Contents lists available at ScienceDirect

SoftwareX

Original software publication

Omicron: A tool to characterize transient noise in gravitational-wave detectors

Florent Robinet [a,*], Nicolas Arnaud [a], Nicolas Leroy [a], Andrew Lundgren [b], Duncan Macleod [c], Jessica McIver [d]

[a] Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France
[b] Institute of Cosmology & Gravitation, University of Portsmouth, Portsmouth PO1 3FX, UK
[c] Cardiff University, Cardiff CF24 3AA, UK
[d] University of British Columbia, Vancouver V6T 1Z4, Canada

ARTICLE INFO

ABSTRACT

The Omicron software is a tool developed to perform a multi-resolution time–frequency analysis of data from gravitational-wave detectors: the LIGO, Virgo, and KAGRA detectors. Omicron generates spectrograms from whitened data streams, offering a visual representation of transient detector noises and gravitational-wave events. In addition, these events can be parameterized with an optimized resolution. They can be written to disk to conduct offline noise characterization and gravitational-wave event validation studies. Omicron is optimized to process, in parallel, thousands of data streams recorded by gravitational-wave detectors. The Omicron software plays an important role in vetting gravitational-wave detection candidates and characterization of transient noise.

Code metadata

Current code version	2.4.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_128
Code Ocean compute capsule	
Legal Code License	GNU General Public License 3.0
Code versioning system used	git and gitlab
Software code languages, tools, and services used	C++
Compilation requirements, operating environments & dependencies	CMake, GSL, FFTW, ROOT, HDF5
If available Link to developer documentation/manual	https://tds.virgo-gw.eu/ql/?c=10651
Support email for questions	florent.robinet@ijclab.in2p3.fr

1. Motivation and significance

In 2016, the LIGO [1] and Virgo [2] collaborations announced the first detection of gravitational waves [3]. The signal detected by the LIGO interferometers, and labeled GW150914, originated from a system of two black holes spiraling down and merging into a single and final black hole. This event is of very short duration, ∼200 ms, and had to be extracted from a very large population of transient noise events, also called glitches, polluting the data of interferometric detectors.

Understanding glitches is critical to confidently claim the astrophysical nature of a gravitational-wave candidate. This noise investigation work relies on a monitoring of the instrument and its environment. For that purpose, tens of thousands of auxiliary data streams are recorded. They include data from environmental sensors (thermal, acoustic, seismic, magnetic...), photodiodes, actuators, electronic devices or feedback control loops. Many of those auxiliary channels are insensitive to gravitational waves and are used to witness disturbances from noise sources. More specifically, transient events must be searched for in thousands of auxiliary channels in order to identify correlations with the detector's output data and to understand coupling mechanisms leading to glitches. In this context, the Omicron algorithm, implementing a fast Q transform [4], was developed to detect and characterize transient events in data from gravitational-wave detectors [5,6].

Before 2011, at the time of the first generation of LIGO and Virgo detectors [7–9], the Q transform was already implemented

---

* Corresponding author.
E-mail address: robinet@lal.in2p3.fr (F. Robinet).

to search for transient gravitational-wave signals [10–12] and for detector characterization work [13–15] but the ©MATLAB implementation was too slow to perform a multi-channel analysis required for detector characterization. Another approach, based on a wavelet decomposition [16], was much faster but offered poor precision on the parameter reconstruction. In this context, the Omicron algorithm was first developed in 2012 and much improved over the years. The code is written in C++ for performance reasons and lives in the LIGO–Virgo git repository [17]. A detailed description of the algorithm and its implementation can be found in the technical note [18].

2. Software description

The Omicron software (see Code metadata.) is built upon GWOLLUM [19], a set of libraries used to perform every analysis step of Omicron processing. The GWOLLUM package depends on several external libraries. The FrameL [20] library is used to perform IO operations on data frame files.[1] Mathematical routines developed in the GNU Scientific Library [21] are commonly used by GWOLLUM functions. Discrete Fourier transforms are performed using the FFTW [22] algorithm. Finally, the GWOLLUM and Omicron codes deeply rely on C++ classes developed in the CERN ROOT [23] framework. For example, ROOT classes are used for plotting purposes, event storage or data access. To build Omicron and to manage dependencies, the CMake [24,25] configuration tool was chosen.

