

High-Contrast Imaging in the Cloud with klipReduce and Findr

Asher Haug-Baltzell^a, Jared R. Males^b, Katie M. Morzinski^b, Ya-Lin Wu^b, Nirav Merchant^{a,c}, Eric Lyons^{a,c}, Laird M. Close^b

^aCyVerse, University of Arizona, Tucson, AZ, USA

^bDepartment of Astronomy, University of Arizona, Tucson, AZ, USA

^cBIO5 Institute, University of Arizona, Tucson, AZ, USA

ABSTRACT

Astronomical data sets are growing ever larger, and the area of high contrast imaging of exoplanets is no exception. With the advent of fast, low-noise detectors operating at 10 to 1000 Hz, huge numbers of images can be taken during a single hours-long observation. High frame rates offer several advantages, such as improved registration, frame selection, and improved speckle calibration. However, advanced image processing algorithms are computationally challenging to apply. Here we describe a parallelized, cloud-based data reduction system developed for the Magellan Adaptive Optics VisAO camera, which is capable of rapidly exploring tens of thousands of parameter sets affecting the Karhunen-Loève image processing (KLIP) algorithm to produce high-quality direct images of exoplanets. We demonstrate these capabilities with a visible-wavelength high contrast data set of a hydrogen-accreting brown dwarf companion.

Keywords: Cyberinfrastructure, cloud computing, exoplanet imaging, adaptive optics, KLIP, PSF subtraction, VisAO

1. INTRODUCTION

Direct imaging is essential for fully understanding exoplanet orbits and determining atmospheric compositions. However, successfully imaging these planets requires tackling two main challenges: the small angular separation between stars and their companion planets, and a stark difference in contrast between signals from the bright stars and faint neighboring planets¹. Fortunately, the development of larger telescopes and adaptive optics (AO) systems is helping to resolve the issue of small angular separation, but challenges still exist in removing residual light from the star while leaving the planet signal intact, a step which is necessary even when advanced coronagraphs or other starlight-suppression systems are used². Determining the optimum point spread function (PSF) for subtraction is a mathematically and computationally difficult problem, challenges which are further compounded by the fact that the optimum PSF varies with time and thus must be calculated for each image rather than once for a time series. Here, we present a parallelized software implementation of the Karhunen-Loève image projection algorithm suitable for conducting statistically-optimal PSF subtraction (klipReduce) and an associated software suite for distributed execution of these calculations (Findr). Together these utilities can be used to generate exoplanet images for further analysis and publication, which we demonstrate with visible-wavelength images of a hydrogen-accreting brown dwarf companion taken with the VisAO camera on the Magellan Clay telescope. All tools discussed are available under the MIT open-source license, and are available both as source code and installed on a Docker container.

2. METHODS

2.1 Magellan Adaptive Optics (MagAO)

The Magellan Adaptive Optics (MagAO) system is a second generation adaptive secondary mirror (ASM) installed on the 6.5m Magellan Clay telescope at Las Campanas Observatory (LCO) in Chile³⁻⁵. The primary purpose of the MagAO system is to enable direct imaging of exoplanets. MagAO has two science cameras: the VisAO camera working in the visible (0.6-1.1 μm imaging)^{6-8,2} and the Clio2 camera working in the NIR (1-5.3 μm)^{9,10}. Whereas all NIR light is sent to the Clio2 camera, a selectable beamsplitter divides the visible light between the VisAO camera and a pyramid wavefront sensor (PWFS) for turbulence detection. The 585 actuator ASM is controlled with ~ 300 modes at a rate of 1kHz. Working in the visible, MagAO routinely achieves 20mas resolution.

