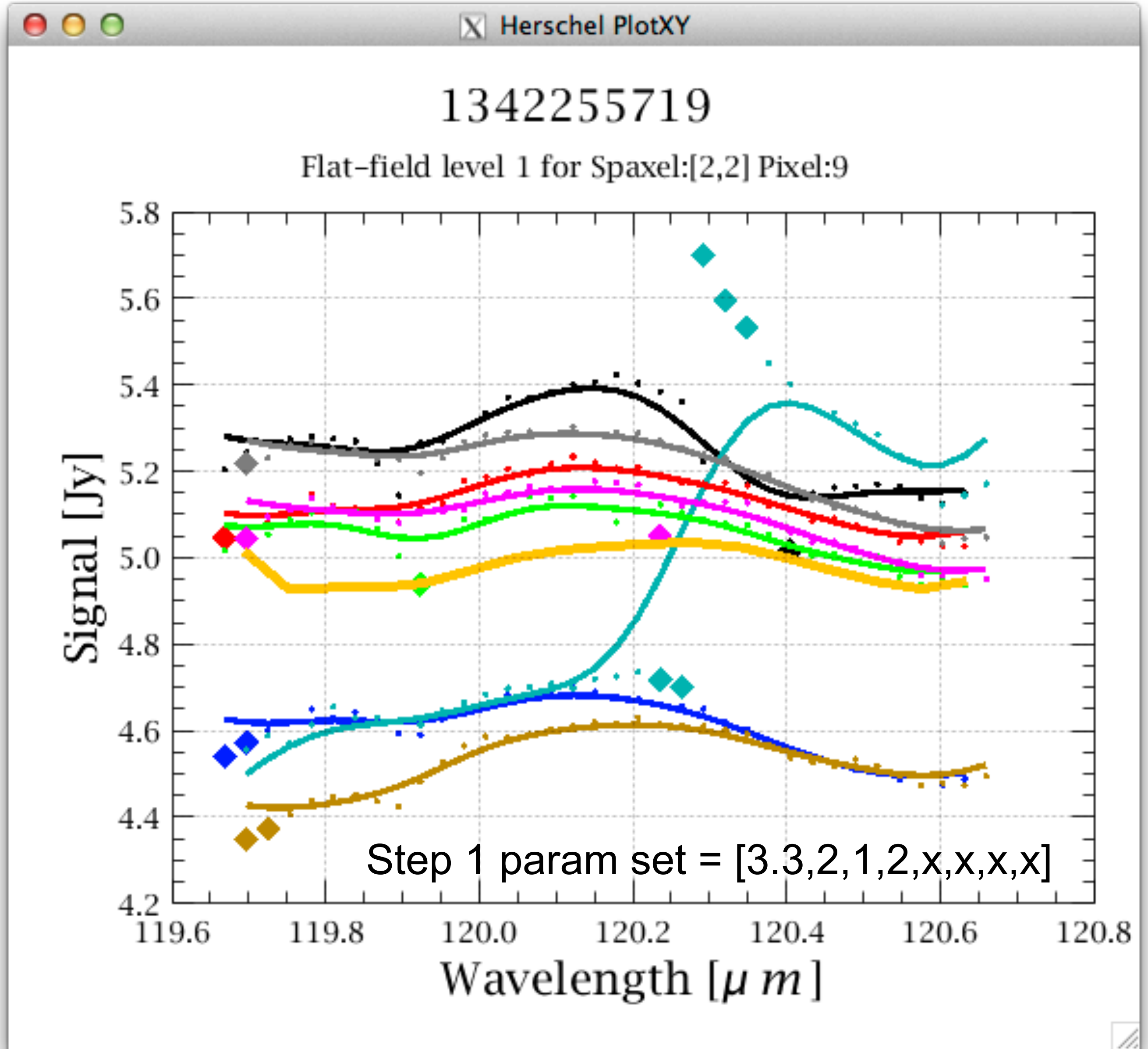
FF2 – stage 1 examples (FF per pixel)



FF2 – stage 1 examples (FF per pixel)

