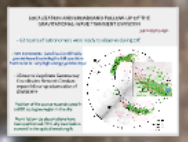
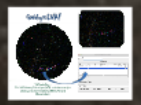
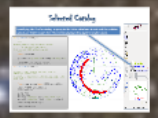
EM Follow-up of Gravitational-Wave Triggers: Current Status and Future Outlook

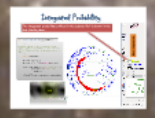
G. Greco, E. Chassande-Mottin, M. Branchesi, G. Stratta, Thomas Boch and many others

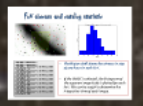
ASTERICS DADI Technology Forum 2





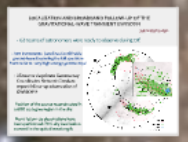
Pollock Halls campus in the Pollock Room in St Leonard's Hall, Edinburgh, 7 & 8 March, 2016

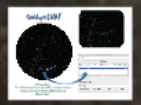
prezi: http://prezi.com/3kxhf-vgnjog/?utm_campaign=share&utm_medium=copy&rc=ex0share

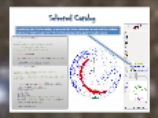
EM Follow-up of Gravitational-Wave Triggers: Current Status and Future Outlook

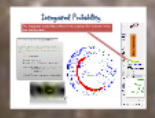
G. Greco, E. Chassande-Mottin, M. Branchesi, G. Stratta, Thomas Boch and many others

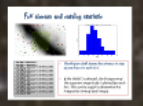
ASTERICS DADI Technology Forum 2





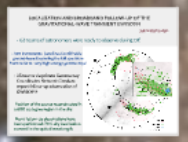
Pollock Halls campus in the Pollock Room in St Leonard's Hall, Edinburgh, 7 & 8 March, 2016

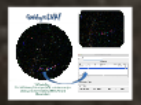
prezi: http://prezi.com/3kxhf-vgnjog/?utm_campaign=share&utm_medium=copy&rc=ex0share

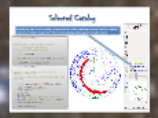
EM Follow-up of Gravitational-Wave Triggers: Current Status and Future Outlook

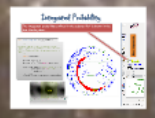
G. Greco, E. Chassande-Mottin, M. Branchesi, G. Stratta, Thomas Boch and many others

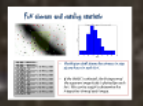
ASTERICS DADI Technology Forum 2





Pollock Halls campus in the Pollock Room in St Leonard's Hall, Edinburgh, 7 & 8 March, 2016

prezi: http://prezi.com/3kxhf-vgnjog/?utm_campaign=share&utm_medium=copy&rc=ex0share