2.2 Point Spread Function Subtraction: klipReduce

A critical step in direct imaging of exoplanets is accurately subtracting the PSF of the star while leaving behind the image of the planet. This task is generally image specific (i.e. varies with time). "klipReduce" is a custom, fully-parallelized implementation of the Karhunen-Loève Image Processing (KLIP) algorithm¹ for identifying the optimal PSF for subtraction. A key concern of this implementation is computational performance when reducing large numbers ($N =$ several hundred to $N =$ several thousand) of images, which requires eigendecomposition of an $N \times N$ covariance matrix. We use optimized basic linear algebra system (BLAS) libraries, and the Eigen template library to make use of compile time optimization. The eigendecomposition of covariance matrices is performed using the LAPACK SYEVR routine, which only solves for the desired number of eigenvectors. As this number is usually less than N (50 % or less), solving for only these eigenvectors is typically more efficient than, say, SVD based methods which solve for all eigenvectors. Because this implementation is fully parallelized, time needed for calculations decreases nearly linearly with added CPUs.

2.3 klipReduce Parameters and Configuration

The klipReduce algorithm takes a number of parameters that affect the final results (Table 1). The optimum parameters are, in general, unique for each observation night and for each planet, as seeing, weather conditions, elevation, etc., play an important role forming the star PSF and thus in determining the "best" (least-squares sense) star PSF for subtraction. As such, a thorough and statistically-sound analysis of parameters should be conducted using fake-planet injection to maximize the signal-to-noise (S/N) ratio and ensure accurate and non-biased results. To enable this process, a distributed computing solution has been implemented (see 2.4) that allows users to rapidly generate configuration files containing permutations of defined parameter values and simultaneously execute the calculations across distributed ("cloud") compute resources.

Table 1: Important klipReduce parameters. For full documentation including all parameters, visit <http://goo.gl/9GfyaH>.

Option	Type	Description
Pre-processing Parameters		
fakeContrast	vector float	Contrast of fake planet
fakePA	vector float	Position angle to insert fake planet.
fakeSep	vector float	Separation of fake planet.
Processing Parameters		
minRadius	float	Minimum radius of reduced region.
maxRadius	float	Minimum radius of reduced region.
minDPx	float	Angular exclusion.
NModes	vector int	Number of modes to include in PSF.
includeRefNum	int	Number of reference images to use in covariance matrix, chosen by correlation coefficient. If 0, all are used.

2.3.1 Fake Planet injection

The main goal of using Findr with klipReduce is to optimize reduction parameters using fake planets. This is necessary because optimizing on a putative detection risks biasing both the S/N measurement and photometry, as it may optimize speckles rather than the planet. Fake planets are modeled using either unsaturated PSF measurements, or (if unsaturated) the central star itself. In either case the PSF is scaled by a multiplicative contrast factor, set by the "fakeContrast" configuration parameter. If an estimate of the contrast of the actual planet is known, we typically inject a negative planet at its location in an attempt to minimize cross-talk between the real planet and the fake planet. For the same reason we inject only one fake positive planet at a time, at a position angle and separation specified by "fakePA" and "fakeSep" respectively.

2.3.2 Preprocessing parameters

After fake planet injection, we then can apply several pre-processing steps. These include applying a binary mask, e.g. to mask saturated pixels or ghosts, and centered radial profile subtraction. We also employ a Gaussian unsharp mask (high pass filter) specified by the kernel full-width at half-maximum (FWHM). In standard VisAO bandpasses speckles are chromatically elongated radially, so we often apply an unsharp mask with an asymmetric kernel specified with a radial and an azimuthal width. Typical values are 15 and 1 pixels, respectively.

2.3.3 KLIP Parameter Grid Search

Once fake planets are injected and the pre-processing steps are carried out, Findr conducts a grid search through a user-specified set of parameters. `klipReduce` processes images in annular search regions. These are defined in terms of the minimum and maximum radius of the annulus, specified by the "minRadius" and "maxRadius" arguments. When employing angular differential imaging (ADI)¹¹, `klipReduce` employs an angular exclusion parameter. This is equivalent to δ_{\min} from Lafreniere et al (2007)¹², and the `klipReduce` configuration parameter is "minDPx". This specifies the minimum amount of rotation that must have occurred on the inner edge of the search region between the target image and any image included in the reference set. Normally, this is in units of pixels. The "excludeMethod" parameter can be used to change this to degrees or images. The parameter "includeRefNum" is used to limit the number of reference images included in the covariance matrix. If "includeRefNum" < N, the "includeRefNum" images with the highest correlation with relation to the target image are included in the covariance matrix (see the SOSIE algorithm¹³). The parameter "Nmodes" specifies the number of eigenvectors to include in the PSF estimate. This is equivalent to the K_{klip} parameter from Soummer et al (2012)¹.