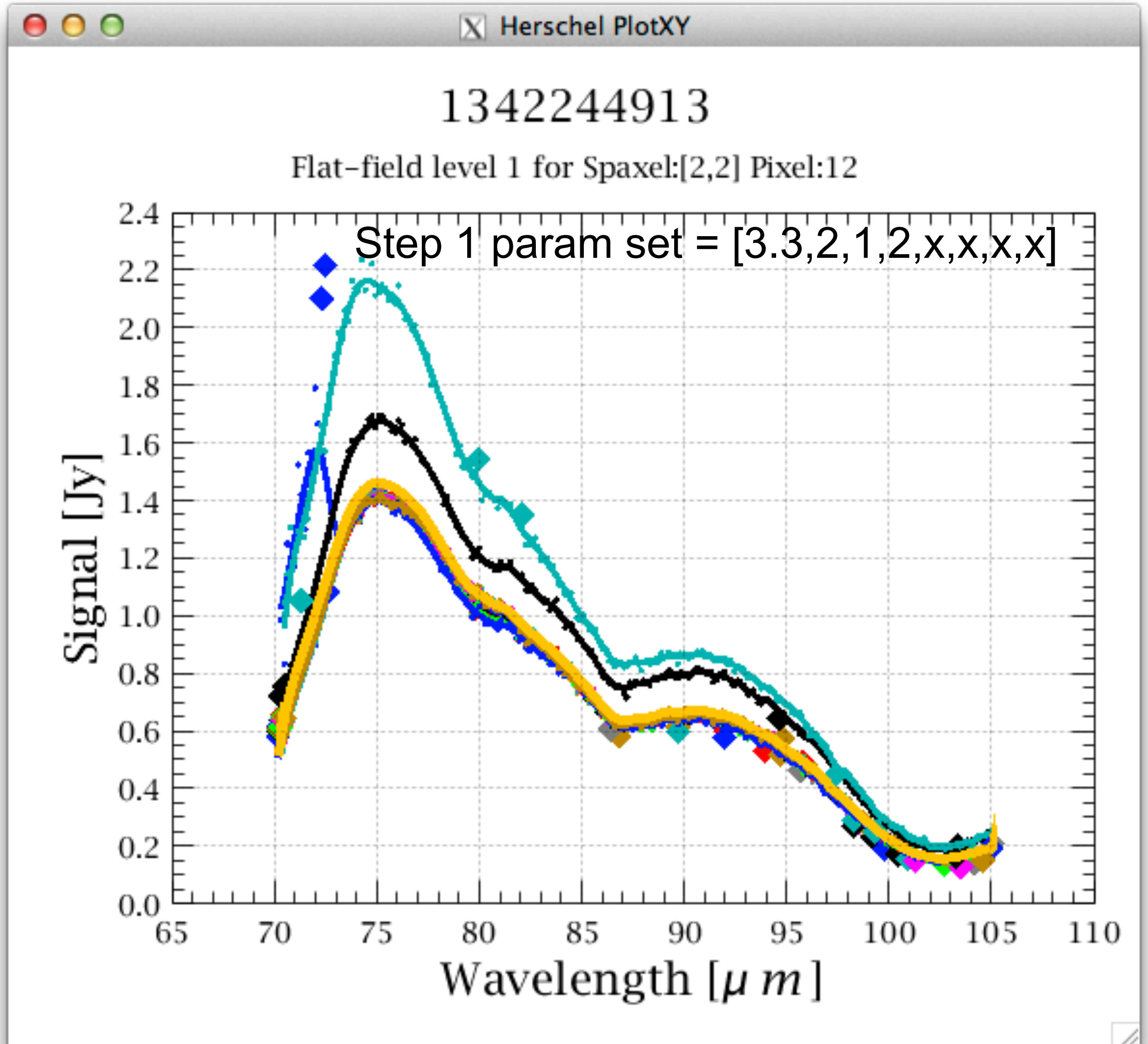


FF2 – stage 1 examples (FF per pixel)

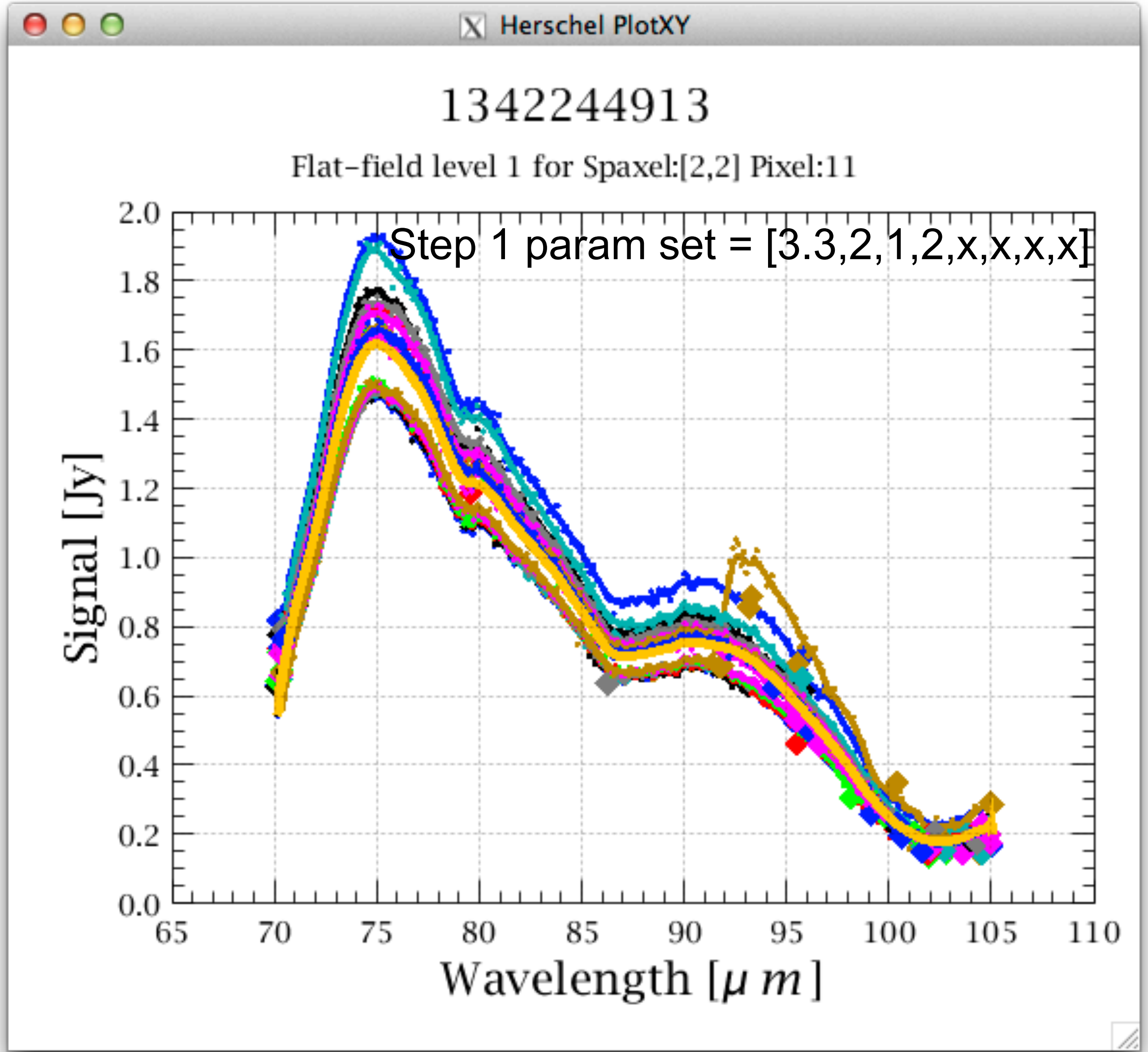


HERSCHEL SPACE OBSERVATORY

FF2 – stage 1 examples (FF per pixel)

