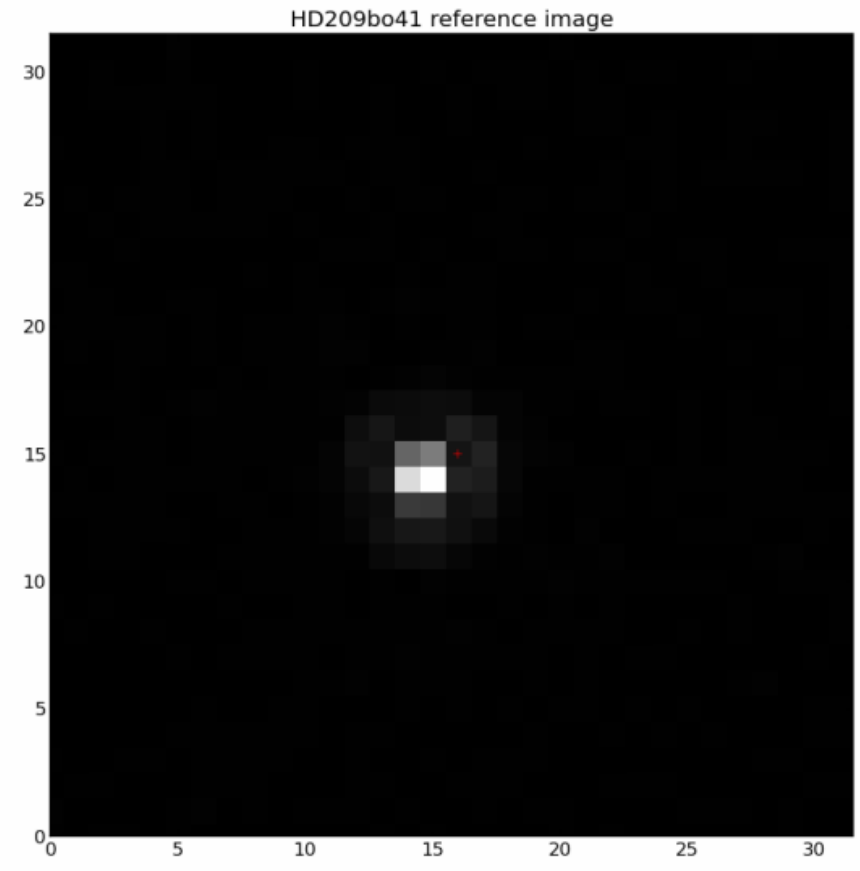
Exoplanet Data Reduction Tool

Hannah Diamond-Lowe & Zakir Gowani

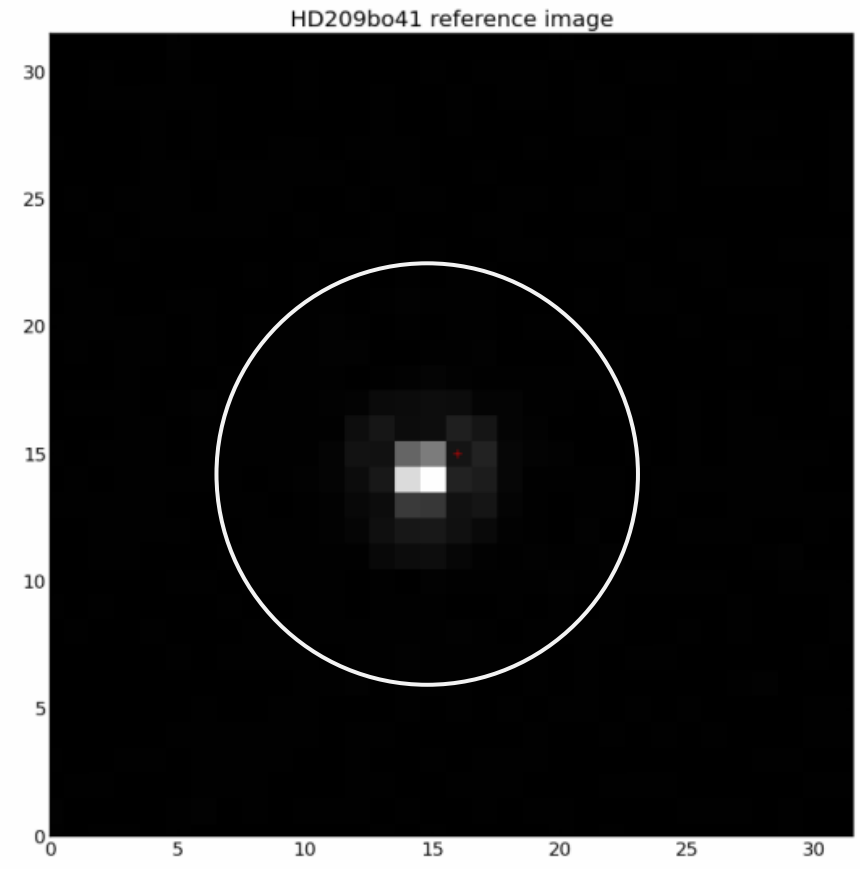
Overview

- Determine best aperture size to use
- Quick, efficient method for guessing boundaries



