The Journal of Open Source Software

pygwb: a Python-based library for gravitational-wave background searches

Arianna I. Renzini[1,2¶], **Alba Romero-Rodriguez**[3], **Colm Talbot**[4], **Max Lalleman**[5], **Shivaraj Kandhasamy**[6], **Kevin Turbang**[3,5], **Sylvia Biscoveanu**[4,7], **Katarina Martinovic**[8], **Patrick Meyers**[9], **Leo Tsukada**[10,11], **Kamiel Janssens**[5,12], **Derek Davis**[1,2], **Andrew Matas**[13], **Philip Charlton**[14], **Guo-chin Liu**[15], **and Irina Dvorkin**[16]

1 LIGO Laboratory, California Institute of Technology, Pasadena, California 91125, USA **2** Department of Physics, California Institute of Technology, Pasadena, California 91125, USA **3** Theoretische Natuurkunde, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium **4** Kavli Institute for Astrophysics and Space Research, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA **5** Universiteit Antwerpen, Prinsstraat 13, 2000 Antwerpen, België **6** Inter-University Centre for Astronomy and Astrophysics, Pune 411007, India **7** LIGO Laboratory, Massachusetts Institute of Technology, 185 Albany St, Cambridge, MA 02139, USA **8** Theoretical Particle Physics and Cosmology Group, Physics Department, King's College London, University of London, Strand, London WC2R 2LS, United Kingdom **9** Theoretical Astrophysics Group, California Institute of Technology, Pasadena, CA 91125, USA **10** Department of Physics, The Pennsylvania State University, University Park, Pennsylvania 16802, USA **11** Institute for Gravitation and the Cosmos, The Pennsylvania State University, University Park, Pennsylvania 16802, USA **12** Université Côte d'Azur, Observatoire Côte d'Azur, ARTEMIS, Nice, France **13** Max Planck Institute for Gravitational Physics (Albert Einstein Institute), D-14476 Potsdam, Germany **14** OzGrav, Charles Sturt University, Wagga Wagga, New South Wales 2678, Australia **15** Department of Physics, Tamkang University, Danshui Dist., New Taipei City 25137, Taiwan **16** Institut d'Astrophysique de Paris, Sorbonne Université & CNRS, UMR 7095, 98 bis bd Arago, F-75014 Paris, France ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

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

In partnership with

AMERICAN ASTRONOMICAL SOCIETY

This article and software are linked with research article DOI 10.3847/xxxxx <- update this with the DOI from AAS once you know it., published in the Astrophysical Journal <- The name of the AAS journal..

Introduction

A gravitational-wave background (GWB) is expected from the superposition of all gravitational waves (GWs) too faint to be detected individually, or by the incoherent overlap of a large number of signals in the same band (A. I. Renzini et al., 2022). A GWB is primarily characterized by its spectral emission, usually parameterized by the GW fractional energy density spectrum $\Omega_{\rm GW}(f)$, which is the target for stochastic GW searches (Allen & Romano, 1999),

$$\Omega_{\rm GW}(f) = \frac{1}{\rho_c}\frac{d\rho_{\rm GW}(f)}{d\ln f},$$

where $d\rho_{\rm GW}$ is the energy density of GWs in the frequency band f to $f+df$, and ρ_c is the critical energy density of the Universe. Different categories of GW sources may be identified by the unique spectral shape of their background emission; hence, the detection of a GWB will provide invaluable information about the evolution of the Universe and the population of GW sources within it.

Due to the considerable amount of data to analyze, and the vast panorama of GWB models to test, the detection and characterization of a GWB requires a community effort, justifying the need for an accessible and user-friendly open-source code.

