VOEvent Documentation: How to read and interpret a VOEvent

Min-A Cho

1. What is a VOEvent?

A VOEvent (Virtual Observatory Event) is a packet of information summarizing the who, what, where, when, and how of a transient astronomical event. Data in the VOEvent is presented in a structured format rather than in natural language so that automated systems, such as robotic telescopes, can effectively interpret and use the information to follow up on the event.

In the aLIGO (Advanced Laser Interferometer Gravitational-Wave Observatory) setting, VOEvents will be the automated mode of communication between aLIGO experimenters and the astrophysics community. Different pipelines that analyze the recorded data at aLIGO will reveal information about these candidate events and successive VOEvents will transmit the newfound information.

This documentation will cover under which circumstances different version VOEvents are generated and what kind of information will be contained in each one.

2. Generation of Alerts

Multiple pipelines analyze data recorded by aLIGO and VIRGO. In the search for Compact Binary Coalescences (CBCs), matched filtering is used and in the search for Coherent Wave Bursts (CWBs), excess power method is used. If these pipelines find anything interesting, they generate a trigger that is recorded in the Gravitational-wave Candidate Event Database (GraCEDb) with its **Unique Identifier (UID)**, in the form G##### and an **Event Time** in GPS. This is now a candidate event that must go through a series of approvals before a VOEvent is generated and issued.

2.1. Preliminary Checks. I. Candidate events can be categorized with **Labels**. The first automated check makes sure that the candidate does not have the injection label, INJ.

II. The second automated check is to make sure the candidate's **False Alarm Rate (FAR)** is below a set threshold limit. If its FAR is above the limit, no alert is created.

If both of these checks are passed, we have a potential real gravitational-wave event and a Preliminary Alert is created, given there is no human objection to data.

2.2. VOEvent_0: Preliminary Alert i.e. 'Pre-Alert'.

The reasons for issuing a Pre-Alert are two-fold. I. In the survey for Gamma Ray Bursts (GRBs), time coincidences are interesting and the faster we issue an alert, the quicker we can see if there were any GRBs detected electromagnetically around the same time. II. The Low-Frequency Array for radio astronomy (LOFAR) requested to be alerted since it can only buffer one minute at a time to save data.

The Pre-Alert contains only basic candidate event information. This includes: UID, Event Time, Labels, FAR, and **Instruments**. Under the Instruments header, only the instruments that recorded data leading to the candidate event are listed. H1, L1, and V1 stand for the detectors LIGO Hanford, LIGO Livingston, and VIRGO respectively.

The Pre-Alert recipients are aLIGO's MoU partners and human checkers at each instrument site listed in the VOEvent.

2.3. Skymaps.

While the Pre-Alert is being issued, there is further analysis of the candidate event to find its location. For CWBs, a link to the probability density map, (**skymap**), is added to GraCEDb right away while CBCs have to wait a little bit longer for skymap generating pipelines to run. These skymaps are FITS files whose links are contained in the VOEvent.

For CBCs, the first map that is uploaded is the Rapid Localization skymap. As the name suggests, this map rapidly calculates the probability that the candidate event could be located in a specific region in the sky. When more information about the candidate event is revealed through further analysis, a second map known as the Full Parameter Estimation skymap is uploaded.

Opening and interpreting the skymaps is discussed under its own section of this document, under Skymaps.

2.4. Data Quality (iDQ) Checks.

Meanwhile, GraCEDb software begins data quality checks to prepare for the next VOEvent. Software that searches for glitches for aLIGO is called **iDQ**. It takes the event GPS time and checks for any glitches near that time in the iDQ database. The results are then recorded numerically in GraCEDb as a real number between 0 and 1. This is the minimum False Alarm Probability, (**minFAP**). If this number is above a threshold value, then the candidate event passed the iDQ check, and there is low probability that the candidate is a glitch. The closer the the value is to 0, the more likely the candidate event was a glitch.

GraCEDb software sets the label **EM_READY** for a candidate event that has a skymap link and passes the automated iDQ check. In the case where human sign offs are part of the approval process, the label EM_READY is applied after human scientific monitors from each relevant detector site also sign off on the event. In either case, the setting of the label EM_READY will generate an Initial Localization Alert.

The human sign-off will be recorded in an uploaded text file as

$$Pass: \text{(Passed sign-off)}$$
$$Fail: \text{(Failed sign-off)}$$

with an optional comment per instrument site.

