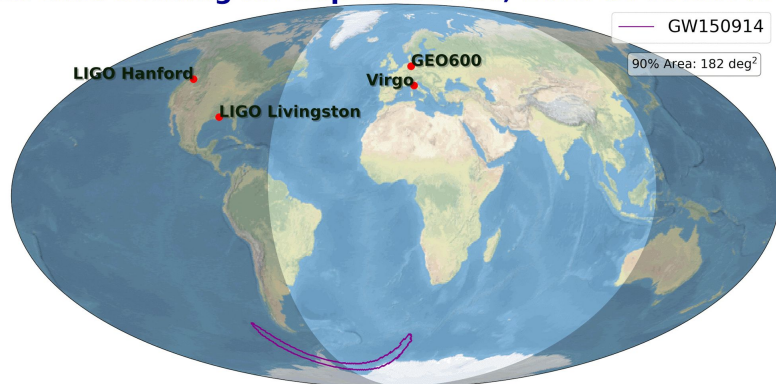


Multi Order Coverage data structure to plan multi-messenger observations

Night time shading for September 14, 2015 at 09:50:45 UTC



GWTC-1/2: Gravitational-Wave Transient Catalogs - Sky Localizations

Powered by Cartopy; Projection: Mollweide.

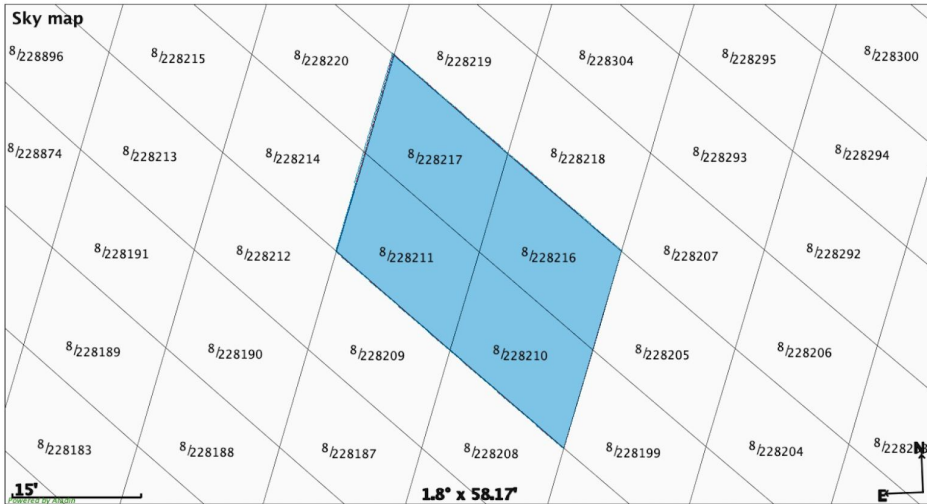
Giuseppe Greco

INFN-Perugia

WP4 Technology Forum 2 - Observatoire de Strasbourg



A Brief Description of MOC



The shaded area highlights a MOC map consisting of 4 cells at order = 8.

A JSON MOC is written following the syntax:

```
{"order": [npix, npix,...], "order": [npix, npix...], ...}
```

→ {"8": [228210, 228211, 228216, 228217]}

Ada Nebod opened an issue to standardise the JSON format.

The MOC encoding method based on HEALPix tessellation was originally developed at the Centre de Données astronomiques de Strasbourg (CDS) and has been adopted as a recommendation by the International Virtual Observatory Alliance (IVOA). Initially designed for manipulating sky coverages from astronomical surveys, MOC has been extended to support both temporal and spatial coverage, known as Space-Time MOC (ST-MOC); *Pierre Fernique et al., 2014 and 2020.*

Determining multi-observability within a MOC coverage

Flatten MOC at a fixed order

From MOC cells to sky coordinates

Computing visibility
from observatory sites

Define visibility MOC regions

A MOC is a collection of cells (from HEALPix) stored at different order. To calculate the airmass in sky regions with equal areas we flat the MOC at a predefined order. Predefined MOC order >= max order presented in the original MOC.

Convert HEALPix indices to celestial coordinates using `healpix_to_skycoord` method from `cds-healpix-python`. The input is a 1-D array of HEALPix indices and the resulting celestial coordinates are returned in a **SkyCoord** high-level object as defined in astropy.

The module `astroplan` is used to compute the visibility of each SkyCoord from different observatories by specifying the location and the airmass and/or moon illumination/separation values.

