Overview on the EM-followUP of Gravitational Wave Events

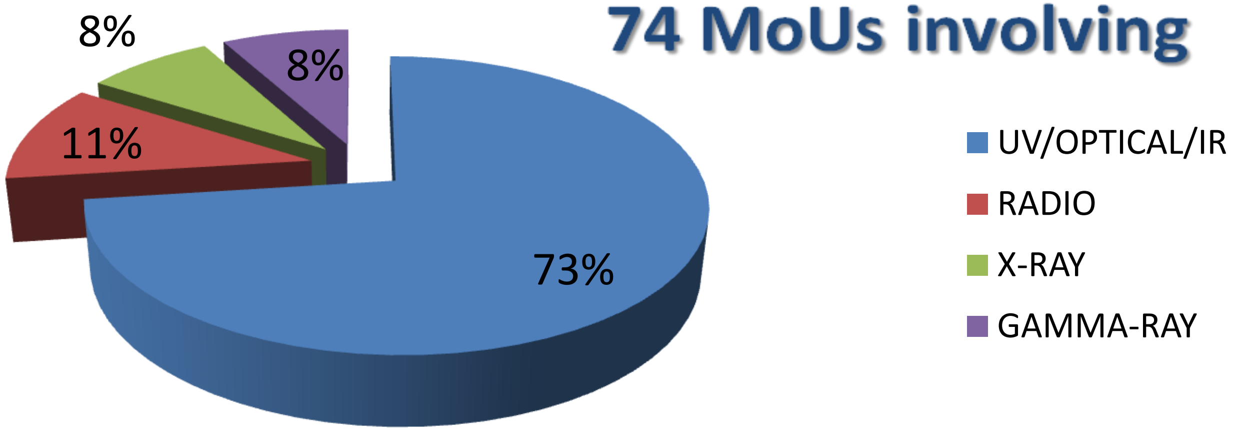
G. Greco, E. Chassande-Mottin,

M. Branchesi, G. Stratta and many others

LSC

VIRGO

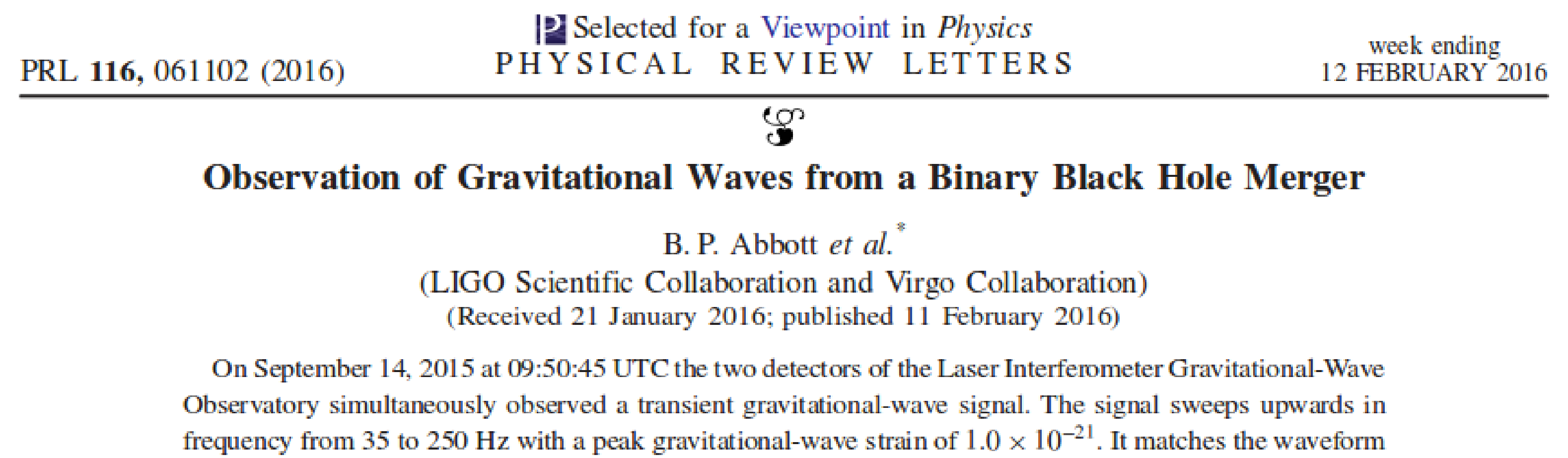
LVC GW-EM follow-up program



160 instruments covering the full spectrum from radio to very high-energy gamma-rays.

63 teams of astronomers were ready to observe during O1.

After four GW events have been published, further event candidates <u>with high confidence</u> will be shared immediately with the entire astronomy community, while lower-significance candidates will continue to be shared promptly only with partners who have signed an MoU.



Selected for a Viewpoint in *Physics*

PRL **116,** 061102 (2016) PHYSICAL REVIEW LETTERS week ending 12 FEBRUARY 2016

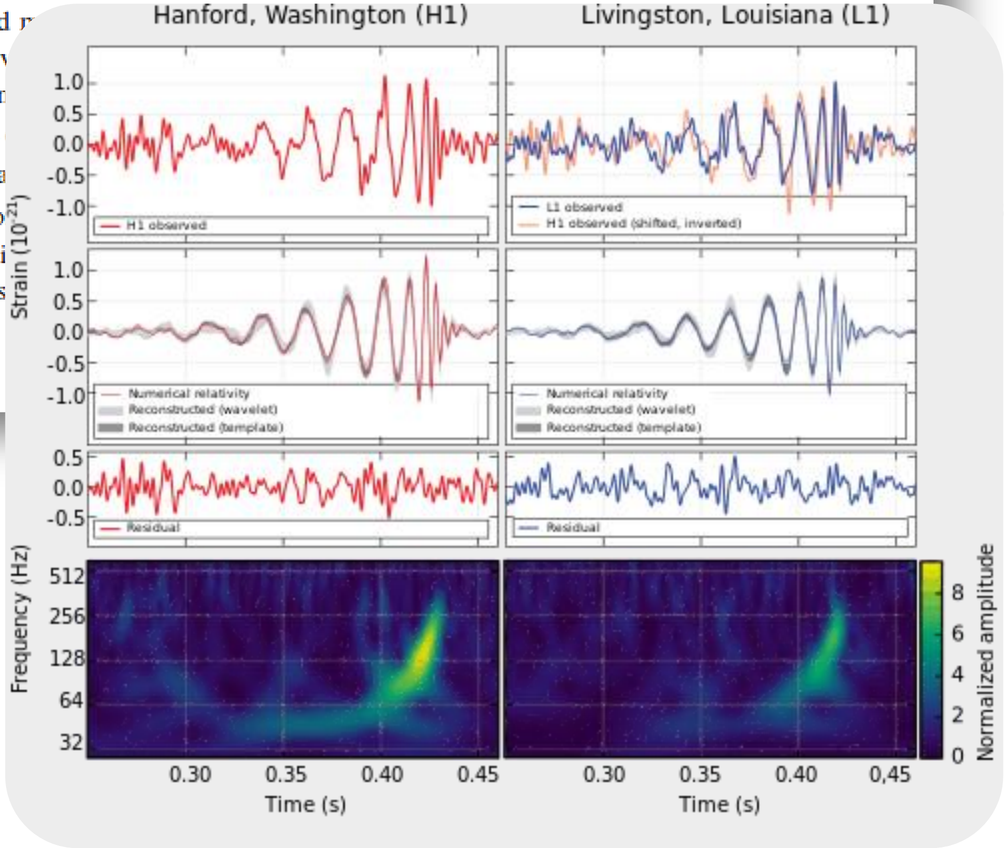
Observation of Gravitational Waves from a Binary Black Hole Merger

B. P. Abbott *et al.*[*]
(LIGO Scientific Collaboration and Virgo Collaboration)

(Received 21 January 2016; published 11 February 2016)

On September 14, 2015 at 09:50:45 UTC the two detectors of the Laser Interferometer Gravitational-Wave Observatory simultaneously observed a transient gravitational-wave signal. The signal sweeps upwards in frequency from 35 to 250 Hz with a peak gravitational-wave strain of 1.0×10^{-21}. It matches the waveform
predicted by general relativity for the inspiral and
resulting single black hole. The signal was obser
false alarm rate estimated to be less than 1 even
than 5.1σ. The source lies at a luminosity distance
In the source frame, the initial black hole masses a
$62^{+4}_{-4} M_\odot$, with $3.0^{+0.5}_{-0.5} M_\odot c^2$ radiated in gravitatio
These observations demonstrate the existence of bi
detection of gravitational waves and the first obs

DOI: 10.1103/PhysRevLett.116.061102



The Birth of the Gravitational Wave Astronomy

GW150914: first detection of gravitational waves!

On September 14, 2015 09:50:45 UTC the Advanced LIGOs detected the GW signal GW150914, originating from the coalescence of a binary black hole system.

Abbott et al. 2016, PhRvL, 116

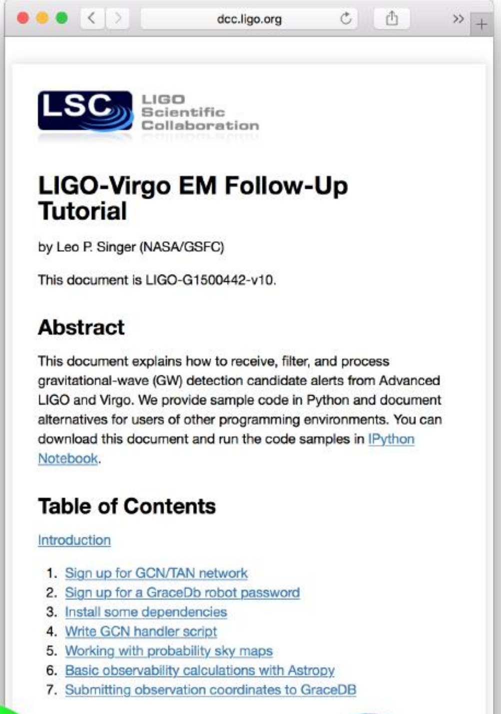
- Clear signal observed in coincidence by two LIGO detectors.
- The source is the merger of two stellar mass black holes.

• total mass:	65 Msun
• primary black hole:	32 Msun to 41 Msun
• secondary black hole:	25 Msun to 33 Msun
• remnant black hole:	62 Msun
• redshift:	0.054 to 0.136

Provides the first robust confirmation that:

- "Heavy" stellar-mass BHs exist
- Binary BHs (BBH) are formed in nature
- BBHs inspiral and merge within the age of the Universe

https://gw-astronomy.org/wiki/LV_EM/TechInfo



LIGO-Virgo EM Follow-Up Tutorial

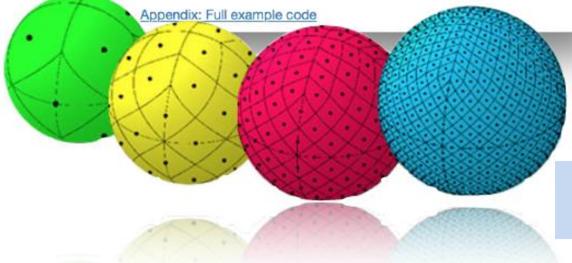
by Leo P. Singer (NASA/GSFC)

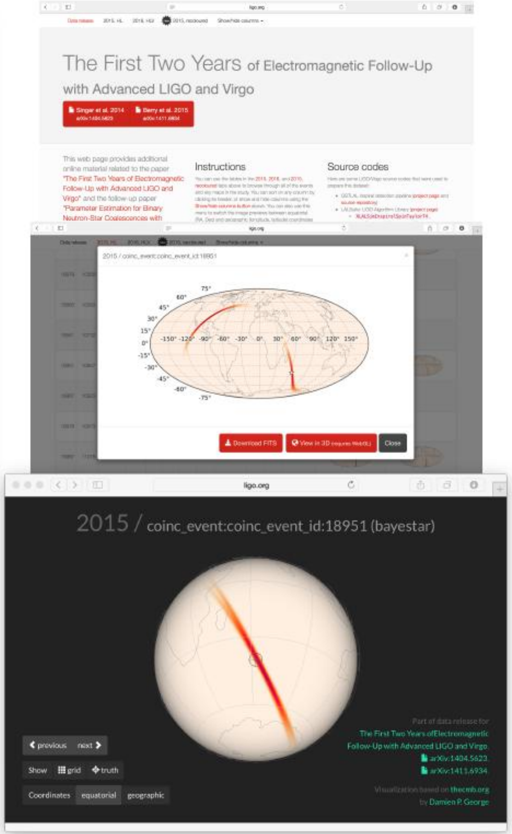
This document is LIGO-G1500442-v10.

Abstract

This document explains how to receive, filter, and process gravitational-wave (GW) detection candidate alerts from Advanced LIGO and Virgo. We provide sample code in Python and document alternatives for users of other programming environments. You can download this document and run the code samples in IPython Notebook.

