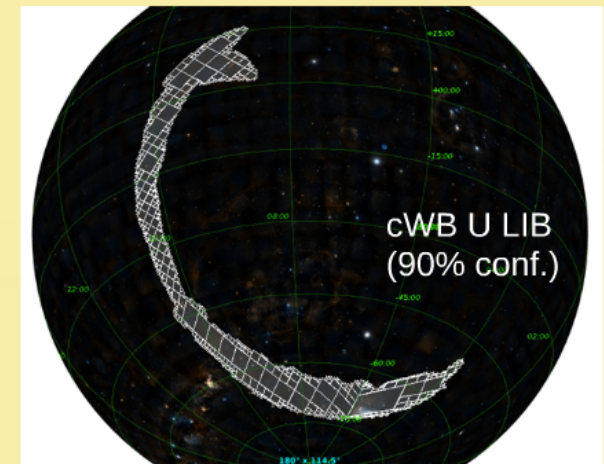
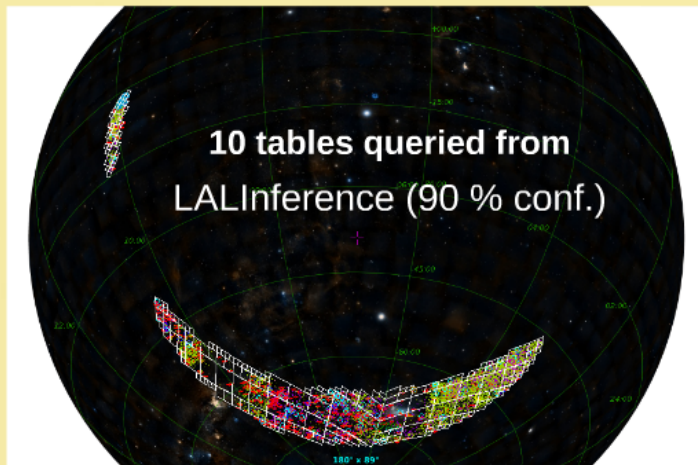


Multi-Order Coverage Map (MOC)

Fernique et al. (2015); <http://arxiv.org/pdf/1505.02937v1.pdf>

- **Effective comparison** mechanisms between different sky maps



- **Simultaneous queries** of all VizieR tables (16.000 MOCs)

MOC provides a **dynamic** approach to manage probability sky maps

MOC and EM-FollowUP

The MOC method is based on the **HEALPix** tessellation algorithm and it is essentially a simple way to **map irregular and complex sky regions** into hierarchically grouped predefined cells.

Probability sky map

Sky localization of several hundred square degrees

Source location can be characterized by elliptical sky areas

Ring shape as a result of triangulation based on arrival time delay between detector sites

We can describe complex sky region using just a list of cells based on Healpix

From Singer et al. 2016, The Astrophysical Journal, 762

From Abbott et al. 2016, Living Reviews in Relativity, 19, 2

GW skymap: from 2009 to 2015

The trigger detection, based on individual detectors, of each candidate GW event on a skymap, a list of probability elements assigned to pixels, is a grid covering the sky.

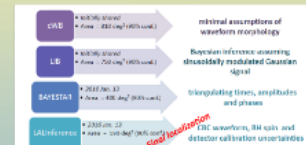
HEALPix tessellation is introduced in the Advanced LIGO/ Virgo implementation as **mapskid**.

From Abbott et al. 2016, Living Reviews in Relativity, 19, 2

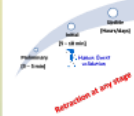
The **operations** between the MOC maps (union, intersection, subtraction, difference) are extremely **simple and fast** (generally a few milliseconds) even for very complex sky regions.

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

Distributed sky maps of GW 150914



GW Alerts and Elapsed Time



We need fast tool to compare them!

The GCN Circular (18858) shows the overlap regions between the sky maps published with the GW 150914. The same regions are visualized applying the MOC method.

GW 150914: The first detection of a gravitational wave event by the LIGO and Virgo detectors. The signal was detected by the LIGO Livingston detector on September 14, 2015, at 09:50:45 UTC. The event was followed by a series of smaller events, including GW150914-09:50:45, GW150914-09:50:45, and GW150914-09:50:45.

The event was followed by a series of smaller events, including GW150914-09:50:45, GW150914-09:50:45, and GW150914-09:50:45.

The event was followed by a series of smaller events, including GW150914-09:50:45, GW150914-09:50:45, and GW150914-09:50:45.

Some datasever, such as **VizieR**, can be **queried by MOC** in order to return data (galaxy catalogs/list of images) only inside the MOC coverage.

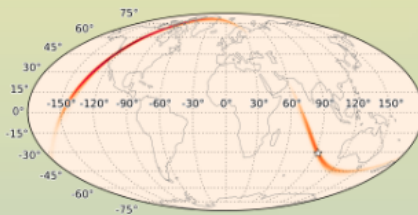
To identify likely host galaxies of a GW event we need to collect **as much information as possible**

TOPCAT post-processing: to organize data from queries.

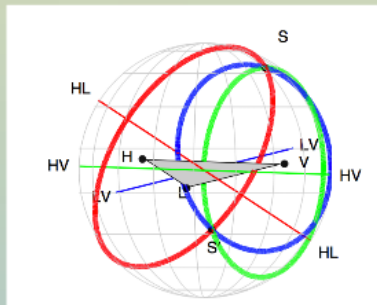


The MOC method is based on the **HEALPix** tessellation algorithm and it is essentially a simple way to map irregular and complex sky regions into hierarchically grouped predefined cells.

Probability sky map



From Singer et al. 2014, *The Astrophysical Journal*, 795



From Abbott et al. 2016, *Living Rev. Relativity* 19, 1

Sky localization of several hundred square degrees

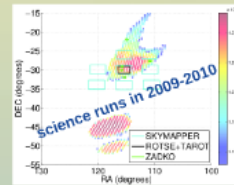
Source location can be characterized by disjoint sky areas

Ring shape as a result of "triangulation" based on arrival time delay between detector sites.

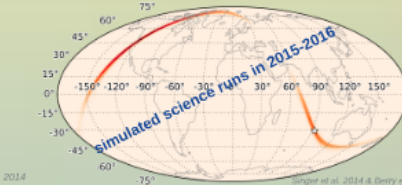
We can describe complex sky region using just a list of cells based on Healpix

GW skymap: from 2009 to 2015

The trigger pipelines report the estimated position of each candidate GW event as a skymap, a list of probability densities assigned to pixels in a grid covering the sky.



The gravitational wave skymap for event G19377 which occurred on 16th September 2010. This event was later revealed to be a blind injection.

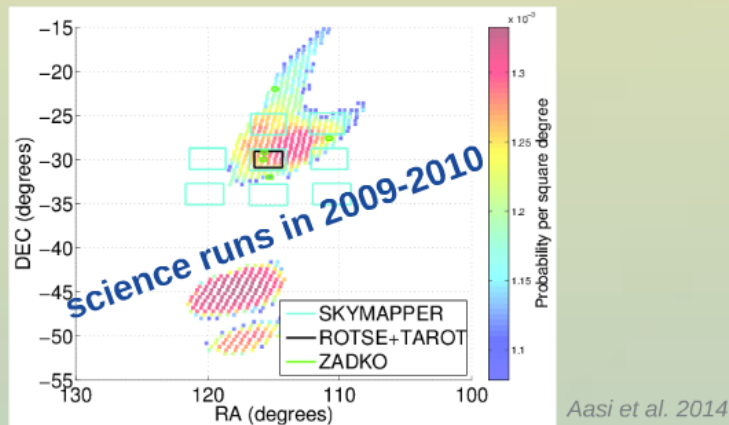


ID 4532; data release for The First Two Years of Electromagnetic Follow-Up with Advanced LIGO and Virgo.

