The Journal of Open Source Software

CWInPy: A Python package for inference with continuous gravitational-wave signals from pulsars

Matthew Pitkin [1,2]

1 Department of Physics, Lancaster University, Lancaster, UK, LA1 4YB **2** School of Physics and Astronomy, University of Glasgow, University Avenue, Glasgow, UK, G12 8QQ

DOI: 10.21105/joss.04568

Software

- Review
- Repository
- Archive

Editor: Dan Foreman-Mackey

Reviewers:

- @GregoryAshton
- @ColmTalbot

Submitted: 14 June 2022
Published: 29 September 2022

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

Continuous quasi-monochromatic gravitational-waves are expected to be emitted from non-axisymmetric rapidly rotating neutron stars (see, e.g., Riles, 2022 for a review). There are thought to be on the order of $10^8 - 10^9$ neutron stars within the Milky Way (Sartore et al., 2010). At the time of writing, around 3000 such stars, have been found through electromagnetic (primarily radio) observations of their highly regular pulsing profiles (Manchester et al., 2005). These are known as pulsars. A proportion of these pulsars with rotation frequencies $\gtrsim 10\,\mathrm{Hz}$ make enticing targets for the current and future generation of ground-based gravitational-wave observatories such as LIGO, Virgo and KAGRA (B. P. Abbott et al., 2019). The detection of gravitational-waves from such a source would reveal information on the size of any non-axisymmetry, or more colloquially a "mountain", on the star. This in turn provides valuable and novel information on neutron star matter and structure (see, e.g., Lasky, 2015).

Detecting the very weak gravitational-wave signal from a pulsar requires the coherent integration of long (months-to-years) gravitational-wave data sets from multiple detectors. CWInPy implements an analysis pipeline enabling the user to search for and characterise these signals. It implements the preprocessing of the time-domain strain, $h(t)$, measured by gravitational-wave detectors for a user-provided set of pulsars (defined through Tempo2-style parameter files, Hobbs et al., 2006; Hobbs & Edwards, 2006); this entails heterodyning the data with the expected signal phase evolution, followed by aggressive filtering and down-sampling of the data (Dupuis & Woan, 2005). These much compressed datasets are then used to perform Bayesian inference on the unknown signal parameters, including the gravitational-wave amplitude.

CWInPy can be used to perform this full pipeline or implement the data preprocessing stages and inferences stages separately. These are all accessible via convenient command line executables (making use of configuration files) or, equivalently, through a Python API. The pipelines provided by CWInPy can perform analyses on an individual machine, but the amount of data being processed generally requires that the preprocessing be parallelised over multiple machines. Therefore, CWInPy will create jobs that can be submitted over a computing cluster running the HTCondor job management system (Thain et al., 2005) via the `htcondor` package. These can also be run over the Open Science Grid (Pordes et al., 2007) using open gravitational-wave data provided via the Gravitational-wave Open Science Center (GWOSC) (R. Abbott, Abbott, et al., 2021; Vallisneri et al., 2015).

For the heterodyne preprocessing stage, CWInPy makes use of algorithms written in `C` within LALSuite (LIGO Scientific Collaboration, 2018) to calculate the phase evolution of each pulsar, which must account for the slowly changing rotation frequency and include Doppler and relativistic effects related to the position and motion of the detector with respect to the pulsar. These are accessible within Python through a SWIG interface to LALSuite (Wette, 2020). The Bayesian inference stage makes use of the `bilby` package (Ashton et al., 2019), which provides a convenient interface to a wide variety of packages for using the Markov Chain

Monte Carlo [MCMC; see, e.g., Sharma (2017) for a review of MCMC with particular reference to astronomy] or nested sampling algorithms (see, e.g., Ashton et al., 2022; Buchner, 2021 for reviews). By default, CWInPy uses the dynesty package (Speagle, 2020) for inference using nested sampling, producing both posterior probability distributions for the parameters of interest and the Bayesian evidence for the data given the signal model.