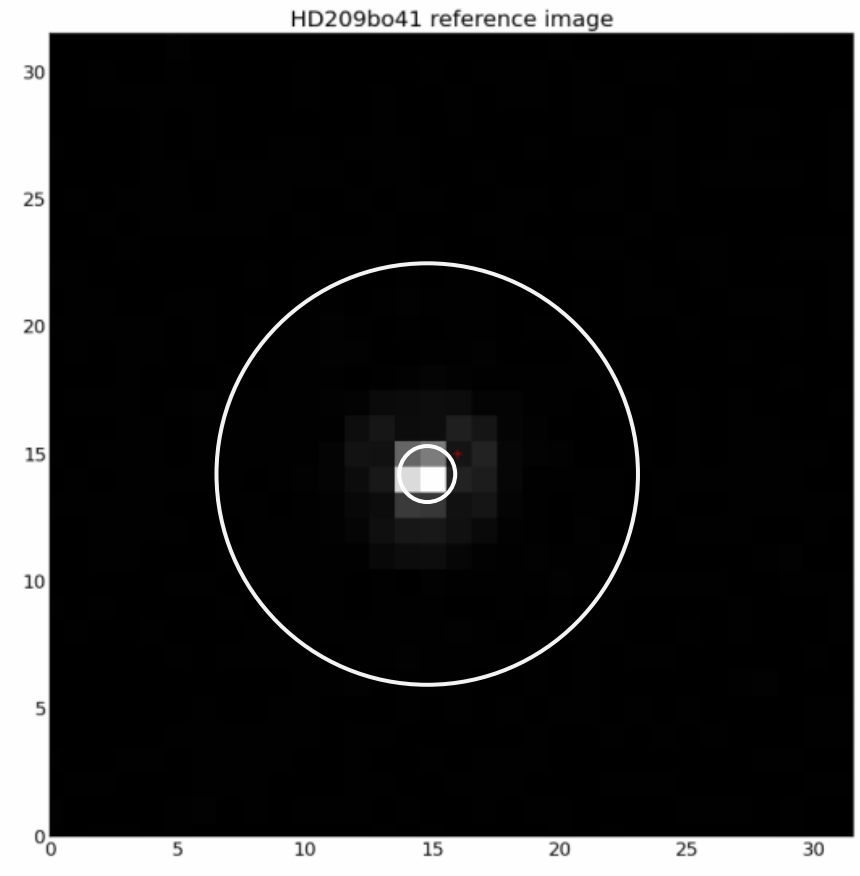
Motivation

- Too large = too much noise



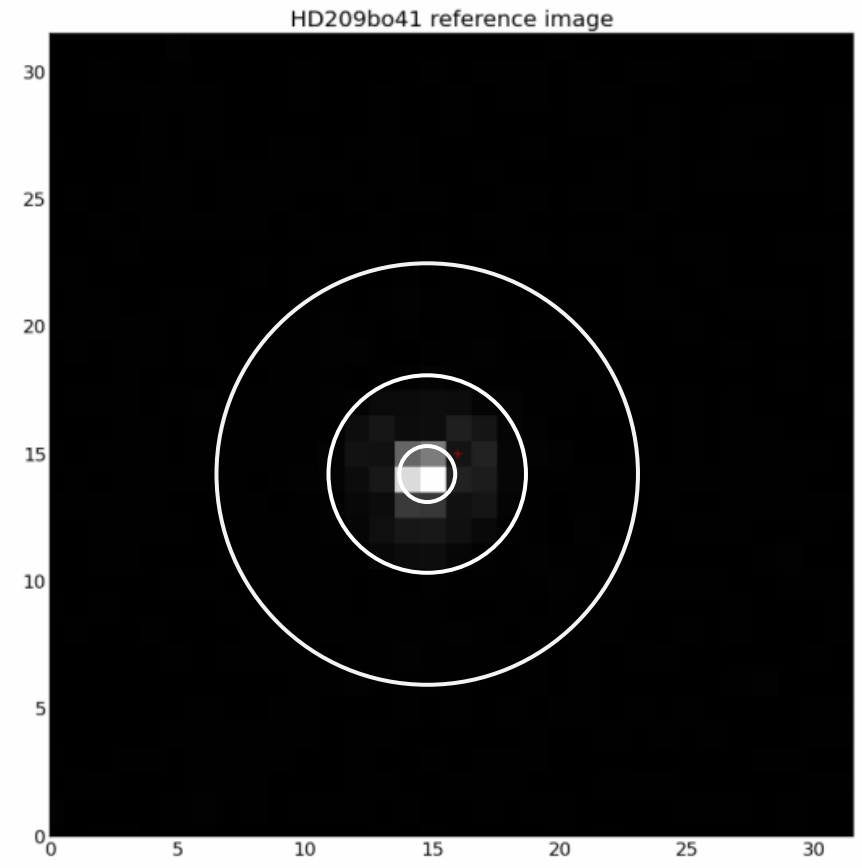
Motivation

- Too small = loss of information



Motivation

- Just right = the most information with the least amount of background noise
- If this step is not done properly the whole analysis will be wrong