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

LOCALIZATION AND BROADBAND FOLLOW-UP OF THE GRAVITATIONAL-WAVE TRANSIENT GW150914

Submitted to ApJL

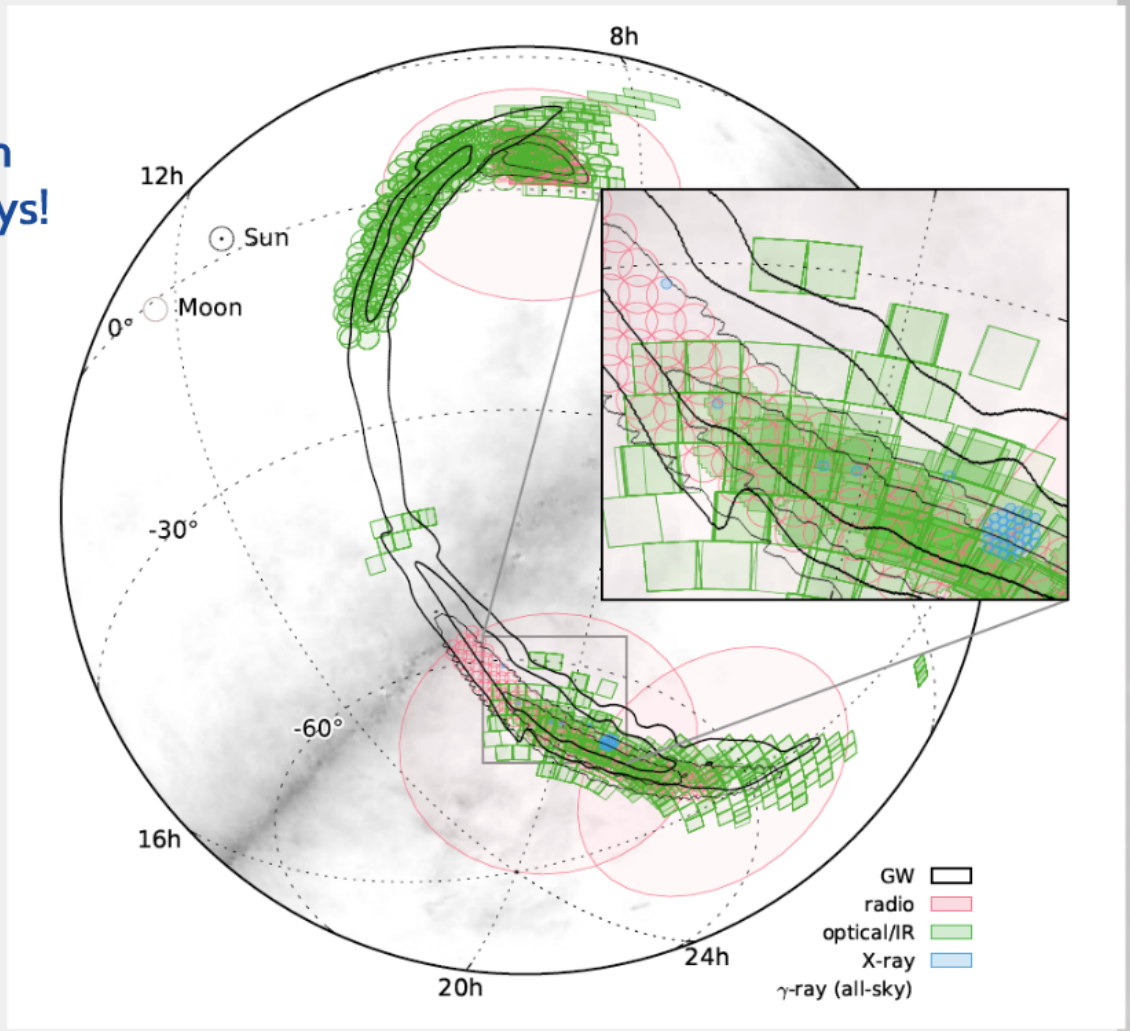
- 63 teams of astronomers were ready to observe during O1!

- 160 instruments (satellites/world-wide ground-based) covering the full spectrum from radio to very high-energy gamma-rays!

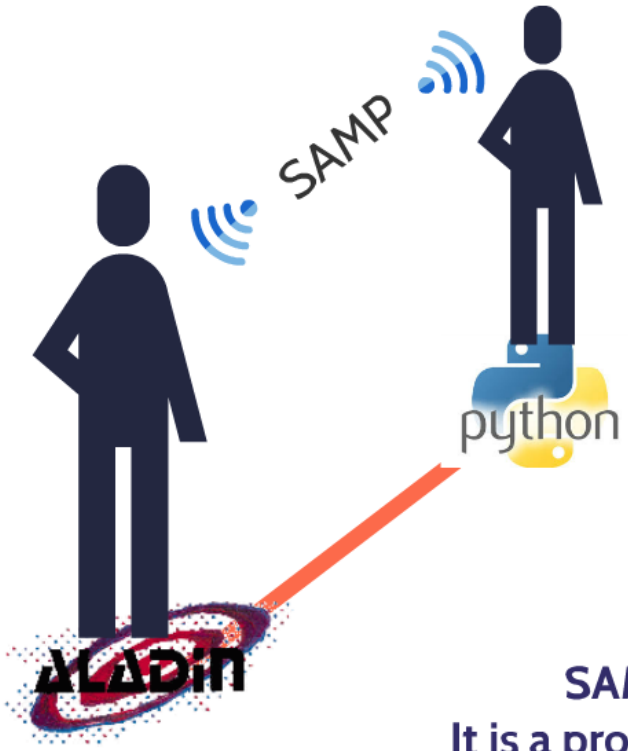
- 25 teams via private Gamma-ray Coordinates Network Circulars report follow-up observation of GW150914

Position of the source reconstructed in a 600 sq degree region in the sky.