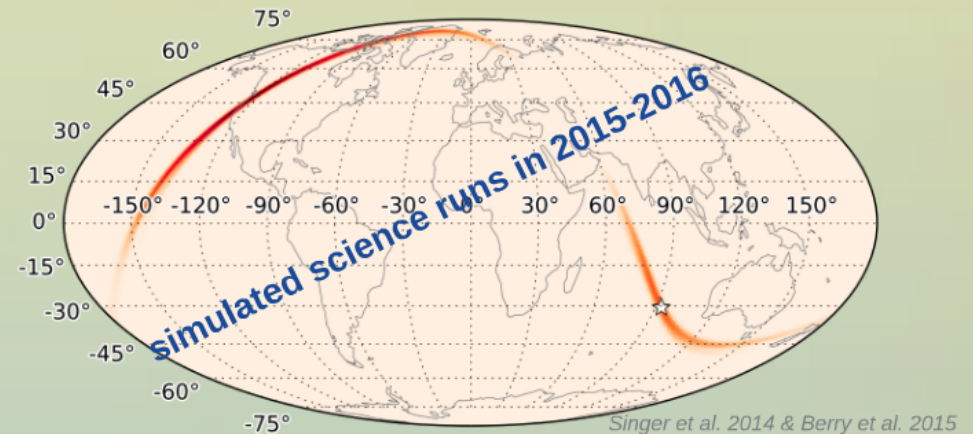
HEALPix pixelization is introduced in the Advanced LVC run. The implementation is manageable!

GW skymap: from 2009 to 2015

The trigger pipelines report the estimated position of each candidate GW event as a skymap, a list of probability densities assigned to pixels in a grid covering the sky.



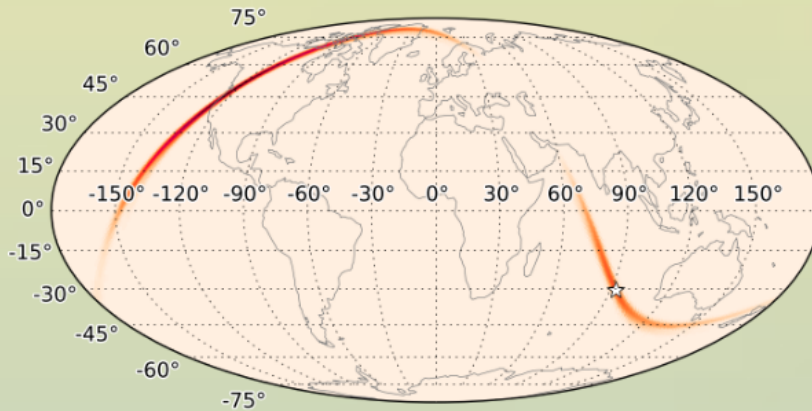
The gravitational wave skymap for event G19377 which occurred on 16th September 2010. This event was later revealed to be a blind injection.



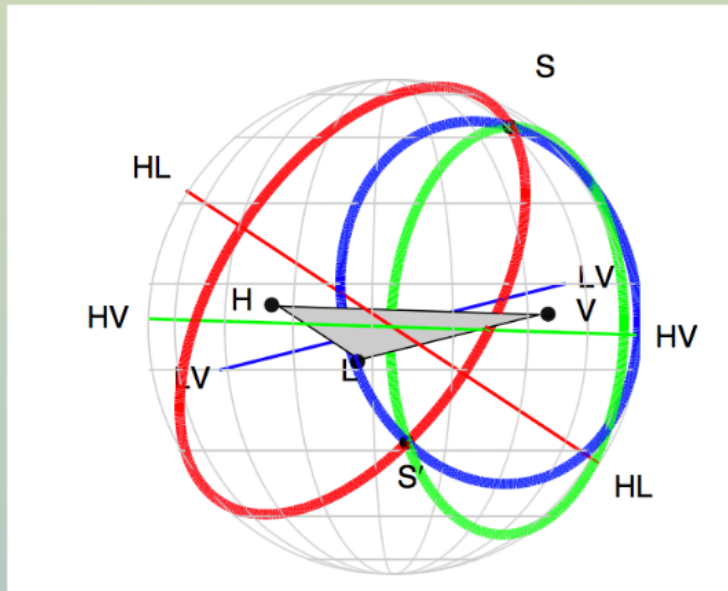
ID 4532; data release for The First Two Years of Electromagnetic Follow-Up with Advanced LIGO and Virgo.

***HEALPix pixelization is introduced in the Advanced LVC run.
The implementation is manageable!***

Probability sky map



From Singer et al. 2014, *The Astrophysical Journal*, 795



From Abbott et al. 2016, *Living Rev. Relativity* 19, 1

Sky localization of several hundred square degrees

Source location can be characterized by disjoint sky areas

Ring shape as a result of “triangulation” based on arrival time delay between detector sites.

We can describe complex sky region using just a list of cells based on Healpix

The sky

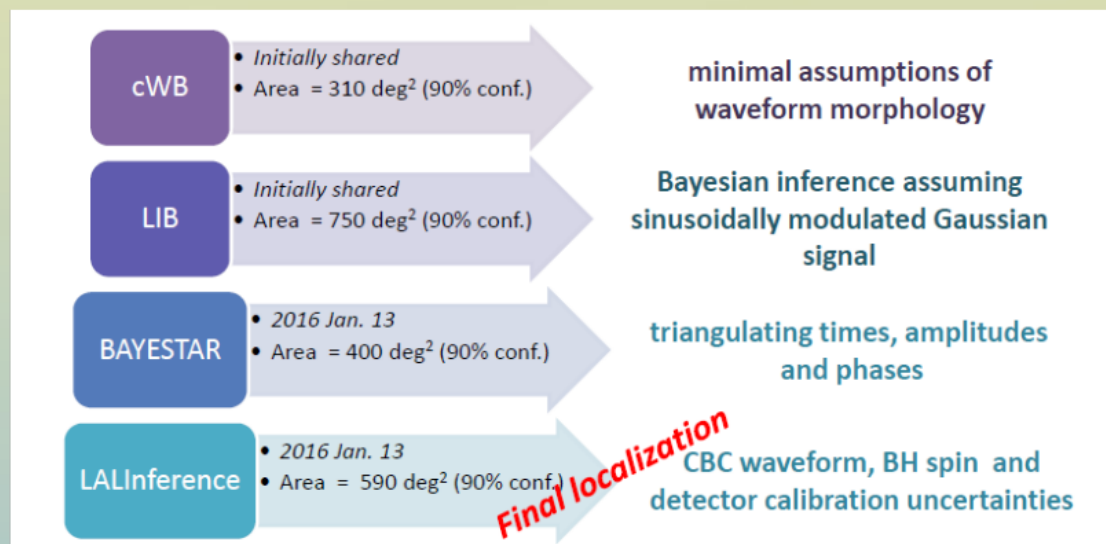
The
whi
was



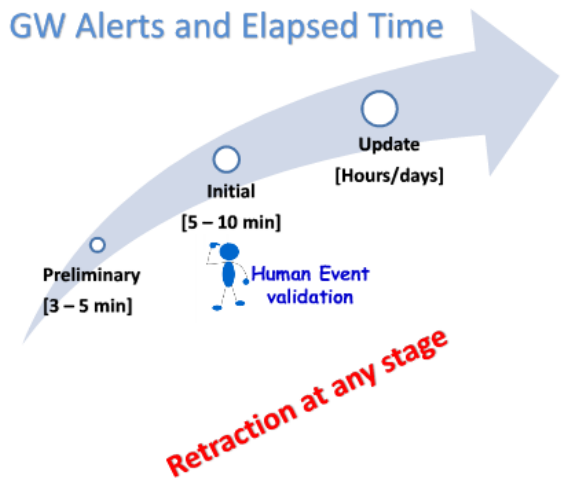
The **operations** between the MOC maps (union, intersection, subtraction, difference) are extremely **simple and fast** (generally a few milliseconds) even for very complex sky regions.

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

Distributed sky maps of GW 150914



GW Alerts and Elapsed Time



We need fast tool to compare them!