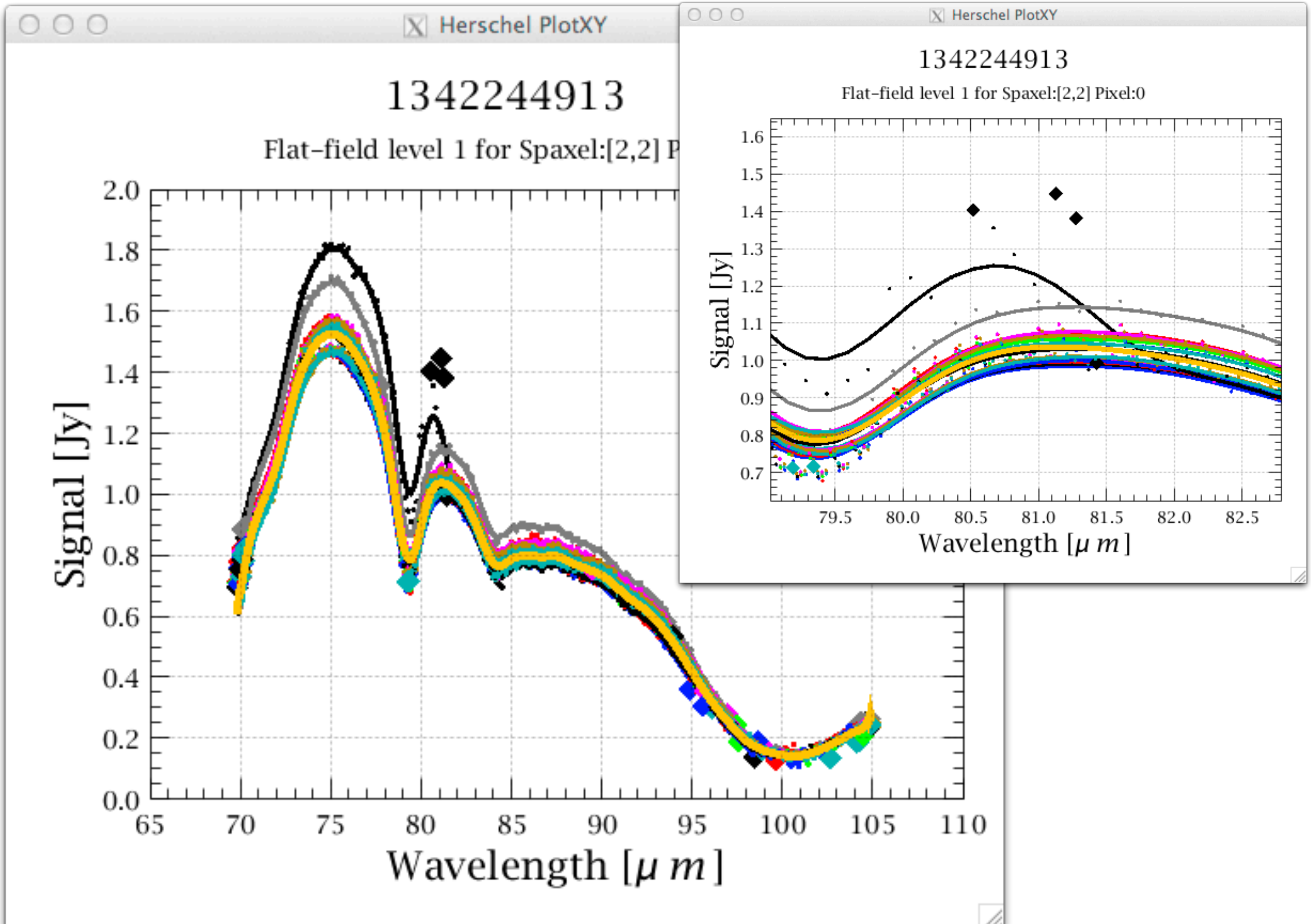


FF2 – stage 1 examples (FF per pixel)



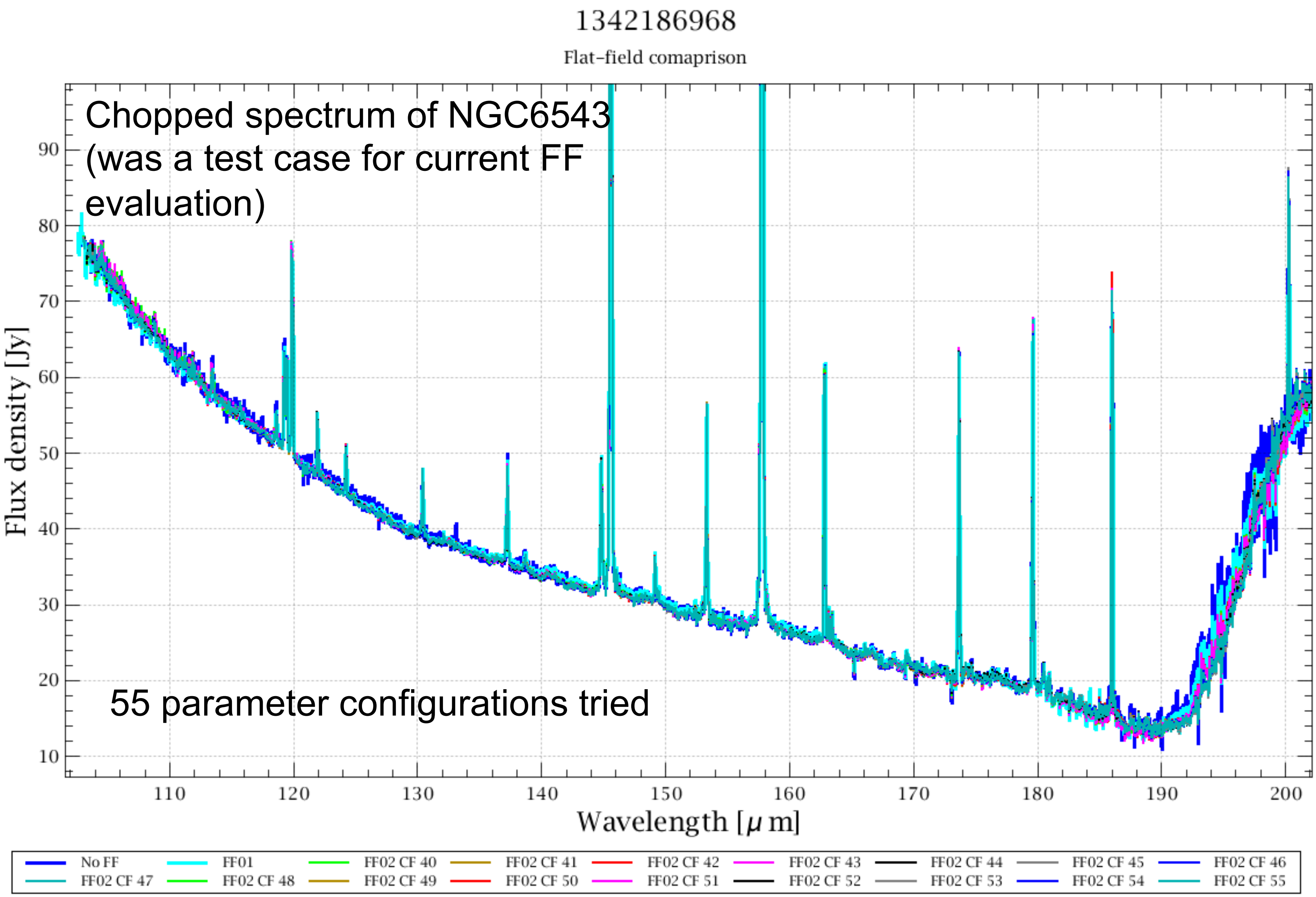
FF2 – stage 1 examples (FF per pixel)