Omicron [18] is designed to search for power excess in data time series using the Q transform [4]. The Q transform is a modification of the standard short Fourier transform in which the analysis window duration, σ_t, varies inversely with frequency, ϕ, and is parameterized by a quality factor Q: $\sigma_t = Q/\sqrt{8}\pi\phi$. The data are projected onto a basis of complex-valued sinusoidal Gaussian functions, offering an optimal time–frequency decomposition of the signal x:

$$X(\tau, \phi, Q) = \int_{-\infty}^{+\infty} x(t)w(t - \tau, \phi, Q)e^{-2i\pi\phi t}dt. \quad (1)$$

The transform coefficient, X, measures the average signal amplitude and phase within a time–frequency region, called a *tile*, centered on time τ and frequency ϕ, whose shape and area are determined by the requested quality factor Q and the Gaussian window, w, with a width σ_t. The parameter space (τ, ϕ, Q) is tiled following the strategy developed in [10], where the tiles are distributed over a cubic lattice using a mismatch metric to guarantee a high detection efficiency. This tiling technique offers a multi-resolution time–frequency analysis that consists of logarithmically-spaced Q planes, logarithmically-spaced frequency rows, and linearly-spaced tiles in time.

Before applying the Q transform, the input time series, $x(t)$, is normalized (or whitened) to $x^{wh}(t)$ such that, for pure stationary noise, the expectation value of the power spectral density is frequency-independent and is equal to 2. This whitening step offers a statistical interpretation for the Q transform: the resulting transform coefficient, X^{wh}, is used to derive a signal-to-noise ratio estimator,

$$\hat{\rho}(\tau, \phi, Q) = \begin{cases} \sqrt{\left|X^{wh}(\tau, \phi, Q)\right|^2 - 2}, & \text{if } \left|X^{wh}(\tau, \phi, Q)\right|^2 \geq 2 \\ 0, & \text{otherwise.} \end{cases}$$

$$(2)$$

2.1. Software architecture

The Omicron code is designed with a modular approach so that new features can easily be added. C++ objects are created to achieve specific analysis tasks, and are orchestrated by a master object from the Omicron class. These objects can be divided in three categories as represented in Fig. 1. Input data are managed by a so-called ffl object which is used to read frame files and extract discrete time series. The time series is first conditioned using various signal analysis filters and windows. Additionally, a Spectrum object is created to dynamically measure the average noise spectrum used to whiten the data. The Q transform is performed by a class called Otile. This class manages a collection of time–frequency planes called Q planes. A Q plane is represented by a Omap object which can be seen as a two-dimensional histogram[2] with bins shaped to describe the time–frequency tiling introduced above. Fig. 2 displays two Q planes, illustrating the multi-resolution analysis performed by Omicron. In virtue of the uncertainty principle, the time resolution increases (while the frequency resolution decreases) when the Q value decreases. The Q plane bins are filled with the tile signal-to-noise ratio estimated with Eq. (2) to produce characteristic spectrograms.

Finally, power excesses as measured by Omicron can be saved to disk. Tiles with a signal-to-noise ratio above a given threshold are collected and clustered over time. Resulting events are called triggers and are given parameters, e.g. time, frequency, Q, and signal-to-noise ratio, given by the tile with the highest signal-to-noise ratio in the cluster. Tiles and triggers are operated by multiple GWOLLUM classes. Tiles are represented by an object from the Triggers class which inherits from the ROOT TTree [27] class developed to manage Ntuples. The Triggers object is supervised by the MakeTriggers class which is designed to save the tiles to disk along with auxiliary data like processing meta-data and analyzed time segments.