Table of Contents





The First Two Years of Electromagnetic Follow-Up with Advanced LIGO and Virgo

Singer et al. 2014 arXiv:1404.5623
Berry et al. 2015 arXiv:1411.6934

This web page provides additional online material related to the paper "The First Two Years of Electromagnetic Follow-Up with Advanced LIGO and Virgo" and the follow-up paper "Parameter Estimation for Binary Neutron-Star Coalescences with

Instructions

Source codes

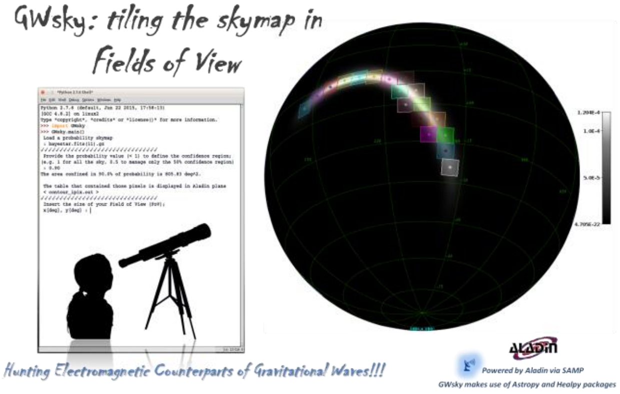
2015 / coinc_event:coinc_event_id:18951

2015 / coinc_event:coinc_event_id:18951 (bayestar)

Part of data release for The First Two Years of Electromagnetic Follow-Up with Advanced LIGO and Virgo
arXiv:1404.5623
arXiv:1411.6934

Visualization based on thecmb.org by Damien P. George

Singer+ 2014 (arXiv:1404.5623)
Berry+ 2015 (arXiv:1411.6934)
Essick+ 2015 (arXiv:1409.2435)
LVC+ 2016 (arXiv:1304.0670)



GWsky: tiling the skymap in Fields of View

Hunting Electromagnetic Counterparts of Gravitational Waves!!!

Powered by Aladin via SAMP

GWsky makes use of Astropy and Healpy packages

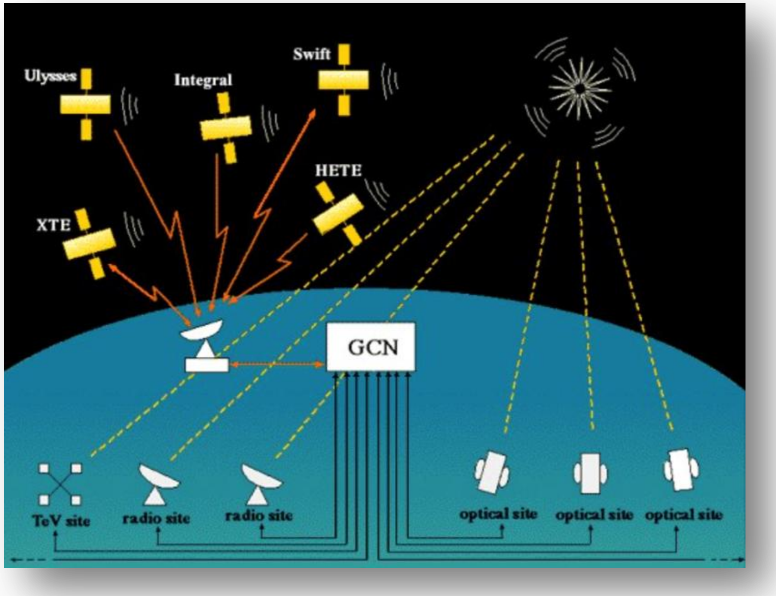
Tools: Astropy, Healpy, PyGCN, GWsky , mocpy

This wiki supports efforts by astronomers to follow up and search for electromagnetic counterparts to gravitational wave event candidates identified by LIGO and Virgo.

LVC GCN Notice/Circulars

A. Restricted to the MOU astronomer partners until the publication of the event.

B. LVC GCN notices do not contain a position (RA, Dec, error radius) instead they point to an URL to a FITS file containing a probability sky map in the HEALPix all-sky projection.



Gamma-ray Coordinates Network/Transient Astronomy Network (GCN/TAN)

The LIGO-Virgo data are analyzed in real time to search for GW transients (see next slide).

For each detection candidate, a series of alerts are produced and distributed (machine-readable GCN Notice).

EM-followUP parterns communicate the results of observations via short bullettins and GCN Circulars.

O1 low-latency pipelines configuration

- **cWB: Coherent WaveBurst**

  un-modeled GW bursts

  Klimenko et al. 2016, Phys. Rev. D 93, 042004

- **oLIB: Omicron + LALInference**

  un-modeled GW bursts

  Lynch et al. 2015, LIGO-P150022

- **GSTLAL: Gstreamer + LAL**

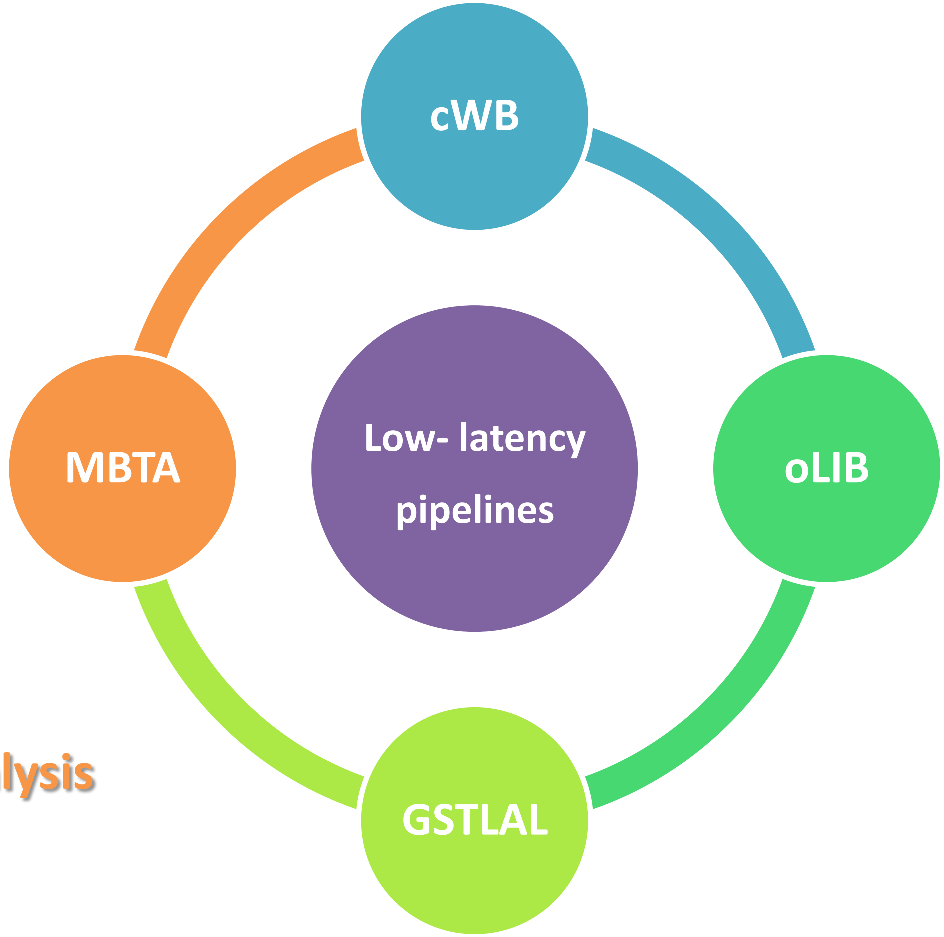
  NS binary mergers

  Cannon et al. 2012, ApJ, 748, 136

- **MBTA: Multi-Band Template Analysis**

  NS binary mergers

  Adams et al. 2015, arXiv:1512.02864



All four detection pipelines report candidates within a few minutes of data acquisition.

They continually search for transients that are coincident in the two detectors within 10 ms.

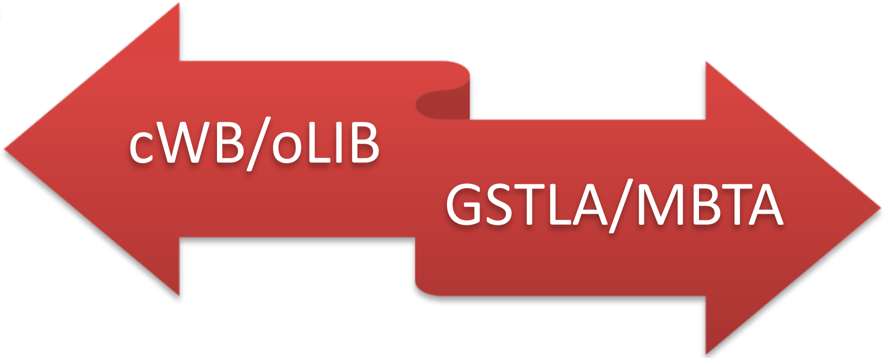
Alert of GW 150914

LIGO Calibration was complete by September 12 and O1 was scheduled to begin on September 18.

On 2015 September 14 at 09:50:45

Candidate

- **S/N = 23.45**
- **FAR < 0.371 yr^{-1}**



cWB/oLIB

GSTLA/MBTA

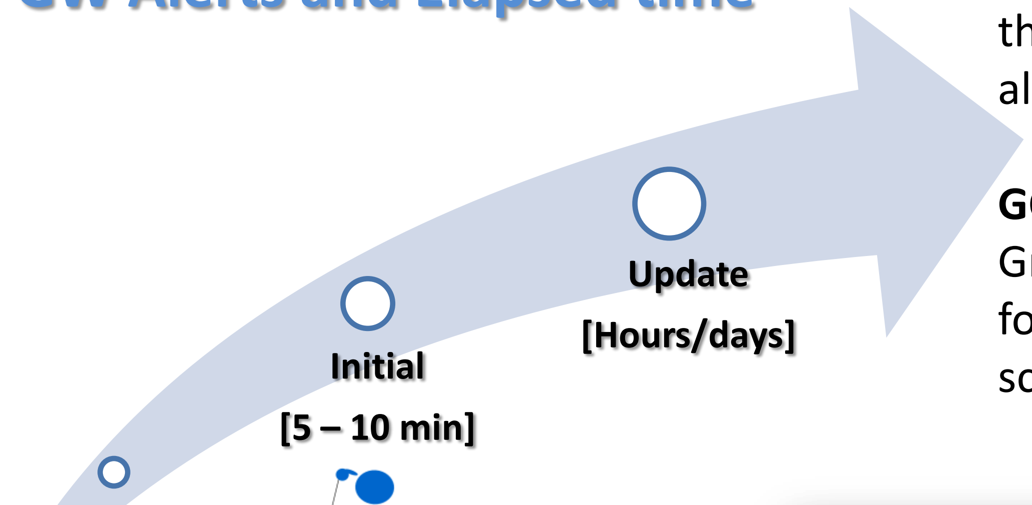
No Candidate

- **Ruling out a BNS or NSBH merger**

The data were re-analyzed offline with two independent matched-filter searches using a template bank which includes both NS binary and BBH merges. The waveform was confirmed to be consistent with a BBH merger and this information was shared with observers about 3 weeks after the event (GCN 18388).

LVC GCN notices

GW Alerts and Elapsed time



Preliminary

[3 – 5 min]

Initial

[5 – 10 min]

Update

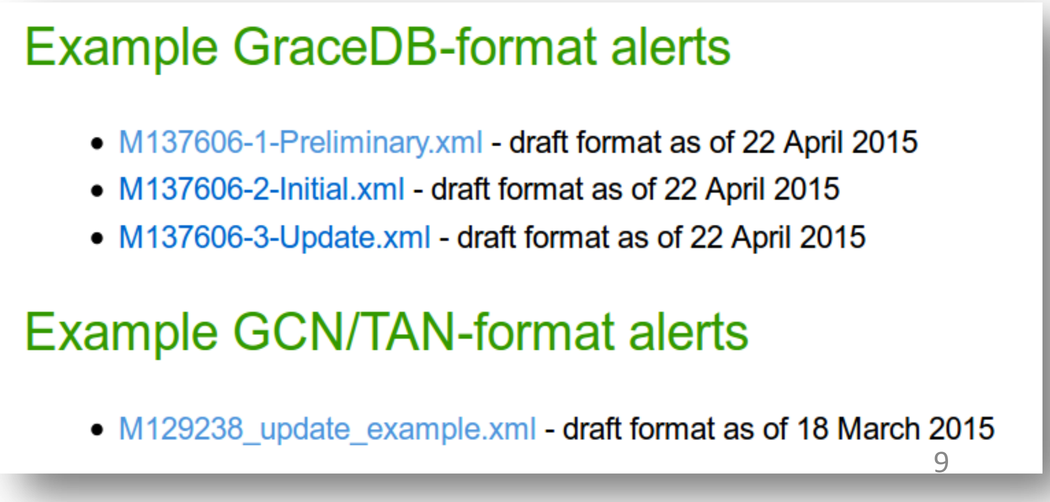
[Hours/days]

Human Event validation

Retraction at any stage

GraceDB produces one VOEvent format and makes it directly available through the GraceDB web page for the event, and also through a certain query syntax.

GCN/TAN receives that format from GraceDB but translates it into a different format which follows the GCN/TAN schema.