Step 1 param set = [3.3,2,1,2,x,x,x,x]



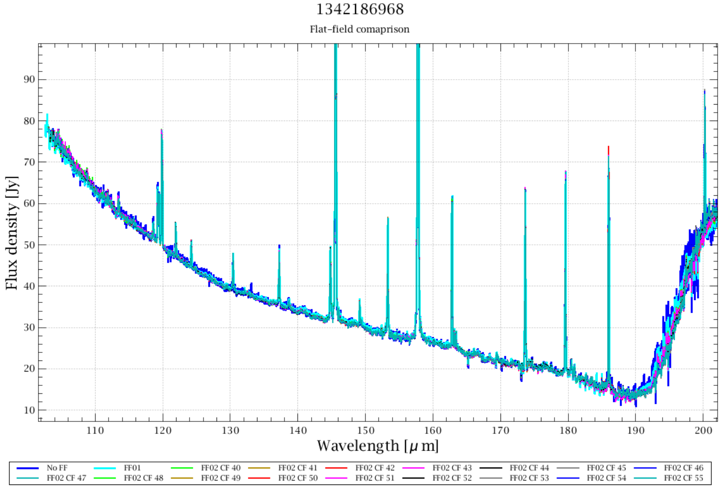
HERSCHEL SPACE OBSERVATORY

Flat-field parameter optimization

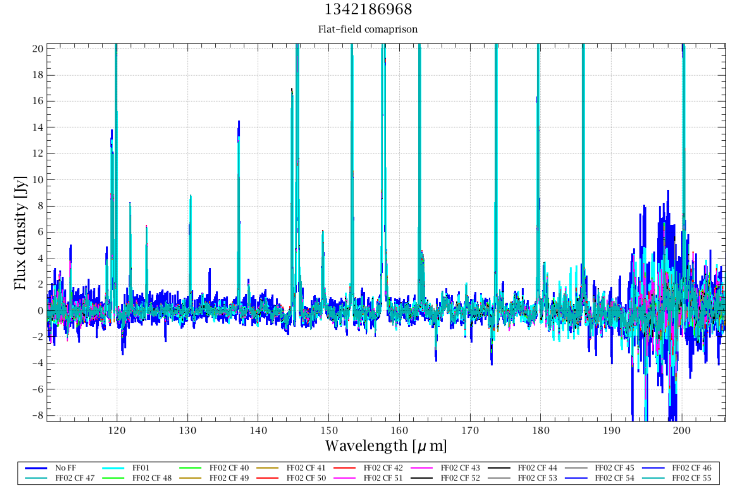


Flat-field parameter optimization