2.4 Distributed Execution Framework: Findr (FITS Image Normalization by Distributed Resources)

When all parameters and fake planet injections are considered, it is easy to reach tens of thousands of combinations that need to be assessed. Each parameter set requires a full reduction of the entire dataset, leading to a computationally expensive and time intensive analysis. To address this challenge and enable efficient reduction using numerous parameter combinations, a distributed execution framework for `klipReduce` was developed by the University of Arizona 2015 Applied Concepts in Cyberinfrastructure class (<https://pods.iplantcollaborative.org/wiki/display/ACIC2015/>). Over the course of the Fall 2015 semester, a mixed graduate/undergraduate class was divided into four teams mimicking the organization of a standard software development organization: (1) project management, (2) computation (programming), (3) data management, and (4) user interface and documentation. The teams worked together to produce software suite that addresses the challenges of parameter assessment in KLIP-based reduction, ultimately simplifying and improving the overall workflow (Figure 1). The Findr ("FITS Image Normalization by Distributed Resources") suite complements `klipReduce` by automating image preprocessing steps and configuration file generation and by leveraging cloud computing resources to parallelize the execution of `klipReduce` with different sets of parameters. A key design feature of the Findr suite is generality of use applications: although originally developed for `klipReduce`, the utilities can be easily adapted to function with a variety of programs making them useful in distributing a variety of analysis tasks.

The Findr suite is composed of three tools - FindrPrepare, ConfigGenerator, and FindrReduce - which together largely automate the exploration of parameter effects on `klipReduce`. The first program in the suite is FindrPrepare, which fully automates preprocessing steps of observation images. Given a folder of raw observation images, FindrPrepare performs quality thresholding (removal of poor quality images), generates a master dark image(s), conducts dark subtraction from each science image, and centers each science image on the target star. Default dark subtraction and centering utilities are included with the suite, but it is possible for researchers to substitute their own preprocessing tools if desired. Automation of preprocessing steps is important as it saves time and prevents errors that can arise when conducted manually. The final result of running FindrPrepare is a new directory of quality-checked centered and dark subtracted images which are ready for PSF subtraction, de-rotation and merging by `klipReduce`.

After preprocessing the observation images, ConfigGenerator is used to generate configuration files that cover a grid-search of all combinations of specified parameter values. Parameter names and their desired values to analyze are specified in a simple configuration file using new-line separated parameter=["values", "to", "explore"] definitions (example configuration file included in with sourcecode, see 5. Availability and Requirements). Values for each parameter are specified using Python 2.7 syntax lists or list comprehensions (<https://docs.python.org/2/tutorial/datastructures.html>). The final outputs of ConfigGenerator are a directory containing

all configuration files, and a file listing the relative path to each file and the expected output filename from that configuration file. It is important to note that although initially developed for use with *klipReduce*, *ConfigGenerator* is a general-purpose utility and can be set up to generate standard-format configuration files for any program.

Finally, *FindrReduce* is the distributed computing coordinator and handles data transfer, job execution, and results collection to and from worker nodes. *FindrReduce* is used in a typical master-worker environment, where *FindrReduce* is executed on a master node and workers are initialized on distributed computing resources to perform calculations. Distributed computing, including worker initialization, is handled by *WorkQueue*, which provides a highly scalable framework capable of utilizing 1 to tens of thousands of worker nodes¹⁴⁻¹⁶. Worker nodes can be started on most Linux compute resources, including scientific clouds (i.e. *CyVerse Atmosphere*¹⁷⁻¹⁹, *Chameleon Cloud*²⁰), commercial clouds (i.e. Amazon EC2, Google Cloud, Rackspace), HPC facilities (i.e. TACC, XSEDE, campus clusters), or even local machines, with the only requirements being (1) a working installation of *klipReduce* and *CCTools-WorkQueue* (<http://ccl.cse.nd.edu/software/workqueue/>) and (2) permissions to form a network connection to the resource.