An astropy table merges SkyCoord and measurements in the previous step. Next, the table is filtered to specify the user-defined constraints. Using `from_skycoords` method in `mocpy`, a new MOC is recreated: "the visibility MOC".

A practical example: an ideal observation campaign

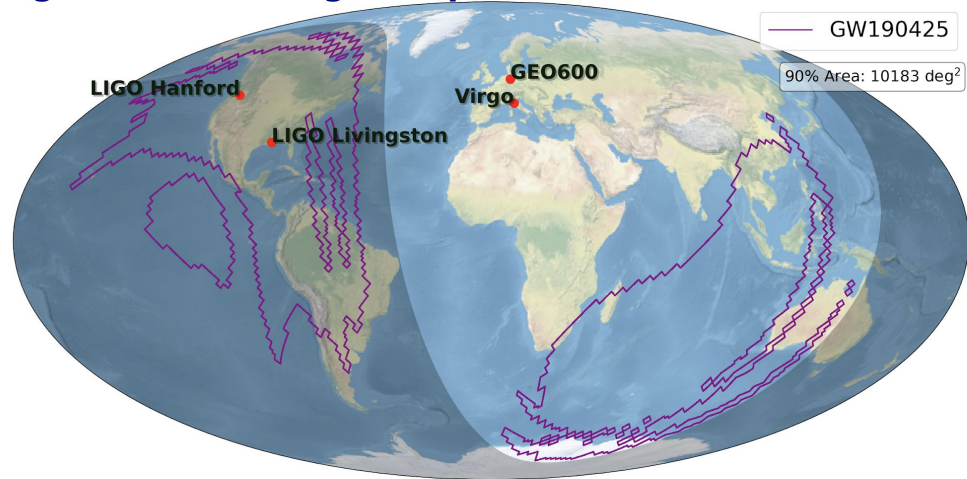
Multi-messenger team:

1. Haleakala Observatories in Hawaii, USA
2. Paranal Observatory in Chile
3. Siding Spring Observatory (SSO) in Australia

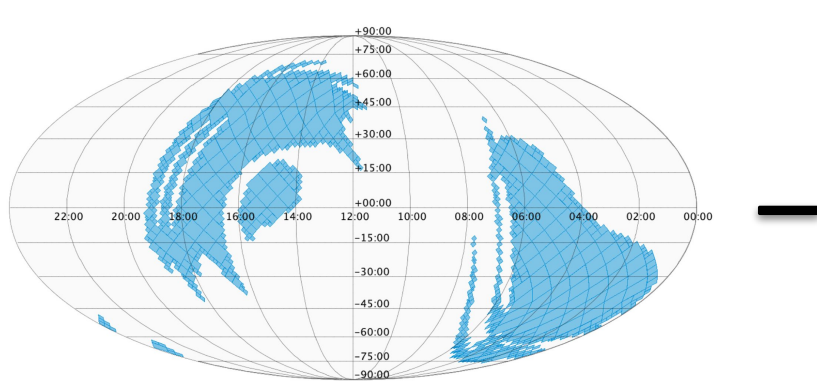
With `astroplan` we define three **Observer** classes. `Observer` is an `astroplan` container class for information about an observer's site

Observation of GW190425. The signal has been observed on 2019 April 25, 08:18:05 UTC, during the third observing run (O3) of the LIGO–Virgo network. The network consists of two Advanced LIGO interferometers in Hanford, Washington, USA (LHO) and Livingston, Louisiana, USA (LLO) and the Advanced Virgo interferometer in Cascina, Italy. At the time of GW190425, LHO was temporarily offline with only LLO and Virgo taking data.

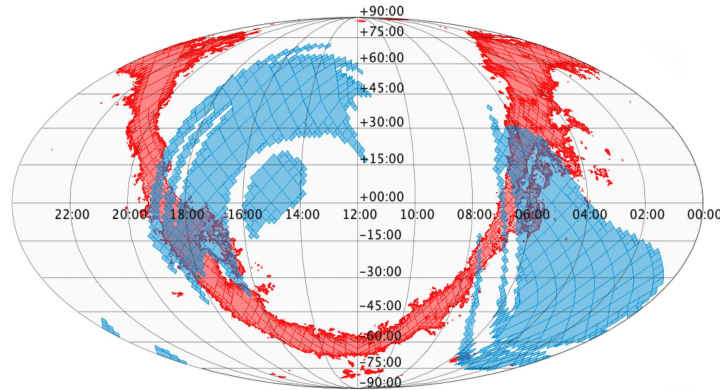
Night time shading for April 25, 2019 at 08:18:05 UTC