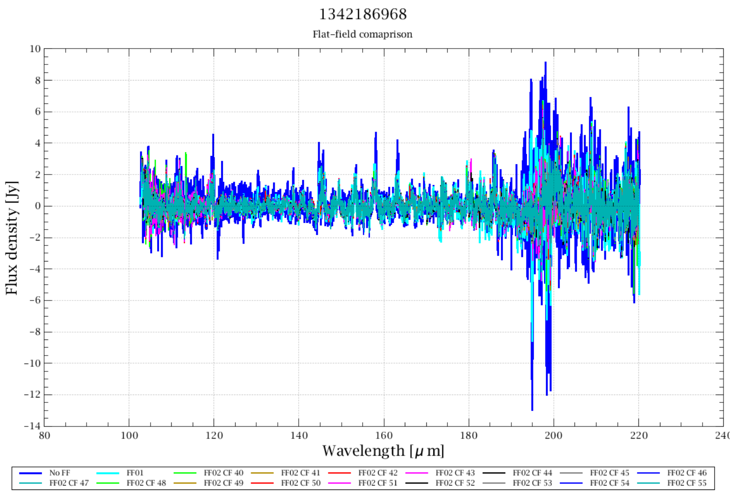
FF-ed spectrum



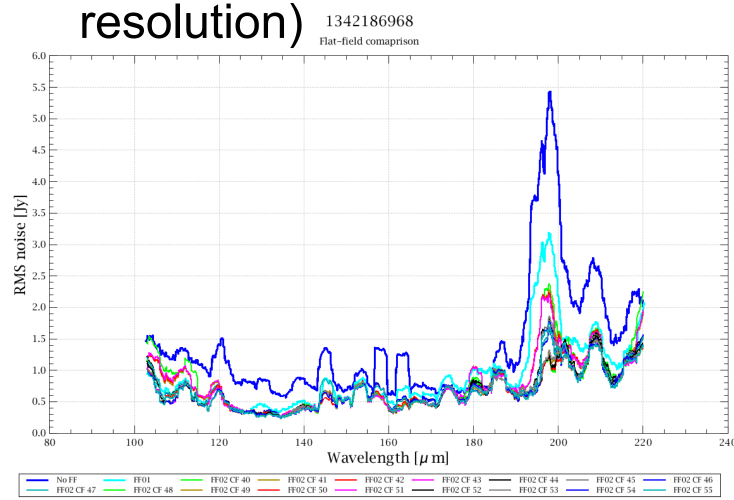
Subtract baseline



Create noise cube



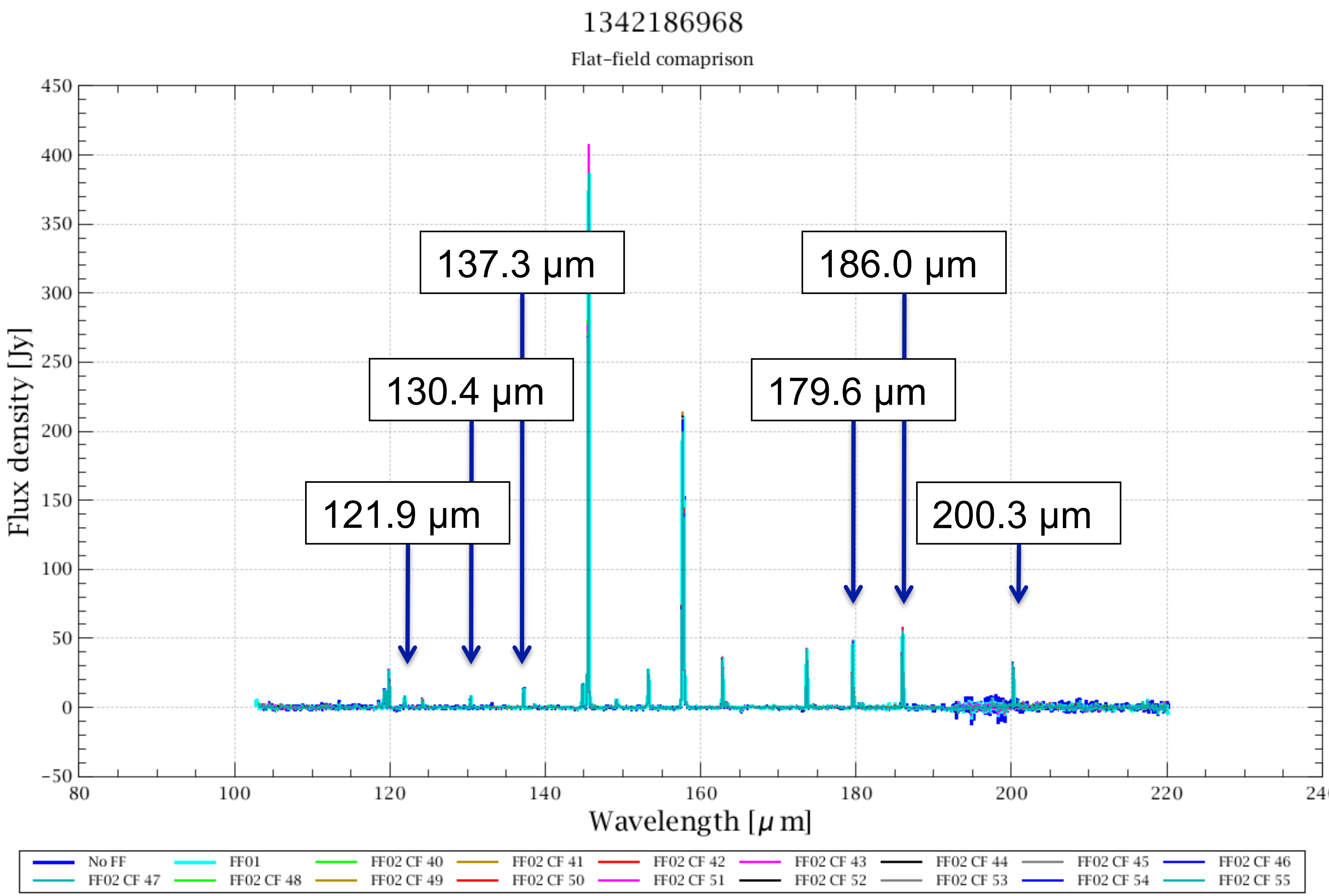
RMS noise function (3 microns resolution)