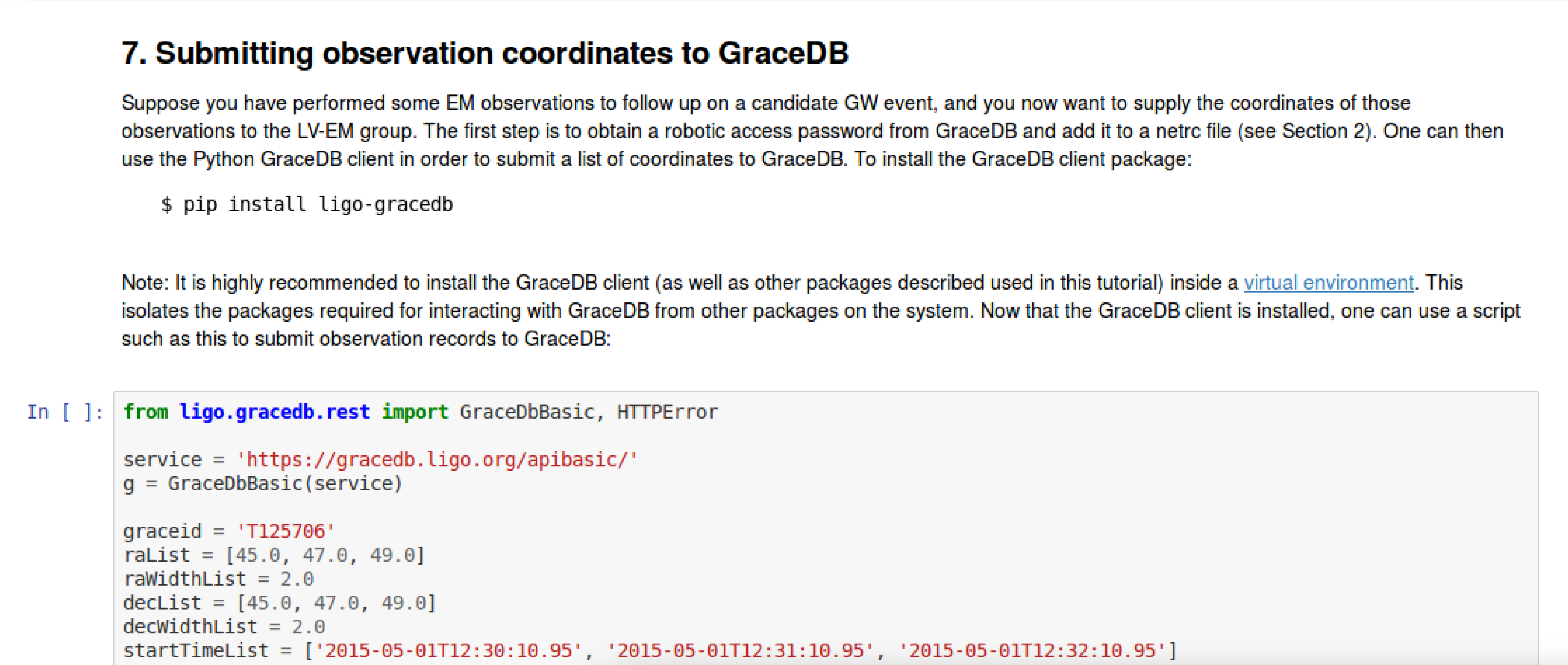


Example GraceDB-format alerts

- M137606-1-Preliminary.xml - draft format as of 22 April 2015
- M137606-2-Initial.xml - draft format as of 22 April 2015
- M137606-3-Update.xml - draft format as of 22 April 2015

Example GCN/TAN-format alerts

- M129238_update_example.xml - draft format as of 18 March 2015



7. Submitting observation coordinates to GraceDB

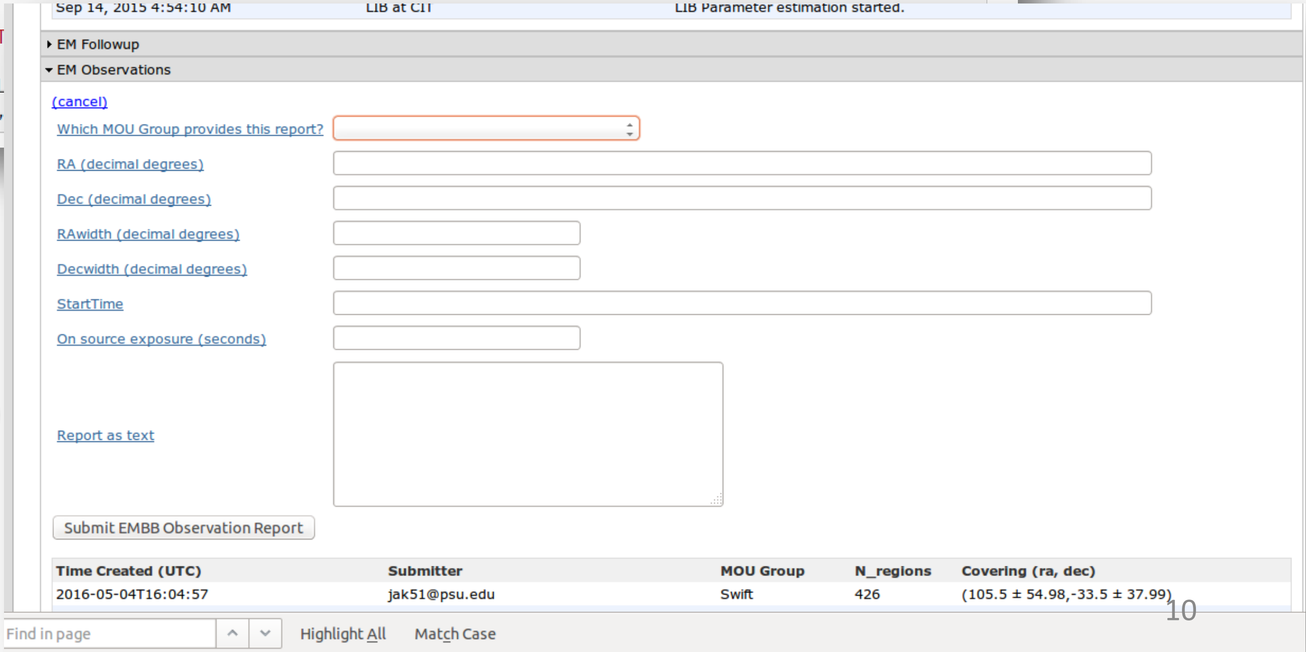
Suppose you have performed some EM observations to follow up on a candidate GW event, and you now want to supply the coordinates of those observations to the LV-EM group. The first step is to obtain a robotic access password from GraceDB and add it to a netrc file (see Section 2). One can then use the Python GraceDB client in order to submit a list of coordinates to GraceDB. To install the GraceDB client package:

```
$ pip install ligo-gracedb
```

Note: It is highly recommended to install the GraceDB client (as well as other packages described used in this tutorial) inside a virtual environment. This isolates the packages required for interacting with GraceDB from other packages on the system. Now that the GraceDB client is installed, one can use a script such as this to submit observation records to GraceDB:

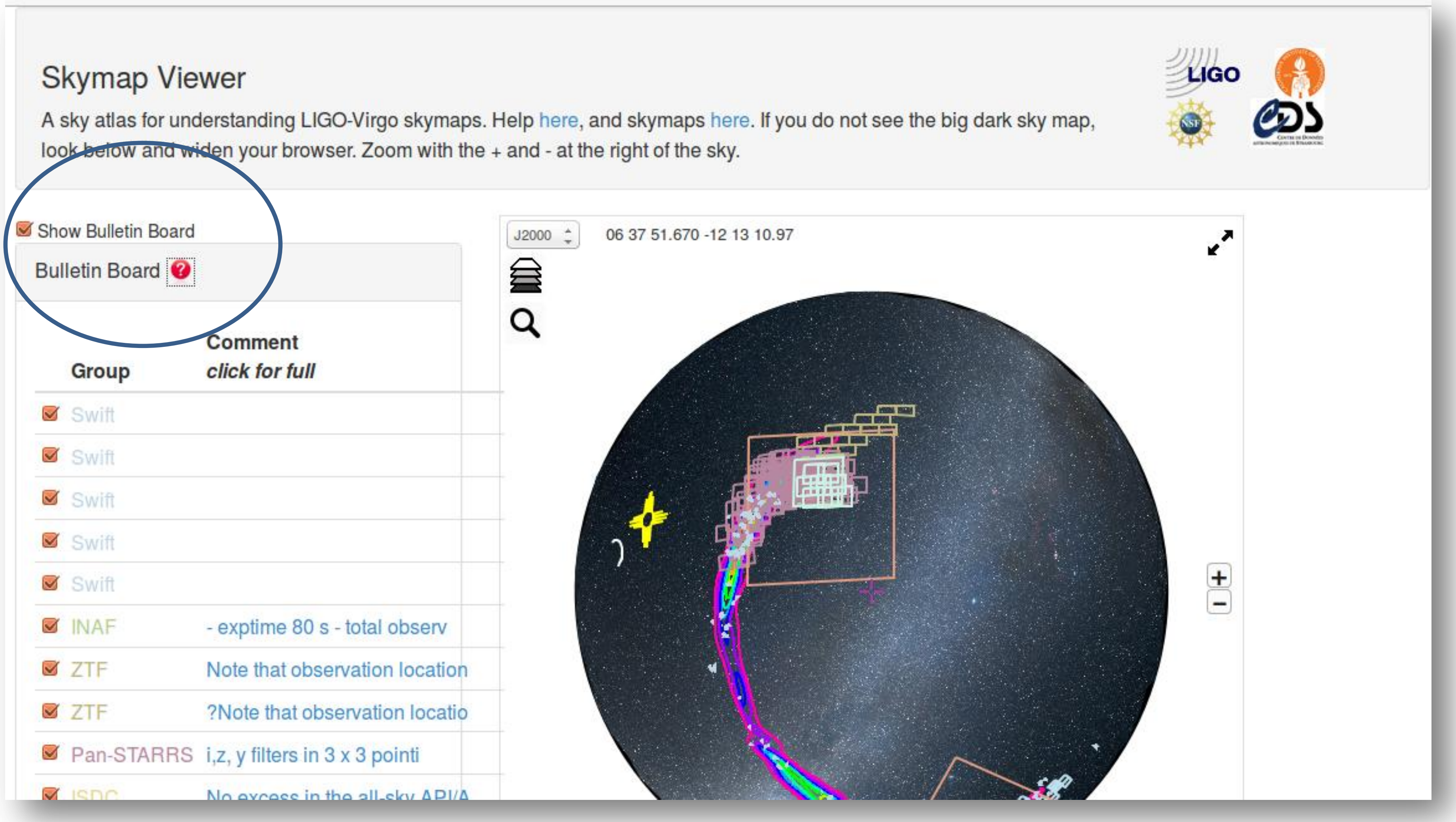
```
In [ ]: from ligo.gracedb.rest import GraceDbBasic, HTTPError

        service = 'https://gracedb.ligo.org/apibasic/'
        g = GraceDbBasic(service)

        graceid = 'T125706'
        raList = [45.0, 47.0, 49.0]
        raWidthList = 2.0
        decList = [45.0, 47.0, 49.0]
        decWidthList = 2.0
        startTimeList = ['2015-05-01T12:30:10.95', '2015-05-01T12:31:10.95', '2015-05-01T12:32:10.95']
        durationList = 100.0
        comment = 'Some text comment goes here. T

        g.writeEMObservation(graceid, 'Test', raL
            decList, decWidthList, startTimeList,
```

As agreed in MoU **within 12 hours** the observations must be reported in **GraCEDb** using a GUI (EMBB) or ligo-gracedb python package.



Sep 14, 2015 4:54:10 AM LIB at CIT LIB Parameter estimation started.

Time Created (UTC)	Submitter	MOU Group	N_regions	Covering (ra, dec)
2016-05-04T16:04:57	jak51@psu.edu	Swift	426	(105.5 ± 54.98,-33.5 ± 37.99)



Skymap Viewer

A sky atlas for understanding LIGO-Virgo skymaps. Help here, and skymaps here. If you do not see the big dark sky map, look below and widen your browser. Zoom with the + and - at the right of the sky.

