Original software publication

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

Florent Robinet¹,∗, Nicolas Arnaud², Nicolas Leroy³, Andrew Lundgren⁴, Duncan Macleod⁵, Jessica McIver⁶

¹ Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France
² Institute of Cosmology & Gravitation, University of Portsmouth, Portsmouth PO1 3FX, UK
³ Cardiff University, Cardiff CF24 3AA, UK
⁴ University of British Columbia, Vancouver V6T 1Z4, Canada

A B S T R A C T

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.

Crown Copyright © 2020 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

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.
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 im-
proved 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 rou-
tines 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 GWOL-
LUM 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: σ_t = Q/√2πφ. The
Gaussion functions, offering an optimal time–frequency
decomposition of the signal x:

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

The transform coefficient, X, measures the average signal am-
plitude and phase within a time–frequency region, called a tile,
centered on time \( \tau \) and frequency \( \phi \), whose shape and area are
determined by the requested quality factor Q and the Gaussian
window, \( w \), with a width \( \sigma_t \). The parameter space \((\tau, \phi, Q)\) 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 fre-
quency 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 sta-
nary noise, the expectation value of the power spectral density is
frequency-independent and is equal to 2. This whitening step of-
fers a statistical interpretation for the Q transform: the resulting
transform coefficient, \( X^{wh}(\tau, \phi, Q) \), is used to derive a signal-to-noise ratio
estimator,

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

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

2 Histograms are managed by TH2D [26] objects from ROOT.

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 ff1 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 per-
formed by a class called Qtile. This class manages a collection of
time–frequency planes called Q planes. A Q plane is represented
by a Qmap object which can be seen as a two-dimensional his-
togram2 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 thresh-
old 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 oper-
ated 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 de-
signed to save the tiles to disk along with auxiliary data like
data 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 seg-
ments 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.
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.
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

Andrew Lundgren is supported by STFC, UK grant ST/N000668/1. Duncan Macleod is supported by the European Union Horizon 2020 Framework Programme under grant agreement no. 663380-CU073. This research has made use of data, software and/or...
[44] Abbott B, et al., LIGO Scientific, Virgo, Fermi GBM, INTEGRAL, IceCube, AstroSat 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, I-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, IGI-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.