HERSCHEL SPACE OBSERVATORY

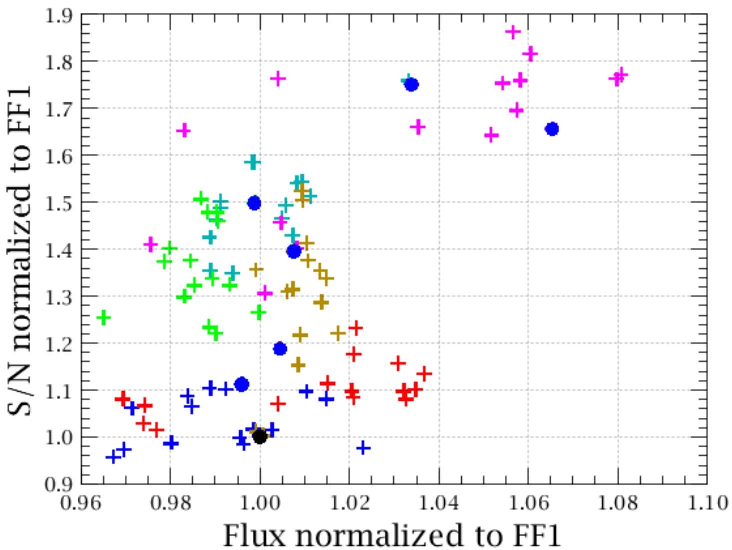
Flat-field parameter optimization

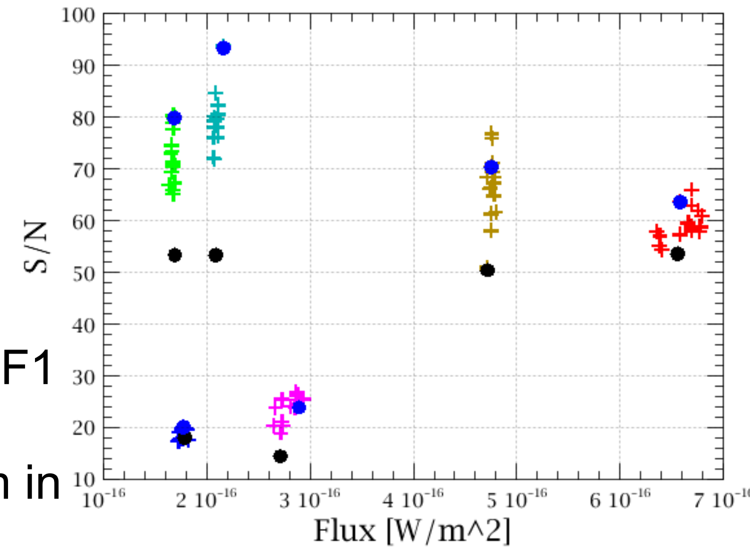
Lines selected for performance evaluation



HERSCHEL SPACE OBSERVATORY

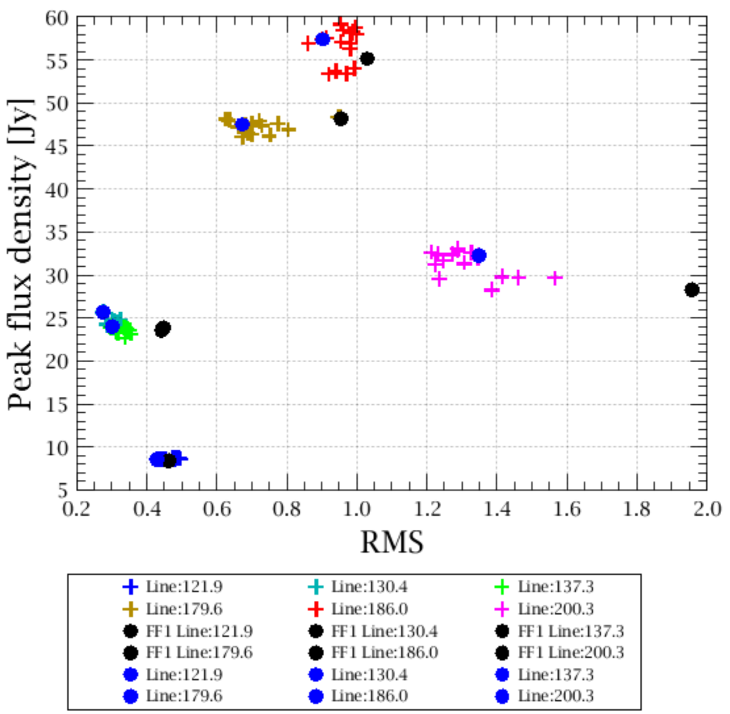
Flat-field parameter optimization

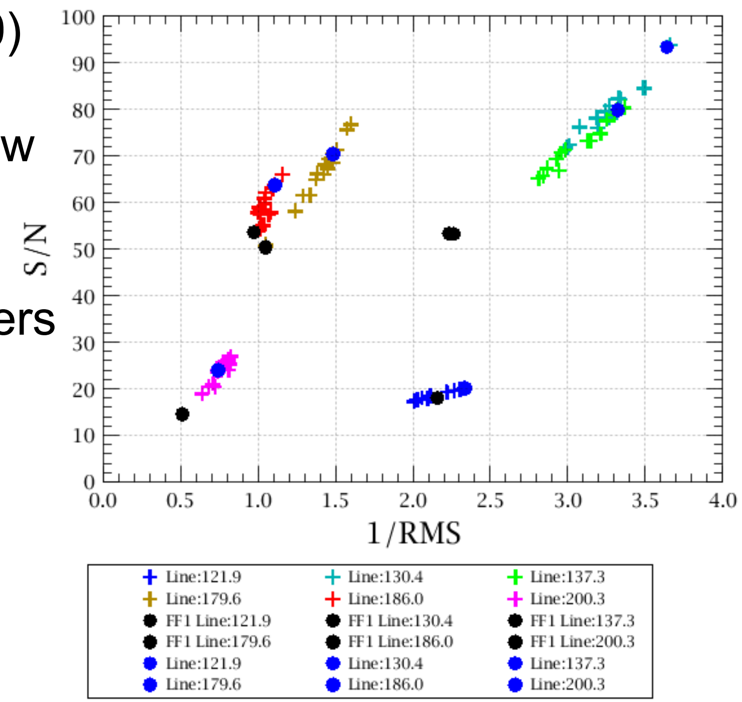




Black: FF1 (current algorithm in pipeline HIPE 8.0)