Many follow-up observations have been performed: 70% sky localization covered in the optical wavelength.



Aladin Sky Atlas and Python



GWsky's Motivations

LIGO-Virgo 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.

The error box of a GW event is very large typically several hundred of square degrees.

SAMP: Simple Application Messaging Protocol.
It is a protocol for astronomical applications to collaborate.

A easy tool to display the projections of the Field of View in the GW error box and get some relevant information *combining Aladin and Python via SAMP.*

GWsky: an interactive Python script

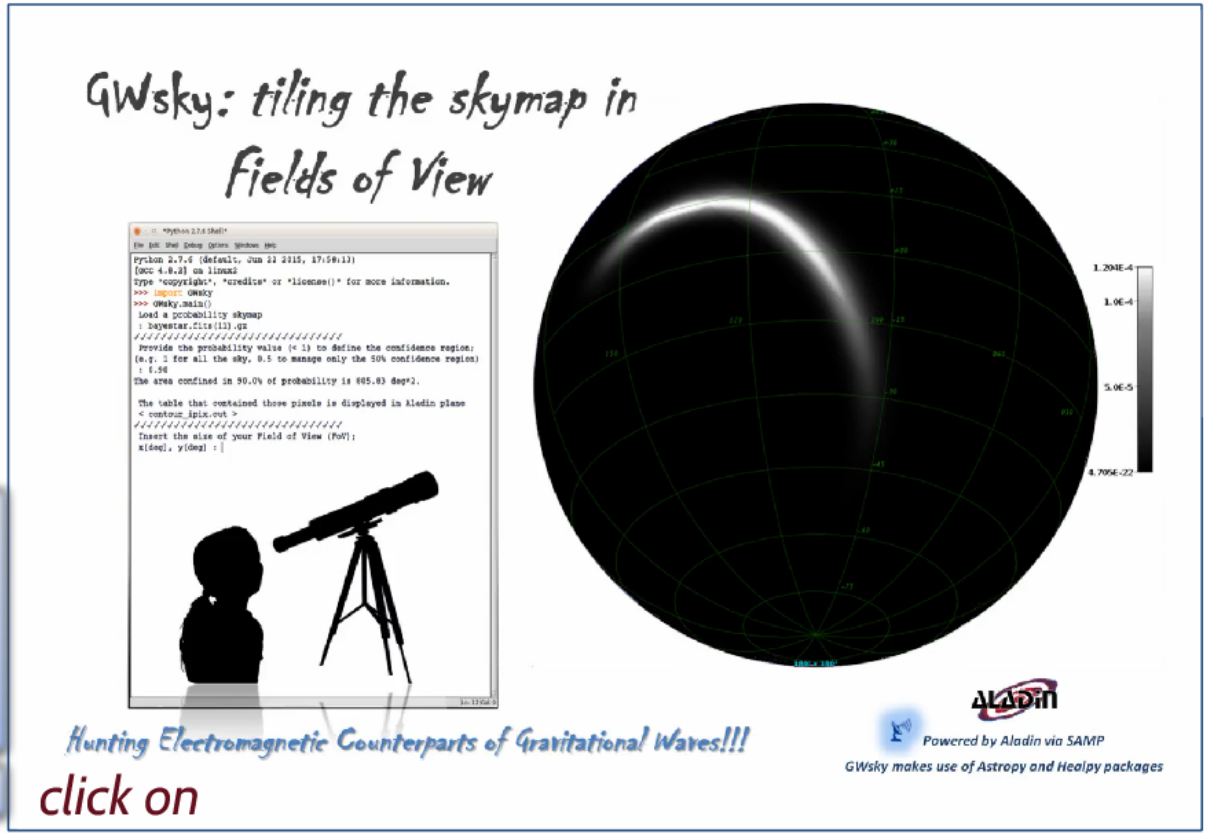
GWsky is an interactive Python script to generate a sequence of pointings given a specific Field of View (FoV). It aims to split the large GW sky localization into several independent areas.

The airmass, the integrate probability and a query to the Vizier database are provided in real time.

