1. Introduction

2. Automation Program

The rc3 class creates an rc3 object containing its "basic info" [1] for each galaxy in the RC3 catalog. The practical purpose of the class (other than the use of OOP that actually makes sense) is to keep tract of the number of iterations in the recursive step using the instance variable num_iterations.

Some parts of the program needs to be adjusted for survey-specific , but the core concept (and bulk of the code) should stay the same.

2.1. Technical Details

The program was written in Python 2.7.6 to survey-independent and interacts It uses IPAC's Montage [1] using the AstroPy Montage wrapper[2]. The final g,r,i image is created using Astromatic STIFF [2] . Our choice of program reflects best feature from both program : STIFF provides the flexibility of changing many variables for the final g,r,i; Montage creates scientifically-calibrated images by retaining astrometric accuracy during image reprojection. The use of two different program in the mosaicing step is due to Montage's ability to create scientifically-calibrated mosaic FITS, but STIFF provided more flexible parameters for combining all bands into color images, so we get the best of both worlds. The STIFF setting needs to be adjusted for each survey depending on specs on the telescope's CCD dependent parameters such as imaging bands. (?) The source extraction is done using SExtractor. If the mosaicing field is chosen correctly, then SExtractor's skylevel estimation is fairly accurate.

2.2. Getting the Data

2.2.1. Search

Figure out how to convert positional values (ra,dec) to record-keeping parameters dependent on the survey's telescope. (tiles, frames...etc) [3] Usually this step can be done using the SQL search. Most surveys have an API that enables you to acess data using SQL querys so that the mosaicing program can interact with so that it doesn't have to click buttons or type in textboxes in the web GUI.

3. Algorithms

4. Pipeline

4.1. Class Hierarchy

5. Result from SDSS run

5.1. Known Errors

Even though a series of exception handling and error prevention mechanicsms were put in place,

5.2. Performance

accelerate the process by "testing" only on the a single band fit file. For SDSS, that is the r band, best quality (longest wavelength =¿ most photons). Most of the time is spent on downloading the raw FITS files from the survey's specific server. This process can be dramtically sped up if the investigator has imaging data stored locally on a hardisk. We anticipate that this consist of simply write a subclass of the Server class and fill in some path-dependent detials (where images are stored) to enable this feature.

6. Algorithm

7. Conclusion

[1] G. B. Berriman, E. Deelman, J. C. Good, J. C. Jacob, D. S. Katz, C. Kesselman, A. C. Laity, T. A. Prince, G. Singh, and M.-H. Su. Montage: a grid-enabled engine for delivering custom science-grade mosaics on demand. In P. J. Quinn and A. Bridger, editors, *Optimizing Scientific Return for Astronomy through Information Technologies*, volume 5493 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 221–232, Sept. 2004. doi: 10.1117/12.550551.

[2] E. Bertin. Displaying Digital Deep Sky Images. In P. Ballester, D. Egret, and N. P. F. Lorente, editors, *Astronomical Data Analysis Software and Systems XXI*, volume 461 of *Astronomical Society of the Pacific Conference Series*, page 263, Sept. 2012.

[1] ra,dec,radius, pgc

[2] http://www.astropy.org/montage-wrapper/

[3] Since SDSS images are taken in drift-scan mode, we need the run,camcol,field values to acess each iamge frame