Method

The GWB spectrum estimation implemented in pygwb is based on the unbiased minimum variance cross-correlation estimator (Romano & Cornish, 2017),

$$\hat{\Omega}_{\mathrm{GW},f} = \frac{\mathrm{Re}[C_{IJ,f}]}{\gamma_{IJ}(f) S_0(f)}.$$

Here, $C_{IJ,f}$ is the cross-correlation spectral density between two detectors I and J, γ_{IJ} is the overlap reduction function (Allen & Romano, 1999), and $S_0(f) = \frac{3H_0^2}{10\pi^2}\frac{1}{f^3}$, where H_0 is the Hubble constant today (Aghanim & others, 2020). The variance of the estimator is given by

$$\sigma^2_{\mathrm{GW},f} = \frac{1}{2T\Delta f} \frac{P_{I,f} P_{J,f}}{\gamma^2_{IJ}(f) S_0^2(f)},$$

where $P_{I,f}$ is the power spectral density from detector I and T is the duration of data used to produce the above spectral densities. This estimator is optimal and unbiased under the assumption that the signal is Gaussian, isotropic, and continuous. Details on how the estimation is carried out, as well as the implementation of the estimator on large datasets and with many potentially overlapping datasegments can be found in our companion methods paper (Arianna I. Renzini & others, 2023).

Model testing in pygwb is performed through Bayesian inference on a select set of parameters, given a parametric GWB model and a likelihood p of observing the data given the model. Concretely, the above cross-correlation estimator is input data to a Gaussian residual likelihood,

$$p\left(\hat{\Omega}^{IJ}_{\mathrm{GW},f} | \lambda\right) \propto \exp\left[-\frac{1}{2}\sum_{IJ}^{B}\sum_{f}\left(\frac{\hat{\Omega}^{IJ}_{\mathrm{GW},f} - \Omega_{\mathrm{M}}(f|\lambda)}{\hat{\sigma}^{IJ}_{\mathrm{GW},f}}\right)^2\right],$$

where $\Omega_{\mathrm{M}}(f|\lambda)$ is the GWB model and λ are its parameters. pygwb currently admits a variety of GWB models, compatible with the Gaussian likelihood above. More information about the parameter estimation and the implemented models can be found in our companion methods paper (Arianna I. Renzini & others, 2023).

pygwb

pygwb is a Python-based, open-source stochastic GW analysis package specifically tailored to searches for isotropic GWBs with current ground-based interferometers, namely the Laser Interferometer Gravitational-wave Observatory (LIGO), the Virgo observatory, and the KAGRA detector.

The pygwb package is class-based and modular to facilitate the evolution of the code and to increase flexibility of the analysis pipeline. The advantage of the Python language lies in rapid code execution, while maintaining a certain level of user-friendliness, which results in a shallow learning curve and will encourage future contributions to the code from the whole GW community. A summary of all pygwb modules and its main external dependencies can be found in the pygwb schema Figure 1.

The package is compatible with GW frame files in a variety of formats, relying on the I/O functionality of gwpy (Macleod et al., 2021). NumPy (Harris et al., 2020) is heavily used within the pygwb code, as well as matplotlib (Hunter, 2007) for plotting purposes. Some of the frequency-related computations rely on functionalities of the scipy (Virtanen et al., 2020) package. The astropy (Astropy Collaboration et al., 2022) package is employed for cosmology-related computations. The parameter estimation module included in pygwb is based on Bilby (Ashton et al., 2019) and the dynesty (Speagle, 2020) sampler package.

A customizable pipeline script, pygwb_pipe, is provided with the package and can be run in default mode, which reproduces the methodology of the LIGO-Virgo-KAGRA Collaboration (LVK) isotropic analysis implemented on the most recent observation run (Abbott et al., 2021). On the other hand, the modularity of the package allows users to develop custom pygwb pipelines to fit their needs. A set of simple statistical checks can be performed on the data after a pygwb run by using the statistical_checks module. In addition, a parameter estimation script, pygwb_pe, is also included and allows to test a subset of default models with user-defined parameters. pygwb_pe is based on the pygwb parameter estimation module, pe, which allows the user to test both predefined and user-defined models and obtain posterior distributions on the parameters of interest. Users are encouraged to develop and test their own models within the pe module. The pygwb package also contains built-in support for running on HTCondor-supported servers using dag files to parallelize the analysis of long stretches of data. Using the dedicated pygwb_combine script, the output can be combined into an overall estimation of the GWB for the whole data set.