Statement of need

CWInPy is designed to supersede the current analysis pipeline, known as lalapps_knope (Pitkin et al., 2017), largely based on executables written in C, which has been used for several searches in LIGO and Virgo data (B. P. Abbott et al., 2017, 2019; R. Abbott et al., 2020; R. Abbott, Abe, et al., 2021). The reasons behind CWInPy's development, and its enhancements over existing software, include:

- the Python API allows easy access to the full range of functionality from data preprocessing to source parameter estimation, with greater ability to control various aspects of the analysis;
- the parallelisation of the analysis for multiple pulsars over long observing runs has been changed to improve its efficiency and robustness;
- the intermediate preprocessed data products (the heterodyned data files) have their own HeterodynedData class, based on a GWPy TimeSeries (Macleod et al., 2021), and can be written to and read from HDF5 files; this class includes full information on the data provenance, useful plotting routines in both time- and frequency-domain, and access to data statistics;
- the use of bilby for the signal parameter estimation allows easy access to a wide range of packages for Bayesian inference;
- the pipeline can be run over the Open Science Grid, allowing a wider range of computing resources to be used;
- the pipeline can make use of open gravitational-wave data provided over CVMFS from the Gravitational-wave Open Science Center (GWOSC) (R. Abbott, Abbott, et al., 2021; Vallisneri et al., 2015);
- it provides tools to generate simulated signals injected in real or fake data for individual sources, or for user specified populations of sources;
- it provides tools to perform hierarchical inference on the underlying ellipticity distribution of a population of pulsars (Pitkin et al., 2018);
- it provides tools for determining the significance of a recovered signal using the sky-shifting method (Isi et al., 2020);
- the pulsar phase evolution can be calculated using the standard pulsar timing package Tempo2 (Hobbs et al., 2006) via the libstempo Python package (Vallisneri, 2020) in addition to being able to use the independent routines within LALSuite (LIGO Scientific Collaboration, 2018).

Validation

To validate CWInPy, extensive testing against previous codes has been performed and is provided with the documentation. CWInPy has been used to successfully extract simulated signals directly added into the LIGO data (Biwer et al., 2017) as well as multiple signals simulated via software.

The inference of source parameters has been validated through the use of simulation-based calibration (Talts et al., 2018), with the posterior probability distributions shown to provide well-calibrated (Dawid, 1982) credible intervals.

As evidence for this consistency, Figure 1 shows the posterior probability distributions—plotted using matplotlib (Hunter, 2007), and corner.py (Foreman-Mackey, 2016), via an interface with PESummary, (Hoy & Raymond, 2021)—for the four unknown parameters of a simulated gravitational-wave signal from a pulsar. The figure compares the posterior samples extracted using CWInPy (with the `dynesty` sampler) with those calculated (again with CWInPy) over a uniform grid in the parameters space and those produced by the previously used `lalapps_pulsar_parameter_estimation_nested` code (Pitkin et al., 2017). Consistent posteriors, and also Bayesian odds values comparing the evidence for the signal model versus the data containing pure noise, are found.