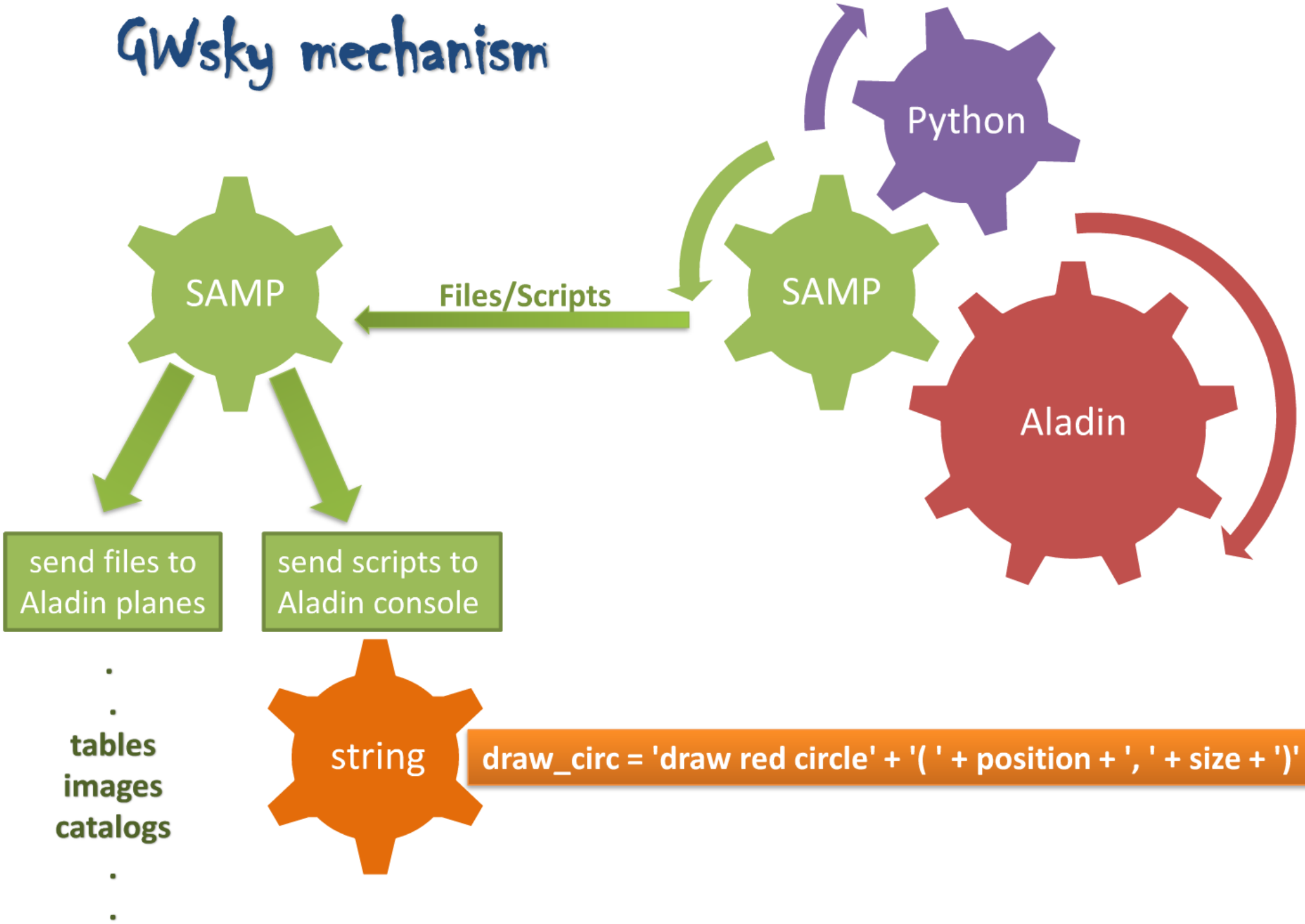
The Italian National Institute for Astrophysics (INAF) made used of GWsky during the first science run **O1.**

The skymap of a GW event, represented by white region of interest on a dark background, is tiled with multiple EM observations, each one targeting a colored tile.