The source code can be found at https://github.com/a-renzini/pygwb, and can be installed from PyPi via pip install pygwb. The online documentation, tutorials and examples are hosted at https://pygwb.docs.ligo.org/pygwb/index.html. The package includes a unit test suite which currently covers 80% of the modules. pygwb is released under a OSI Approved :: MIT License.

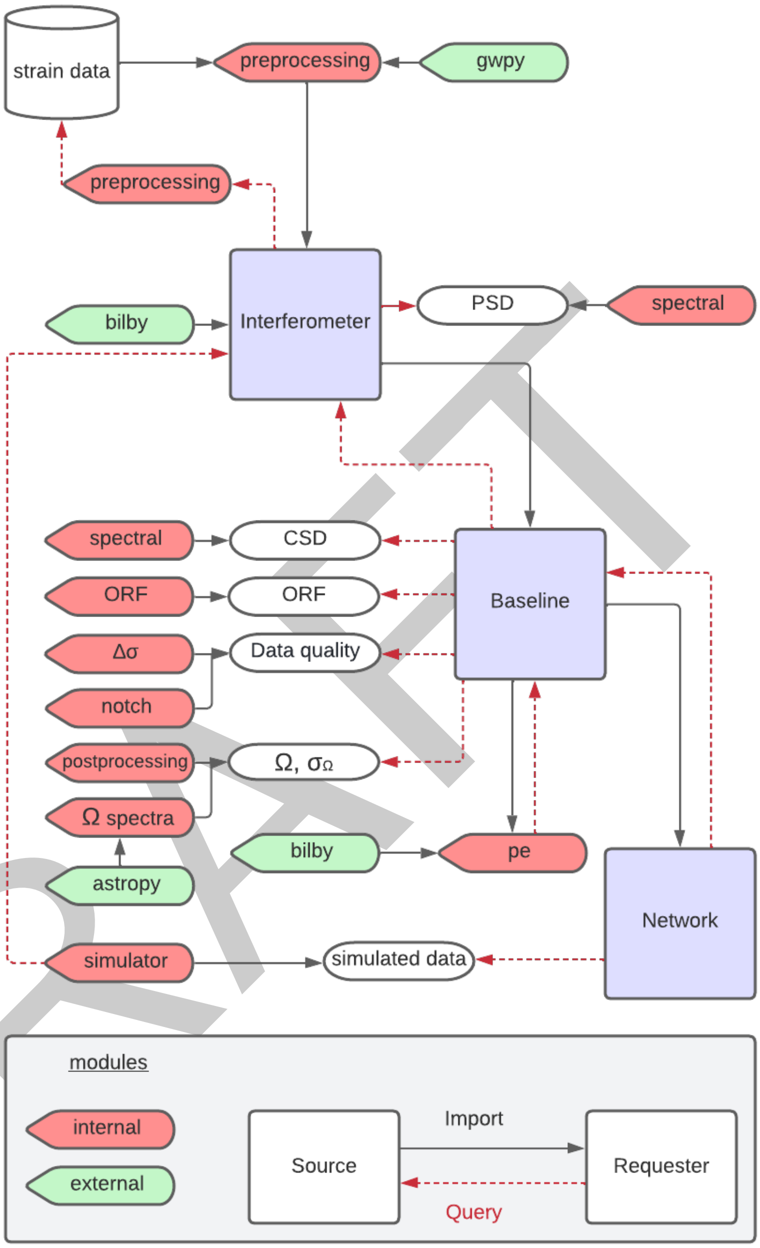

Figure 1: pygwb schema.