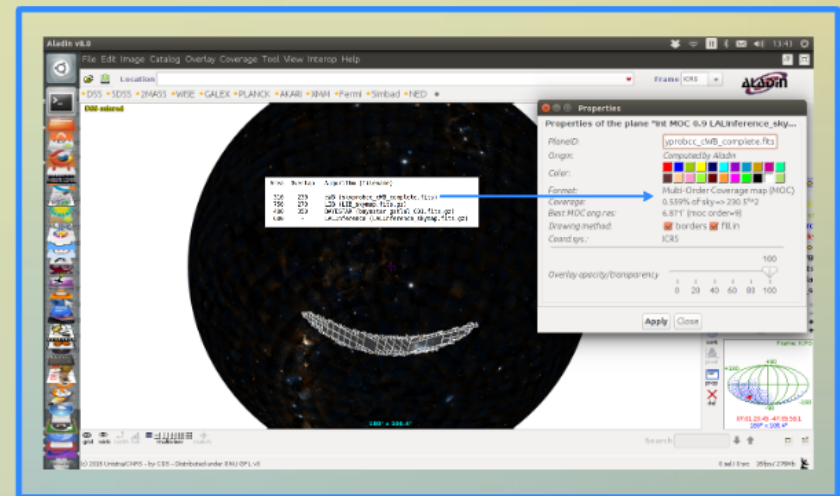
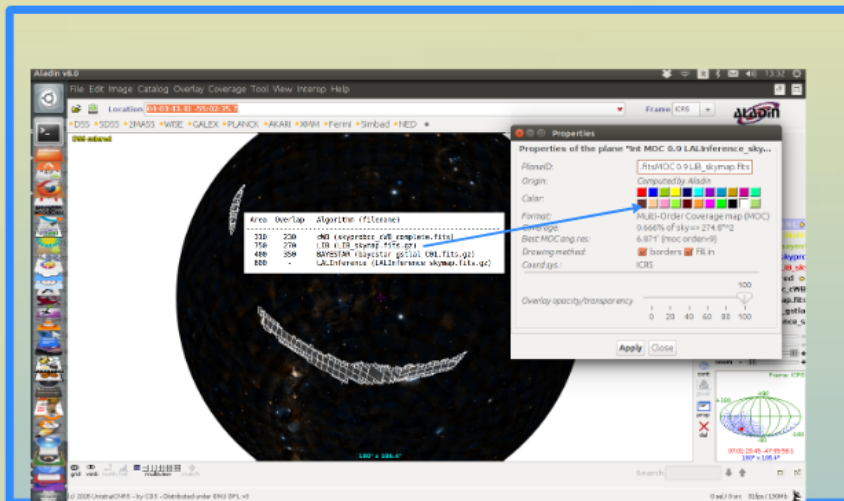
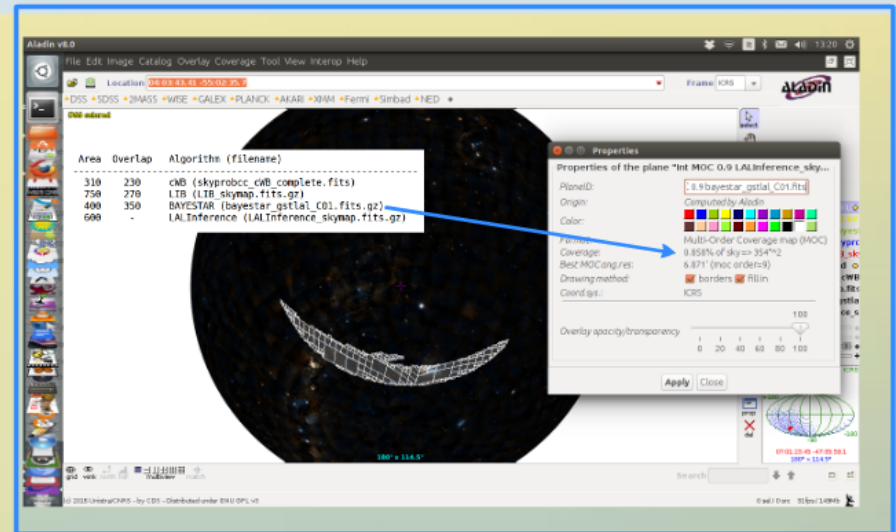
The GCN Circular (18858) shows the overlap regions between the sky maps published with the GW 150914.

The same regions are visualized applying the MOC method.

Both of the sky maps above agree with the initial LALInference Burst (LIB) localization (GCN 18330, LIB_skymap.fits.gz) on favoring the southern portion of the annulus determined by an arrival time difference between LIGO Hanford and LIGO Livingston of about 7 ms.

The table below presents a quantitative comparison of the available localizations along the lines of Sec. 4.5 of Essick et al. (2015, <http://adsabs.harvard.edu/abs/2015ApJ...800...81E>). The first column gives the area in deg² of the 90% credible region, and the second column gives the area in deg² of the overlap with the LALInference 90% credible region.

Area	Overlap	Algorithm (filename)
310	230	cWB (skyprobcc_cWB_complete.fits)
750	270	LIB (LIB_skymap.fits.gz)
400	350	BAYESTAR (bayestar_gstlal_C01.fits.gz)
600	-	LALInference (LALInference_skymap.fits.gz)



Skymap Viewer

A sky atlas for understanding LIGO-Virgo skymaps. Help here, or view skymaps here. If you do not see the big dark sky map, look below.

Skymap Viewer. Plenty simulated skymaps with the + and - at the right of the toolbar.



ed applying the MOC method.

The screenshot shows the Aladin v8.0 interface. The main window displays a sky map with a white wireframe overlay representing a Multi-Order Coverage map (MOC). A table in the upper left lists the algorithms used for the MOC. A 'Properties' dialog box is open, showing details for the selected MOC plane. A blue arrow points from the 'LALInference' entry in the table to the 'Properties' dialog.

Area	Overlap	Algorithm (filename)
310	230	cWB (skyprobcc_cWB_complete.fits)
750	270	LIB (LIB_skymap.fits.gz)
400	350	BAYESTAR (bayestar_gstlal_C01.fits.gz)
600	-	LALInference (LALInference_skymap.fits.gz)

Properties of the plane "Int MOC 0.9 LALInference_sky..."

- PlaneID: 0.9 bayestar_gstlal_C01.fits
- Origin: Computed by Aladin
- Color: [Color selection palette]
- Format: Multi-Order Coverage map (MOC)
- Coverage: 0.858% of sky => 354°^2
- Best MOCang.res: 6.871' (moc order=9)
- Drawing method: borders fill in
- Coord.sys.: ICRS
- Overlay opacity/transparency: 100

0 sel / 0 src 31fps / 149Mb

400 350
600 -

BAYESTAR (bayestar_gstlal_C01.fits.gz)
LALInference (LALInference_skymap.fits.gz)

The screenshot shows the Aladin v8.0 interface. The main window displays a sky map with a grid overlay. A table in the center lists the data sources used for the map:

Area	Overlap	Algorithm (filename)
310	230	cWB (skyprobcc_cWB_complete.fits)
750	270	LIB (LIB_skymap.fits.gz)
400	350	BAYESTAR (bayestar_gstlal_C01.fits.gz)
600	-	LALInference (LALInference_skymap.fits.gz)

The Properties dialog box is open, showing the following details for the plane "Int MOC 0.9 LALInference_sky...":

- PlaneID: .fitsMOC 0.9 LIB_skymap.fits
- Origin: Computed by Aladin
- Color: (Color selection palette)
- Format: Multi-Order Coverage map (MOC)
- Coverage: 0.666% of sky => 274.8°^2
- Best MOCang.res: 6.871' (moc order=9)
- Drawing method: borders fill in
- Coord.sys.: ICRS
- Overlay opacity/transparency: 100

The interface also shows the location 04:03:43.41 -55:02:35.7, a list of data sources (DSS, SDSS, 2MASS, WISE, GALEX, PLANCK, AKARI, XMM, Fermi, Simbad, NED), and a zoomed-in view of the sky map in the bottom right corner.

Aladin v8.0

File Edit Image Catalog Overlay Coverage Tool View Interop Help

Location Frame ICRS

DSS SDSS 2MASS WISE GALEX PLANCK AKARI XMM Fermi Simbad NED +

DSS entered

Area	Overlap	Algorithm (filename)
310	230	cWB (skyprobcc_cWB_complete.fits)
750	270	LIB (LIB_skymap.fits.gz)
400	350	BAYESTAR (bayestar_gstlal_C01.fits.gz)
600	-	LALInference (LALInference_skymap.fits.gz)

Properties

Properties of the plane "Int MOC 0.9 LALInference_sky..."

PlaneID: yprobcc_cWB_complete.fits

Origin: Computed by Aladin

Color:

Format: Multi-Order Coverage map (MOC)

Coverage: 0.559% of sky => 230.5^2

Best MOC ang.res: 6.871' (moc order=9)

Drawing method: borders fill in

Coord.sys.: ICRS

Overlay opacity/transparency

Apply Close

180° x 106.4°

grid wink north hdr multiview match

Search

cont pixel prop del

Frame: ICRS

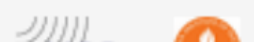
+180 +90 -180 -90

07:01:28.45 -47:35:59.1

180° x 106.4°

0 sel / 0 src 26fps / 276Mb

(c) 2016 Unistra/CNRS - by CDS - Distributed under GNU GPL v3



Skymap Viewer

A sky atlas for understanding LIGO-Virgo skymaps. Help [here](#), or with [this](#) Skymap Viewer. Plenty simulated skymaps [here](#). If you do not see the big dark sky map, look below. Zoom in on the sky with the + and - at the right of the sky.