2.2. Software functionalities

The Omicron class offers many methods for users to conduct their own Q transform analysis. The Omicron package also comes with an out-of-the-box user program which can be run at the command line:

```
# run an omicron analysis over one single time segment:
omicron 1234567890 1234568890 ./my_parameters.txt

# run an omicron analysis over a list of time segments:
omicron ./my_segments.txt ./my_parameters.txt
```

where the Omicron analysis is performed over either one single time segment defined by two GPS times or many GPS time segments listed in a text file (./my_segments.txt). The analysis parameters are listed in a text file (./my_parameters.txt) using specific tags and keywords. The input data files must be listed in another text file [20,28] the path of which is specified in the parameter file. One important parameter is the list of output products the Omicron analysis is to generate. Triggers can be saved to disk as described in Section 2.1. In addition, graphical plots can be generated: spectrograms (single or combined Q planes), time series before and after whitening (including audio tracks), average noise spectra (before and after whitening) and a html report.

---

[1] The frame format was specifically developed for interferometric gravitational-wave detector data.

[2] Histograms are managed by TH2D [26] objects from ROOT.
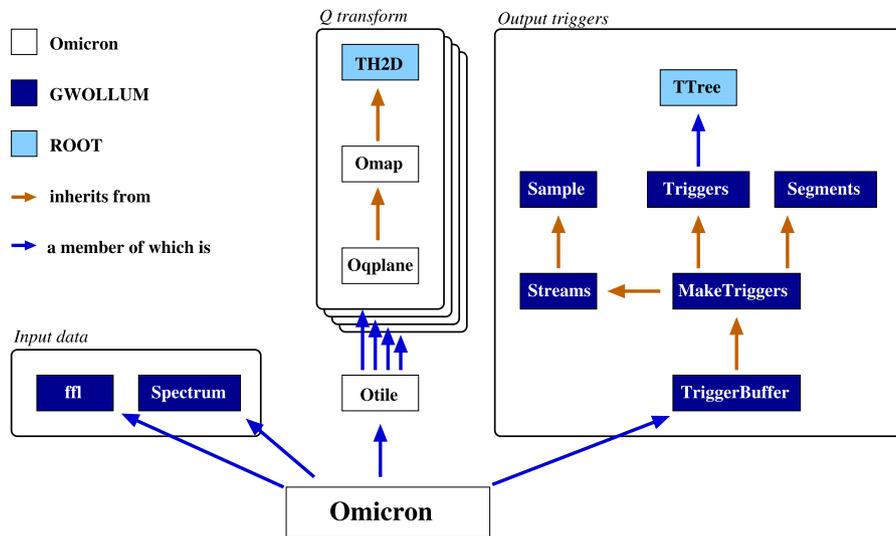
Fig. 1. Omicron class organization. See Section 2.1 for a detailed description.
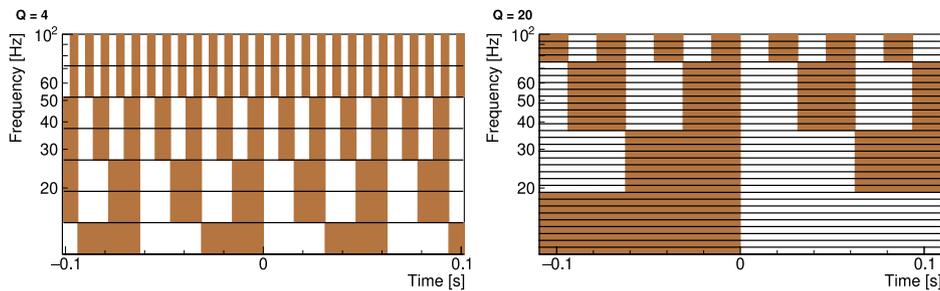
Fig. 2. Example of time–frequency planes for two values of Q: $Q = 4$ (left) and $Q = 20$ (right). Each Q plane is tiled (brown and white rectangles) along frequency rows delimited by the horizontal black lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3. Illustrative examples

The upper plot in Fig. 3 presents the Omicron spectrogram of GW150914 as detected by the LIGO-Hanford interferometer. The frequency evolution of the signal is typical of gravitational waves produced by a binary system of stellar black holes. The different Q planes are stacked up and combined in a single picture, which explains why the power deposits appear star-shaped.