Removing high dust extinction areas



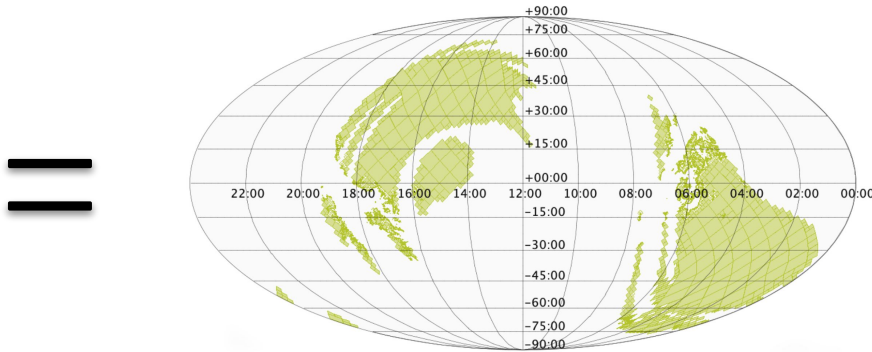
A gravitational-wave sky localization - 90% credible region.



A selected high Galactic dust extinction region is overlaid.



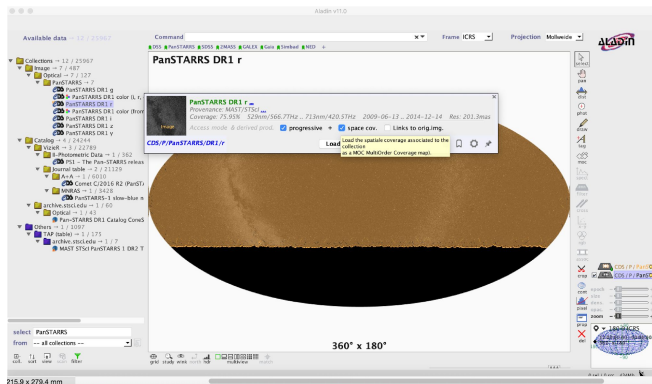
The extinction skymap will be provided by CADE - Analysis Center for Extended Data



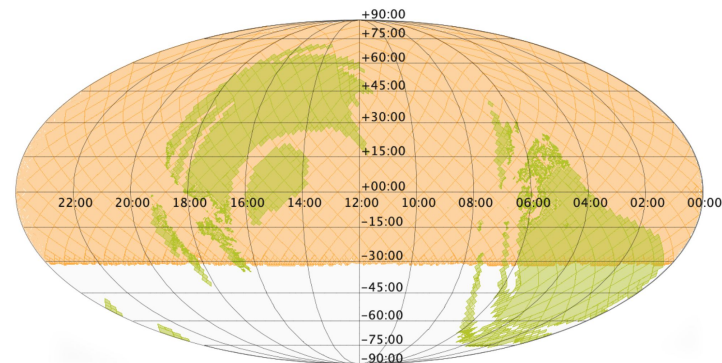
The resulting sky map with the dust absorption fields subtracted.

Generally, it could be realistic to exclude regions of high estimates of Galactic dust extinction or areas with a stellar density too high for specific transient search.

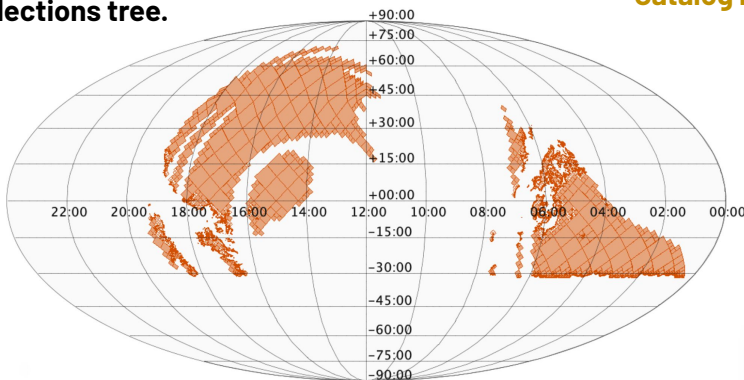
Searching for reference images



Exploring the Aladin data collections tree.