LIGO-Virgo Skymaps

This is skymap
GW150914:CWB.

50% area = 98.19 sq deg
90% area = 308.2 sq deg



Show Weighted Galaxies (or [table](#)).

Time and Place

Universal time

2015-09-14T09:50:45

E Longitude Latitude

Sun = and = Moon

Catalog Sources

Click the Layers icon to switch on catalogs. If you click on the sources on the sky, information will appear here with links to Simbad and NED.

Zoomable Multiwavelength Sky

Zoom in on the sky with the mouse or the +/- icons

J2000

09 06 55.75 76 44 48.12



For astronomers might be useful to visualize the MOC operations in astronomical context in the Skymap Viewer



FoV: 180°





Some datasever, such as Vizier, can be **queried by MOC** in order to return data (galaxy catalogs/list of images) only inside the MOC coverage.

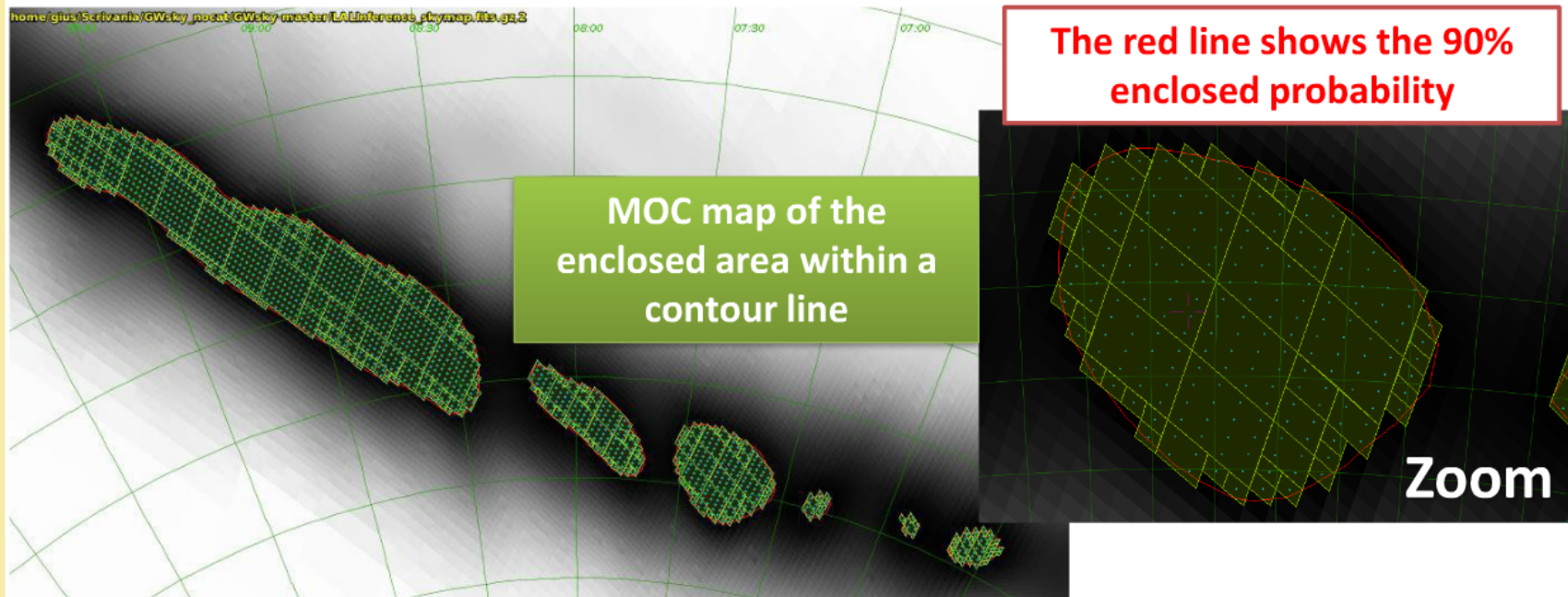
*To identify likely host galaxies of a GW event
we need to collect **as much information as possible***



post-processing: to organize data from queries.

TOPCAT

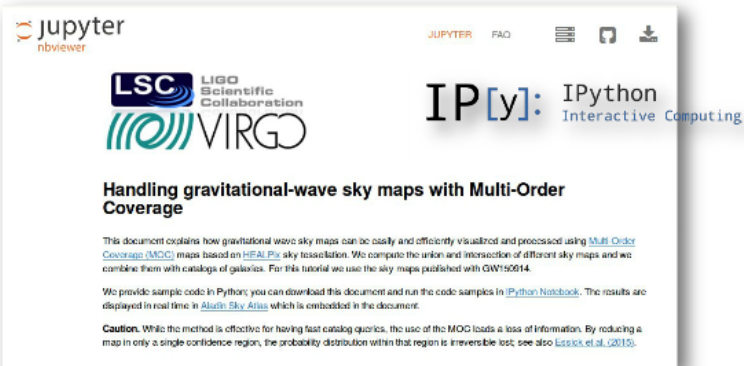
MOC representation of sky areas enclosed into iso-contour lines



`MOC_area_prob(infile, percentage, 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.

MOCs at work



jupyter nbviewer JUPYTER FAQ

LSC LIGO Scientific Collaboration

VIRGO

IP[y]: IPython Interactive Computing

Handling gravitational-wave sky maps with Multi-Order Coverage

This document explains how gravitational wave sky maps can be easily and efficiently visualized and processed using [Multi-Order Coverage \(MOC\)](#) maps based on [HEALPix](#) sky tessellation. We compute the union and intersection of different sky maps and we combine them with catalogs of galaxies. For this tutorial we use the sky maps published with GW150914.

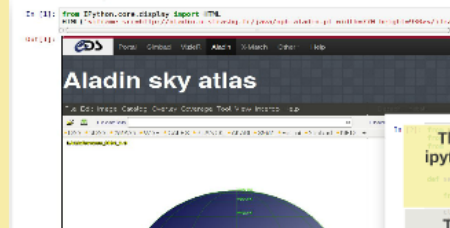
We provide sample code in Python; you can download this document and run the code samples in [IPython Notebook](#). The results are displayed in real time in [Aladin Sky Atlas](#) which is embedded in the document.

Caution. While the method is effective for having fast catalog queries, the use of the MOC loads a loss of information. By reducing a map in only a single confidence region, the probability distribution within that region is irreversible lost; see also [Lesicki et al., 2015](#).

ipython tutorial:
<http://nbviewer.jupyter.org/gist/ggreco77/5fc0cc2777f9edd446b459459db830e9>

Video Tutorial:
<https://vimeo.com/167173587>

IPython Notebook powered by Aladin Sky Atlas



SAMP:
Simple Application Messaging Protocol

The Aladin java applet is embedded in the ipython cell, as the code runs the results are displayed in real time in Aladin planes

The interoperability between Aladin and ipython notebook is obtained using the SAMPIntegredClient with a set of dedicated functions

The Aladin Console commands are "converted" in python string

```
rename_plane = 'rename' + ' ' + plane
```





IP[y]: IPython
Interactive Computing

Handling gravitational-wave sky maps with Multi-Order Coverage

This document explains how gravitational wave sky maps can be easily and efficiently visualized and processed using [Multi-Order Coverage \(MOC\)](#) maps based on [HEALPix](#) sky tessellation. We compute the union and intersection of different sky maps and we combine them with catalogs of galaxies. For this tutorial we use the sky maps published with GW150914.

We provide sample code in Python; you can download this document and run the code samples in [IPython Notebook](#). The results are displayed in real time in [Aladin Sky Atlas](#) which is embedded in the document.

Caution. While the method is effective for having fast catalog queries, the use of the MOC leads a loss of information. By reducing a map in only a single confidence region, the probability distribution within that region is irreversible lost; see also [Essick et al. \(2015\)](#).