Although originally developed specifically for optimizing the *klipReduce* workflow, *Findr* is developed and distributed in a manner that makes it easily extensible and applicable for other uses. *FindrPrepare* allows users to substitute alternative dark subtraction and centering utilities, and can be easily extended to include additional preprocessing steps as needed. *ConfigGenerator* can be used to generate configuration files for any program, and is especially useful for assessing any algorithm with complex interacting parameters. Finally, *FindrReduce* can be used to distribute any program which uses configuration files with only a single command line argument change, and can be modified to accept more complex programs if needed.

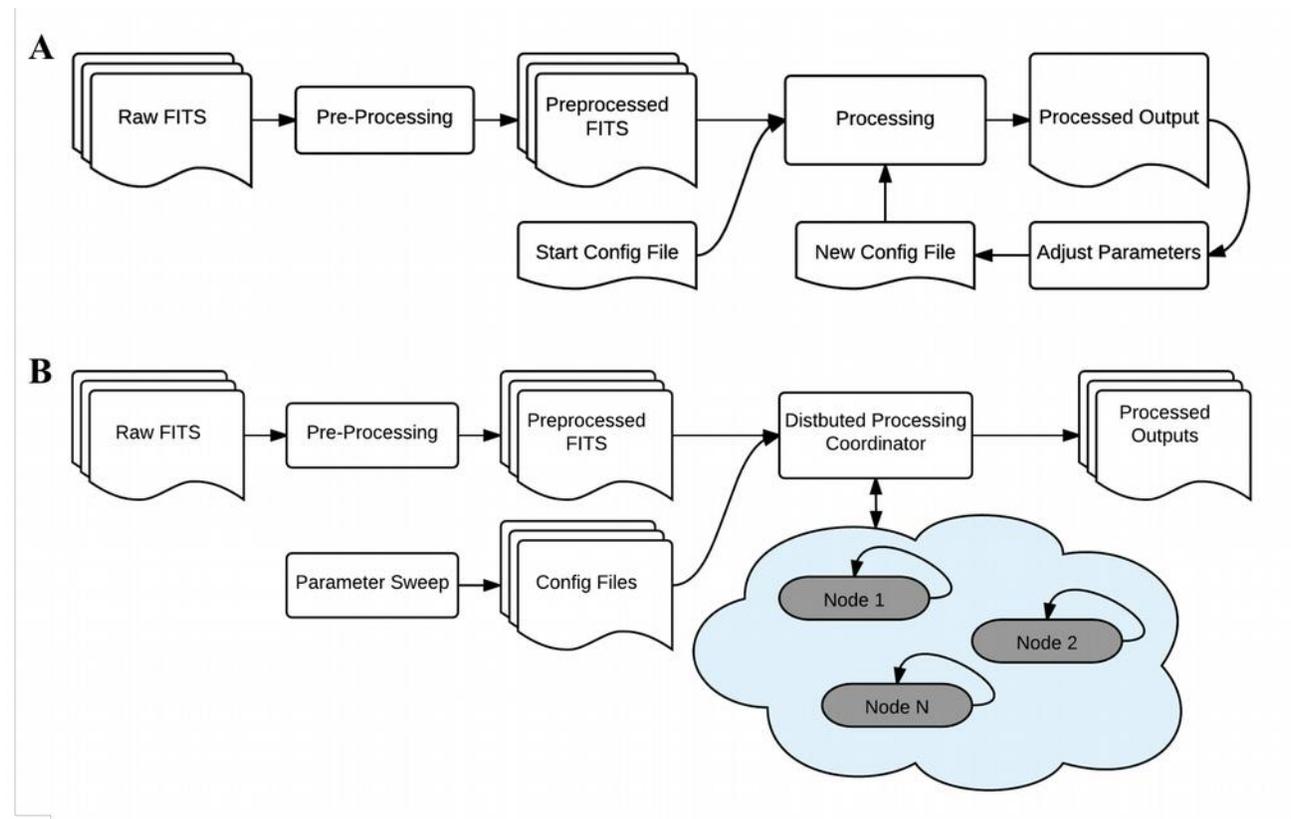


Figure 1. PSF subtraction workflows before (A) and after (B) cloud distribution. (A) Before cloud distribution, determining optimum parameters for KLIP reduction was a slow, manually intensive process. After each reduction, parameters would be adjusted and results compared until a high signal-to-noise ratio was achieved. (B) Distribution of the reduction process to cloud computing resources leads to a simplified workflow and enables full sweeps investigating all parameter combinations. This workflow is faster, less prone to bias, requires less manual input, and leads to statistically sound conclusions.