2.5. VOEvent_1: Initial Localization Alert.

aLIGO's approval process software (**Approval Processor**) is prompted when the candidate event gets the label EM_READY. When this happens, an Initial Localization Alert is generated and issued to MoU partners. This VOEvent is the first alert that has gone through some data quality validation and contains the link to a most probable location map. Human sign offs are only part of the approval process if aLIGO specifically asks for it in Approval Processor's configuration file.

Additionally, more information about the candidate event is included in the Initial Localization VOEvent. For CBCs this might include estimations of the chirp mass, ratio of reduced mass to total mass, and the maximum distance to the event. For CWBs this might include estimations of the gravitational-wave burst's peak frequency, measured duration, and fluence. This information will be resolved by various analysis pipelines running in the background as the data quality checks are conducted.

2.6. VOEvent_2, VOEvent_3, ...: Revised VOEvent Alerts.

Once another skymap is created for a candidate event and its link is uploaded to GraCEDb, the candidate event gets an additional label **PE_READY** where PE stands for Parameter Estimation. This skymap will have improved estimations on the location of the candidate event. Furthermore, if the values describing the candidate event changes, the next VOEvent that is generated will be updated with the most recent values. All VOEvents come with a link to the GraCEDb webpage to provide access to evolving candidate event statuses.

2.7. Retraction Alerts.

At any point of the validation and analysis process, if there is objection to the data, a Retraction Alert can be issued. This alert lets MoU partners know which data quality check failed and/or from which site(s) there was human objection to the candidate event.

3. Sample VOEvents with Explanation

3.1. VOEvent_0: Preliminary Alert.

3.2. VOEvent_1: Initial Localization Alert for a CBC.

The following is a sample VOEvent generated for candidate event G96195.

```xml
<?xml version="1.0" ?>
<voe:VOEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:voe="http://www.ivoa.net/xml/VOEvent/v2.0"
xsi:schemaLocation="http://www.ivoa.net/xml/VOEvent/v2.0 http://www.ivoa.net/xml/VOEvent/VOEvent-v2.0.xsd"
 version="2.0" role="test" ivorn="ivo://gwnet/G96195-1">
    <Who>
```

```
        <Date>2014-12-03T16:03:34</Date>
        <Author>
            <contactName>LIGO Scientific Collaboration and Virgo Collaboration</contactName>
        </Author>
    </Who>
    <What>
        <Param name="GraceID" dataType="string" value="G96195" ucd="meta.id" unit="">
            <Description>Identifier in the GraceDb database</Description>
        </Param>
        <Param name="AlertType" dataType="string" value="Initial" ucd="meta.version" unit="">
            <Description>VOEvent alert type</Description>
        </Param>
        <Param name="FAR" dataType="float" value="7.80321138878e-12" ucd="arith.rate;stat.falsealarm" unit="Hz">
            <Description>False alarm rate for GW candidates with this strength or greater</Description>
        </Param>
        <Param name="EventPage" dataType="string" value="https://gracedb.ligo.org/events/G96195" ucd="meta.ref.url" unit="">
            <Description>Web page for evolving status of this candidate event</Description>
        </Param>
        <Param name="Pipeline" dataType="string" value="gstlal-spiir" ucd="meta.code" unit="">
            <Description>Low-latency data analysis pipeline</Description>
        </Param>
        <Param name="Search" dataType="string" value="LowMass" ucd="meta.code" unit="">
            <Description>Low-latency search type</Description>
        </Param>
        <Param name="ChirpMass" dataType="float" value="0.912880957127" ucd="phys.mass" unit="solar mass">
            <Description>Estimated CBC chirp mass</Description>
        </Param>
        <Param name="MaxDistance" dataType="float" value="57.5933" ucd="pos.distance" unit="Mpc">
            <Description>Estimated maximum distance for CBC event</Description>
        </Param>
        <Param name="Eta" dataType="float" value="0.2484173" ucd="phys.mass;arith.factor" unit="">
            <Description>Estimated ratio of reduced mass to total mass</Description>
        </Param>
        <Group type="GW_SKYMAP" name="BAYESTAR">
```

```xml
            <Param name="skymap_png_x509" dataType="string" value="https://gracedb.ligo.org/api/events/G96195/files/skymap.png,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map image X509 protected</Description>
            </Param>
            <Param name="skymap_fits_x509" dataType="string" value="https://gracedb.ligo.org/api/events/G96195/files/skymap.fits.gz,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map FITS X509 protected</Description>
            </Param>
            <Param name="skymap_png_shib" dataType="string" value="https://gracedb.ligo.org/events/G96195/files/skymap.png,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map image Shibboleth protected</Description>
            </Param>
            <Param name="skymap_fits_shib" dataType="string" value="https://gracedb.ligo.org/events/G96195/files/skymap.fits.gz,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map FITS Shibboleth protected</Description>
            </Param>
        </Group>
    </What>
    <WhereWhen>
        <ObsDataLocation>
            <ObservatoryLocation id="LIGO Virgo"/>
            <ObservationLocation>
                <AstroCoordSystem id="UTC-FK5-GEO"/>
                <AstroCoords coord_system_id="UTC-FK5-GEO">
                    <Time>
                        <TimeInstant>
                            <ISOTime>2014-03-01T03:57:59</ISOTime>
                        </TimeInstant>
                    </Time>
                    <Position2D>
                        <Value2>
                            <C1>0.000000</C1>
                            <C2>0.000000</C2>
                        </Value2>
                        <Error2Radius>180.000000</Error2Radius>
                    </Position2D>
                </AstroCoords>
            </ObservationLocation>
        </ObsDataLocation>
    </WhereWhen>
    <How>
        <Description>L1: LIGO Livingston 4 km gravitational wave detector</Description>
```

```
79          <Description>V1: Virgo 3 km gravitational wave detector</Description>
80      </How>
81      <Why>
82          <Description>Candidate gravitational wave event identified by low-latency analysis</Description>
83      </Why>
84      <Description>Report of a candidate gravitational wave event</Description>
85 </voe:VOEvent>
```