IPython tutorial:

<http://nbviewer.jupyter.org/gist/ggreco77/5fc0cc2777f9edd446b459459db830e9>

Video Tutorial:

<https://vimeo.com/167173587>

IPython Notebook powered by Aladin Sky Atlas

```
In [1]: from IPython.core.display import HTML
HTML('<iframe src=http://aladin.u-strasbg.fr/java/nph-aladin.pl width=770 height=930></iframe>')
```



SAMP:
Simple Application Messaging Protocol

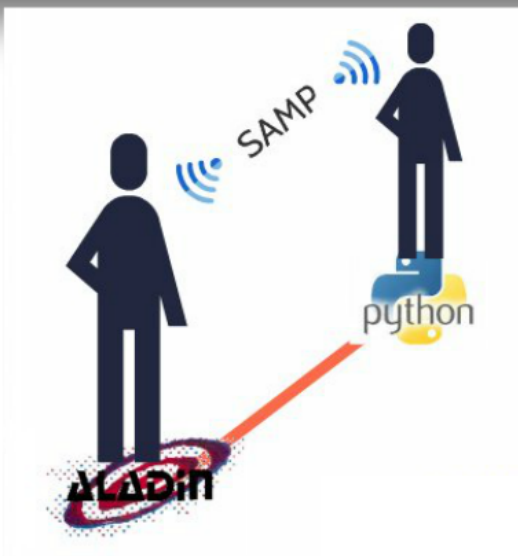
```
In [2]: from astropy.utils.data import download_file
from astropy.utils.scripts import get_list_of_urls
from astropy.utils.scripts import get_list_of_urls
def send_script( script ):
    from astropy.vo.samp import SAMPIntegatedClient
    client = SAMPIntegatedClient()
    client.notify_all( message )
    client.disconnect()
```

The Aladin java applet is embedded in the ipython cell, as the code runs the results are displayed in real time in Aladin planes

The interoperability between Aladin and Ipython notebook is obtained using the SAMPIntegatedClient with a set of dedicated functions

The Aladin Console commands are "converted" in python string

```
rename_plane = 'rename' + ' ' + plane
```



Home x Handling gravitation... x

localhost:8889/notebooks/Handling gravitational-wave skymaps with Multi-Order Co Search

jupyter Handling gravitational-wave skymaps with Multi-Order Coverage Last Checkpoint: 2 minutes ago (autosaved) Python 2

File Edit View Insert Cell Kernel Help

Code CellToolbar



Handling gravitational-wave sky maps with Multi-Order Coverage

This document explains how gravitational wave sky maps can be easily and efficiently visualized and processed using [Multi-Order Coverage \(MOC\)](#) maps based on [HEALPix](#) sky tessellation. We compute the union and intersection of different sky maps and we combine them with catalogs of galaxies. For this illustration we use the sky maps published with GW150914.

We provide sample code in Python; you can download this document and run the code samples in [IPython Notebook](#). The results are in real time displayed in [Aladin Sky Atlas](#) that is embedded in the document.

Caution. While the method is effective for having fast catalog queries, the use of the MOC leads a loss of information. By reducing a map in only a single confidence region, the probability distribution within that region is irreversible lost; see [Essick et al. \(2015\)](#).

Table of Contents

[Forewords about GW150914](#)

- [1. Installation and initialization](#)
 - [A. Sky map visualization with Aladin](#)
 - [B. Python packages](#)
- [2. Multiscale meshes of gravitational-wave sky maps using MOC](#)
 - [A. cWB sky map](#)
 - [B. LIB sky map](#)
 - [C. BAYESTAR sky map](#)
 - [D. LALInference sky map](#)
- [3. Handling and combining multiple sky maps](#)
 - [A. \$cWB \cup LIB\$](#)
 - [B. \$cWB \cap LIB\$](#)
 - [C. \$BAYESTAR \cup LALInference\$](#)
 - [D. \$\(cWB \cup LIB\) \cap \(BAYESTAR \cup LALInference\)\$](#)
- [4. Query Catalogs from MOCs](#)
 - [A. Query a single catalog](#)

MOC Basic Algorithm

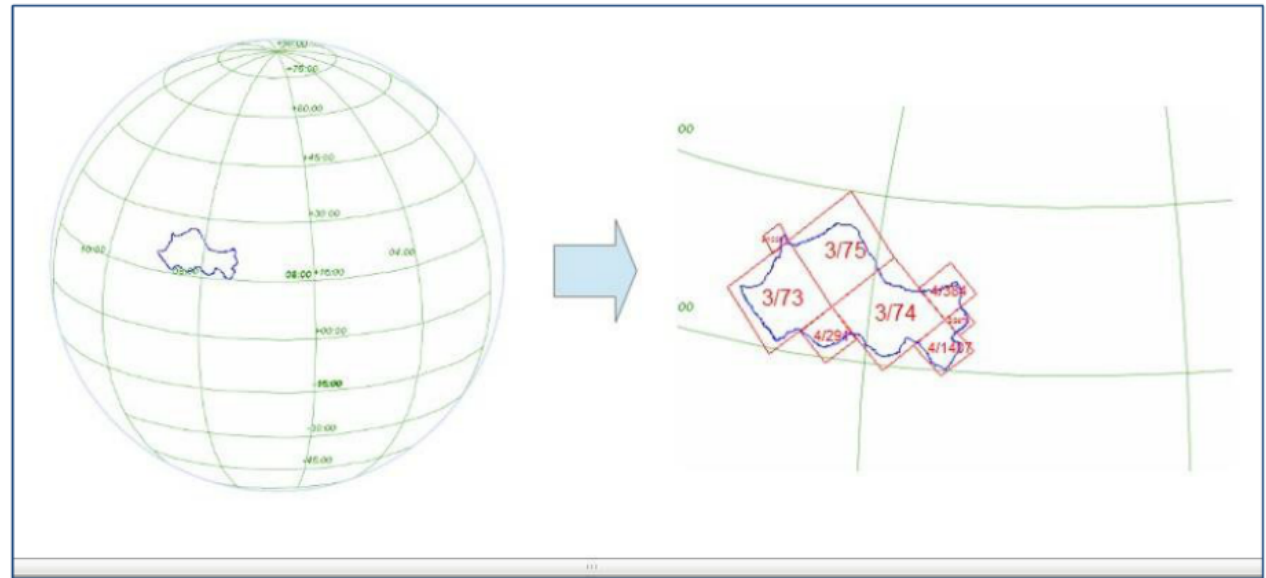
Each MOC cell is defined by two numbers: the hierarchy level (HEALPIX ORDER) and the pixel index (HEALPIX NPIX).

The NUNIQ scheme defines an algorithm for packing an (ORDER, NPIX) pair into a single integer for compactness:

$$uniq = 4 \times 4^{order} + npix$$

Fernique et al. 2014

<http://ivoa.net/documents/MOC/20140602/REC-MOC-1.0-20140602.pdf>





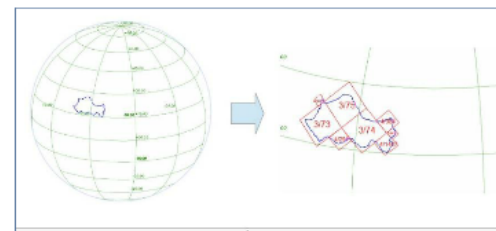
The code correctly traces out the original underlying mesh in the HEALPix image?

MOC: No space projection, No smoothing

The screenshot shows a software window titled "LALInference_MOC_0.txt" with a list of HEALPix cell identifiers. The list is displayed in a text editor format, showing a sequence of integers representing cell IDs. A blue arrow points from the text "List of HEALPix cell identifiers" to the list. The background of the software window shows a dark image of a galaxy with a white grid overlay representing the MOC cells.

MOC Basic Algorithm

Each MOC cell is defined by two numbers: the hierarchy level (HEALPIX ORDER) and the pixel index (HEALPIX NPIX).



The NUNIQ scheme defines an algorithm for packing an (ORDER, NPIX) pair into a single integer for compactness:



$$uniq = 4 \times 4^{order} + npix$$

Fernique et al. 2014

<http://ivoa.net/documents/MOC/20140602/REC-MOC-1.0-20140602.pdf>

MOC: No space projection, No smoothing

The image shows a screenshot of the Aladin v9.0 software interface on the left and a text editor window on the right. The Aladin window displays a star field with a white grid overlay. A blue double-headed arrow points from the text "List of HEALPix cell identifiers" to the grid. The text editor window shows a list of HEALPix cell identifiers, organized into groups labeled "4", "5", "6", and "7".