3. EXAMPLE RESULTS

klipReduce and Findr have now been used on multiple datasets from the VisAO camera, with several publications in preparation based on these reductions. To illustrate the capabilities of Findr, we will discuss one example: the reduction of 143 input images of a hydrogen-accreting brown dwarf companion. The images were obtained using the simultaneous differential imaging (SDI) mode of VisAO⁴. To determine the optimal parameters for maximizing the S/N ratio, 11,880 different parameter combinations were assessed. Using twenty virtual instances running CentOS 7 (8VCPU, 16GB RAM), a complete reduction of the images for each parameter combination was completed in just 1 hour and 11 minutes (190 CPU-hours). These results were analyzed to find the combination of parameters which maximized the mean S/N of the ensemble of fake planets. These optimum parameters were then used to reduce the data with no fake planet injected, shown in Figure 2. The result is a S/N ~ 6 detection of the companion in H- α spectral line, and a non-detection in the continuum filter. Using Findr allows for an unbiased photometry to be conducted on this object. See the forthcoming paper by Wu et al (in prep) for a complete analysis of this data.

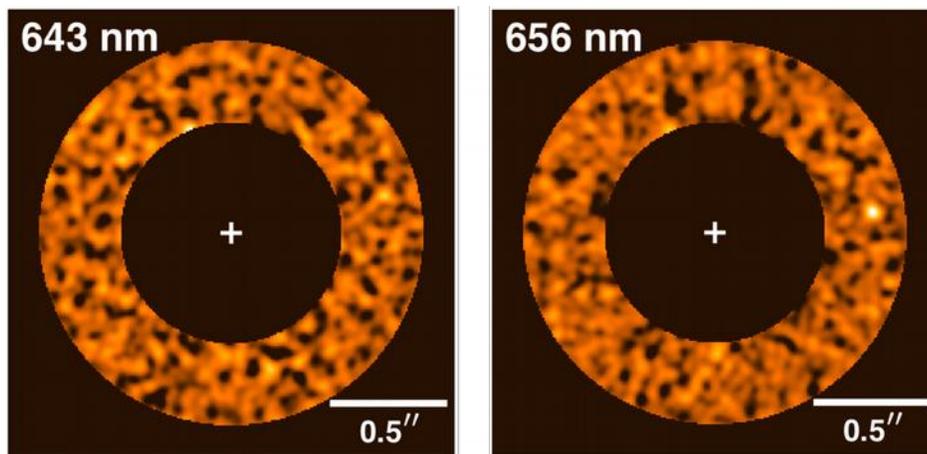


Figure 2. VisAO simultaneous differential imaging (SDI) images of a hydrogen-accreting brown dwarf companion in H α (right) and continuum (left) (Wu et al in prep.). Parameters for maximizing mean signal-to-noise ratio were optimized using an ensemble of 33 fake-planets injections into the H-alpha images, then un-altered images of both bands were reduced using these Findr-optimized parameters. In total, 11,880 parameter sets were investigated over 190 CPU hours (1 hour 11 minutes wall time) to identify the optimum solution.

4. CONCLUSIONS

Used together, klipReduce and Findr provide an efficient solution for converting raw, high-contrast observation images to a final PSF-subtracted exoplanet image. klipReduce overcomes a number of the previous challenges posed by the KLIP algorithm by fully parallelizing and reducing computational burdens of the algorithm and implementing a fake-planet injection mode for assessing parameter effects. Findr enables the assessment of numerous parameter combinations, ensuring that the final result have the lowest possible S/N ratio, are non-biased, and accurately depict the exoplanet rather than an amplification of noise. klipReduce and Findr have been tested and used on multiple datasets from the VisAO camera, and several publications are in preparation based on these reductions. Prior to the implementation of Findr, reduction of a single dataset could take well over a month and required significant manual input, but Findr has shortened this process to frequently less than one day. Further, klipReduce and Findr are designed to scale as computing resources become more powerful and even more widely available. klipReduce is fully parallelized, so additional cores shorten the time required for a single reduction at a near-linear rate, and Findr is developed with scalable distribution technologies that can handle tens of thousands of remote resources¹⁴. These developments will continue to shorten the timeframe needed assess and reduce observation images, enable the inclusion of larger numbers of images, and lead to better overall understanding of how the parameters affect the KLIP algorithm.