Blue: new FF2 with optimal parameters

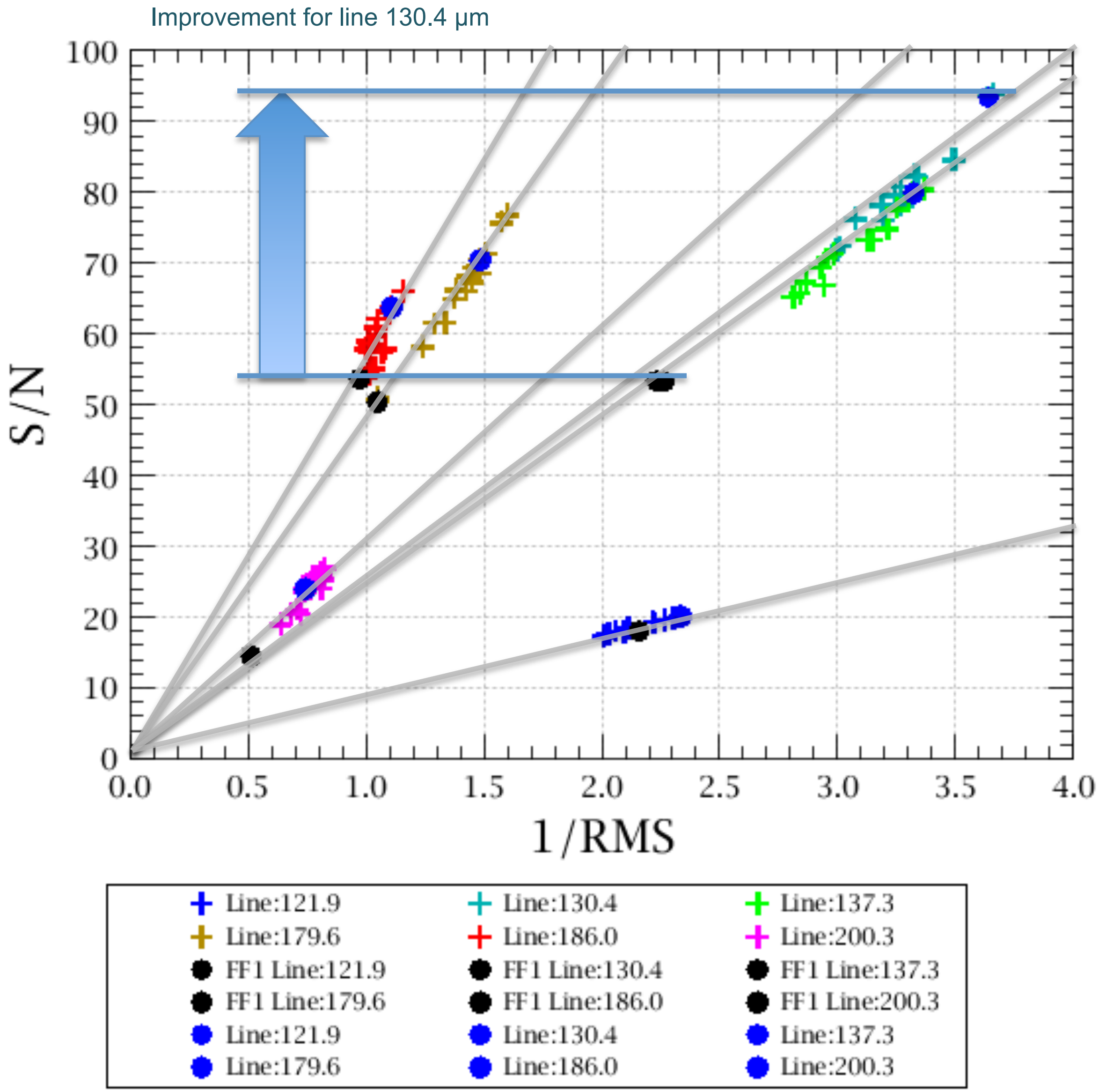




Flat-field parameter optimization

- SNR vs. 1/RMS noise is a good way of mapping FF efficiency
- RMS continuum noise only may be misleading if multiresolution filtering cuts at- or below frequencies corresponding to the line width
- The slope is the signal, ideally options follow a straight line (indicating multiresolution FF does not cut line peaks)
- Best options are further away from (0,0), blue is the adopted solution

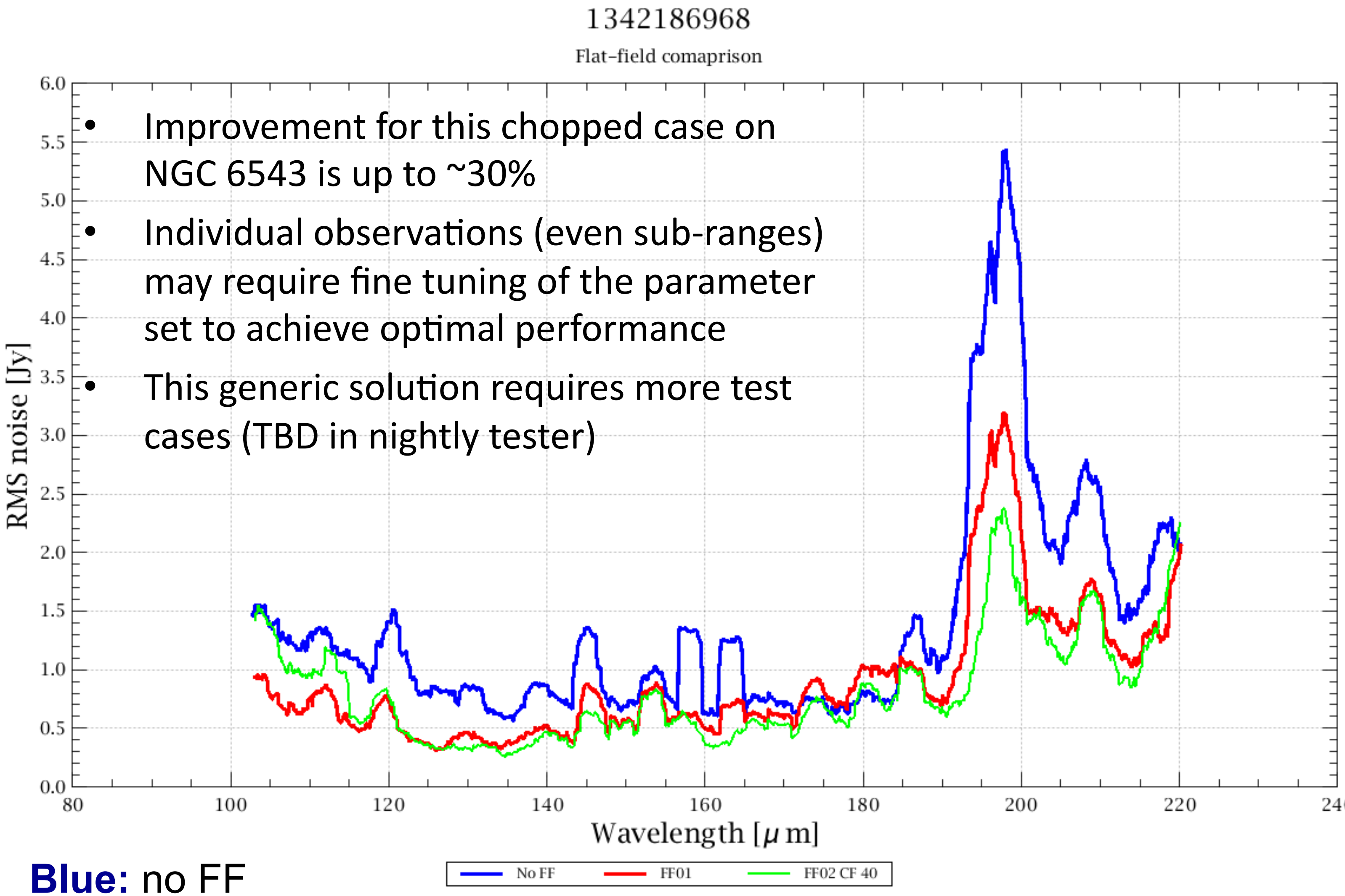
Step 1 & 2 param set = [3.0,5,2,5,3.0,5,2,5]