Aladin v9.0 *** BETA VERSION (based on v9.024) ***
Image Catalog Overlay Coverage Tool View Interop Help

Location
SDSS 2MASS WISE GALEX PLANCK AKARI XMM Fermi Simbad NE

red

180° x 106.4°

north hdr multiview match

Aladin/CNRS - by CDS - Distributed under GNU GPL v3

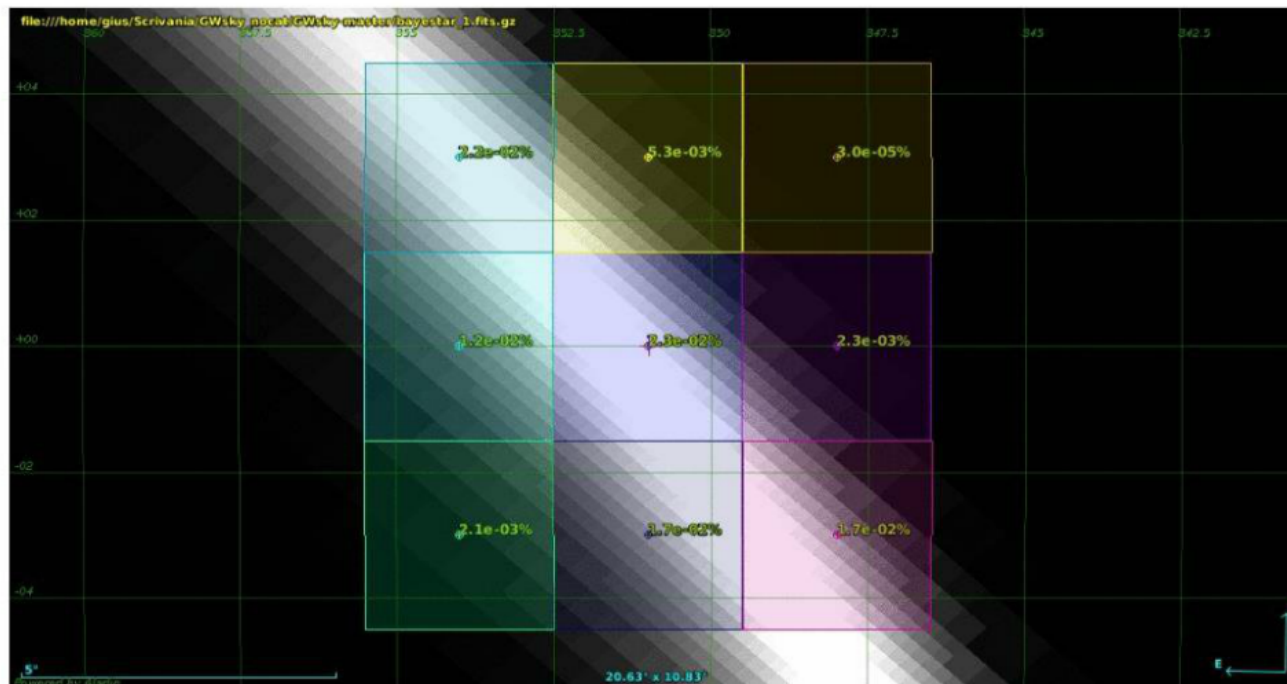
LALInference_MOC_0.txt (~/Scrivania) - gedit