After thresholding the signal-to-noise ratio, tiles are saved in ROOT files. They are typically used to study transient noise in gravitational-wave detectors. For example, in Fig. 4, Omicron triggers generated from Virgo data recorded in August 2017 are distributed in the time–frequency plane. This representation clearly exhibits families of glitches localized in frequency and/or time. This plot is often considered as a starting point for glitch investigations [29–31].

4. Impact

The Omicron software is the primary trigger generator used by the LIGO Scientific Collaboration and Virgo collaboration to study and characterize transient noise in the detectors [32]. It is also widely used to vet gravitational-wave detections [33]. Specific applications [34] have been developed and deployed to perform a continuous Omicron analysis over thousands of LIGO and Virgo channels with a low latency. The triggers are saved to disk a few seconds after detector's data are recorded and are available to third-party applications to study transient noise.

For example, algorithms [35,36] were developed to systematically study correlations between glitches measured in the detector's main gravitational-wave channel and triggers detected in auxiliary channels. These analyses typically identify auxiliary channels witnessing a noise disturbance and lead to a better understanding of the noise coupling mechanisms. Triggers generated by Omicron can also be classified into categories often sourced by the same noise. Glitches can be categorized simply using the trigger parameters estimated by Omicron, e.g. the frequency or the signal-to-noise ratio. For example, the low-frequency and high-SNR triggers, below 30 Hz, visible in Fig. 4 are the result of seismic noise originating from human activity. Indeed, they are found to coincide with times of working hours and many triggers are coincident in time in auxiliary seismic channels. More advanced methods are also used to classify glitches detected by Omicron based on a multi-variate approach [37] or machine-learning techniques [38–41]. The result of these transient noise investigations can lead to the definition of data quality flags. With this information, LIGO–Virgo gravitational-wave searches can ignore flagged time segments which are contaminated by glitches, hence improving the sensitivity to real transient gravitational-wave events [6,33].