5. AVAILABILITY & REQUIREMENTS

5.1 klipReduce

klipReduce is available under the permissive MIT open-source license from the ACIC Utilities repository (<https://bitbucket.org/jaredmales/acic>). All code is written for Linux systems (tested on Ubuntu and CentOS) in C++, and must be compiled from source using a compatible compiler (i.e. GCC). The main dependencies are MXLIB (<https://bitbucket.org/jaredmales/mxlib>), BLAS and LAPACK. To ease installation & setup requirements, a Docker (<http://docker.com>) image preconfigured with klipReduce, Findr and all dependencies is available from the Findr repository (See 5.2).

5.2 Findr

Findr is available as an open-source project under the MIT license, and can be obtained from the ACIC 2015 GitHub repository (<https://github.com/acic2015/findr>). All code is written in Python and requires a functional Python 2.7 interpreter for execution. FindrPrepare requires Astropy²¹ and the ACIC Utilities “darksub”, “darkmaster”, and “fitscent” (<https://bitbucket.org/jaredmales/acic>). FindrReduce requires a functional installation of WorkQueue (<http://ccl.cse.nd.edu/software/workqueue/>) on both master and worker nodes, and klipReduce on worker nodes only. For convenience, a preconfigured Docker image is available in the repository.

ACKNOWLEDGEMENTS

We would like to acknowledge all additional members of the Applied Concepts in Cyberinfrastructure 2015 course (University of Arizona) who contributed to the development, testing and documentation of Findr: Daniel Kapellusch, AJ Stangl, Ian Kaufer, Alex Frank, Alby Chaj, Ryan Jicha, Christian Greyeyes, Erik Goodman, AJ Jicha, David Sidi, Adam Soll, Matt Madrid, Richard Park, Brennen Meland, Daniel Chen, Xavier Ortega, Philipp von Bieberstein, and Roy Zacharias. In addition, we would like to thank the Chameleon Cloud (<http://chameleoncloud.org>) and CyVerse (<http://cyverse.org>: National Science Foundation under Award Numbers DBI-0735191 and DBI-1265383) for providing generous allocations of compute resources and technical support during the development of Findr.

REFERENCES

- [1] Soummer, R., Pueyo, L., Larkin, J., “Detection and Characterization of Exoplanets and Disks using Projections on Karhunen-Loeve Eigenimages,” *Apj*, **755**, L28, (2012).
- [2] Thomas-Osip, J. E., McCarthy, P., Prieto, G., Phillips, M. M., Johns, M., “Giant Magellan Telescope site testing: summary,” *Proc. SPIE*, 7733, 77331L - 10 (2010).
- [3] Close, L. M., Males, J. R., Morzinski, K., Kopon, D., Follette, K., Rodigas, T. J., Hinz, P., Wu, Y.-L., A. Puglisi., et al., “Diffraction-limited Visible Light Images of Orion Trapezium Cluster with the Magellan Adaptive Secondary Adaptive Optics System (MagAO),” *Apj*, **774**(2), 94 (2013).
- [4] Morzinski, K. M., Close, L. M., Males, J. R., Hinz, P. M., “MagAO: status & science,” presented at SPIE: Astronomical Telescopes & Instrumentation, *Proc. SPIE*, Edinburgh, UK (2016).
- [5] Males, J. R., Close, L. M., Kopon, D., Quiros-Pacheco, F., Riccardi, A., Xompero, M., Puglisi, A., Gasho, V., Morzinski, K. M., et al., “Laboratory demonstration of real time frame selection with Magellan AO,” *Proc. SPIE*, 8447, 844742 - 12 (2012).
- [6] Kopon, D., Close, L. M., Males, J. R., Gasho, V., “Design, implementation, and on-sky performance of an advanced apochromatic triplet atmospheric dispersion corrector for the Magellan adaptive optics system and VisAO camera,” *Publ. Astron. Soc. Pac.* **125**(930), 966-975 (2013).
- [7] Close, L. M., Males, J. R., Kopon, D. A., Gasho, V., Follette, K. B., Hinz, P., Morzinski, K., Uomoto, A., Hare, T., et al., “First closed-loop visible AO test results for the advanced