☑ Show Bulletin Board

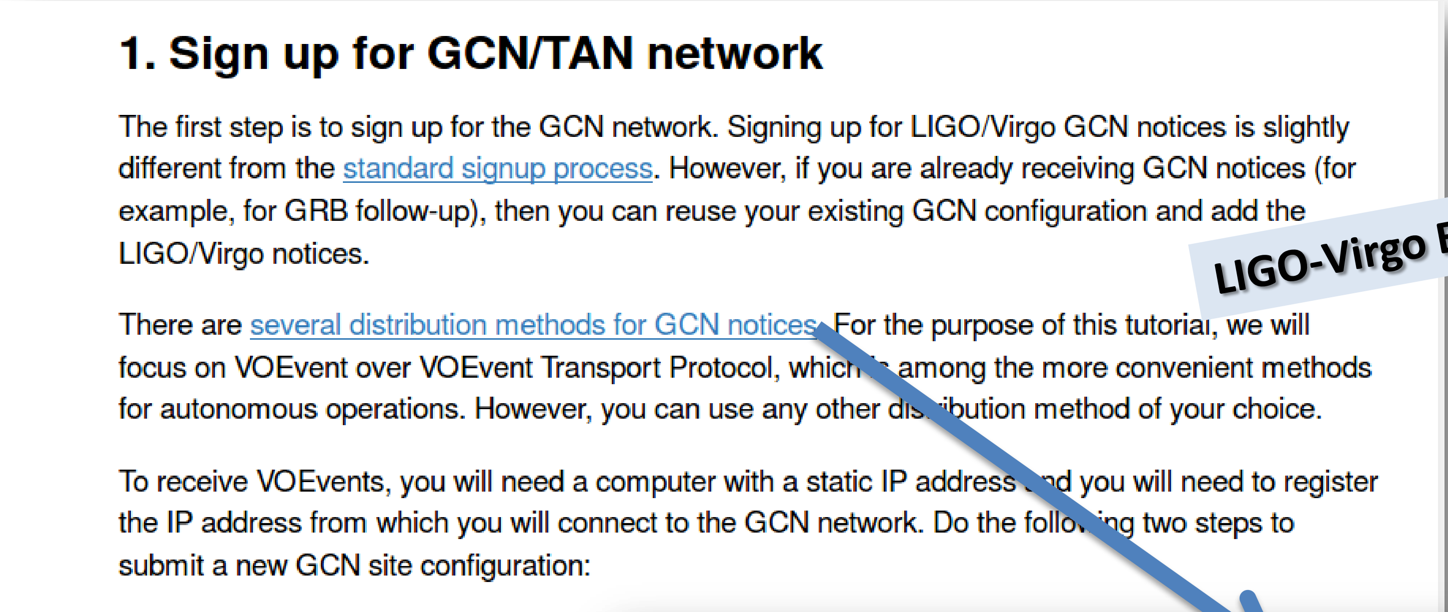
Bulletin Board

	Group	Comment *click for full*
☑	Swift	
☑	Swift	
☑	Swift	
☑	Swift	
☑	Swift	
☑	INAF	- exptime 80 s - total observ
☑	ZTF	Note that observation location
☑	ZTF	?Note that observation locatio
☑	Pan-STARRS	i,z, y filters in 3 x 3 pointi
☑	ISDC	No excess in the all-sky API/A

J2000

06 37 51.670 -12 13 10.97

LIGO-Virgo EM Follow-Up Tutorial



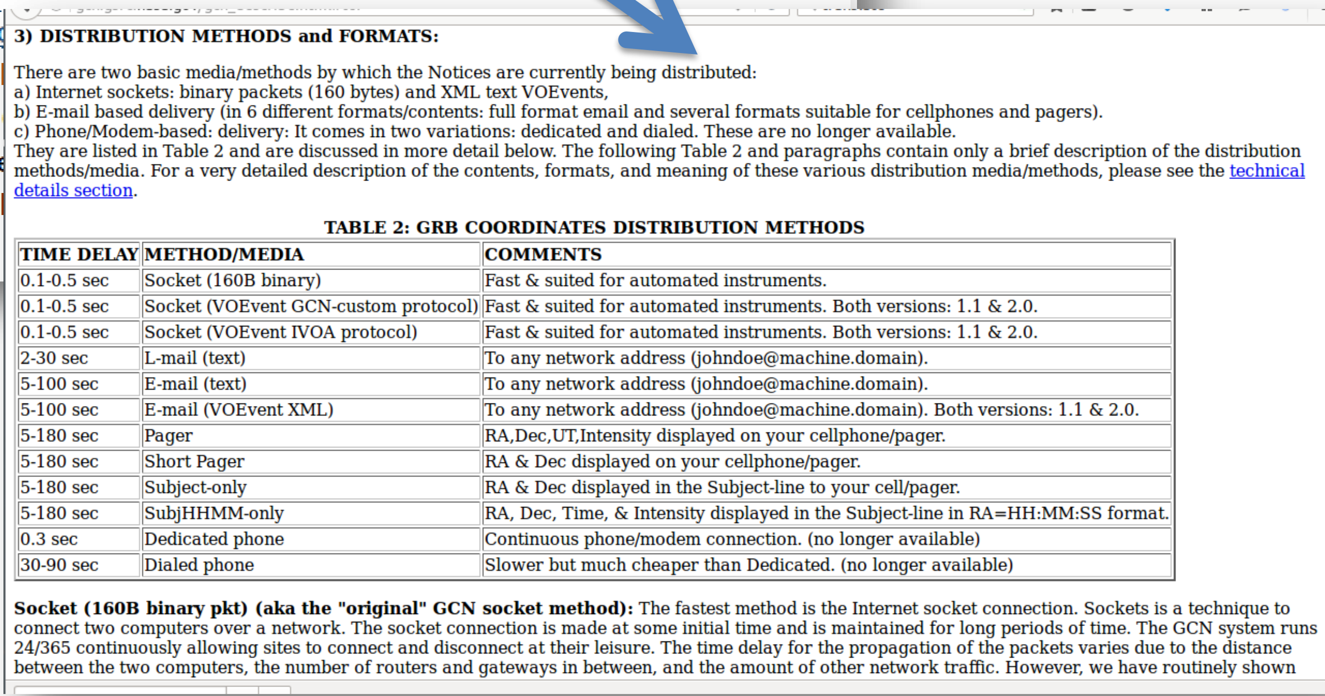
1. Sign up for GCN/TAN network

The first step is to sign up for the GCN network. Signing up for LIGO/Virgo GCN notices is slightly different from the standard signup process. However, if you are already receiving GCN notices (for example, for GRB follow-up), then you can reuse your existing GCN configuration and add the LIGO/Virgo notices.

There are several distribution methods for GCN notices. For the purpose of this tutorial, we will focus on VOEvent over VOEvent Transport Protocol, which is among the more convenient methods for autonomous operations. However, you can use any other distribution method of your choice.

To receive VOEvents, you will need a computer with a static IP address and you will need to register the IP address from which you will connect to the GCN network. Do the following two steps to submit a new GCN site configuration:

1. Go to http://gcn.gsfc.nasa.gov/lv
   "Send LVC Notice request to GC
2. If you are registering as a new G
   an existing site configuration), the
   resulting form. Select the VOEVE



3) DISTRIBUTION METHODS and FORMATS:

There are two basic media/methods by which the Notices are currently being distributed:
a) Internet sockets: binary packets (160 bytes) and XML text VOEvents,
b) E-mail based delivery (in 6 different formats/contents: full format email and several formats suitable for cellphones and pagers).
c) Phone/Modem-based: delivery: It comes in two variations: dedicated and dialed. These are no longer available.
They are listed in Table 2 and are discussed in more detail below. The following Table 2 and paragraphs contain only a brief description of the distribution methods/media. For a very detailed description of the contents, formats, and meaning of these various distribution media/methods, please see the technical details section.

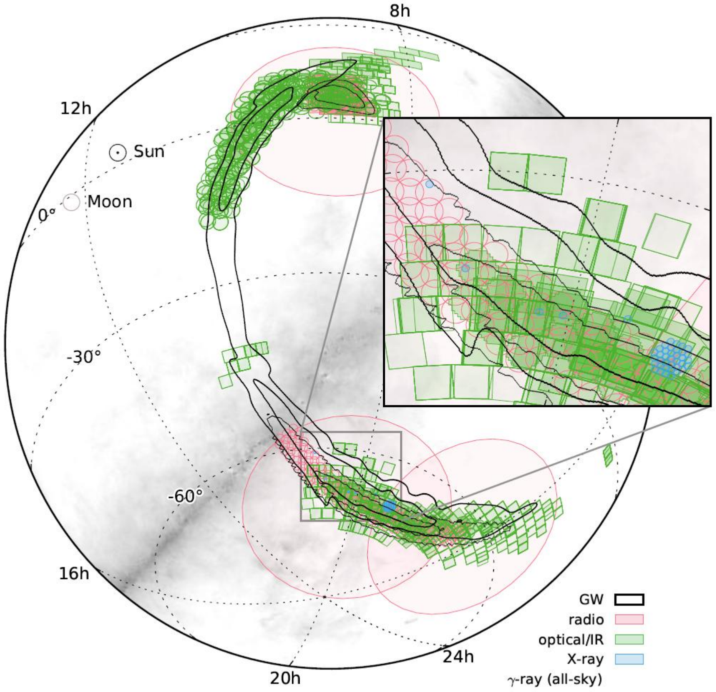
TABLE 2: GRB COORDINATES DISTRIBUTION METHODS

TIME DELAY	METHOD/MEDIA	COMMENTS
0.1-0.5 sec	Socket (160B binary)	Fast & suited for automated instruments.
0.1-0.5 sec	Socket (VOEvent GCN-custom protocol)	Fast & suited for automated instruments. Both versions: 1.1 & 2.0.
0.1-0.5 sec	Socket (VOEvent IVOA protocol)	Fast & suited for automated instruments. Both versions: 1.1 & 2.0.
2-30 sec	L-mail (text)	To any network address (johndoe@machine.domain).
5-100 sec	E-mail (text)	To any network address (johndoe@machine.domain).
5-100 sec	E-mail (VOEvent XML)	To any network address (johndoe@machine.domain). Both versions: 1.1 & 2.0.
5-180 sec	Pager	RA,Dec,UT,Intensity displayed on your cellphone/pager.
5-180 sec	Short Pager	RA & Dec displayed on your cellphone/pager.
5-180 sec	Subject-only	RA & Dec displayed in the Subject-line to your cell/pager.
5-180 sec	SubjHHMM-only	RA, Dec, Time, & Intensity displayed in the Subject-line in RA=HH:MM:SS format.
0.3 sec	Dedicated phone	Continuous phone/modem connection. (no longer available)
30-90 sec	Dialed phone	Slower but much cheaper than Dedicated. (no longer available)

Socket (160B binary pkt) (aka the "original" GCN socket method): The fastest method is the Internet socket connection. Sockets is a technique to connect two computers over a network. The socket connection is made at some initial time and is maintained for long periods of time. The GCN system runs 24/365 continuously allowing sites to connect and disconnect at their leisure. The time delay for the propagation of the packets varies due to the distance between the two computers, the number of routers and gateways in between, and the amount of other network traffic. However, we have routinely shown

The EM-followUP tutorial focus on VOEvent which is among the more convient methods for autonomous operations .

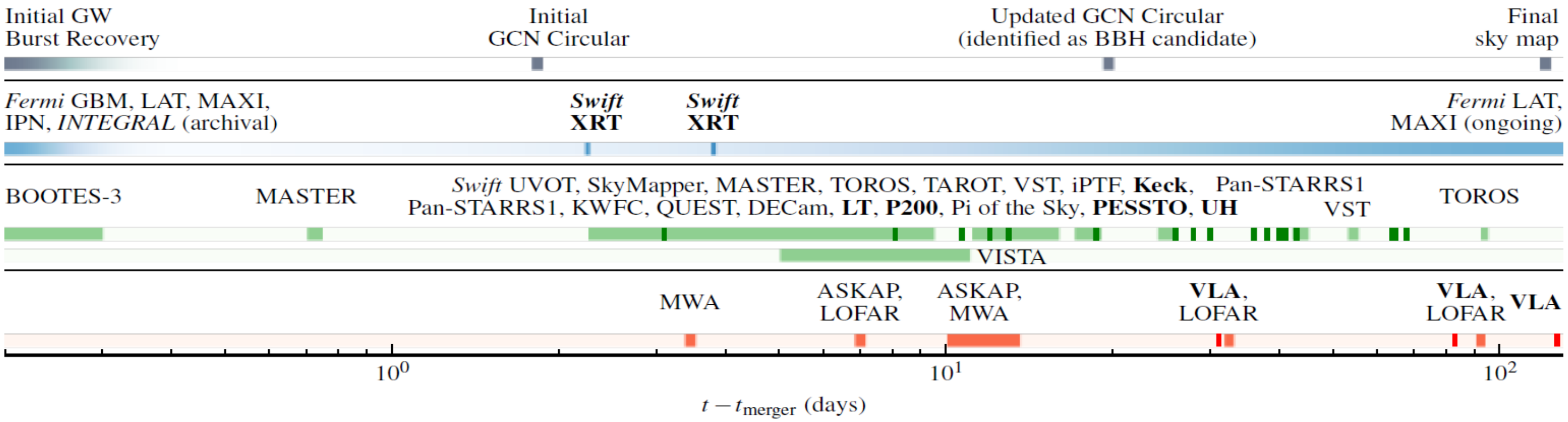
Sky Map Coverage



Follow-up observations are reported by 25 teams via private GCN Circulars.

very extensive sky coverage!

From Abbott et al. 2016, arXiv:1602.08492



Sky Maps of GW 150914

The probability skymap are disseminated using a sequence of algorithms with increasing accuracy and computational cost.

cWB

- *Initially shared*
- Area = 310 deg^2 (90% conf.)

minimal assumptions of waveform morphology

LIB

- *Initially shared*
- Area = 750 deg^2 (90% conf.)