After the discovery of GW150914, it took several months to vet the gravitational-wave candidate and to certify its astrophysical origin [6]. The Omicron software played an important part to check many aspects related to transient noise. The data surrounding the event was scrutinized and every glitch was carefully inspected. Triggers in auxiliary channels were examined and statistical studies were conducted to discard correlations between GW150914 and transient noise in auxiliary channels.
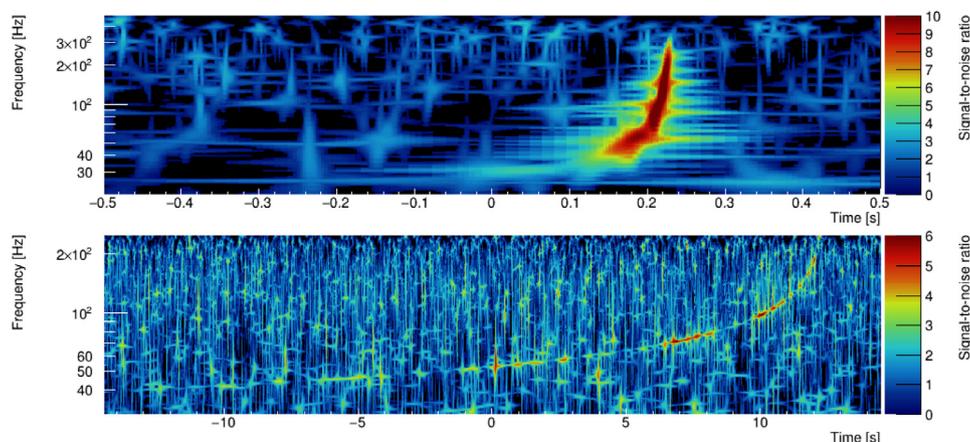
Fig. 3. Top: Omicron spectrogram of LIGO-Hanford detector's data around the time of GW150914. The whitened data is projected in multiple time–frequency planes characterized by a constant Q value and the signal-to-noise ratio is measured for each tile. In this representation, all Q planes are stacked up and combined into one; the tile with the highest signal-to-noise ratio is displayed on top. Bottom: Omicron spectrogram of LIGO-Livingston detector's data around the time of GW170817, using data after glitch subtraction.
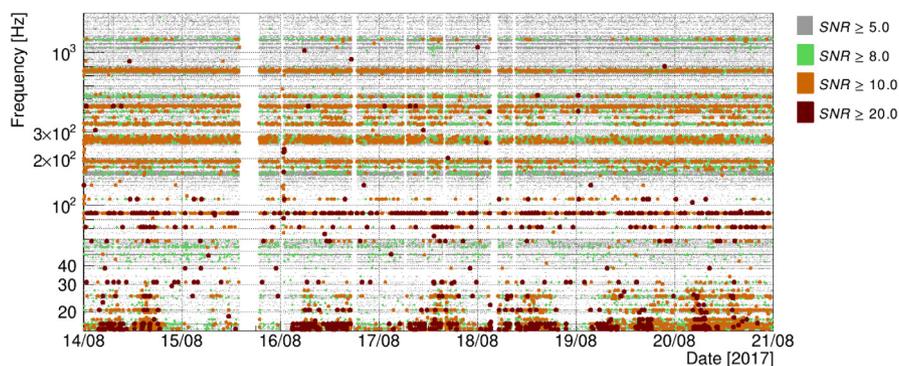
Fig. 4. time–frequency distribution of glitches as detected by Omicron in the Virgo output data over one week in 2017. The glitch signal-to-noise ratio (SNR) is indicated by a color scale. The Omicron processing is interrupted (blank bands) when the detector is not running in nominal conditions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Two years later, another major event, GW170817, was detected [42]. For the first time, an electromagnetic counterpart [43] was associated to the source: a binary system of neutron stars. At the time of the event, a loud glitch contaminated the data of one of the LIGO detectors. As a result, the gravitational-wave signal could not be found in coincidence in multiple detectors by online search pipelines. Shortly after the detection alert, the glitch was manually excised and Omicron was used to visualize the signal in a spectrogram. Having the spectrogram of GW170817 for both LIGO detectors provided one more convincing element to guide the decision of releasing the event to astronomers in charge of conducting the follow-up campaign [44]. The bottom plot in Fig. 3 shows the Omicron spectrogram of GW170817 using the final data set where the glitch was subtracted.

After the first discoveries, the process of vetting gravitational-wave candidates was automatized to cope with the increasing rate of detections. When a candidate is identified by online search pipelines, a series of data quality tests is automatically triggered. An Omicron analysis of all LIGO and Virgo channels with a sampling rate above 1 Hz ($\mathcal{O}(1000)$ per detector) is performed to characterize the transient noise at the time of the event: spectrograms of the strain data and auxiliary channels are generated, trigger distributions are plotted, and correlation analyses with auxiliary channels are conducted. The results of these analyses can be reviewed by the scientists and the public alert can be retracted if the data is found to be of bad quality.

5. Conclusions

The Omicron software occupies a central position in the characterization of transient noise associated to the validation of gravitational-wave events. Omicron combines a high processing speed, high detection efficiency and high resolution to reconstruct parameters of transient noise. Thousands of channels from the LIGO, Virgo and KAGRA [45,46] detectors are continuously processed by Omicron. Resulting triggers offer an invaluable starting point for noise investigations. Over the years, Omicron has served the gravitational-wave community at many levels: it helped to mitigate noise issues in the detectors, it was used to produce data quality flags improving the sensitivity of searches, and it characterized the data quality around the time of gravitational-wave events.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

web tools obtained from the Gravitational Wave Open Science Center (https://www.gw-openscience.org/), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale della Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain.