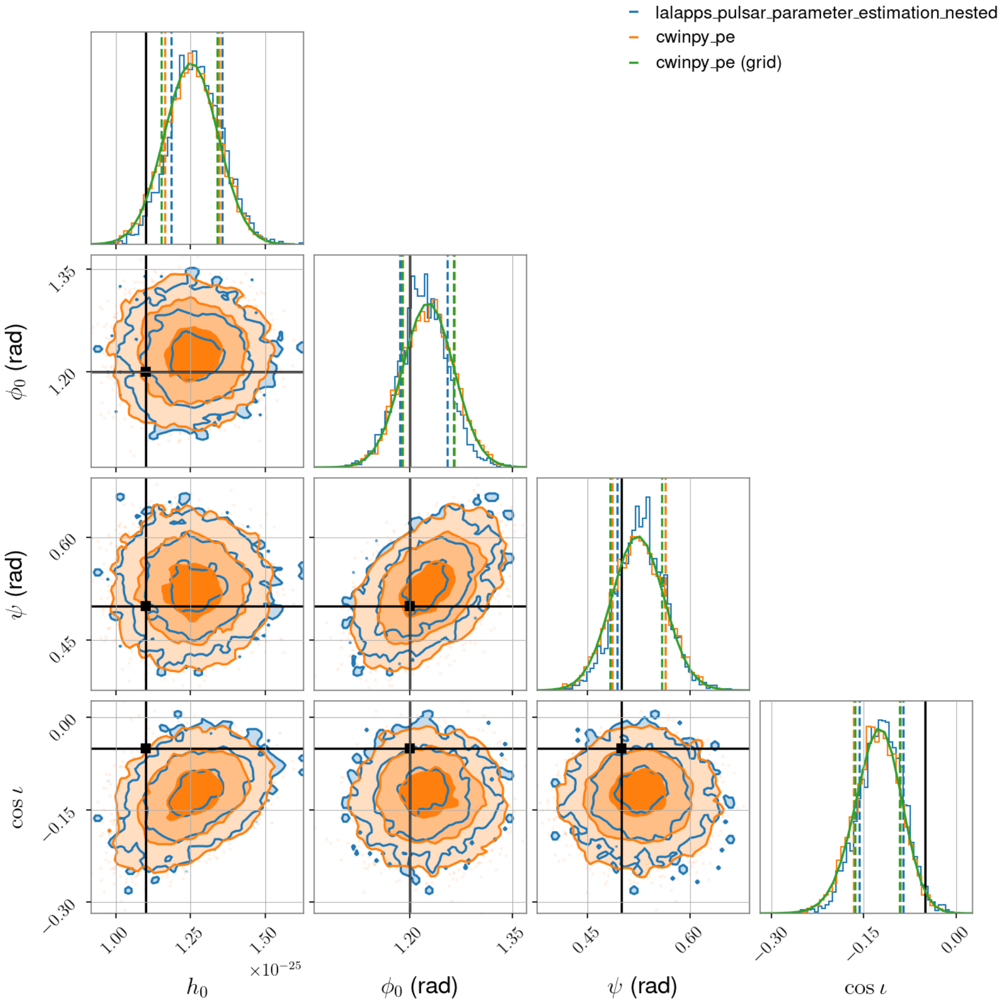

Figure 1: The posterior probability distributions for the parameters of a simulated gravitational-wave signal from a pulsar injected into Gaussian noise.

In Figure 2 CWInPy has been used to heterodyne simulated data containing a gravitational-wave signal and an insignificant amount of noise (to show the the method does not corrupt the signal in any way). The simulated data has been generated using software that is largely independent of CWInPy. The solid lines show the heterodyned time series as produced by CWInPy, whereas the dashed lines show the theoretical expectation for the heterodyned signal, which provide a very good match.

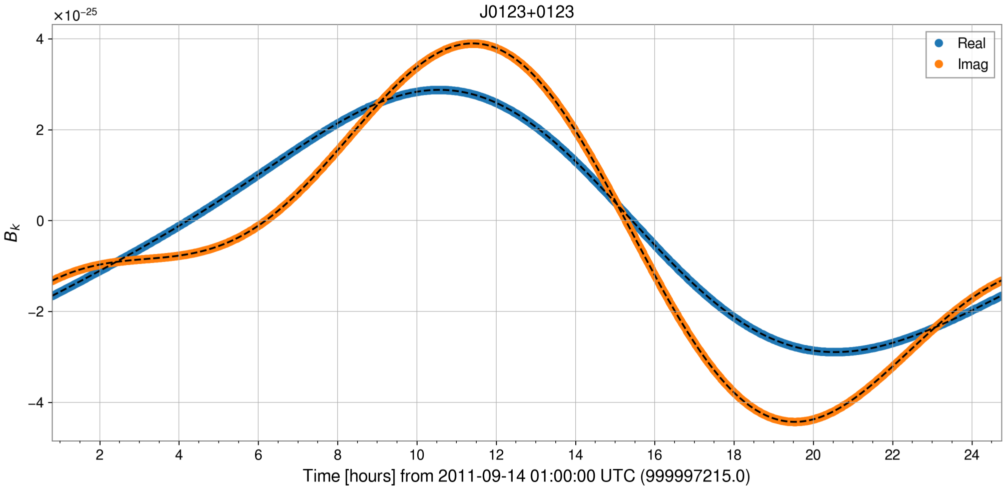

Figure 2: The real and imaginary components of the heterodyned time series for a simulated signal generated using the heterodyne pipeline in CWInPy.

Usage

The main pipelines provided by CWInPy are accessible using command line executables that require a configuration file. The full pipeline, which must be run as an HTCondor directed acyclic graph (DAG), can be run with the `cwinpy_knope_pipeline` executable. The heterodyne preprocessing stage can be run using `cwinpy_heterodyne` or, if running for long stretches of data and multiple pulsars, using a HTCondor DAG via `cwinpy_heterodyne_pipeline`. The latter should be used for most practical purposes, while the former is mainly useful for testing purposes. The parameter estimation stage can be run using `cwinpy_pe` or, if running for multiple pulsars, using an HTCondor DAG via `cwinpy_pe_pipeline`. Full details of all the required configuration file settings are given in the documentation.