Lines 6 through 11 (Who) state when and from where the VOEvent is sent. In this case, it is the LIGO Scientific Collaboration and Virgo Collaboration on 2014-12-03.

Lines 12 through 54 (What) state the parameters that describe the particular candidate event. In this case for a potential CBC, the parameters are: **GraceID**, **AlertType**, **FAR**, **EventPage**, **Pipeline**, **Search**, **ChirpMass**, **MaxDistance**, and **Eta**. There is also a group type **GW_SKYMAP**.

GraceID is the unique candidate event identifier in the database. The entry for it tells you that in this case, the value is the string: G96195.

AlertType tells you what type of VOEvent is issued. In this case, the value is the string 'Initial' meaning that we have included a skymap and concluded data quality checks – this is an Initial Localization Alert.

FAR is the false alarm rate for gravitational wave candidate events with the recorded strength or greater. In this case, we have the value 7.80321138878e-12 Hz which means that one out of 128,152,365,760 = 1/7.80321138878e-12 events of the recorded strength type will be a false event.

EventPage gives us the link to the candidate event's GraCEDb page which contains more information about the event and its evolving status.

Pipeline and **Search** tells you what low-latency search type resulted in finding the candidate event. In this case, it was a gstlal-spiir LowMass pipeline search.

Up to this point, all of the parameters above are common to both CBC and CWB candidate events. The other parameters are specific to potential CBCs.

ChirpMass is the estimated CBC chirp mass given in solar mass units. In our case, the chirp mass is 0.912880957127 $M_\odot$. The chirp mass is equal to $(m_1 m_2)^{3/5}/(m_1 + m_2)^{1/5}$.

MaxDistance tells us the estimated maximum distance to the candidate event in mega parsecs. In this case, the furthest the CBC could be is 57.5933 Mpc.

Eta is the estimated ratio of reduced mass, $(m_1 m_2)/(m_1 + m_2)$ to total mass, $m_1 + m_2$. In this case, this ratio is 0.2484173.

The group **GW_SKYMAP** tells us what pipeline generated the skymap. In this case, it is BAYESTAR. Under this heading, we have four representations of the same skymap. This is to convenience our MoU partners who might use different ways of extracting the skymap. This skymap gives us the probability density of the sky of where the event could come from.

Lines 55 through 76 (WhereWhen) tells us the location of the observatory as a string and where and when the observation was made.

Lines 77 through 80 (How) tells us which detectors were involved with the candidate event. In this case, it was Ligo Livingston (**L1**) and Virgo (**V1**).