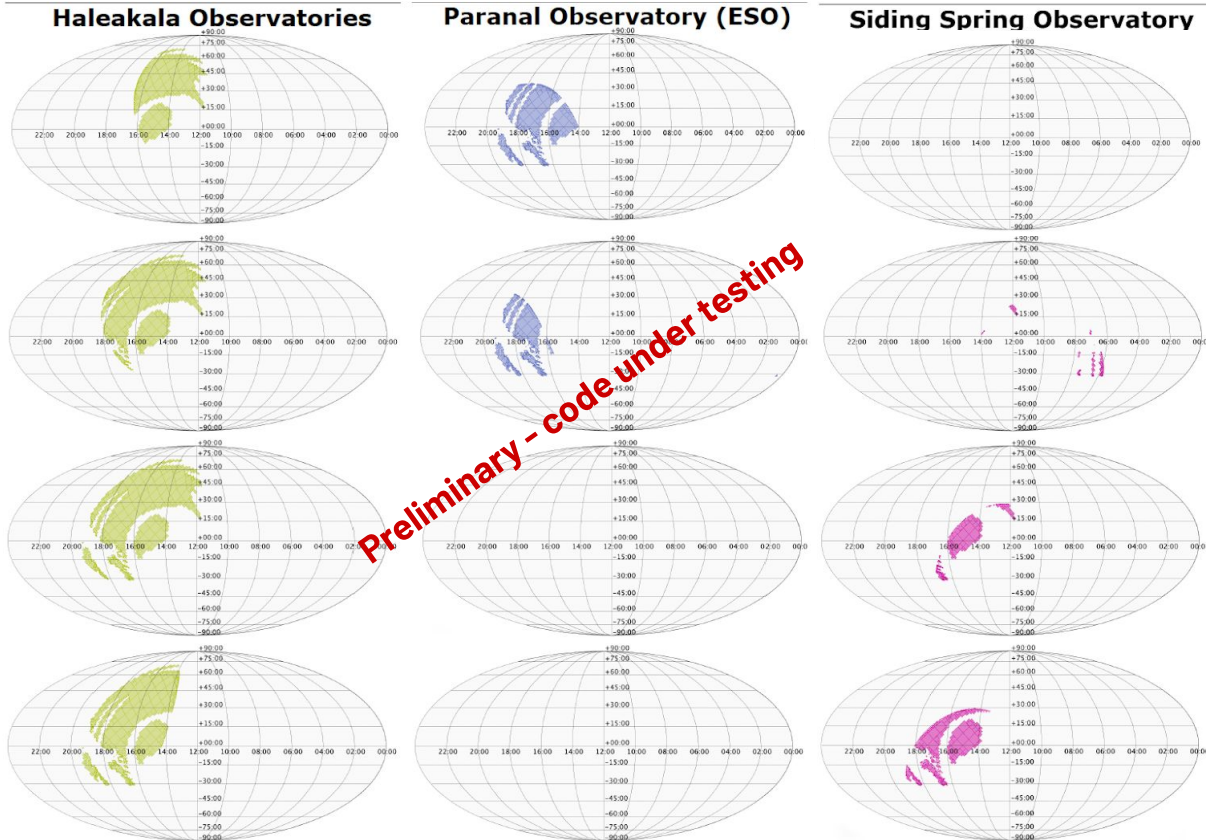
Catalog image selection: PanSTARRS DR1.



The resulting intersection area between the gravitational wave sky localization in which the high Galactic dust regions are removed and PanSTARRS DR1.

To search for convenient reference images, we explore the public surveys from the Aladin data collections tree - https://www.youtube.com/watch?v=IG_6Eh9EKkk

GW sky localization Visibility



MOC observabilities of the gravitational-wave sky localization of GW190425.

The original sky map was previously processed taking into account an all-sky Galactic reddening map and overlapping the PanSTARRS DR1 survey as reference images.

The visibility refers to three astronomical observatories: **Haleakala Observatories** in Hawaii (USA), **Paranal Observatory** in Chile and **Siding Spring Observatory** (SSO) in Australia.

The time interval is defined from **08:18:05** UTC to 14:18:05 UTC in two hour steps with airmass ranging from 1 to 2.

Tables and Planning

Table 1: MOC visibility areas [deg²].