Problem

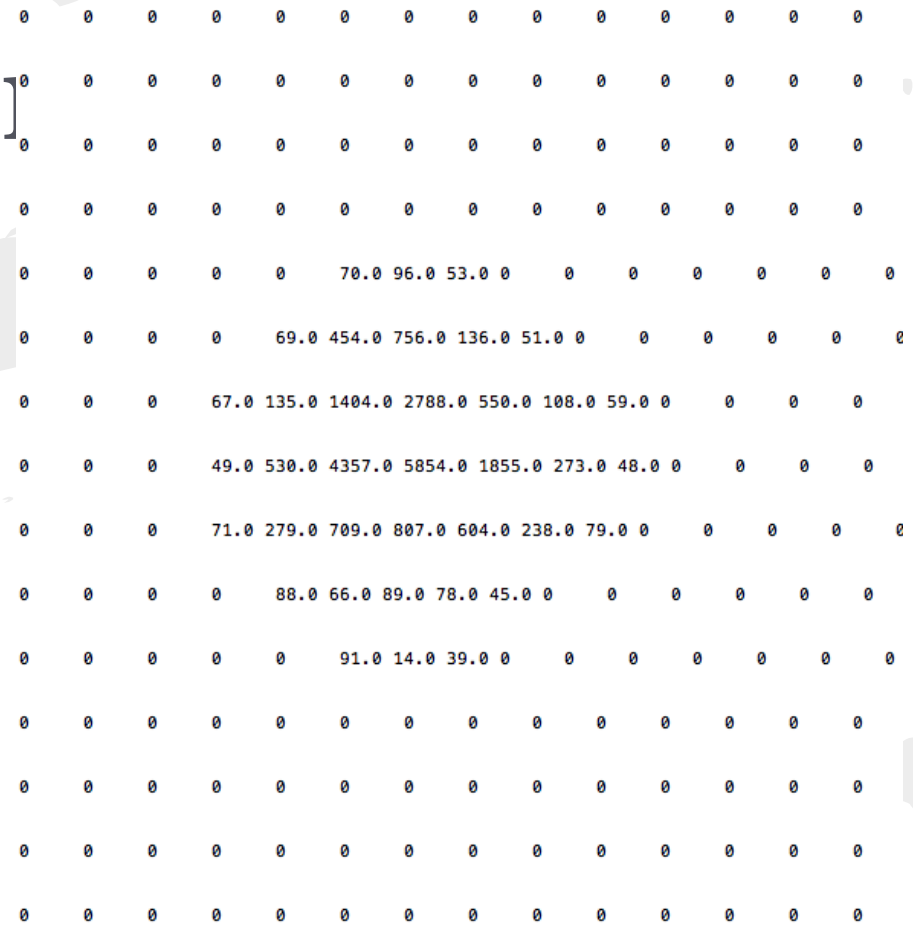
- Finding the best aperture is lengthy and time consuming
- The best solution will provide a guess range
- Solution example: best aperture from 3 - 4 pixels in radius

Raw Data

- FITS files (pictures)
- 64 data frames per FITS file
- ~4 - 8GB per dataset

Data Reductio

- Eliminate noise spikes
- Determine how much information to keep
- Test runtimes for 10 FITS files (~8000 in whole



Data Reduction: Method A

- Determine the best aperture size for every data frame file
- Each thread contributes to analysis of single FITS file
- 64 data frames spread evenly across available threads
- Runtime for test: 288.1 s (extrapolate: 64.02 hr)

Data Reduction: Method B

- Determine the best aperture size for every data frame file
- Main thread obtains full list of FITS files and distributes whole FITS files to other threads
- Runtime for test: 430.0 s (extrapolate: 95.6 hr)

Data Reduction: Method C

- Determine the best aperture size per FITS file
- Create an averaged, composite data frame
- Good for data frames taken with short exposure times
- Runtime for test: 11.5 s (extrapolate: 2.6 hr)

Data Reduction: Method D

- Determine best aperture size for a sample of data frames in each FITS file
- Choose 4 out of 64 data frames per FITS file
- Good for data frames with long exposure times
- Runtime for test: 21.8 s (extrapolate: 4.84 hr)

And the winner is…

Method C!

- Fastest
- Inherently deals with short-timescale noise spikes

Results