3.3. VOEvent_1: Initial Localization Alert for a CWB.

The following is a sample VOEvent generated for candidate event G120903.

```xml
<?xml version="1.0" ?>
<voe:VOEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:voe="http://www.ivoa.net/xml/VOEvent/v2.0"
xsi:schemaLocation="http://www.ivoa.net/xml/VOEvent/v2.0 http://www.ivoa.net/xml/VOEvent/VOEvent-v2.0.xsd"
 version="2.0" role="test" ivorn="ivo://gwnet/G120903-1">
    <Who>
        <Date>2015-03-11T18:05:35</Date>
        <Author>
            <contactName>LIGO Scientific Collaboration and Virgo Collaboration</contactName>
        </Author>
    </Who>
    <What>
        <Param name="GraceID" dataType="string" value="G120903" ucd="meta.id" unit="">
            <Description>Identifier in the GraceDb database</Description>
        </Param>
        <Param name="AlertType" dataType="string" value="Initial" ucd="meta.version" unit="">
            <Description>VOEvent alert type</Description>
        </Param>
        <Param name="FAR" dataType="float" value="0.000386414" ucd="arith.rate;stat.falsealarm" unit="Hz">
            <Description>False alarm rate for GW candidates with this strength or greater</Description>
        </Param>
        <Param name="EventPage" dataType="string" value="https://gracedb.ligo.org/events/G120903" ucd="meta.ref.url" unit="">
            <Description>Web page for evolving status of this candidate event</Description>
        </Param>
```

```
        <Param name="Pipeline" dataType="string" value="CWB" ucd="meta.code" unit="">
            <Description>Low-latency data analysis pipeline</Description>
        </Param>
        <Param name="Search" dataType="string" value="AllSky" ucd="meta.code" unit="">
            <Description>Low-latency search type</Description>
        </Param>
        <Param name="PeakFreq" dataType="float" value="1497.077393" ucd="gw.frequency" unit="Hz">
            <Description>Peak frequency of GW burst signal</Description>
        </Param>
        <Param name="Duration" dataType="string" value="0.042969" ucd="time.duration" unit="s">
            <Description>Measured duration of GW burst signal</Description>
        </Param>
        <Param name="Fluence" dataType="float" value="1.06665771962" ucd="gw.fluence" unit="erg/cm^2">
            <Description>Estimated fluence of GW burst signal</Description>
        </Param>
        <Group type="GW_SKYMAP" name="BAYESTAR">
            <Param name="skymap_png_x509" dataType="string" value="https://gracedb.ligo.org/api/events/G120903/files/skymap.png,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map image X509 protected</Description>
            </Param>
            <Param name="skymap_fits_x509" dataType="string" value="https://gracedb.ligo.org/api/events/G120903/files/skymap.fits.gz,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map FITS X509 protected</Description>
            </Param>
            <Param name="skymap_png_shib" dataType="string" value="https://gracedb.ligo.org/events/G120903/files/skymap.png,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map image Shibboleth protected</Description>
            </Param>
            <Param name="skymap_fits_shib" dataType="string" value="https://gracedb.ligo.org/events/G120903/files/skymap.fits.gz,0" ucd="meta.ref.url" unit="">
                <Description>Sky Map FITS Shibboleth protected</Description>
            </Param>
        </Group>
    </What>
    <WhereWhen>
        <ObsDataLocation>
            <ObservatoryLocation id="LIGO Virgo"/>
```

```
            <ObservationLocation>
                <AstroCoordSystem id="UTC-FK5-GEO"/>
                <AstroCoords coord_system_id="UTC-FK5-GEO">
                    <Time>
                        <TimeInstant>
                            <ISOTime>2014-05-31T01:36:54</ISOTime>
                        </TimeInstant>
                    </Time>
                    <Position2D>
                        <Value2>
                            <C1>0.000000</C1>
                            <C2>0.000000</C2>
                        </Value2>
                        <Error2Radius>180.000000</Error2Radius>
                    </Position2D>
                </AstroCoords>
            </ObservationLocation>
        </ObsDataLocation>
    </WhereWhen>
    <How>
        <Description>H1: LIGO Hanford 4 km gravitational wave detector<
            /Description>
        <Description>L1: LIGO Livingston 4 km gravitational wave detector<
            /Description>
    </How>
    <Why>
        <Description>Candidate gravitational wave event identified by low-
            latency analysis</Description>
    </Why>
    <Description>Report of a candidate gravitational wave event<
        /Description>
</voe:VOEvent>
```

This VOEvent contains similar information except under the What heading, different parameters describe CWBs: **PeakFreq**, **Duration**, and **Fluence**.