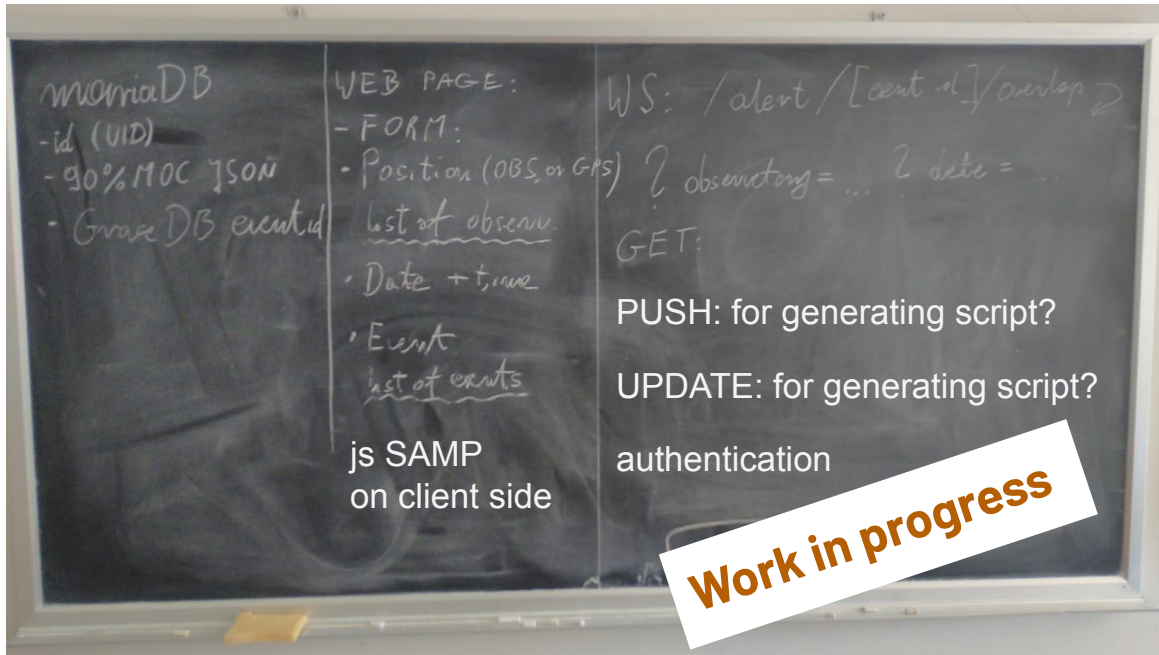
Obs Time (UTC)	Haleakala	Paranal	SSO
2019-04-25 08:18:05.0	2567	2055	–
2019-04-25 10:18:05.0	3996	1089	128
2019-04-25 12:18:05.0	4350	–	771
2019-04-25 14:18:05.0	3728	–	1515

As indicated by the blank values in the second column of **Table 2** there are no simultaneous overlaps between the three observatories at any time, but **columns 3 and 4** show significant intersections between Haleakala and Paranal, and Haleakala and SSO. The generations of these simple tables allow to organize strategies among several observers. Continuing the illustration of this example, to optimize the observations planning in such case, the intersection areas can be totally subtracted from Haleakala schedule and redistributed between Paranal and SSO observatories.

Table 2: MOC visibility intersection areas [deg²].

Obs Time (UTC)	Haleakala \cap Paranal \cap SSO	Haleakala \cap Paranal	Haleakala \cap SSO	Paranal \cap SSO
2019-04-25 08:18:05.0	–	657	–	–
2019-04-25 10:18:05.0	–	706	31	–
2019-04-25 12:18:05.0	–	–	765	–
2019-04-25 14:18:05.0	–	–	1465	–

A MOC Visibility Server



- Sky map selection.
- MOC generation.
- Selection observatory site(s).
- Selection constraints.
- A SAMP Web page.

SAMP is the Simple Application Messaging Protocol used for communicating between applications. It was developed within the [IVOA](#).

What Next for MMA&VO

Support mocpy and
ipyladin in
conda-forge.



Support ipyladin in
google colab.



On fly statistic report and pixel
extractions for Box and Circle in
Aladin (see next slide).



Conditional distribution along the
line of sight implemented in Aladin
as a plugin.