Bayesian inference assuming sinusoidally modulated Gaussian signal

BAYESTAR

- *2016 Jan. 13*
- Area = 400 deg^2 (90% conf.)

triangulating times, amplitudes and phases

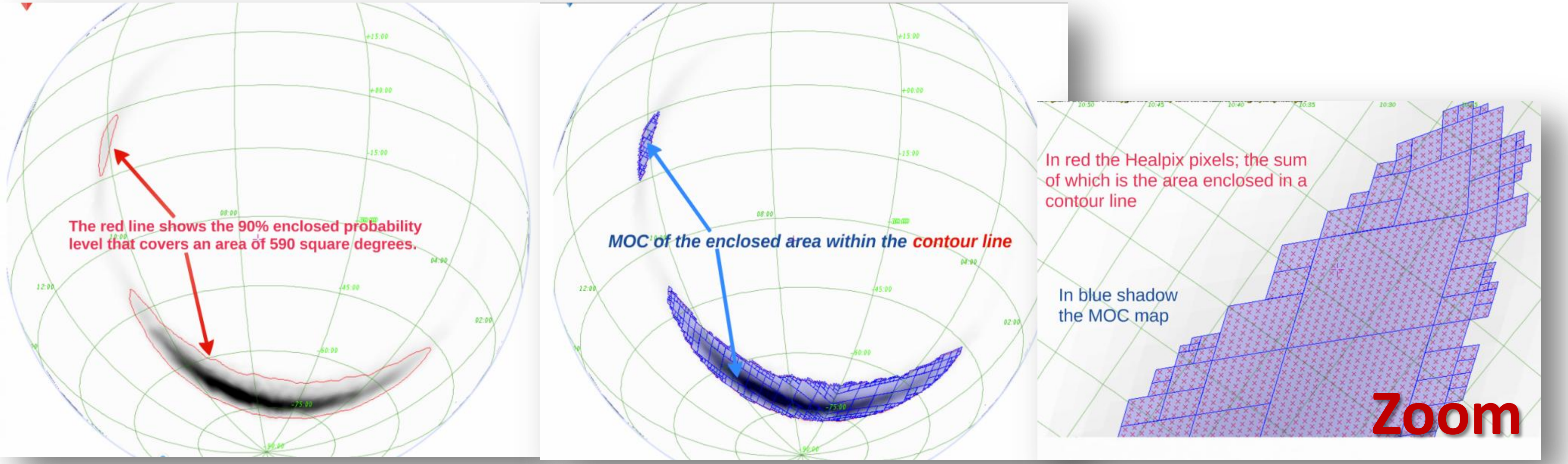
LALInference

- *2016 Jan. 13*
- Area = 590 deg^2 (90% conf.)

Final localization

CBC waveform, BH spin and detector calibration uncertainties

MOC representation of sky areas enclosed into iso-contour lines



The red line shows the 90% enclosed probability level that covers an area of 590 square degrees.

MOC of the enclosed area within the contour line

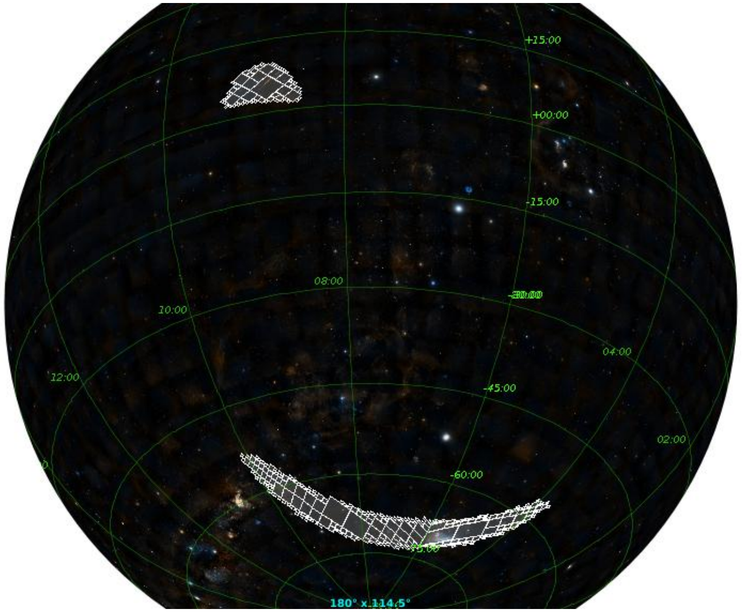
In red the Healpix pixels; the sum of which is the area enclosed in a contour line

In blue shadow the MOC map

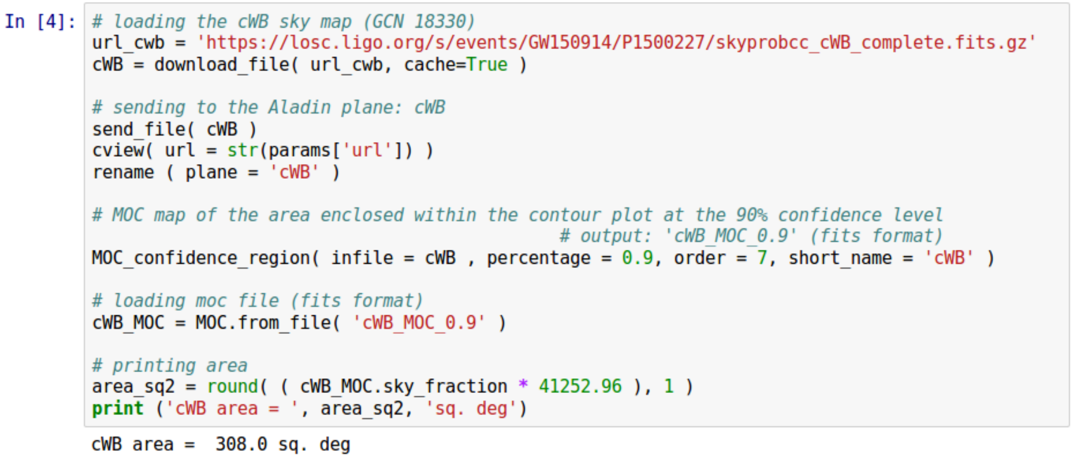
Zoom

MOC_area_prob(infile, percentage, order, output) based on MOCpy module

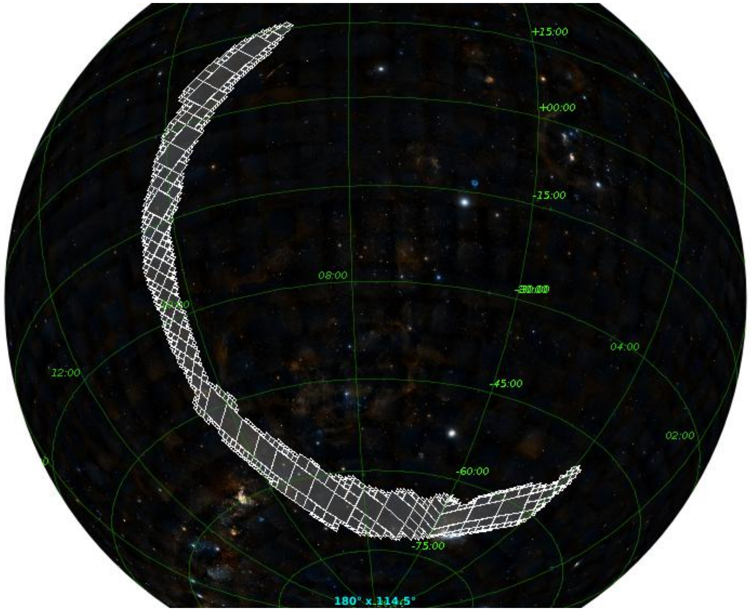
The enclosed area within a given probability level of a GW sky map can be effectively described through the Multi-Order Coverage (MOC) method.



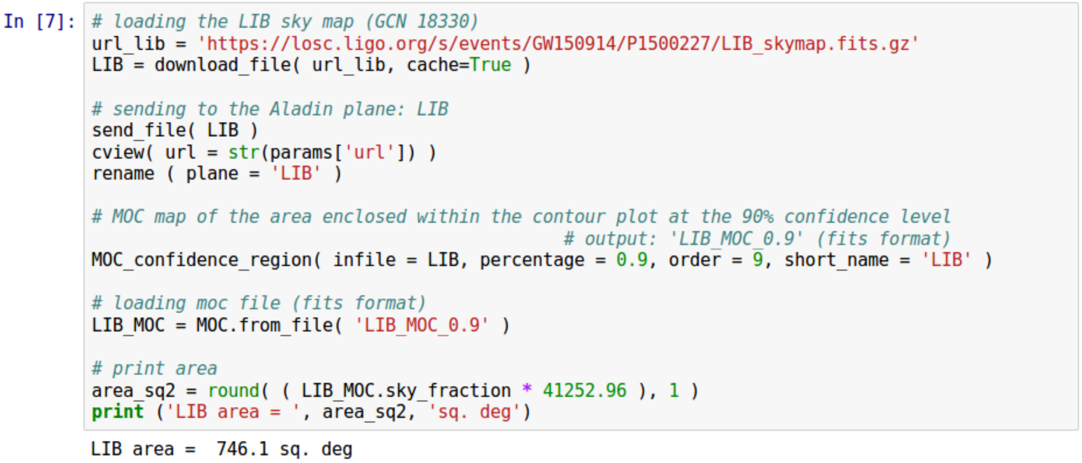
3.A cWB sky map



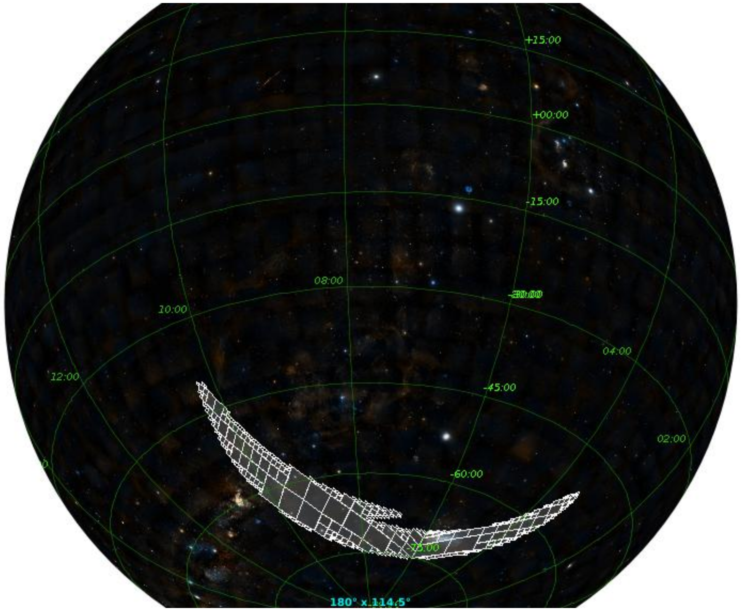
```
In [4]: # loading the cWB sky map (GCN 18330)
        url_cwb = 'https://losc.ligo.org/s/events/GW150914/P1500227/skyprobcc_cWB_complete.fits.gz'
        cWB = download_file( url_cwb, cache=True )

        # sending to the Aladin plane: cWB
        send_file( cWB )
        cview( url = str(params['url']) )
        rename ( plane = 'cWB' )

        # MOC map of the area enclosed within the contour plot at the 90% confidence level
                                                      # output: 'cWB_MOC_0.9' (fits format)
        MOC_confidence_region( infile = cWB , percentage = 0.9, order = 7, short_name = 'cWB' )

        # loading moc file (fits format)
        cWB_MOC = MOC.from_file( 'cWB_MOC_0.9' )

        # printing area
        area_sq2 = round( ( cWB_MOC.sky_fraction * 41252.96 ), 1 )
        print ('cWB area = ', area_sq2, 'sq. deg')

        cWB area =  308.0 sq. deg
```

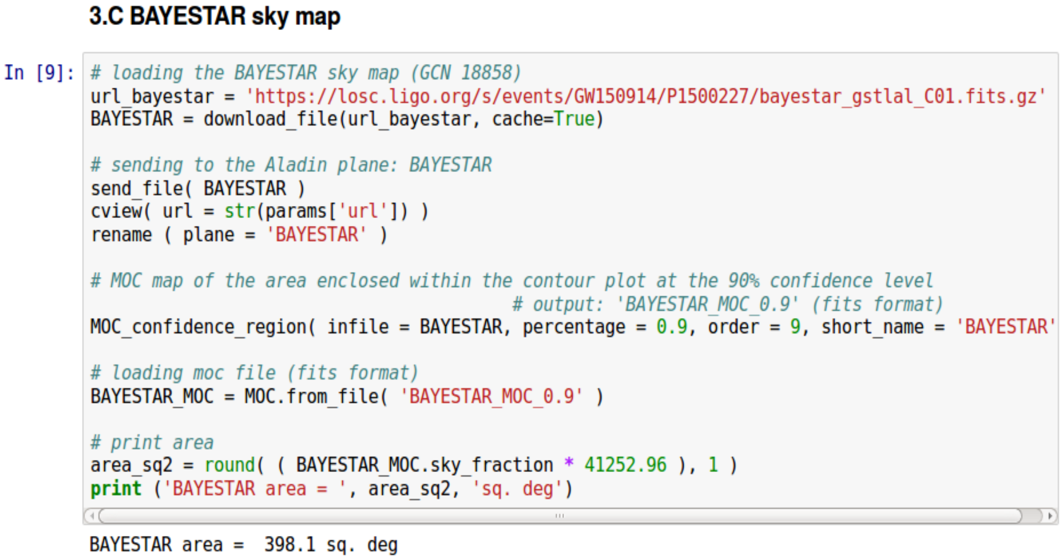


3.B LIB sky map



```
In [7]: # loading the LIB sky map (GCN 18330)
        url_lib = 'https://losc.ligo.org/s/events/GW150914/P1500227/LIB_skymap.fits.gz'
        LIB = download_file( url_lib, cache=True )

        # sending to the Aladin plane: LIB
        send_file( LIB )
        cview( url = str(params['url']) )
        rename ( plane = 'LIB' )

        # MOC map of the area enclosed within the contour plot at the 90% confidence level
                                                      # output: 'LIB_MOC_0.9' (fits format)
        MOC_confidence_region( infile = LIB, percentage = 0.9, order = 9, short_name = 'LIB' )

        # loading moc file (fits format)
        LIB_MOC = MOC.from_file( 'LIB_MOC_0.9' )

        # print area
        area_sq2 = round( ( LIB_MOC.sky_fraction * 41252.96 ), 1 )
        print ('LIB area = ', area_sq2, 'sq. deg')

        LIB area =  746.1 sq. deg
```