PeakFreq tells us in Hz the peak frequency of the gravitational wave burst signal. In this case, it was 1,497.077393 Hz.

Duration tells us the measured duration of the gravitational wave burst signal in seconds. In this case, the signal lasted 0.042969 seconds.

Fluence is an estimate of how much energy was emitted per area due to the gravitational wave burst. It is actually calculated from the values for strain and central frequency for the burst given from GraCEDb. For this candidate event, it is 1.06665771962 erg/cm^2.

4. Skymaps

Skymaps are provided for both CBCs and CWBs. **There are some questions about the skymaps for CWBs! Some candidate events (for example, G75616) do not have a skymap and others, like G120903, the skymap file link is in an obscure location...** (`https://ldas-jobs.ligo.caltech.edu/~waveburst/ER5_ONLINE_L1H1_noinj_1/JOBS/108553/1085535380-1085535440/OUTPUT_CED/ced_1085535380_60_1085535380-1085535440_slag0_lag0_1_job1/L1H1_1085535430.594_1085535430.594/skyprobcc.fits`) **Is there a way to change this?**

The data contained in the skymaps is pixelized using HEALPix (Hierarchical Equal Area isoLatitude Pixelization). This means that the entire spherical surface that is the sky is translated to an oval 2D image where each pixel in the image covers the same surface area as every other pixel. In our case, we use a Mollweide projection and the color of the pixel determines the likelihood that the candidate event source is located there and there will be a color-coded legend at the bottom of the skymaps to help interpret the maps.

Equally important is getting the coordinates of a given pixel. First, the candidate event skymaps use Celestial (Equatorial) coordinates. On the vertical axis is Declination on the horizontal axis is Right Ascension. In the example below, you will see how to get the coordinates explicitly.

4.1. Sample skymap viewing and interpreting.

Under the sample CBC VOEvent created above for candidate G95149, you can see in line 16 that the link to the skymap is: `https://gracedb.ligo.org/gracedb-files/G95149/private/skymap.fits.gz`.

MoU partners can now save this file from GraCEDb using the link. The data is in the form of a 1 dimensional array whose length is the total number of pixels. Each pixel has an associated probability. Because the majority of pixels in a skymap have probability equal to 0, the skymaps are zipped, hence the 'gz' in the file name. The maps do not have to be unzipped explicitly but if desired, they can be.

There are several ways to open the skymap files. You can install HEALPix itself but that is unnecessary if you use an associated package like healpy (`https://pypi.python.org/pypi/healpy`) for Python. To install healpy there are several options. For Anaconda/Canopy users you can simply run:

```
pip install healpy
```

which will install healpy.

Likewise, Linux users with a stock Python interpreter can type in the shell:

```
pip install –user healpy.
```