References

[1] Aasi J, et al., LIGO Scientific Collaboration Advanced LIGO. Classical Quantum Gravity 2015;32:074001. http://dx.doi.org/10.1088/0264-9381/32/7/074001, arXiv:1411.4547.

[2] Accadia T, et al. Advanced virgo interferometer: a second generation detector for gravitational waves observation. In: Proceedings, 16th Lomonosov conference on elementary particle physics: Particle physics at the year of centenary of Bruno Pontecorvo. 2015, p. 261–70. http://dx.doi.org/10.1142/9789814663618_0049.

[3] Abbott BP, et al., Virgo, LIGO Collaboration Observation of gravitational waves from a binary black hole merger. Phys Rev Lett 2016;116(6):061102. http://dx.doi.org/10.1103/PhysRevLett.116.061102, arXiv:1602.03837.

[4] Brown JC. Calculation of a constant Q spectral transform. J Acoust Soc Am 1991;89(1):425–34. http://dx.doi.org/10.1121/1.400476.

[5] Nuttall L, et al. Improving the data quality of advanced LIGO based on early engineering run results. Classical Quantum Gravity 2015;32(24):245005. http://dx.doi.org/10.1088/0264-9381/32/24/245005, arXiv:1508.07316.

[6] Abbott BP, et al., Virgo, LIGO Collaboration Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914. Classical Quantum Gravity 2016;33(13):134001. http://dx.doi.org/10.1088/0264-9381/33/13/134001, arXiv:1602.03844.

[7] Abbott BP, et al., LIGO Scientific Collaboration LIGO: The Laser interferometer gravitational-wave observatory. Rep Progr Phys 2009;72:076901. http://dx.doi.org/10.1088/0034-4885/72/7/076901, arXiv:0711.3041.

[8] Acernese F, et al. Status of Virgo. In: Proceedings, 18th international conference on general relativity and gravitation (GRG18) and 7th Edoardo Amaldi conference on gravitational waves. Classical Quantum Gravity 2008;25:114045. http://dx.doi.org/10.1088/0264-9381/25/11/114045.

[9] Grote H, LIGO Scientific Collaboration. The status of GEO 600. In: Proceedings, 18th international conference on general relativity and gravitation (GRG18) and 7th Edoardo Amaldi conference on gravitational waves. Classical Quantum Gravity 2008;25:114043. http://dx.doi.org/10.1088/0264-9381/25/11/114043.

[10] Chatterji S. The search for gravitational wave bursts in data from the second LIGO science run [Ph.D. thesis], MIT; 2004.

[11] Abadie J, et al., LIGO Scientific, VIRGO Collaboration All-sky search for gravitational-wave bursts in the first joint LIGO-GEO-Virgo run. Phys Rev 2010;D81:102001. http://dx.doi.org/10.1103/PhysRevD.85.089905, arXiv:1002.1036, http://dx.doi.org/10.1103/PhysRevD.81.102001.

[12] Abbott BP, et al., LIGO Scientific Collaboration Search for gravitational-wave bursts in the first year of the fifth LIGO science run. Phys Rev 2009;D80:102001. http://dx.doi.org/10.1103/PhysRevD.80.102001, arXiv:0905.0020.

[13] Aasi J, et al., VIRGO Collaboration The characterization of Virgo data and its impact on gravitational-wave searches. Classical Quantum Gravity 2012;29:155002. http://dx.doi.org/10.1088/0264-9381/29/15/155002, arXiv:1203.5613.

[14] Robinet F, LIGO Scientific, Virgo Collaboration. Data quality in gravitational wave bursts and inspiral searches in the second Virgo Science Run. In: Proceedings, 14th workshop on gravitational wave data analysis. Classical Quantum Gravity 2010;27:194012. http://dx.doi.org/10.1088/0264-9381/27/19/194012.