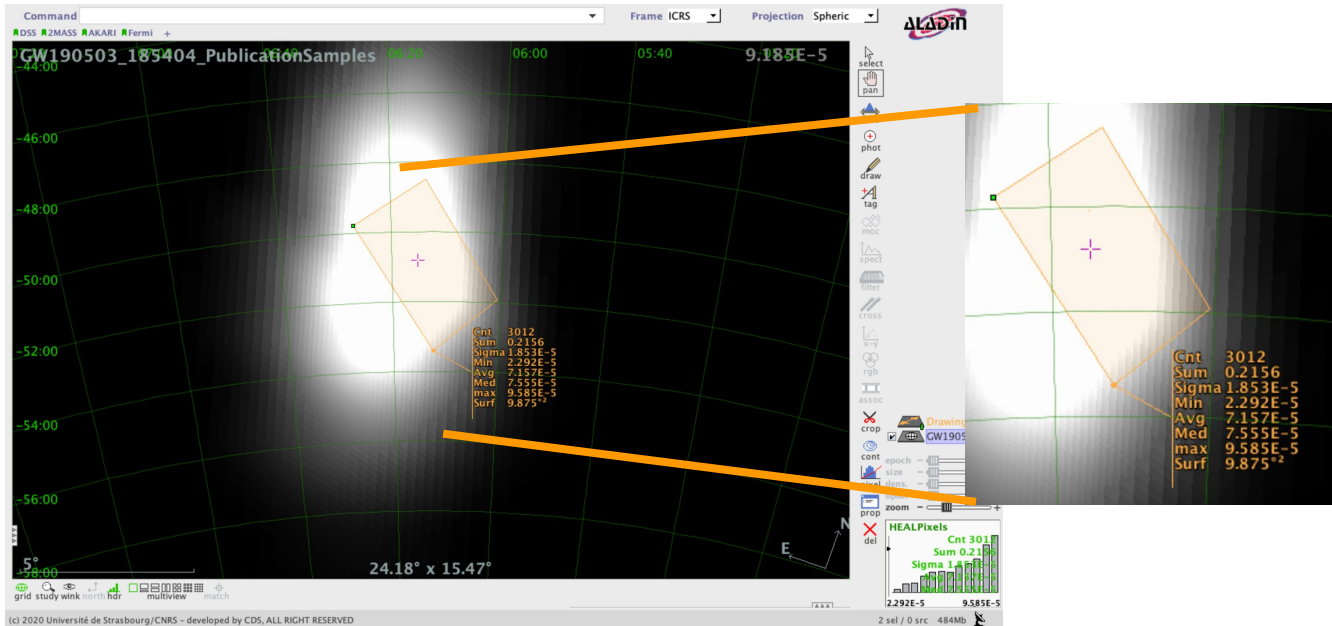
HIPS catalog for
galaxies within the
90% credible volume.



How many disjointed regions
are within a MOC? Split them
as independent MOC.



Integrated Probability and FoV



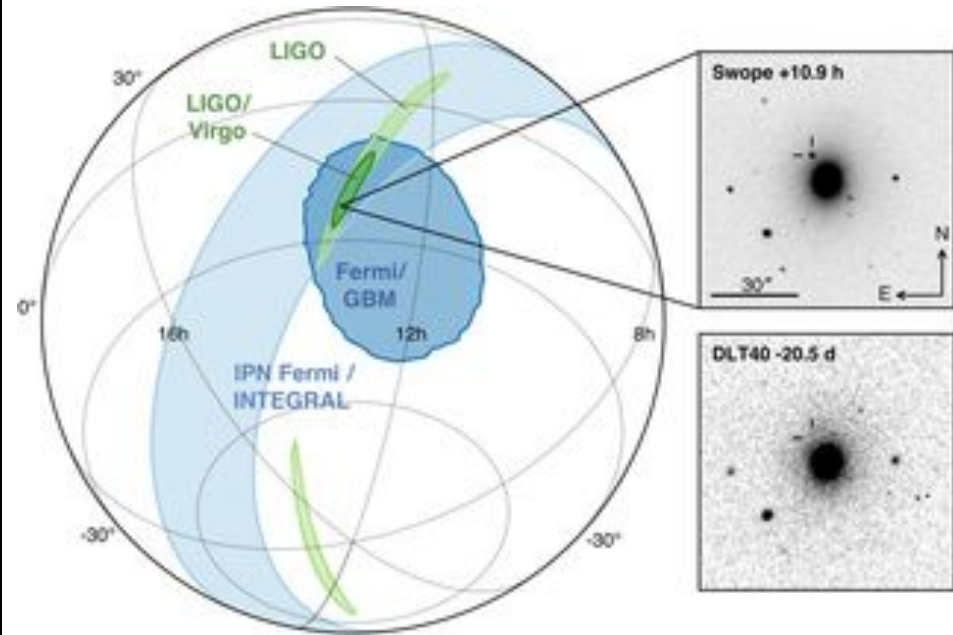
When you draw a polygon, you can get an onfly statistics by selecting it.

If the method is extended to **draw box** and **draw circle**, it will be an useful and fast method to measure the integrated probability within a Field of View!



Live Section: GW170817

Practical Application of ST-MOC



THANKS!

Any questions?