Quick setup

Both the `cwinpy_knope_pipeline` and the `cwinpy_heterodyne_pipeline` executables have several command line arguments that can be used for quickly setting up analyses using open data from GWOSC. These rely on the user having access to a computer, or cluster of computers, with HTCondor installed and CVMFS set up with access to the data. To run the analysis the user just needs to have Tempo2-style pulsar ephemeris files for any pulsars they wish to search for. If one had an ephemeris file for, e.g., PSR J0740+6620, called `J0740+6620.par`, then the pipeline could be run over all data from the first observing run of Advanced LIGO (O1) (R. Abbott, Abbott, et al., 2021), using

```
$ cwinpy_knope_pipeline \
--run O1 \
--pulsar J0740+6620.par \
--output /home/usr/analysis
```

where `/home/usr/analysis` can be substituted for the required final location of the files output by the analysis.

If you do not have access to a pulsar ephemeris file, an ephemeris from the ATNF Pulsar Catalogue (Manchester et al., 2005) can be used by instead just specifying the name of the pulsar (you have to trust that the ephemeris provides a coherent timing solution over the gravitational-wave data period). The ephemeris is extracted from the catalogue using psrqpy

(Pitkin, 2018). For example, to search for PSR J0737-3039A using the ATNF Pulsar Catalogue data in LIGO data from the second observing run, one could use:

```
$ cwinpy_knope_pipeline \
--run O2 \
--pulsar J0737-3039A \
--output /home/usr/analysis
```

In both the above cases the `--pulsar` command can be given multiple times to input multiple pulsars. With these "quick setup" options there is no further control over the pipeline, so default parameters are used in all cases. If more control is required then using a configuration file is highly recommended.

The quick setup can be used to analyse all the signal hardware injections for each observation run (Biwer et al., 2017) by supplying the `--hwinj` command instead of `--pulsar`.

Availability, documentation and development

CWInPy is installable under Linux and MacOS using `pip` via PyPI or using `conda` via conda-forge. Full documentation of the executables and Python API is available on Read the Docs. Development is currently performed in the git.ligo.org/cwinpy/cwinpy git repository, which is openly viewable, but for which write access is only available within the LIGO-Virgo-KAGRA collaborations. The master branch is also mirrored on GitHub. Feedback, bug reports, or development suggestions are welcome and can be contributed via Github issues, the discussion forum, or via email.

Acknowledgements

The author acknowledges support from the UK Science & Technology Facilities Council under grant number ST/V001213/1. The development of this package relies on the original work of Réjean Dupuis and Graham Woan and also has benefited massively from discussions with the LIGO-Virgo-KAGRA continuous waves working group and the developers of the bilby software package.

References

Abbott, B. P., Abbott, R., Abbott, T. D., Abernathy, M. R., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., Adhikari, R. X., Adya, V. B., Affeldt, C., Agathos, M., Agatsuma, K., Aggarwal, N., Aguiar, O. D., Aiello, L., Ain, A., Ajith, P., … Weltevrede, P. (2017). First Search for Gravitational Waves from Known Pulsars with Advanced LIGO. *Astrophysical Journal*, *839*(1), 12. https://doi.org/10.3847/1538-4357/aa677f

Abbott, B. P., Abbott, R., Abbott, T. D., Abraham, S., Acernese, F., Ackley, K., Adams, C., Adhikari, R. X., Adya, V. B., Affeldt, C., Agathos, M., Agatsuma, K., Aggarwal, N., Aguiar, O. D., Aiello, L., Ain, A., Ajith, P., Allen, G., Allocca, A., … Weltervrede, P. (2019). Searches for Gravitational Waves from Known Pulsars at Two Harmonics in 2015-2017 LIGO Data. *Astrophysical Journal*, *879*(1), 10. https://doi.org/10.3847/1538-4357/ab20cb

Abbott, R., Abbott, T. D., Abraham, S., Acernese, F., Ackley, K., Adams, A., Adams, C., Adhikari, R. X., Adya, V. B., Affeldt, C., Agathos, M., Agatsuma, K., Aggarwal, N., Aguiar, O. D., Aiello, L., Ain, A., Ajith, P., Allen, G., Allocca, A., … Weltevrede, P. (2020). Gravitational-wave Constraints on the Equatorial Ellipticity of Millisecond Pulsars. *Astrophysical Journal Letters*, *902*(1), L21. https://doi.org/10.3847/2041-8213/abb655