Credits: The Virgo Collaboration MediaKit



GWsky mechanism



Starting with GWsky

From idle:

```
>>> import GWsky
>>> GWsky.main()
```

From terminal:

```
./GWsky
```

It raises an exception if Aladin Sky Atlas is not run.

```
------------------------------------------------------
***Launch Aladin Sky Atlas for running the script***
http://aladin.u-strasbg.fr/
------------------------------------------------------

Unable to find a running SAMP Hub
.
.
.
.
```

Input values and main functions

HEALPix skymap	• aladinSAMP
Confidence region	• aladinSAMP
CDS Catalog	• aladin_console
Field of View size	• aladin_console
Observation time	• airmass
Observatory Coordinates	• airmass
EM candidates	• aladin_console
Directions	• FOV_sequence

GWsky Command Line

C runs a new sequence *changing* the FoV center

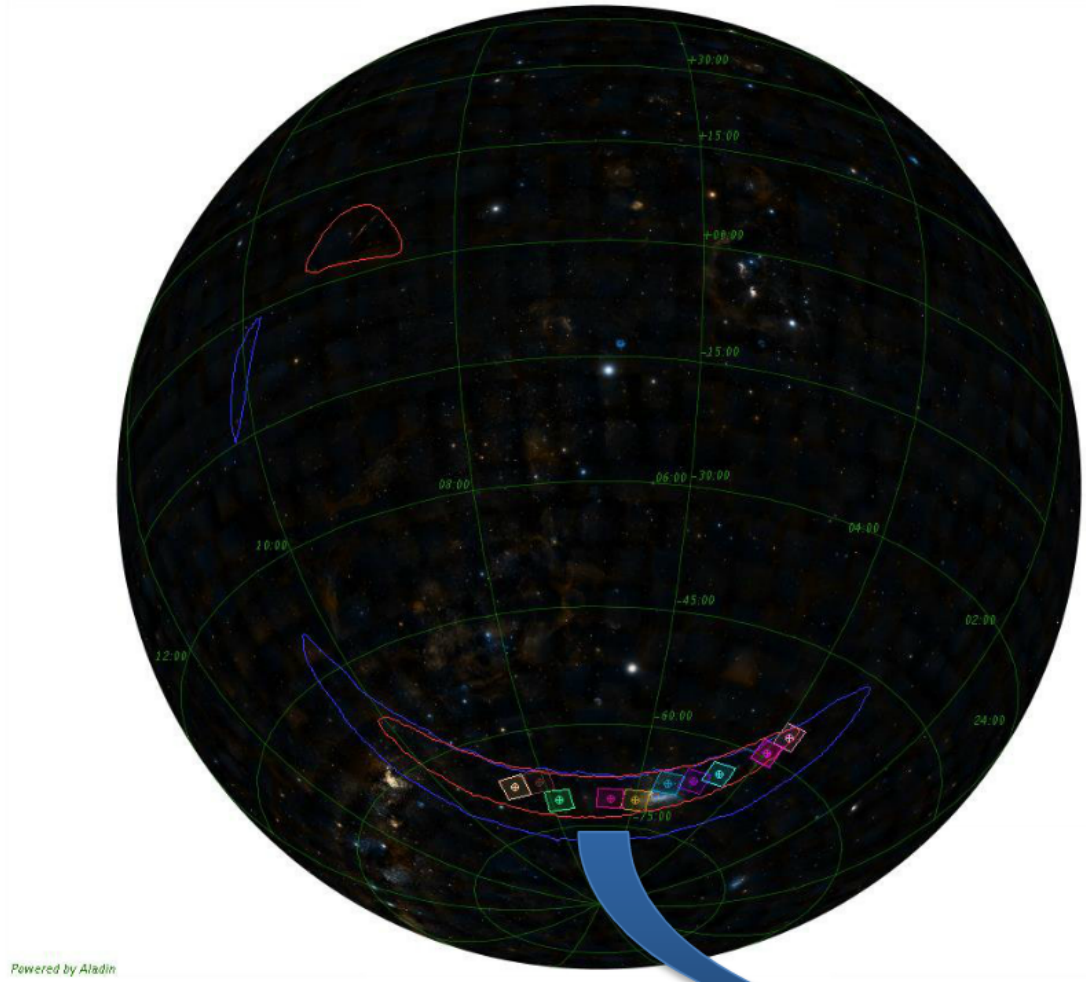
I runs a new sequence without drawing the *input* FoV