Now that Python has the healpy package, here is sample Python code used to open the .fits file and make sense of the data.

```python
# Python imports
# Parts taken from Leo Singer’s bayestar_plot_allsky

import healpy as hp
import numpy as np
import matplotlib.pyplot as plt

# Read in the downloaded skymap file from GraceDb
# Image data is a 1D array of values
hpx = hp.read_map(’bayestar.fits.gz’)

# Create an image of the probability distribution using the Mollweide projection
# The Bayestar skymap is in Celestial (Equatorial) coordinates
skymap = hp.mollview(hpx, coord=’C’, title=’Bayestar Probability Skymap’, unit=’Probability’)

# Draw in graticule with 15 degrees between parallels and 30 degrees between meridians
hp.graticule(dpar=15, dmer=30)
plt.savefig(’skymap.png’)

# We can also create an image of the probability density distribution, which shows you the probability per square degree
# From the length of the healpix array we can find the number of square degrees per pixel
# Dividing the probability per pixel skymap by this number will give us the probability per square degree skymap
# There are two ways to do this -- Method 1
sky_area = 4*180**2/np.pi # This is the number of square degrees in an entire sphere
npix = len(hpx) # The number of pixels in the skymap
squaredegreesperpixel = sky_area/npix # The number of square degrees per pixel

# Method 2
nside = hp.npix2nside(npix) # The lateral resolution, npix = 12*nside**2 where nside is a power of 2
squaredegreesperpixel = hp.nside2pixarea(nside, degrees=True)

# Converting our skymap from probability to probability per square degree
probperdeg2 = hpx/squaredegreesperpixel
skymap2 = hp.mollview(probperdeg2, coord =’C’, title=’Bayestar Probability Density Skymap’, unit=’Probability per square degree’)
hp.graticule(dpar=15, dmer=30)
plt.savefig(’skymap2.png’)
```

```

# Now below are some basic tools for working with the skymaps.
# We assume all the angle parameters in the functions are given as degrees.
# Our functions will also returns angles in degrees.

# 1. Finding the coordinates of a given pixel in (RA, Dec) in degrees
def coordsforpix(nside, ipix):
    theta, phi = hp.pix2ang(nside, ipix) # Currently theta and phi are in radians
    ra = np.rad2deg(phi) # Right ascension is the same as the azimuthal angle
    dec = np.rad2deg(0.5*np.pi - theta) # Declination is the complement of the polar angle
    return (ra, dec)

# 2. Finding the pixel given (RA, Dec) in degrees
def pixforcoords(ra, dec):
    theta = 0.5*np.pi - np.deg2rad(dec)
    phi = np.deg2rad(ra)
    ipix = hp.ang2pix(nside, theta, phi)
    return ipix

# 3. Finding the probability and the probability per degree squared at a given pixel
probatipix = hpx[ipix]
probperdeg2atipix = probperdeg2[ipix]

# 4. Finding the highest probability pixel, the probability there, and the (RA, Dec) coordinates of this pixel
ipix_max = np.argmax(hpx)
hpx[ipix_max]
coordsforpix(nside, ipix_max)

# 5. Finding the probability contained within a cone (the probability contained in a circle in the sky)
# healpy's query_disc function returns a list of pixels located inside a cone lying along a 3-D vector with a certain base radius in radians
def probincone(ra, dec, radius):
    # As written, ra, dec, and radius are all in degrees
    # First we need to determine the vector to this point in the sky, so determine theta and phi
    # They will be in radians
    theta = 0.5*np.pi - np.deg2rad(dec)
    phi = np.deg2rad(ra)
    vec = hp.ang2vec(theta, phi)
    # Now convert the radius to radians
    radius = np.deg2rad(radius)
    # List of pixels inside the base circle
    pixes = hp.query_disc(nside, vec, radius)
```

```
    # Adding up the probability at each of these pixels
    prob = hpx[pixes].sum()
    return prob

# 6. Finding the probability contained in a polygon whose vertices are a list of pixels
# healpy's query_polygon function returns a list of pixels located inside a polygon
# The polygon is specified by its vertices given as a list of vectors
# To get the vector to a particular pixel in the sky, use the healpy function pix2vec
def probinpolygon(listofpixels):
    listofvectors = []
    for pix in listofpixels:
        vecforpix = hp.pix2vec(nside, pix)
        listofvectors.append(vecforpix)
    # List of pixels inside polygon
    pixes = hp.query_polygon(nside, listofvectors)
    # Adding up the probability at each of these pixels
    prob = hpx[pixes].sum()
    return prob

# Note that similar functions can be created for 5 and 6 to determine the probability per square degree contained in a circle or polygon in the sky
# To do that we substitue probperdeg2 for hpx
```

The number of pixels is expressed by the parameter **npix**. The resolution is expressed by the parameter **nside** which must be a power of 2 and can be calculated from npix as $\text{npix} = 12 * \text{nside}^2$.

4.2. The Skymap Viewer Tool. There is also a tool called the Skymap Viewer installed on GraCEDb that will provide another view of the candidate event. For example, for candidate event T125706, you could go to the url `https://gracedb.ligo.org/skmapViewer#T125706` to see this. **The link isn't working right now.**