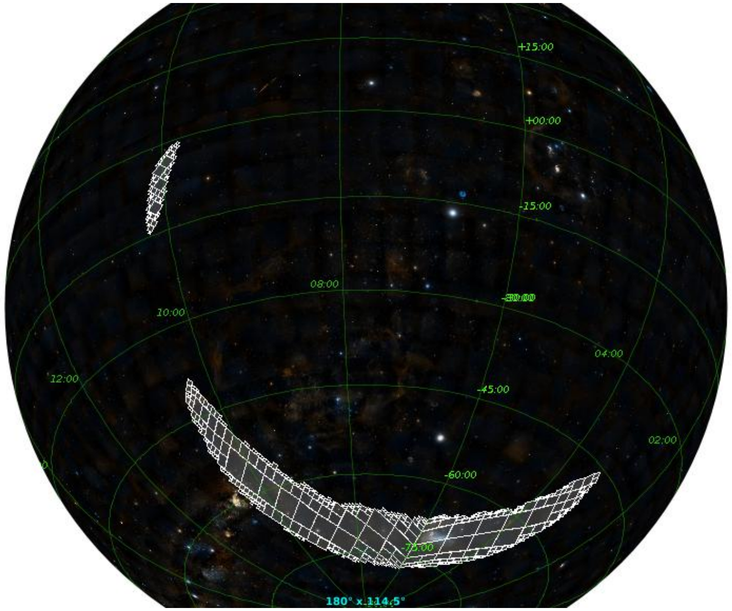


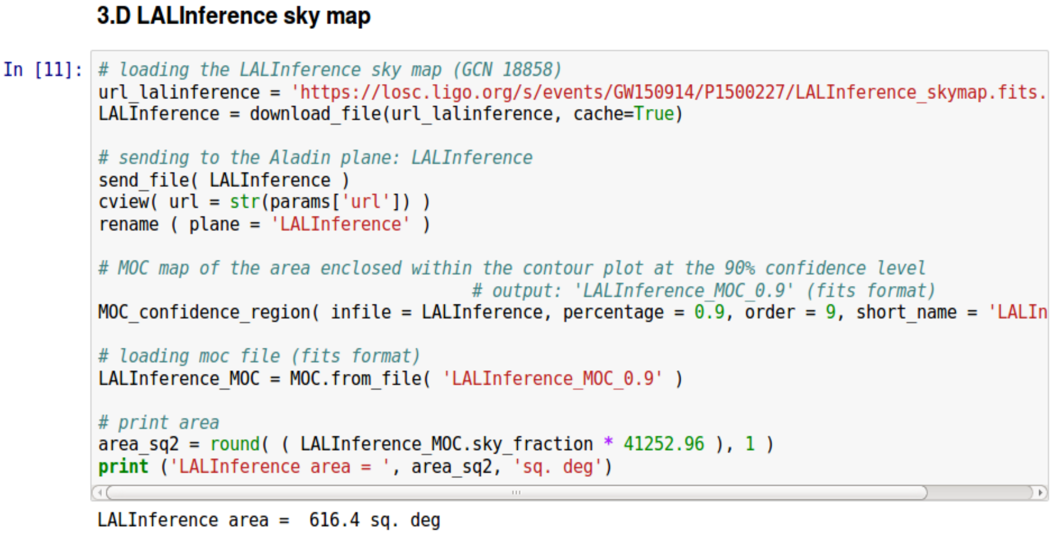
3.C BAYESTAR sky map

In [9]:

```
# loading the BAYESTAR sky map (GCN 18858)
url_bayestar = 'https://losc.ligo.org/s/events/GW150914/P1500227/bayestar_gstlal_C01.fits.gz'
BAYESTAR = download_file(url_bayestar, cache=True)

# sending to the Aladin plane: BAYESTAR
send_file( BAYESTAR )
cview( url = str(params['url']) )
rename ( plane = 'BAYESTAR' )

# MOC map of the area enclosed within the contour plot at the 90% confidence level
                                            # output: 'BAYESTAR_MOC_0.9' (fits format)
MOC_confidence_region( infile = BAYESTAR, percentage = 0.9, order = 9, short_name = 'BAYESTAR'

# loading moc file (fits format)
BAYESTAR_MOC = MOC.from_file( 'BAYESTAR_MOC_0.9' )

# print area
area_sq2 = round( ( BAYESTAR_MOC.sky_fraction * 41252.96 ), 1 )
print ('BAYESTAR area = ', area_sq2, 'sq. deg')
```

```
BAYESTAR area =  398.1 sq. deg
```



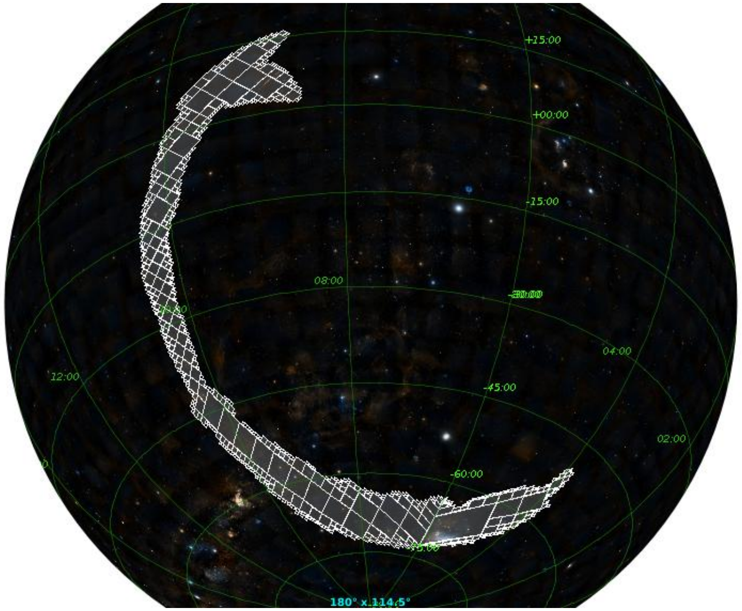


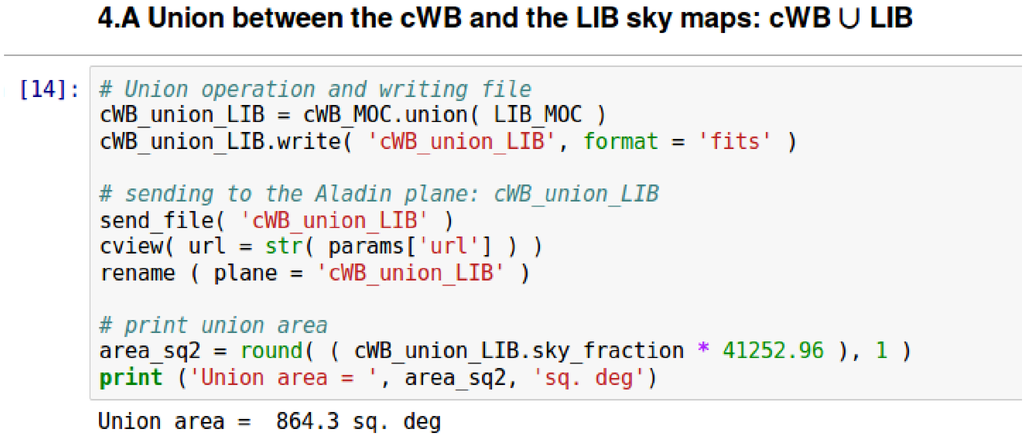
3.D LALInference sky map

In [11]:

```
# loading the LALInference sky map (GCN 18858)
url_lalinference = 'https://losc.ligo.org/s/events/GW150914/P1500227/LALInference_skymap.fits.
LALInference = download_file(url_lalinference, cache=True)

# sending to the Aladin plane: LALInference
send_file( LALInference )
cview( url = str(params['url']) )
rename ( plane = 'LALInference' )

# MOC map of the area enclosed within the contour plot at the 90% confidence level
                                          # output: 'LALInference_MOC_0.9' (fits format)
MOC_confidence_region( infile = LALInference, percentage = 0.9, order = 9, short_name = 'LALIn

# loading moc file (fits format)
LALInference_MOC = MOC.from_file( 'LALInference_MOC_0.9' )

# print area
area_sq2 = round( ( LALInference_MOC.sky_fraction * 41252.96 ), 1 )
print ('LALInference area = ', area_sq2, 'sq. deg')
```

```
LALInference area =  616.4 sq. deg
```



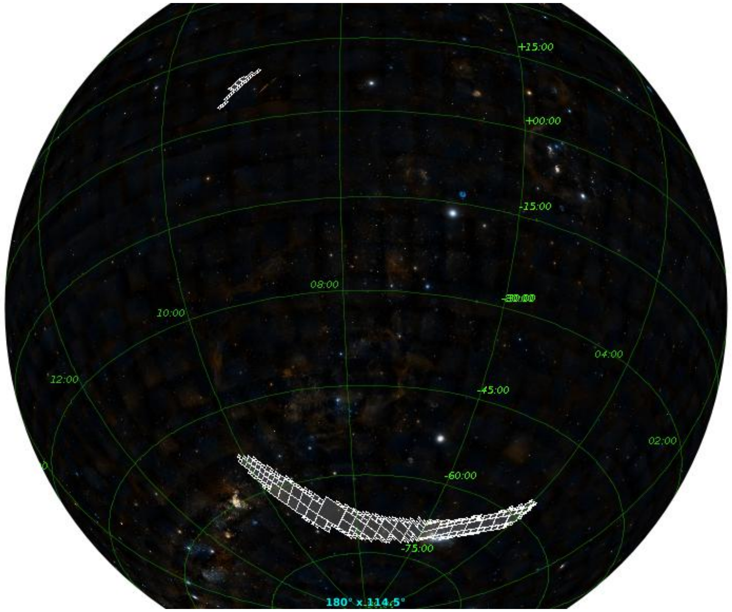


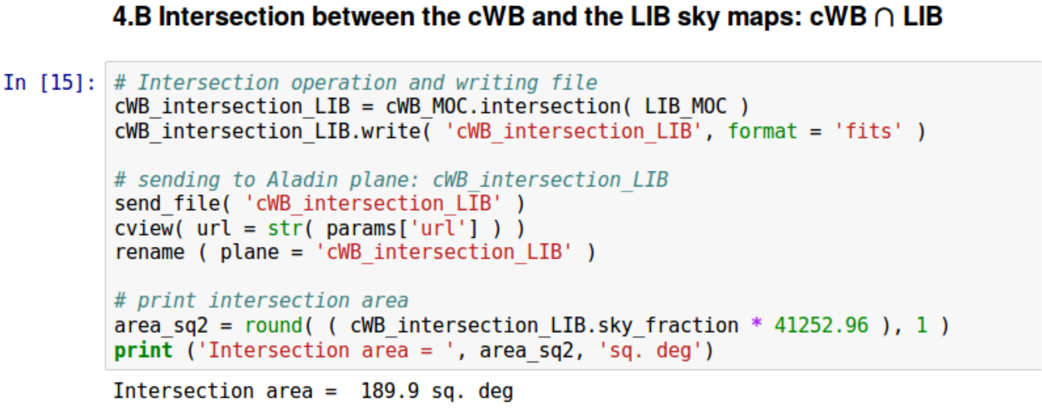
4.A Union between the cWB and the LIB sky maps: cWB ∪ LIB

[14]:

```
# Union operation and writing file
cWB_union_LIB = cWB_MOC.union( LIB_MOC )
cWB_union_LIB.write( 'cWB_union_LIB', format = 'fits' )

# sending to the Aladin plane: cWB_union_LIB
send_file( 'cWB_union_LIB' )
cview( url = str( params['url'] ) )
rename ( plane = 'cWB_union_LIB' )

# print union area
area_sq2 = round( ( cWB_union_LIB.sky_fraction * 41252.96 ), 1 )
print ('Union area = ', area_sq2, 'sq. deg')
```

```
Union area =  864.3 sq. deg
```