HERSCHEL SPACE OBSERVATORY

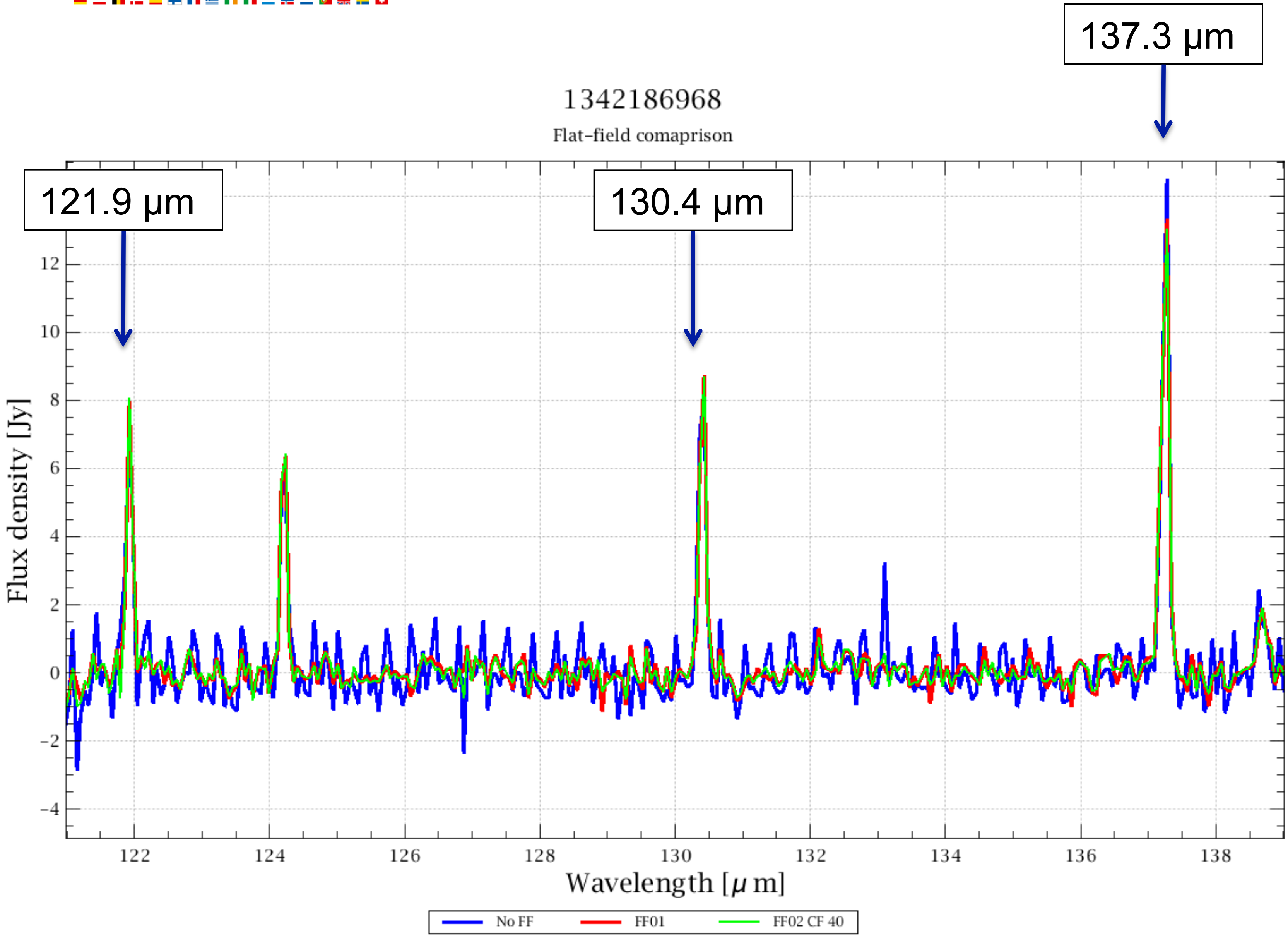
Flat-field parameter optimization

- Improvement for this chopped case on NGC 6543 is up to ~30%
- Individual observations (even sub-ranges) may require fine tuning of the parameter set to achieve optimal performance
- This generic solution requires more test cases (TBD in nightly tester)



Blue: no FF
Red: FF1 (current algorithm in ipipe scripts of HIPE 8.0)
Green: new FF2 with optimal parameters

Flat-field parameter optimization



HERSCHEL SPACE OBSERVATORY

Flat-field parameter optimization