Acknowledgements

We thank S. Banagiri, S. Bose, T. Callister, F. De Lillo, L. D'Onofrio, F. Garufi, G. Harry, J. Lawrence, V. Mandic, A. Macquet, I. Michaloliakos, S. Mitra, K. Pham, R. Poggiani, T. Regimbau, J. Romano, N. van Remortel, and H. Zhong for contributing to the review and tests of the code.

We thank the LVK stochastic group for its support. AIR is supported by the NSF award 1912594. ARR is supported in part by the Strategic Research Program "High-Energy Physics" of the Research Council of the Vrije Universiteit Brussel and by the iBOF "Unlocking the

Dark Universe with Gravitational Wave Observations: from Quantum Optics to Quantum Gravity" of the Vlaamse Interuniversitaire Raad and by the FWO IRI grant I002123N "Essential Technologies for the Einstein Telescope". KT is supported by FWO-Vlaanderen through grant number 1179522N. PMM is supported by the NANOGrav Physics Frontiers Center, National Science Foundation (NSF), award number 2020265. LT is supported by the National Science Foundation through OAC-2103662 and PHY-2011865. KJ is supported by FWO-Vlaanderen via grant number 11C5720N. DD is supported by the NSF as a part of the LIGO Laboratory.

This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation. LIGO was constructed by the California Institute of Technology and Massachusetts Institute of Technology with funding from the National Science Foundation, and operates under cooperative agreement PHY-1764464. Advanced LIGO was built under award PHY-0823459. The authors are grateful for computational resources provided by the LIGO Laboratory and supported by NSF Grants PHY-0757058 and PHY-0823459.

References

Abbott, R., Abbott, T. D., Abraham, S., Acernese, F., Ackley, K., Adams, A., Adams, C., Adhikari, R. X., Adya, V. B., Affeldt, C., Agarwal, D., Agathos, M., Agatsuma, K., Aggarwal, N., Aguiar, O. D., Aiello, L., Ain, A., Akutsu, T., Aleman, K. M., … and, J. Z. (2021). Upper limits on the isotropic gravitational-wave background from advanced LIGO and advanced virgo's third observing run. *Physical Review D*, *104*(2). https://doi.org/10.1103/physrevd.104.022004

Aghanim, N., & others. (2020). Planck 2018 results. VI. Cosmological parameters. *Astron. Astrophys.*, *641*, A6. https://doi.org/10.1051/0004-6361/201833910

Allen, B., & Romano, J. D. (1999). Detecting a stochastic background of gravitational radiation: Signal processing strategies and sensitivities. *Phys. Rev. D*, *59*, 102001. https://doi.org/10.1103/PhysRevD.59.102001

Ashton, G., Hübner, M., Lasky, P. D., Talbot, C., Ackley, K., Biscoveanu, S., Chu, Q., Divakarla, A., Easter, P. J., Goncharov, B., Vivanco, F. H., 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. *The Astrophysical Journal Supplement Series*, *241*(2), 27. https://doi.org/10.3847/1538-4365/ab06fc

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., … Astropy Project Contributors. (2022). The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. *935*(2), 167. https://doi.org/10.3847/1538-4357/ac7c74

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

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

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

DRAFT

Renzini, A. I., Goncharov, B., Jenkins, A. C., & Meyers, P. M. (2022). Stochastic gravitational-wave backgrounds: Current detection efforts and future prospects. *Galaxies*, *10*(1). https://doi.org/10.3390/galaxies10010034

Renzini, Arianna I., & others. (2023). *pygwb: Python-based library for gravitational-wave background searches*. https://arxiv.org/abs/2303.15696

Romano, J. D., & Cornish, Neil. J. (2017). Detection methods for stochastic gravitational-wave backgrounds: A unified treatment. *Living Reviews in Relativity*, *20*(1). https://doi.org/10.1007/s41114-017-0004-1

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

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

DRAFT