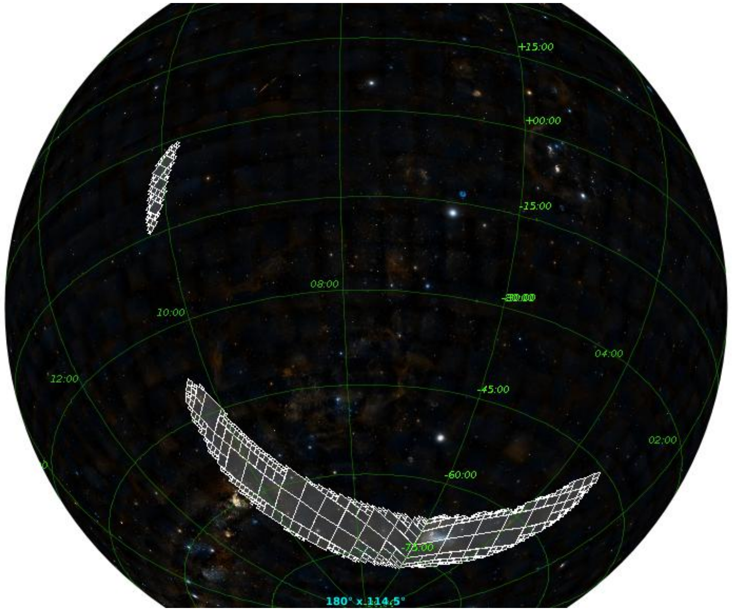


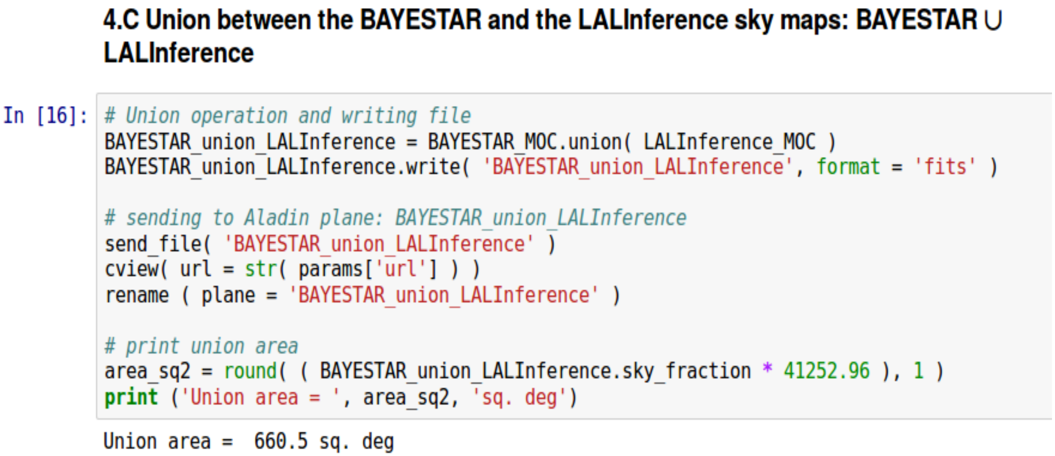
4.B Intersection between the cWB and the LIB sky maps: cWB ∩ LIB

In [15]:

```
# Intersection operation and writing file
cWB_intersection_LIB = cWB_MOC.intersection( LIB_MOC )
cWB_intersection_LIB.write( 'cWB_intersection_LIB', format = 'fits' )

# sending to Aladin plane: cWB_intersection_LIB
send_file( 'cWB_intersection_LIB' )
cview( url = str( params['url'] ) )
rename ( plane = 'cWB_intersection_LIB' )

# print intersection area
area_sq2 = round( ( cWB_intersection_LIB.sky_fraction * 41252.96 ), 1 )
print ('Intersection area = ', area_sq2, 'sq. deg')
```

```
Intersection area =  189.9 sq. deg
```



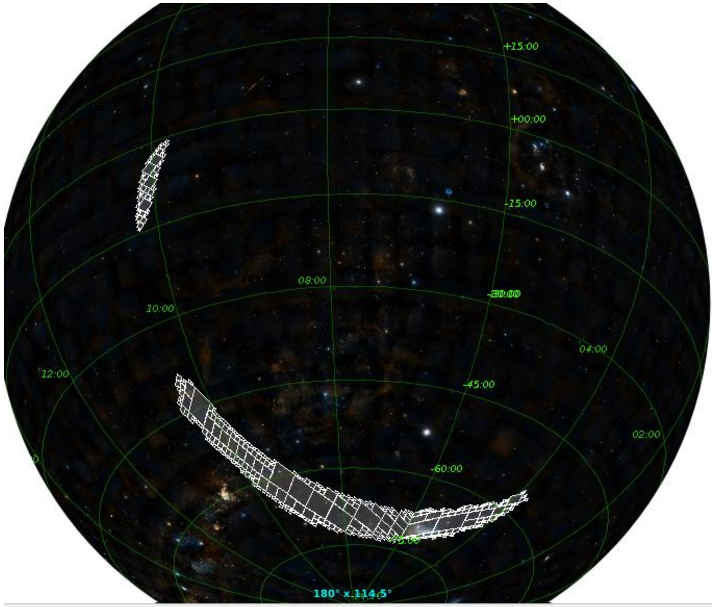


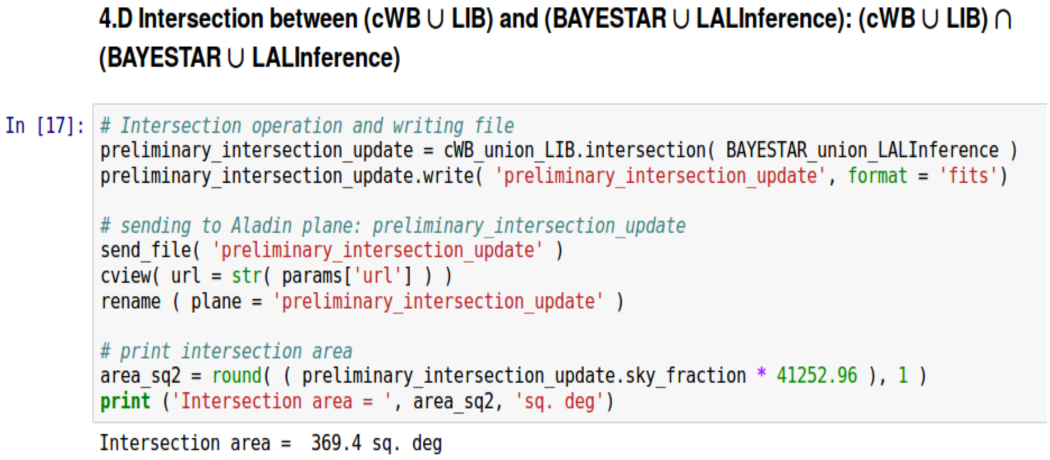
4.C Union between the BAYESTAR and the LALInference sky maps: BAYESTAR ∪ LALInference

In [16]:

```
# Union operation and writing file
BAYESTAR_union_LALInference = BAYESTAR_MOC.union( LALInference_MOC )
BAYESTAR_union_LALInference.write( 'BAYESTAR_union_LALInference', format = 'fits' )

# sending to Aladin plane: BAYESTAR_union_LALInference
send_file( 'BAYESTAR_union_LALInference' )
cview( url = str( params['url'] ) )
rename ( plane = 'BAYESTAR_union_LALInference' )

# print union area
area_sq2 = round( ( BAYESTAR_union_LALInference.sky_fraction * 41252.96 ), 1 )
print ('Union area = ', area_sq2, 'sq. deg')
```

```
Union area =  660.5 sq. deg
```



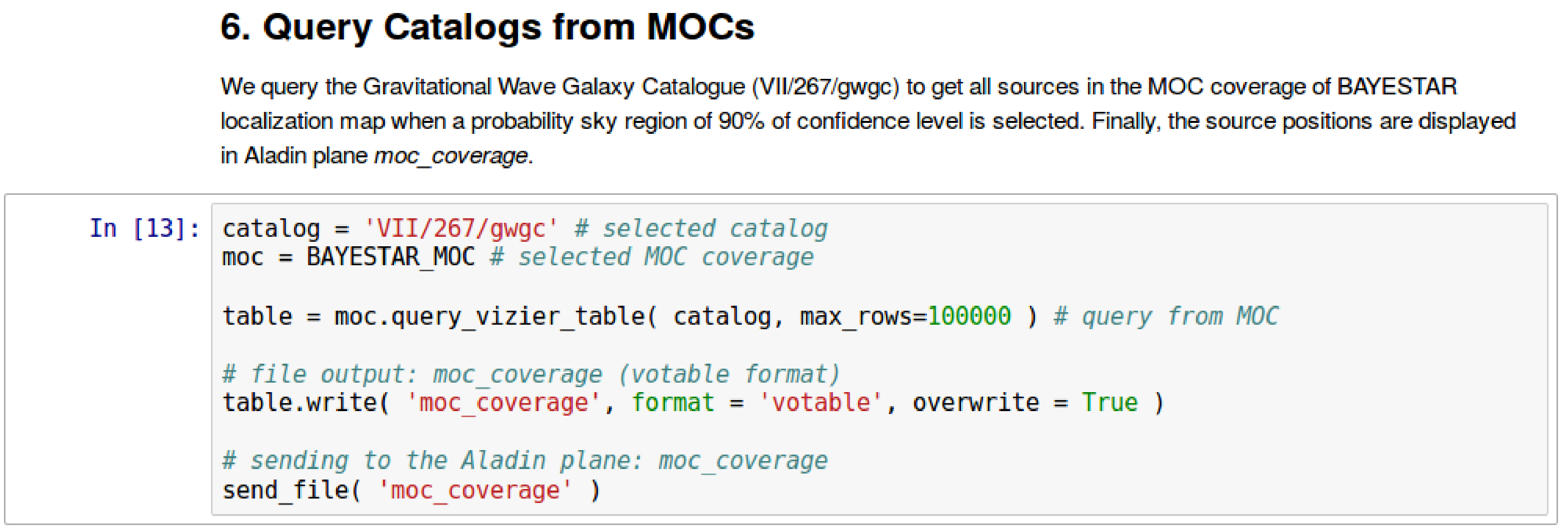


4.D Intersection between (cWB ∪ LIB) and (BAYESTAR ∪ LALInference): (cWB ∪ LIB) ∩ (BAYESTAR ∪ LALInference)

In [17]:

```
# Intersection operation and writing file
preliminary_intersection_update = cWB_union_LIB.intersection( BAYESTAR_union_LALInference )
preliminary_intersection_update.write( 'preliminary_intersection_update', format = 'fits')

# sending to Aladin plane: preliminary_intersection_update
send_file( 'preliminary_intersection_update' )
cview( url = str( params['url'] ) )
rename ( plane = 'preliminary_intersection_update' )

# print intersection area
area_sq2 = round( ( preliminary_intersection_update.sky_fraction * 41252.96 ), 1 )
print ('Intersection area = ', area_sq2, 'sq. deg')
```

```
Intersection area =  369.4 sq. deg
```

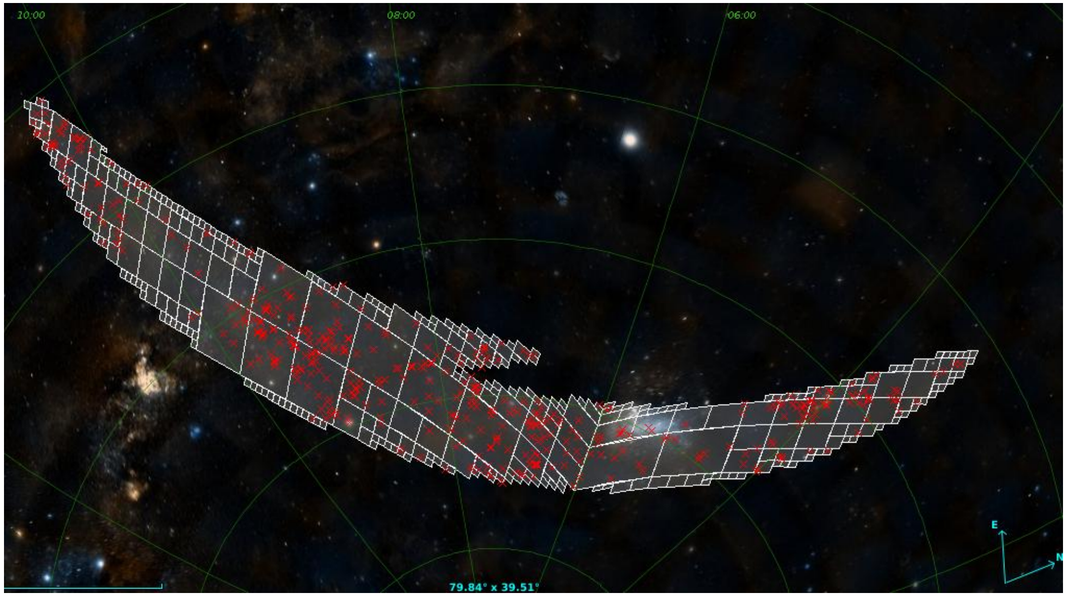


6. Query Catalogs from MOCs

We query the Gravitational Wave Galaxy Catalogue (VII/267/gwgc) to get all sources in the MOC coverage of BAYESTAR localization map when a probability sky region of 90% of confidence level is selected. Finally, the source positions are displayed in Aladin plane *moc_coverage*.