LALInference_MOC_0.txt x

```
["4":  
[2065, 2067, 2068, 2070, 2073, 2075, 2076, 2078, 2100, 2318, 2319, 2330, 2331, 2334, 2335,  
2337, 2338, 2340, 2378],  
"5":  
[6702, 8257, 8259, 8265, 8267, 8276, 8278, 8289, 8291, 8389, 8404, 8406, 8408, 8409, 8411,  
8412, 8414, 9261, 9262, 9263, 9314, 9315, 9318, 9319, 9345, 9346, 9347, 9356, 9357, 9358,  
9364, 9365, 9366, 9368, 9376, 9377, 9378, 9408, 9409, 9412, 9413, 9424, 9425, 9428, 9429,  
9518, 9519, 9530, 9600, 9601, 9604, 9605, 9616, 9617, 9620],  
"6":  
[26758, 26764, 26765, 26767, 26778, 26789, 26800, 26802, 26803, 26814, 27153, 27156, 27158, 30  
27165, 27167, 27189, 33027, 33033, 33034, 33035, 33056, 33057, 33059, 33065, 33067, 33108, 33  
33138, 33144, 33146, 33147, 33153, 33188, 33189, 33191, 33232, 33234, 33240, 33266, 33272, 33  
33553, 33555, 33564, 33565, 33567, 33641, 33652, 33654, 33660, 33662, 33744, 33745, 33747, 33  
37023, 37037, 37038, 37039, 37042, 37043, 37066, 37067, 37070, 37071, 37082, 37083, 37085, 37  
37322, 37323, 37326, 37327, 37338, 37339, 37342, 37343, 37377, 37378, 37379, 37436, 37437, 37  
37469, 37476, 37477, 37478, 37480, 37516, 37517, 37520, 37521, 37522, 37536, 37538, 37640, 37  
37645, 37656, 38026, 38064, 38066, 38067, 38070, 38071, 38126, 38127, 38138, 38484, 38486, 38  
"7":  
[106668, 106670, 106671, 107012, 107013, 107015, 107021, 107024, 107026, 107027, 107036, 10  
107106, 107153, 107164, 107165, 107167, 107189, 107191, 107206, 107248, 107250, 107262, 108  
108623, 108628, 108630, 108645, 108664, 108665, 108667, 108753, 108764, 108765, 108767, 108  
132103, 132107, 132129, 132130, 132131, 132233, 132235, 132257, 132265, 132464, 132548, 132  
132620, 132621, 132623, 132645, 132647, 132653, 132655, 132761, 132788, 132789, 132791, 132  
132970, 133058, 133092, 133094, 134209, 134245, 134265, 134267, 134353, 134356, 134357, 134  
134522, 134523, 134561, 134573, 134614, 134620, 134622, 134644, 134646, 134652, 134654, 134  
134984, 134985, 134996, 134998, 135004, 135006, 135024, 135025, 135027, 135028, 135030, 148  
148143, 148145, 148146, 148147, 148165, 148166, 148167, 148275, 148278, 148279, 148321, 148  
148327, 148337, 148338, 148339, 149007, 149019, 149022, 149023, 149066, 149280, 149282, 149  
149299, 149302, 149303, 149346, 149347, 149350, 149351, 149362, 149363, 149507, 149756, 149  
149882, 149884, 149885, 149916, 149924, 149925, 149926, 149936, 149952, 150072, 150073, 150  
150092, 150093, 150096, 150097, 150098, 150112, 150148, 150149, 150150, 150176, 150628, 150  
150821, 150832, 150833, 150836, 150837, 150880, 150881, 152108, 152110, 152111, 152122, 152  
152263, 152274, 152275, 152278, 152456, 152458, 152459, 152462, 152496, 152498, 152499, 152  
152558, 152559, 152570, 153713, 153716, 153717, 153888, 153889, 153892, 153893, 153895, 153  
153911, 153912, 153913, 153914, 153915, 153916, 153917, 153918, 153919, 153920, 153921, 153922, 153923, 153924, 153925, 153926, 153927, 153928, 153929, 153930, 153931, 153932, 153933, 153934, 153935, 153936, 153937, 153938, 153939, 153940, 153941, 153942, 153943, 153944, 153945, 153946, 153947, 153948, 153949, 153950, 153951, 153952, 153953, 153954, 153955, 153956, 153957, 153958, 153959, 153960, 153961, 153962, 153963, 153964, 153965, 153966, 153967, 153968, 153969, 153970, 153971, 153972, 153973, 153974, 153975, 153976, 153977, 153978, 153979, 153980, 153981, 153982, 153983, 153984, 153985, 153986, 153987, 153988, 153989, 153990, 153991, 153992, 153993, 153994, 153995, 153996, 153997, 153998, 153999, 154000, 154001, 154002, 154003, 154004, 154005, 154006, 154007, 154008, 154009, 154010, 154011, 154012, 154013, 154014, 154015, 154016, 154017, 154018, 154019, 154020, 154021, 154022, 154023, 154024, 154025, 154026, 154027, 154028, 154029, 154030, 154031, 154032, 154033, 154034, 154035, 154036, 154037, 154038, 154039, 154040, 154041, 154042, 154043, 154044, 154045, 154046, 154047, 154048, 154049, 154050, 154051, 154052, 154053, 154054, 154055, 154056, 154057, 154058, 154059, 154060, 154061, 154062, 154063, 154064, 154065, 154066, 154067, 154068, 154069, 154070, 154071, 154072, 154073, 154074, 154075, 154076, 154077, 154078, 154079, 154080, 154081, 154082, 154083, 154084, 154085, 154086, 154087, 154088, 154089, 154090, 154091, 154092, 154093, 154094, 154095, 154096, 154097, 154098, 154099, 154100, 154101, 154102, 154103, 154104, 154105, 154106, 154107, 154108, 154109, 154110, 154111, 154112, 154113, 154114, 154115, 154116, 154117, 154118, 154119, 154120, 154121, 154122, 154123, 154124, 154125, 154126, 154127, 154128, 154129, 154130, 154131, 154132, 154133, 154134, 154135, 154136, 154137, 154138, 154139, 154140, 154141, 154142, 154143, 154144, 154145, 154146, 154147, 154148, 154149, 154150, 154151, 154152, 154153, 154154, 154155, 154156, 154157, 154158, 154159, 154160, 154161, 154162, 154163, 154164, 154165, 154166, 154167, 154168, 154169, 154170, 154171, 154172, 154173, 154174, 154175, 154176, 154177, 154178, 154179, 154180, 154181, 154182, 154183, 154184, 154185, 154186, 154187, 154188, 154189, 154190, 154191, 154192, 154193, 154194, 154195, 154196, 154197, 154198, 154199, 154200, 154201, 154202, 154203, 154204, 154205, 154206, 154207, 154208, 154209, 154210, 154211, 154212, 154213, 154214, 154215, 154216, 154217, 154218, 154219, 154220, 154221, 154222, 154223, 154224, 154225, 154226, 154227, 154228, 154229, 154230, 154231, 154232, 154233, 154234, 154235, 154236, 154237, 154238, 154239, 154240, 154241, 154242, 154243, 154244, 154245, 154246, 154247, 154248, 154249, 154250, 154251, 154252, 154253, 154254, 154255, 154256, 154257, 154258, 154259, 154260, 154261, 154262, 154263, 154264, 154265, 154266, 154267, 154268, 154269, 154270, 154271, 154272, 154273, 154274, 154275, 154276, 154277, 154278, 154279, 154280, 154281, 154282, 154283, 154284, 154285, 154286, 154287, 154288, 154289, 154290, 154291, 154292, 154293, 154294, 154295, 154296, 154297, 154298, 154299, 154300, 154301, 154302, 154303, 154304, 154305, 154306, 154307, 154308, 154309, 154310, 154311, 154312, 154313, 154314, 154315, 154316, 154317, 154318, 154319, 154320, 154321, 154322, 154323, 154324, 154325, 154326, 154327, 154328, 154329, 154330, 154331, 154332, 154333, 154334, 154335, 154336, 154337, 154338, 154339, 154340, 154341, 154342, 154343, 154344, 154345, 154346, 154347, 154348, 154349, 154350, 154351, 154352, 154353, 154354, 154355, 154356, 154357, 154358, 154359, 154360, 154361, 154362, 154363, 154364, 154365, 154366, 154367, 154368, 154369, 154370, 154371, 154372, 154373, 154374, 154375, 154376, 154377, 154378, 154379, 154380, 154381, 154382, 154383, 154384, 154385, 154386, 154387, 154388, 154389, 154390, 154391, 154392, 154393, 154394, 154395, 154396, 154397, 154398, 154399, 154400, 154401, 154402, 154403, 154404, 154405, 154406, 154407, 154408, 154409, 154410, 154411, 154412, 154413, 154414, 154415, 154416, 154417, 154418, 154419, 154420, 154421, 154422, 154423, 154424, 154425, 154426, 154427, 154428, 154429, 154430, 154431, 154432, 154433, 154434, 154435, 154436, 154437, 154438, 154439, 154440, 154441, 154442, 154443, 154444, 154445, 154446, 154447, 154448, 154449, 154450, 154451, 154452, 154453, 154454, 154455, 154456, 154457, 154458, 154459, 154460, 154461, 154462, 154463, 154464, 154465, 154466, 154467, 154468, 154469, 154470, 154471, 154472, 154473, 154474, 154475, 154476, 154477, 154478, 154479, 154480, 154481, 154482, 154483, 154484, 154485, 154486, 154487, 154488, 154489, 154490, 154491, 154492, 154493, 154494, 154495, 154496, 154497, 154498, 154499, 154500, 154501, 154502, 154503, 154504, 154505, 154506, 154507, 154508, 154509, 154510, 154511, 154512, 154513, 154514, 154515, 154516, 154517, 154518, 154519, 154520, 154521, 154522, 154523, 154524, 154525, 154526, 154527, 154528, 154529, 154530, 154531, 154532, 154533, 154534, 154535, 154536, 154537, 154538, 154539, 154540, 154541, 154542, 154543, 154544, 154545, 154546, 154547, 154548, 154549, 154550, 154551, 154552, 154553, 154554, 154555, 154556, 154557, 154558, 154559, 154560, 154561, 154562, 154563, 154564, 154565, 154566, 154567, 154568, 154569, 154570, 154571, 154572, 154573, 154574, 154575, 154576, 154577, 154578, 154579, 154580, 154581, 154582, 154583, 154584, 154585, 154586, 154587, 154588, 154589, 154590, 154591, 154592, 154593, 154594, 154595, 154596, 154597, 154598, 154599, 154600, 154601, 154602, 154603, 154604, 154605, 154606, 154607, 154608, 154609, 154610, 154611, 154612, 154613, 154614, 154615, 154616, 154617, 154618, 154619, 154620, 154621, 154622, 154623, 154624, 154625, 154626, 154627, 154628, 154629, 154630, 154631, 154632, 154633, 154634, 154635, 154636, 154637, 154638, 154639, 154640, 154641, 154642, 154643, 154644, 154645, 154646, 154647, 154648, 154649, 154650, 154651, 154652, 154653, 154654, 154655, 154656, 154657, 154658, 154659, 154660, 154661, 154662, 154663, 154664, 154665, 154666, 154667, 154668, 154669, 154670, 154671, 154672, 154673, 154674, 154675, 154676, 154677, 154678, 154679, 154680, 154681, 154682, 154683, 154684, 154685, 154686, 154687, 154688, 154689, 154690, 154691, 154692, 154693, 154694, 154695, 154696, 154697, 154698, 154699, 154700, 154701, 154702, 154703, 154704, 154705, 154706, 154707, 154708, 154709, 154710, 154711, 154712, 154713, 154714, 154715, 154716, 154717, 154718, 154719, 154720, 154721, 154722, 154723, 154724, 154725, 154726, 154727, 154728, 154729, 154730, 154731, 154732, 154733, 154734, 154735, 154736, 154737, 154738, 154739, 154740, 154741, 154742, 154743, 154744, 154745, 154746, 154747, 154748, 154749, 154750, 154751, 154752, 154753, 154754, 154755, 154756, 154757, 154758, 154759, 154760, 154761, 154762, 154763, 154764, 154765, 154766, 154767, 154768, 154769, 154770, 154771, 154772, 154773, 154774, 154775, 154776, 154777, 154778, 154779, 154780, 154781, 154782, 154783, 154784, 154785, 154786, 154787, 154788, 154789, 154790, 154791, 154792, 154793, 154794, 154795, 154796, 154797, 154798, 154799, 154800, 154801, 154802, 154803, 154804, 154805, 154806, 154807, 154808, 154809, 154810, 154811, 154812, 154813, 154814, 154815, 154816, 154817, 154818, 154819, 154820, 154821, 154822, 154823, 154824, 154825, 154826, 154827, 154828, 154829, 154830, 154831, 154832, 154833, 154834, 154835, 154836, 154837, 154838, 154839, 154840, 154841, 154842, 154843, 154844, 154845, 154846, 154847, 154848, 154849, 154850, 154851, 154852, 154853, 154854, 154855, 154856, 154857, 154858, 154859, 154860, 154861, 154862, 154863, 154864, 154865, 154866, 154867, 154868, 154869, 154870, 154871, 154872, 154873, 154874, 154875, 154876, 154877, 154878, 154879, 154880, 154881, 154882, 154883, 154884, 154885, 154886, 154887, 154888, 154889, 154890, 154891, 154892, 154893, 154894, 154895, 154896, 154897, 154898, 154899, 154900, 154901, 154902, 154903, 154904, 154905, 154906, 154907, 154908, 154909, 154910, 154911, 154912, 154913, 154914, 154915, 154916, 154917, 154918, 154919, 154920, 154921, 154922, 154923, 154924, 154925, 154926, 154927, 154928, 154929, 154930, 154931, 154932, 154933, 154934, 154935, 154936, 154937, 154938, 154939, 154940, 154941, 154942, 154943, 154944, 154945, 154946, 154947, 154948, 154949, 154950, 154951, 154952, 154953, 154954, 154955, 154956, 154957, 154958, 154959, 154960, 154961, 154962, 154963, 154964, 154965, 154966, 154967, 154968, 154969, 154970, 154971, 154972, 154973, 154974, 154975, 154976, 154977, 154978, 154979, 154980, 154981, 154982, 154983, 154984, 154985, 154986, 154987, 154988, 154989, 154990, 154991, 154992, 154993, 154994, 154995, 154996, 154997, 154998, 154999, 155000, 155001, 155002, 155003, 155004, 155005, 155006, 155007, 155008, 155009, 155010, 155011, 155012, 155013, 155014, 155015, 155016, 155017, 155018, 155019, 155020, 155021, 155022, 155023, 155024, 155025, 155026, 155027, 155028, 155029, 155030, 155031, 155032, 155033, 155034, 155035, 155036, 155037, 155038, 155039, 155040, 155041, 155042, 155043, 155044, 155045, 155046, 155047, 155048, 155049, 155050, 155051, 155052, 155053, 155054, 155055, 155056, 155057, 155058, 155059, 155060, 155061, 155062, 155063, 155064, 155065, 155066, 155067, 155068, 155069, 155070, 155071
```