Abbott, R., Abbott, T. D., Abraham, S., Acernese, F., Ackley, K., Adams, C., Adhikari, R. X., Adya, V. B., Affeldt, C., Agathos, M., Agatsuma, K., Aggarwal, N., Aguiar, O. D., Aich, A., Aiello, L., Ain, A., Ajith, P., Allen, G., Allocca, A., … LIGO Scientific Collaboration. (2021). Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo. *SoftwareX*, *13*, 100658. https://doi.org/10.1016/j.softx.2021.100658

Abbott, R., Abe, H., Acernese, F., Ackley, K., Adhikari, N., Adhikari, R. X., Adkins, V. K., Adya, V. B., Affeldt, C., Agarwal, D., Agathos, M., Agatsuma, K., Aggarwal, N., Aguiar, O. D., Aiello, L., Ain, A., Ajith, P., Akutsu, T., Albanesi, S., … Weltevrede, P. (2021). Searches for Gravitational Waves from Known Pulsars at Two Harmonics in the Second and Third LIGO-Virgo Observing Runs. *arXiv e-Prints*, arXiv:2111.13106. https://doi.org/10.48550/arXiv.2111.13106

Ashton, G., Bernstein, N., Buchner, J., Chen, X., Csányi, G., Fowlie, A., Feroz, F., Griffiths, M., Handley, W., Habeck, M., Higson, E., Hobson, M., Lasenby, A., Parkinson, D., Pártay, L. B., Pitkin, M., Schneider, D., Speagle, J. S., South, L., … Yallup, D. (2022). Nested sampling for physical scientists. *Nature Reviews Methods Primers*, *2*(39). https://doi.org/10.1038/s43586-022-00121-x

Ashton, G., Hübner, M., Lasky, P. D., Talbot, C., Ackley, K., Biscoveanu, S., Chu, Q., Divakarla, A., Easter, P. J., Goncharov, B., Hernandez Vivanco, F., Harms, J., Lower, M. E., Meadors, G. D., Melchor, D., Payne, E., Pitkin, M. D., Powell, J., Sarin, N., … Thrane, E. (2019). BILBY: A User-friendly Bayesian Inference Library for Gravitational-wave Astronomy. *Astrophysical Journal Supplement Series*, *241*(2), 27. https://doi.org/10.3847/1538-4365/ab06fc

Biwer, C., Barker, D., Batch, J. C., Betzwieser, J., Fisher, R. P., Goetz, E., Kandhasamy, S., Karki, S., Kissel, J. S., Lundgren, A. P., Macleod, D. M., Mullavey, A., Riles, K., Rollins, J. G., Thorne, K. A., Thrane, E., Abbott, T. D., Allen, B., Brown, D. A., … Sigg, D. (2017). Validating gravitational-wave detections: The Advanced LIGO hardware injection system. *Physical Review D*, *95*(6), 062002. https://doi.org/10.1103/PhysRevD.95.062002

Buchner, J. (2021). Nested Sampling Methods. *arXiv e-Prints*, arXiv:2101.09675. https://arxiv.org/abs/2101.09675

Dawid, A. P. (1982). The Well-Calibrated Bayesian. *Journal of the American Statistical Association*, *77*(379), 605–610. https://doi.org/10.1080/01621459.1982.10477856

Dupuis, R. J., & Woan, G. (2005). Bayesian estimation of pulsar parameters from gravitational wave data. *Physical Review D*, *72*(10), 102002. https://doi.org/10.1103/PhysRevD.72.102002

Foreman-Mackey, D. (2016). Corner.py: Scatterplot matrices in python. *The Journal of Open Source Software*, *1*(2), 24. https://doi.org/10.21105/joss.00024

Hobbs, G. B., & Edwards, R. (2006). *TEMPO2 user manual*. https://www.jb.man.ac.uk/~pulsar/Resources/tempo2_manual.pdf

Hobbs, G. B., Edwards, R. T., & Manchester, R. N. (2006). TEMPO2, a new pulsar-timing package - I. An overview. *Monthly Notices of the Royal Astronomical Society*, *369*(2), 655–672. https://doi.org/10.1111/j.1365-2966.2006.10302.x

Hoy, C., & Raymond, V. (2021). PESummary: the code agnostic Parameter Estimation Summary page builder. *SoftwareX*, *15*, 100765. https://doi.org/10.1016/j.softx.2021.100765

