Pippin: A pipeline for supernova cosmology

Samuel Hinton[1] **and Dillon Brout**[2, 3]

1 University of Queensland **2** University of Pennsylvania **3** NASA Einstein Fellow

Summary

Pippin is a python pipeline for supernova cosmology analysis. It is designed to allow a user to specify a cosmological analysis via a configuration file, and then run that analysis, start to finish, in a single execution. Whilst other tools exist to perform various aspects of supernova cosmology analyses independently, connecting the required tasks has so far been a manual process. For the DES-3YR analysis (Abbott et al., 2019), we implemented a basic bash pipeline to reduce manual labour. For the DES-5YR analysis, which greatly increases complexity, we implement this pipeline to increase researcher productivity.

To this end, Pippin interfaces with multiple external programs in the execution of its derived tasks. These tasks include data preparation, supernova simulation using SNANA (Kessler et al., 2009), light curve fitting tasks, machine learning classification of transients, computing the required bias corrections and feeding the results into CosmoMC (Lewis & Bridle, 2002). The MCMC outputs are then analysed in ChainConsumer (Hinton, 2016).

This functionality allows Pippin to run end-to-end, taking telescope data as input and producing cosmological constraints in under a day, with no input required from the user past the initial configuration file. It is currently in use in multiple projects, including investigating the impact of classifiers on cosmology, determining template tuning of the supernova core-collapse rate, assessing the impact of spectral features on classification, and determining the reduction in cosmological bias from improved calibration of the SALT2 supernova model.

Additonal plots are performed via the matplotlib library (Hunter, 2007), and makes use of various numpy (Walt, Colbert, & Varoquaux, 2011), scipy (Jones, Oliphant, Peterson, & others, 2001) and pandas (McKinney & others, 2010) functions. Several machine learning functions make use of scikit-learn (Pedregosa et al., 2011).

Any bugs or feature requests can be opened as issues on the GitHub development page (Hinton, 2020).

References

Abbott, T. M. C., Allam, S., Andersen, P., Angus, C., Asorey, J., Avelino, A., Avila, S., et al. (2019). First Cosmology Results using Type Ia Supernovae from the Dark Energy Survey: Constraints on Cosmological Parameters, *872*(2), L30. doi:10.3847/2041-8213/ab04fa

Hinton, S. (2020). Pippin. Retrieved February 12, 2020, from https://github.com/samreay/Pippin

Hinton, S. R. (2016). ChainConsumer. *The Journal of Open Source Software*, *1*(4), 00045. doi:10.21105/joss.00045

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science and engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Jones, E., Oliphant, T., Peterson, P., & others. (2001). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/

Kessler, R., Bernstein, J. P., Cinabro, D., Dilday, B., Frieman, J. A., Jha, S., Kuhlmann, S., et al. (2009). SNANA: A Public Software Package for Supernova Analysis, *121*(883), 1028. doi:10.1086/605984

Lewis, A., & Bridle, S. (2002). Cosmological parameters from CMB and other data: A Monte Carlo approach, *66*(10), 103511. doi:10.1103/PhysRevD.66.103511

McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (Vol. 445, pp. 51–56). Austin, TX.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830.

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science and Engg.*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37