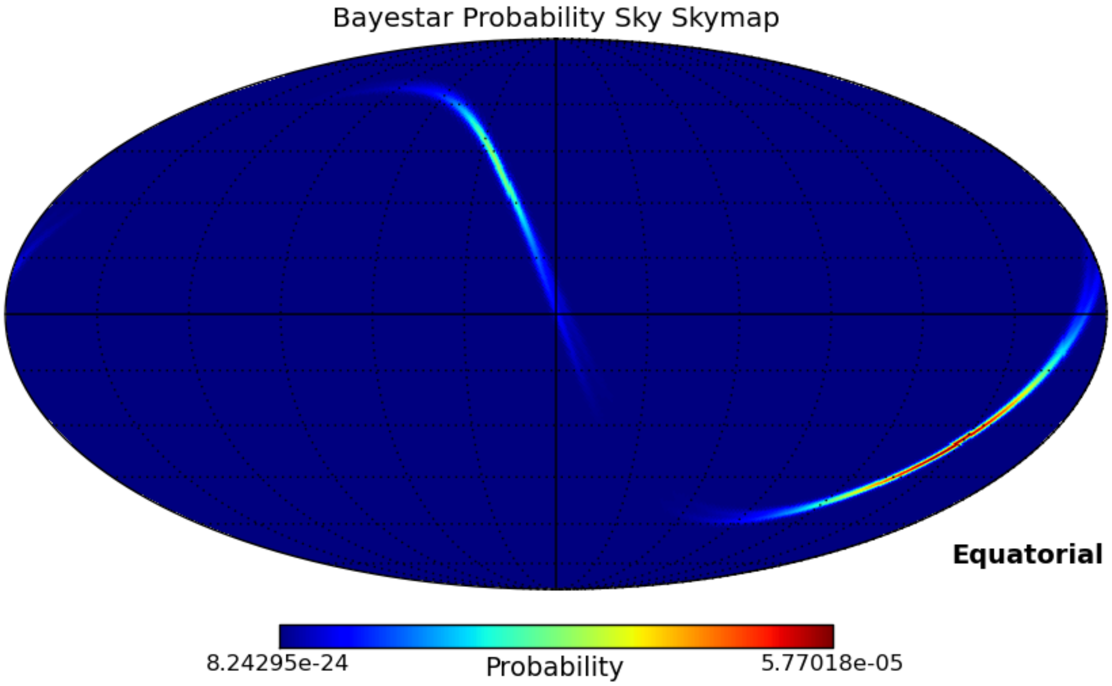

Figure 1. Visualization of the HEALPix probability skymap for T125706

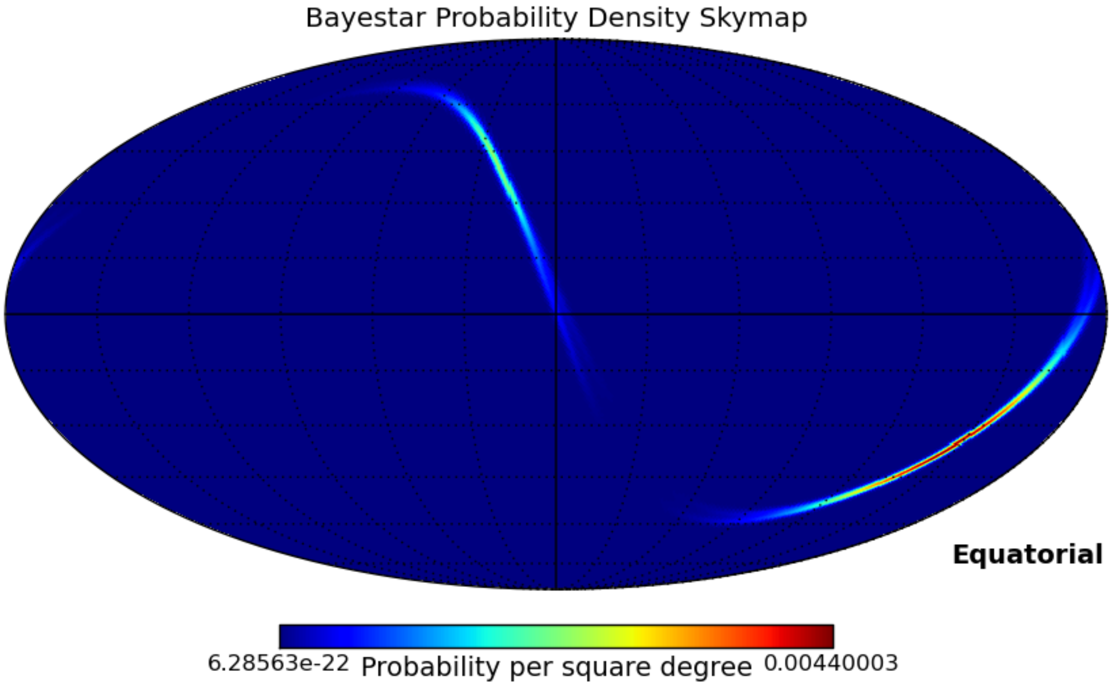

Figure 2. Visualization of the HEALPix probability density skymap for T125706