adaptive secondary AO system for the Magellan Telescope: MagAO's performance and status," Proc. SPIE, 8447, 84470X - 16 (2012).

- [8] Males, J. R., Close, L. M., Morzinski, K. M., Wahhaj, Z., Liu, M. C., Skemer, A. J., Derek Kopon., Follette, K. B., Puglisi, A., et al., "Magellan Adaptive Optics First-light Observations of the Exoplanet β Pic B. I. Direct Imaging in the Far-red Optical with MagAO+VisAO and in the Near-ir with NICI," Apj, **786**(1), 32 (2014).
- [9] Hinz, P. M., Rodigas, T. J., Kenworthy, M. A., Sivanandam, S., Heinze, A. N., Mamajek, E. E., Meyer, M. R., "Thermal Infrared MMTAO Observations of the HR 8799 Planetary System," Apj, **716**(1), 417 (2010).
- [10] Morzinski, K. M., Close, L. M., Males, J. R., Kopon, D., Hinz, P. M., Esposito, S., Riccardi, A., Puglisi, A., Pinna, E., et al., "MagAO: Status and on-sky performance of the Magellan adaptive optics system," Proc. SPIE, 9184, 914804 - 13 (2014).
- [11] Marois, C., Lafreniere, D., Doyon, R., Macintosh, B., Nadeau, D., "Angular Differential Imaging: A Powerful High-Contrast Imaging Technique," Apj, **641**(1), 556-564 (2006).
- [12] Lafreniere, D., Marois, C., Doyon, R., Nadeau, D., Artigau, E., "A New Algorithm for Point-Spread Function Subtraction in High-Contrast Imaging: A Demonstration with Angular Differential Imaging," Apj, **660**(1), 770-780 (2007).
- [13] Marois, C., Macintosh, B., Véran, J.-P., "Exoplanet imaging with LOCI processing: photometry and astrometry with the new SOSIE pipeline," 16 July 2010, 77361J.
- [14] Bui, P., Rajan, D., Abdul-Wahid, B., Izaguirre, J., Thain, D., "Work queue+ python: A framework for scalable scientific ensemble applications," Workshop Python High Perform. Sci. Comput. Sc11 (2011).
- [15] Albrecht, M., Rajan, D., Thain, D., "Making work queue cluster-friendly for data intensive scientific applications," September 2013, 1-8, IEEE.
- [16] Zheng, C., Thain, D., "Integrating Containers into Workflows: A Case Study Using Makeflow, Work Queue, and Docker," Proc. 8th Int. Workshop Virtualization Technol. Distrib. Comput., 31-38, ACM, New York, NY, USA (2015).
- [17] Goff, S. A., Vaughn, M., McKay, S., Lyons, E., Stapleton, A. E., Gessler, D., Matasci, N., Wang, L., Hanlon, M., et al., "The iPlant collaborative: cyberinfrastructure for plant biology," Plant Genet. Genomics **2**, 34 (2011).
- [18] Merchant, N., Lyons, E., Goff, S., Vaughn, M., Ware, D., Micklos, D., Antin, P., "The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for the Life Sciences," PLOS Biol. **14**(1), e1002342 (2016).
- [19] "CyVerse.", <<http://www.cyverse.org/>> (30 June 2016).
- [20] "Chameleon Cloud.", <<https://www.chameleoncloud.org/>> (30 June 2016).
- [21] The Astropy Collaboration., Collaboration, T. A., Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., et al., "Astropy: A community Python package for astronomy," Astron. Astrophys. **558**, 9 (2013).