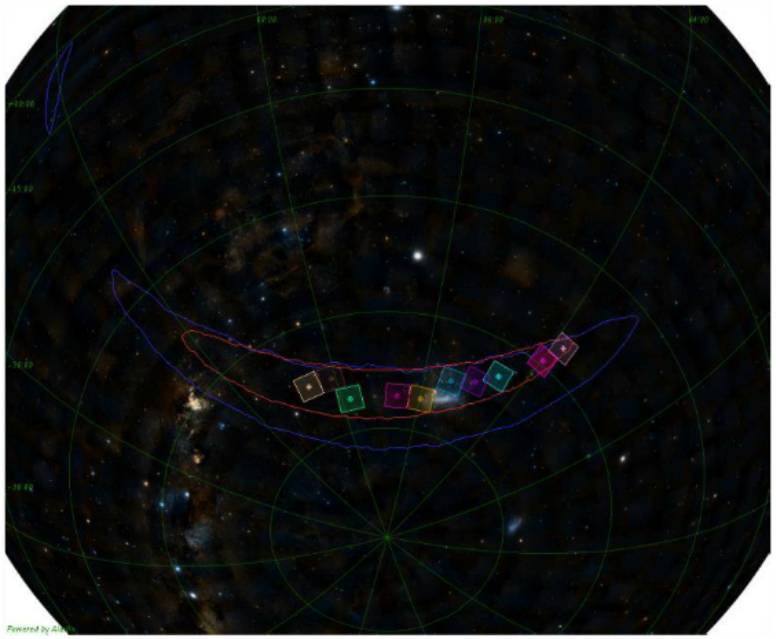
L runs a new sequence starting from the *last* drawn FoV