[15] Christensen N, LIGO Scientific, Virgo Collaboration. LIGO S6 detector characterization studies. In: Proceedings, 14th workshop on gravitational wave data analysis. Classical Quantum Gravity 2010;27:194010. http://dx.doi.org/10.1088/0264-9381/27/19/194010.

[16] Chatterji S, Blackburn L, Martin G, Katsavounidis E. Multiresolution techniques for the detection of gravitational-wave bursts. In: 8th Gravitational wave data analysis workshop. Classical Quantum Gravity 2004;21:S1809–18. http://dx.doi.org/10.1088/0264-9381/21/20/024, arXiv:gr-qc/0412119.

[17] Robinet F. Omicron gitlab project page. 2020, URL https://git.ligo.org/virgo/virgoapp/Omicron.

[18] Robinet F. Omicron: an algorithm to detect and characterize transient events in gravitational-wave detectors. 2018, VIR-0545C-14, URL https://tds.virgo-gw.eu/ql/?c=10651.

[19] Robinet F. GWOLLUM: a toolkit for analysis of gravitational-wave detector's data, URL https://git.ligo.org/virgo/virgoapp/GWOLLUM.

[20] URL http://lappweb.in2p3.fr/virgo/FrameL/.

[21] Galassi M, et al. GNU scientific library reference manual. 2018, URL https://www.gnu.org/software/gsl/.

[22] Frigo M, Johnson SG. The design and implementation of FFTW3. Proc IEEE 2005;93(2):216–31, Special issue on "Program Generation, Optimization, and Platform Adaptation".

[23] Brun R, Rademakers F. ROOT: An object oriented data analysis framework. In: New computing techniques in physics research V. proceedings, 5th international workshop. Nucl Instrum Methods 1997;A389:81–6. http://dx.doi.org/10.1016/S0168-9002(97)00048-X.

[24] URL https://cmake.org/.

[25] Martin K, Hoffman B. An open source approach to developing software in a small organization. IEEE Softw 2007;24:46–53. http://dx.doi.org/10.1109/MS.2007.5.

[26] ROOT: TH2D class reference, URL https://root.cern.ch/doc/master/classTH2D.html.

[27] ROOT: TTree class reference, URL https://root.cern.ch/doc/master/classTTree.html.

[28] URL https://wiki.ligo.org/DASWG/LALSuite.

[29] Verkindt D. Advanced Virgo: from detector monitoring to gravitational wave alerts. In: 54th Rencontres de moriond on gravitation. 2019.

[30] Hemming G, Verkindt D. Virgo interferometer monitor (VIM) web user interface (WUI) user guide. 2016, VIR-0546A-16, URL https://tds.virgo-gw.eu/ql/?c=11869.

[31] Macleod D, Urban A, Isi M, Massinger T, Pitkin M, Nitz A. gwpy/gwsumm, URL https://doi.org/10.5281/zenodo.3590375.

[32] McIver J, Massinger TJ, Robinet F, Smith JR, Walker M. Diagnostic methods for gravitational-wave detectors. In: Reitze D, Grote H, editors. Advanced interferometric gravitational-wave detectors, vol. 1. 2019, p. 373–92. http://dx.doi.org/10.1142/9789813146082_0014.

[33] Abbott BP, et al., LIGO Scientific, Virgo Collaboration GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and virgo during the first and second observing runs. Phys Rev 2019;X9(3):031040. http://dx.doi.org/10.1103/PhysRevX.9.031040, arXiv:1811.12907.

[34] Urban A, Macleod D. gwpy/pyomicron, URL https://doi.org/10.5281/zenodo.2636552.

[35] Isogai T, LIGO Scientific, Virgo Collaboration. Used percentage veto for LIGO and Virgo binary inspiral searches. In: Proceedings, 14th workshop on gravitational wave data analysis, vol. 243. 2010, http://dx.doi.org/10.1088/1742-6596/243/1/012005, 012005.