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

Isi, M., Mastrogiovanni, S., Pitkin, M., & Piccinni, O. J. (2020). Establishing the significance of continuous gravitational-wave detections from known pulsars. *Physical Review D*, *102*(12), 123027. https://doi.org/10.1103/PhysRevD.102.123027

Lasky, P. D. (2015). Gravitational Waves from Neutron Stars: A Review. *Publications of the Astronomical Society of Australia*, *32*, e034. https://doi.org/10.1017/pasa.2015.35

LIGO Scientific Collaboration. (2018). *LIGO Algorithm Library - LALSuite*. free software (GPL). https://doi.org/10.7935/GT1W-FZ16

Macleod, D. M., Areeda, J. S., Coughlin, S. B., Massinger, T. J., & Urban, A. L. (2021). GWpy: A Python package for gravitational-wave astrophysics. *SoftwareX*, *13*, 100657. https://doi.org/10.1016/j.softx.2021.100657

Manchester, R. N., Hobbs, G. B., Teoh, A., & Hobbs, M. (2005). The Australia Telescope National Facility Pulsar Catalogue. *Astronomical Journal*, *129*, 1993–2006. https://doi.org/10.1086/428488

Pitkin, M. (2018). psrqpy: a python interface for querying the ATNF pulsar catalogue. *Journal of Open Source Software*, *3*(22), 538. https://doi.org/10.21105/joss.00538

Pitkin, M., Isi, M., Veitch, J., & Woan, G. (2017). A nested sampling code for targeted searches for continuous gravitational waves from pulsars. *arXiv e-Prints*, arXiv:1705.08978. https://doi.org/10.48550/arXiv.1705.08978

Pitkin, M., Messenger, C., & Fan, X. (2018). Hierarchical Bayesian method for detecting continuous gravitational waves from an ensemble of pulsars. *Physical Review D*, *98*(6), 063001. https://doi.org/10.1103/PhysRevD.98.063001

Pordes, R., Petravick, D., Kramer, B., Olson, D., Livny, M., Roy, A., Avery, P., Blackburn, K., Wenaus, T., Würthwein, F., Foster, I., Gardner, R., Wilde, M., Blatecky, A., McGee, J., & Quick, R. (2007). The open science grid. *Journal of Physics: Conference Series*, *78*, 012057. https://doi.org/10.1088/1742-6596/78/1/012057

Riles, K. (2022). Searches for Continuous-Wave Gravitational Radiation. *arXiv e-Prints*, arXiv:2206.06447. https://doi.org/10.48550/arXiv.2206.06447

Sartore, N., Ripamonti, E., Treves, A., & Turolla, R. (2010). Galactic neutron stars. I. Space and velocity distributions in the disk and in the halo. *Astronomy & Astrophysics*, *510*, A23. https://doi.org/10.1051/0004-6361/200912222

Sharma, S. (2017). Markov Chain Monte Carlo Methods for Bayesian Data Analysis in Astronomy. *Annual Review of Astronomy & Astrophysics*, *55*(1), 213–259. https://doi.org/10.1146/annurev-astro-082214-122339

Speagle, J. S. (2020). DYNESTY: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. *Monthly Notices of the Royal Astronomical Society*, *493*(3), 3132–3158. https://doi.org/10.1093/mnras/staa278

Talts, S., Betancourt, M., Simpson, D., Vehtari, A., & Gelman, A. (2018). Validating Bayesian Inference Algorithms with Simulation-Based Calibration. *arXiv e-Prints*, arXiv:1804.06788. https://arxiv.org/abs/1804.06788

Thain, D., Tannenbaum, T., & Livny, M. (2005). Distributed computing in practice: the Condor experience. *Concurrency - Practice and Experience*, *17*(2-4), 323–356. https://doi.org/10.1002/cpe.938

Vallisneri, M. (2020). *libstempo: Python wrapper for Tempo2* (p. ascl:2002.017).

Vallisneri, M., Kanner, J., Williams, R., Weinstein, A., & Stephens, B. (2015). The LIGO Open Science Center. *Journal of Physics Conference Series*, *610*, 012021. https://doi.org/10.1088/1742-6596/610/1/012021

Wette, K. (2020). SWIGLAL: Python and Octave interfaces to the LALSuite gravitational-wave data analysis libraries. *SoftwareX*, *12*, 100634. https://doi.org/10.1016/j.softx.2020.100634