GWsky: tiling the skymap in FoV

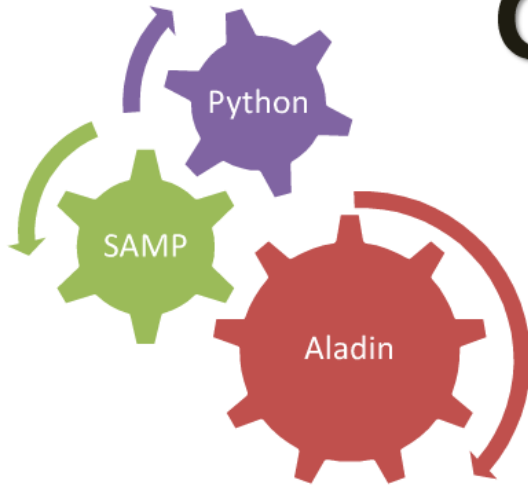
GWsky is an interactive Python script to generate a sequence of pointings given a specific Field of View



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



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

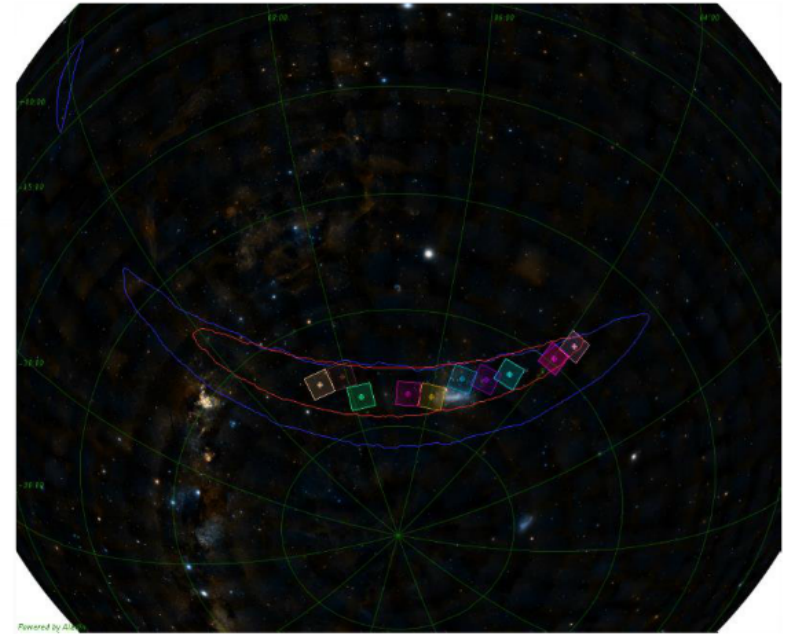
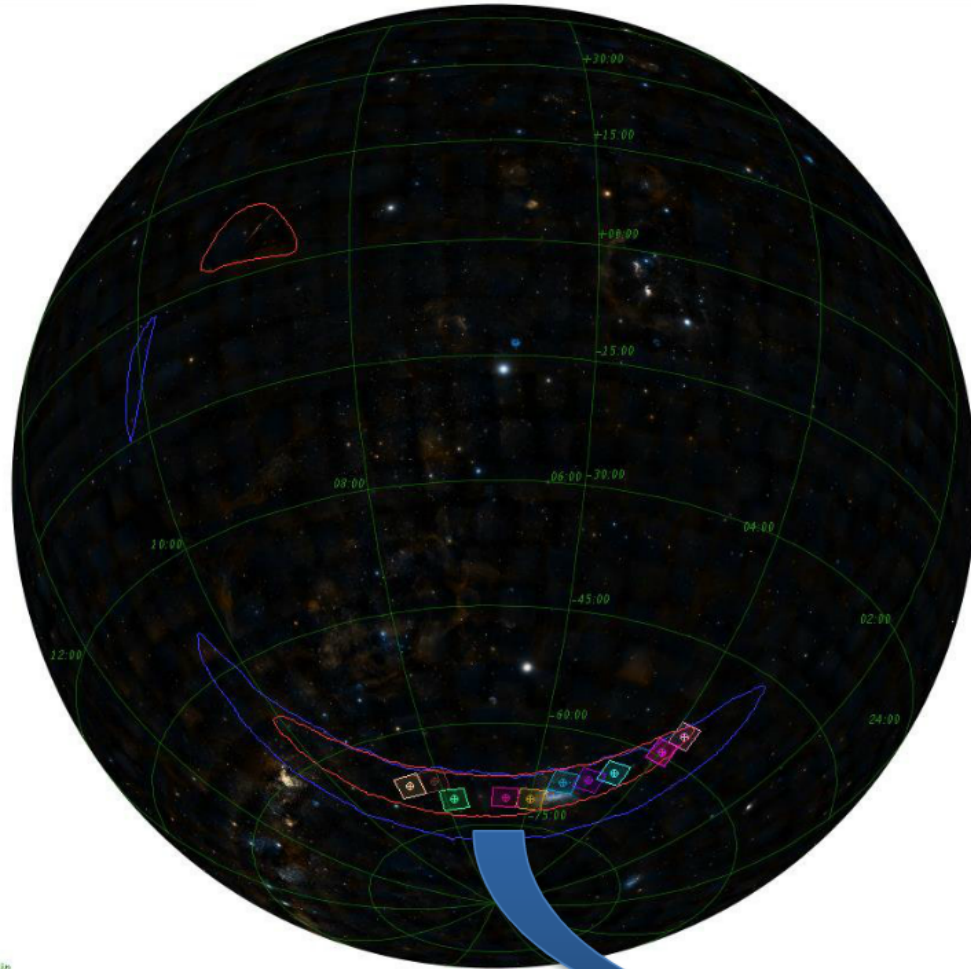


R repeats the last action



Q quit

VST Observation of GW 150914



Survey Area Definition Tool (VST)

File Options Help

Survey ID

Survey Areas

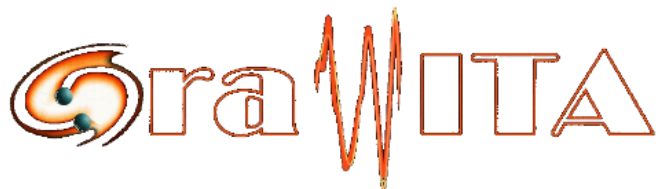
Type	Lon	Lat	Diameter (d...)	---	Angle (d...	System	Exclude
Coordinate Range	30.0	-2.0	35.0	1.2		0 Galactic	<input type="checkbox"/>
Coordinate Range	19:10:00	-02:00:00	19:30:00	+02:00:00		0 FK5 (J20...	<input type="checkbox"/>
Geodesic Rectangle	19:20:00	-07:00:00	5.0	4.0		-20 FK5 (J20...	<input type="checkbox"/>
Circle	26.0	-2.5	4.5			0 Galactic	<input checked="" type="checkbox"/>

Select Dither Pattern.

OMEGACAM_Dither_diag_5

Select Catalogue

GSC-2 at ESO



Conclusion and Future Perspectives

SAMP + Aladin console commands offer the ability to build interactive tools useful to astronomers as **GWsky for Grawita**

MOC offers a dynamic concept for skymap (logical operations and simultaneous queries)

A **python package** for the Aladin console commands (converting in python string) might be useful

The MOC could be implemented in **Skymap Viewer** (to show the MOC operation in astrophysical context)

TOPCAT might be qualified for post-processing analysis to organize the query data