In [13]:

```
catalog = 'VII/267/gwgc' # selected catalog
moc = BAYESTAR_MOC # selected MOC coverage

table = moc.query_vizier_table( catalog, max_rows=100000 ) # query from MOC

# file output: moc_coverage (votable format)
table.write( 'moc_coverage', format = 'votable', overwrite = True )

# sending to the Aladin plane: moc_coverage
send_file( 'moc_coverage' )
```

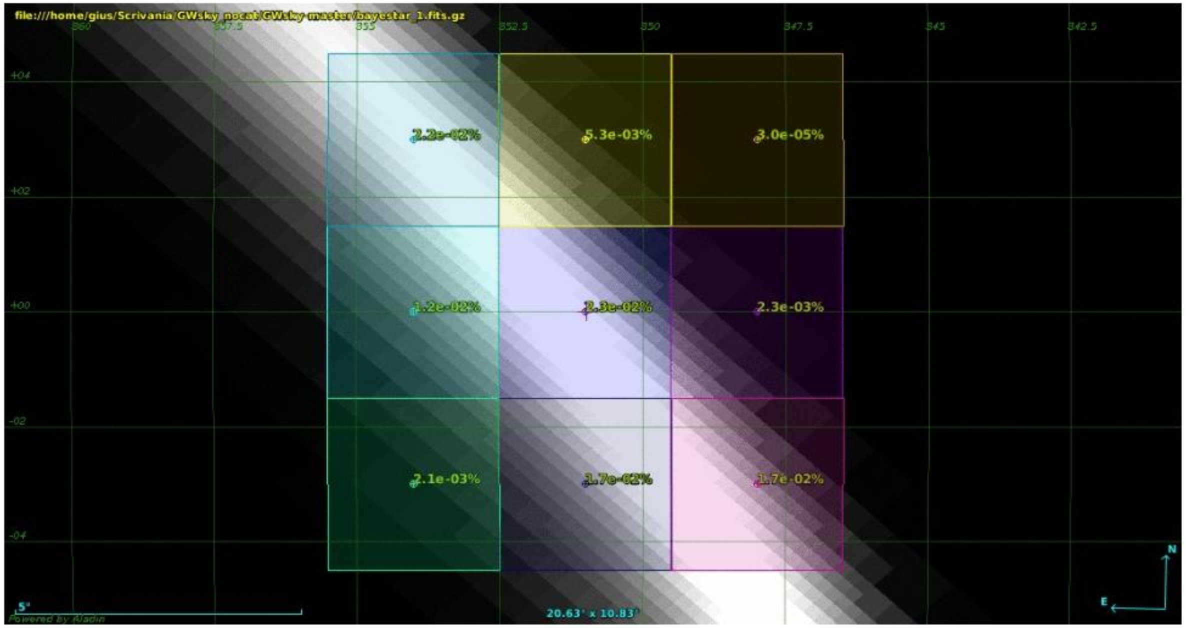


The MOCs of all VizieR tables and CDS pixel surveys (16.000 MOCs) are already available on line, and can be queried simultaneously in few ms.

GWsky: tiling the skymap in FoV

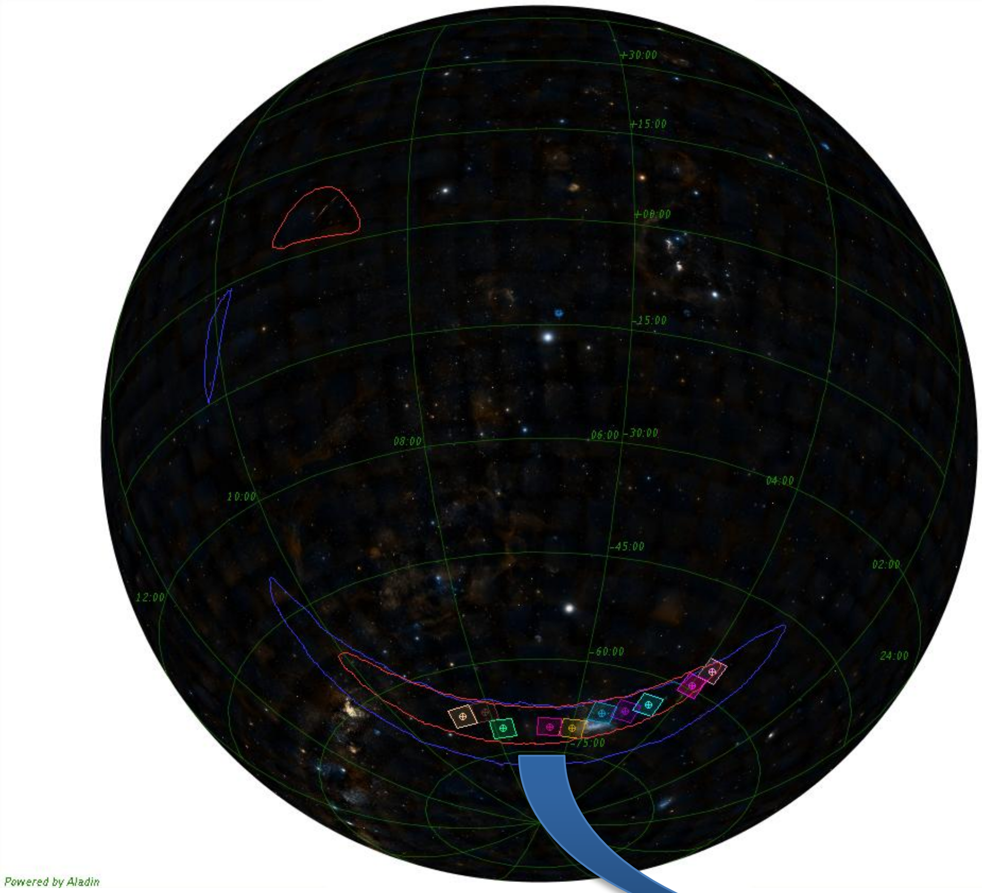
GWsky is an interactive Python script to generate a sequence of pointings given a specific Field of View

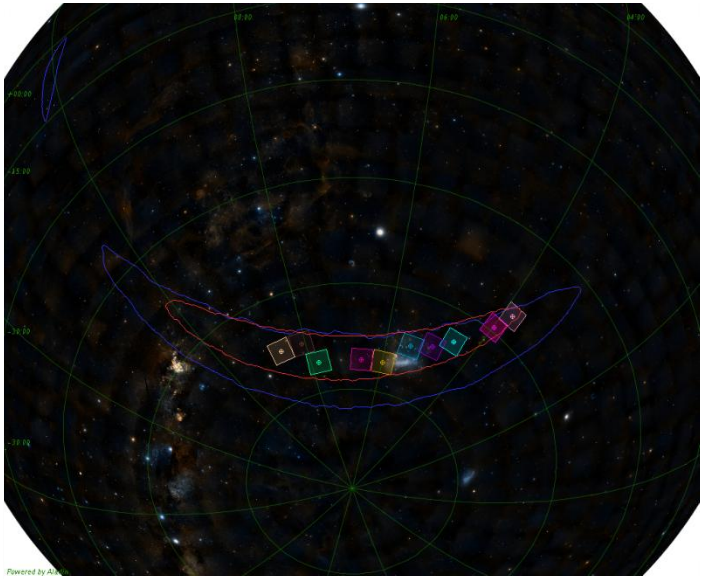
Cardinal and Intercardinal direction → Integrated probability inside each FoV → airmass in step of hour for each FoV → FoV query to catalogs

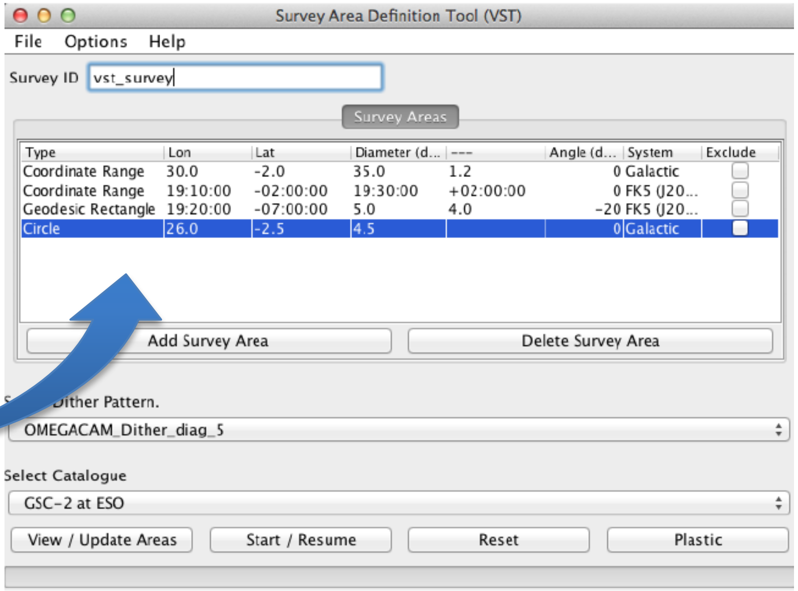


USER OPTION: the FoVs can be overlaid or separated from their default positions

VST Observation of GW 150914







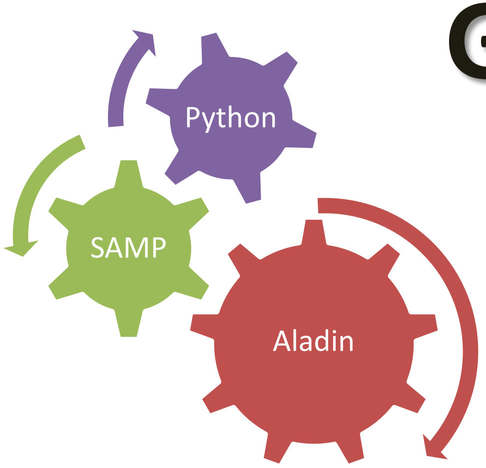
Survey ID vst_survey

Type	Lon	Lat	Diameter (d...	---	Angle (d...	System	Exclude
Coordinate Range	30.0	-2.0	35.0	1.2	0	Galactic	☐
Coordinate Range	19:10:00	-02:00:00	19:30:00	+02:00:00	0	FK5 (J20...	☐
Geodesic Rectangle	19:20:00	-07:00:00	5.0	4.0	-20	FK5 (J20...	☐
Circle	26.0	-2.5	4.5		0	Galactic	☐

Dither Pattern.
OMEGACAM_Dither_diag_5

Select Catalogue
GSC-2 at ESO

GWsky Command Line



C runs a new sequence *changing* the FoV center

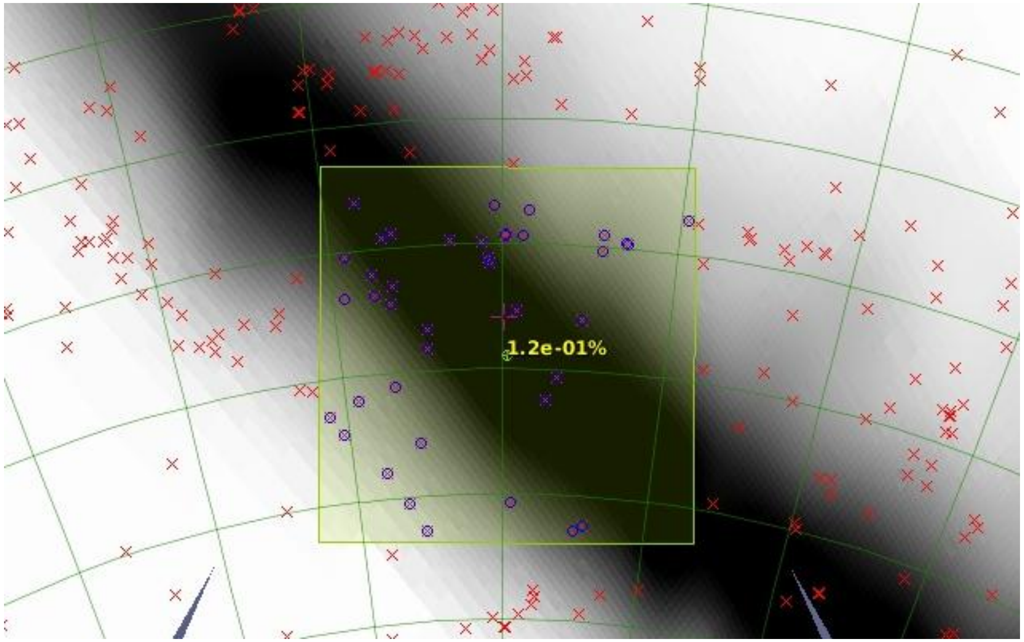
I runs a new sequence without drawing the *input* FoV

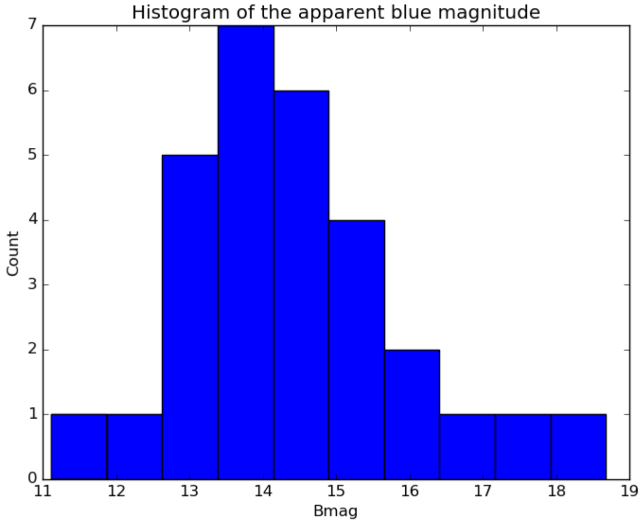
L runs a new sequence starting from the *last* drawn FoV

R *repeats* the last action

Q *quit*

FoV query to catalog: an example





Descriptive statistic is plotted for each FoV

This can be useful to determine the integration time of each image or to avoid bright galaxies or stars

Conclusion & Future Perspectives

- ☐ **GW: new way to observe the Universe**
- ☐ **VO can help shaping multimessenger astronomy with GW**
- ☐ **Need to bring different types of data/observation together**
- ☐ **Establish connection between GW observations and existing data bases (galaxy catalogs, variable stars)**

.