[36] Smith JR, Abbott T, Hirose E, Leroy N, Macleod D, McIver J, et al. A hierarchical method for vetoing noise transients in gravitational-wave detectors. Classical Quantum Gravity 2011;28:235005. http://dx.doi.org/10.1088/0264-9381/28/23/235005, arXiv:1107.2948.

[37] Kapadia SJ, Dent T, Dal Canton T. Classifier for gravitational-wave inspiral signals in nonideal single-detector data. Phys Rev 2017;D96(10):104015. http://dx.doi.org/10.1103/PhysRevD.96.104015, arXiv:1709.02421.

[38] Biswas R, et al. Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data. Phys Rev 2013;D88(6):062003. http://dx.doi.org/10.1103/PhysRevD.88.062003, arXiv:1303.6984.

[39] Zevin M, et al. Gravity spy: Integrating advanced LIGO detector characterization, machine learning, and citizen science. Classical Quantum Gravity 2017;34(6):064003. http://dx.doi.org/10.1088/1361-6382/aa5cea, arXiv:1611.04596.

[40] Coughlin S, et al. Classifying the unknown: discovering novel gravitational-wave detector glitches using similarity learning. Phys Rev D 2019;99(8):082002. http://dx.doi.org/10.1103/PhysRevD.99.082002, arXiv:1903.04058.

[41] Mukund N, Abraham S, Kandhasamy S, Mitra S, Philip NS. Transient classification in LIGO data using difference boosting neural network. Phys Rev 2017;D95(10):104059. http://dx.doi.org/10.1103/PhysRevD.95.104059, arXiv:1609.07259.

[42] Abbott B, et al., LIGO Scientific, Virgo Collaboration GW170817: Observation of gravitational waves from a binary neutron star inspiral. Phys Rev Lett 2017;119(16):161101. http://dx.doi.org/10.1103/PhysRevLett.119.161101, arXiv:1710.05832.

[43] Ajello M, et al. Fermi-LAT observations of LIGO/Virgo event GW170817. Astrophys J 2018;861(2):85. http://dx.doi.org/10.3847/1538-4357/aac515.

[44] Abbott B, et al., LIGO Scientific, Virgo, Fermi GBM, INTEGRAL, IceCube, As-troSat Cadmium Zinc Telluride Imager Team, IPN, Insight-Hxmt, ANTARES, Swift, AGILE Team, 1M2H Team, Dark Energy Camera GW-EM, DES, DLT40, GRAWITA, Fermi-LAT, ATCA, ASKAP, Las Cumbres Observatory Group, OzGrav, DWF (Deeper Wider Faster Program), AST3, CAASTRO, VINROUGE, MASTER, J-GEM, GROWTH, JAGWAR, CaltechNRAO, TTU-NRAO, NuSTAR, Pan-STARRS, MAXI Team, TZAC Consortium, KU, Nordic Optical Telescope, ePESSTO, GROND, Texas Tech University, SALT Group, TOROS, BOOTES, MWA, CALET, IKI-GW Follow-up, H.E.S.S., LOFAR, LWA, HAWC, Pierre Auger, ALMA, Euro VLBI Team, Pi of Sky, Chandra Team at McGill University, DFN, ATLAS Telescopes, High Time Resolution Universe Survey, RIMAS, RATIR, SKA South Africa/MeerKAT Collaboration Multi-messenger observations of a binary neutron star merger. Astrophys J 2017;848(2):L12. http://dx.doi.org/10.3847/2041-8213/aa91c9, arXiv:1710.05833.

[45] Akutsu T, et al., KAGRA Collaboration Construction of KAGRA: an underground gravitational wave observatory. Prog Theor Exp Phys 2018;2018(1):013F01. http://dx.doi.org/10.1093/ptep/ptx180, arXiv:1712.00148.

[46] Akutsu T, et al., KAGRA Collaboration First cryogenic test operation of underground km-scale gravitational-wave observatory KAGRA. Classical Quantum Gravity 2019;36(16):165008. http://dx.doi.org/10.1088/1361-6382/ab28a9, arXiv:1901.03569.