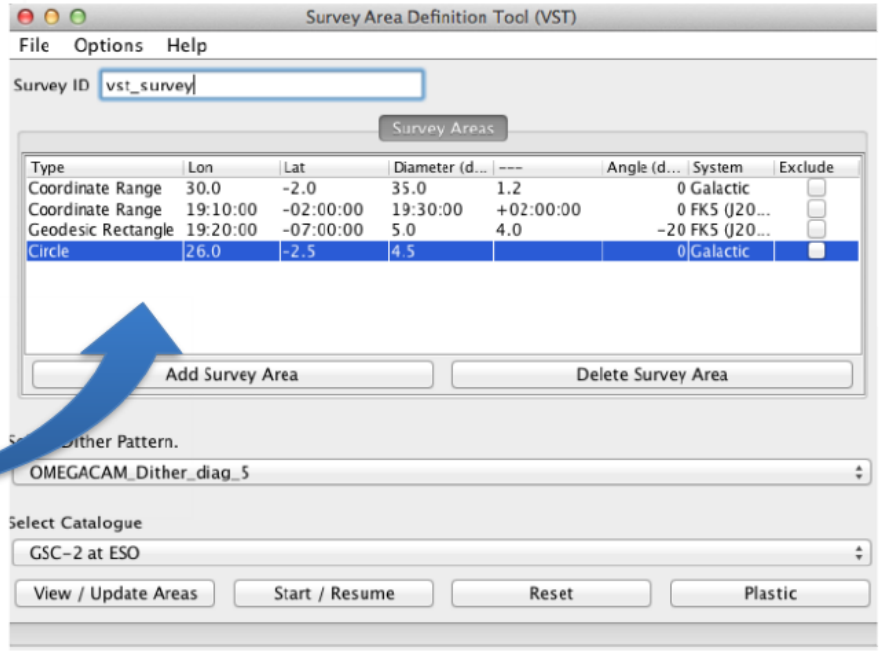
R *repeats* the last action

Q *quit*

GWsky@INAF





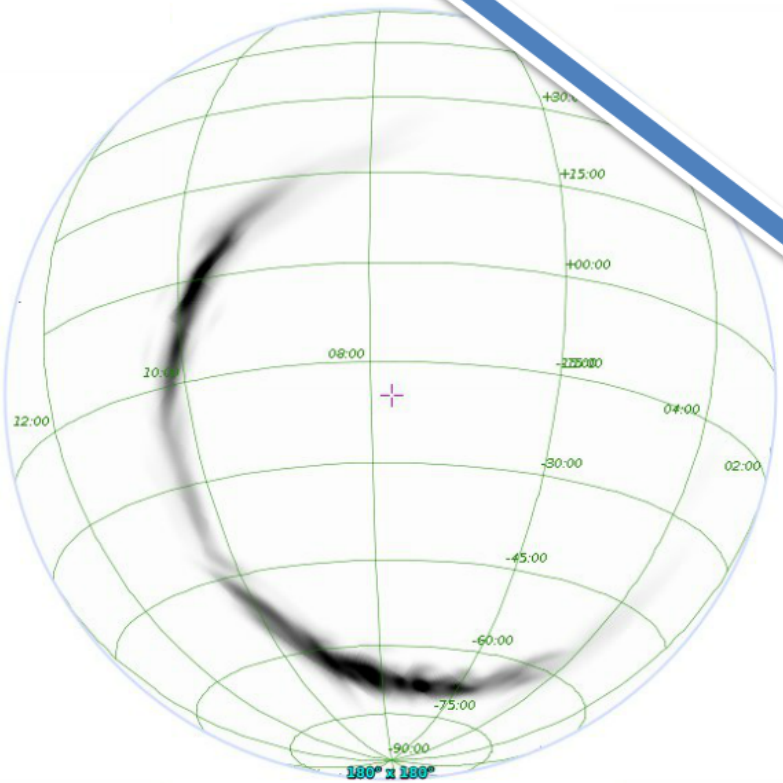


VST pointings.
The VLT Survey Telescope (VST) is the latest major telescope to be installed at ESO's Paranal Observatory.

Probability Skymap

The probability skymap is shown In the first Aladin plane.

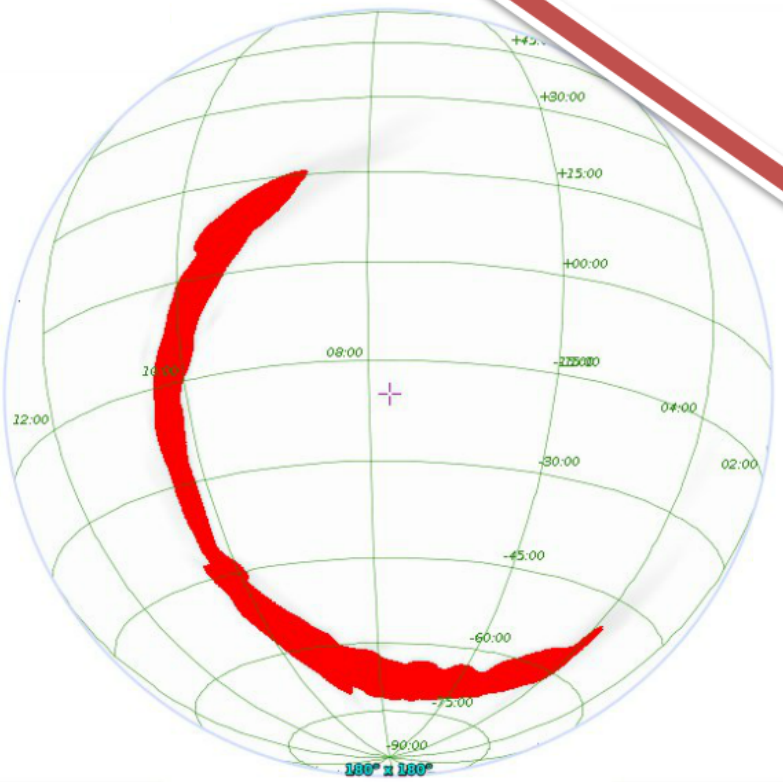
```
def send_file( infile ):

    """
Sending a file (image or table) to Aladin Sky Atlas
using the SAMPIntegratedClient class.

http://docs.astropy.org/en/stable/vo/samp/example_table_image.html
    """

    from astropy.vo.samp import SAMPIntegratedClient

    client = SAMPIntegratedClient()
    client.connect()

    params = {}
    import urlparse
    import os.path
    params[ "url" ] = urlparse.urljoin( 'file:',
                        os.path.abspath( infile ) )

    message = {}
    message[ "samp.mtype" ] = "image.load.fits"
    message[ "samp.params" ] = params

    client.notify_all( message )

    client.disconnect()
```



Confidence Region

The confidence region selected by the user - pixel table.

```
def send_file( infile ):

    """
Sending a file (image or table) to Aladin Sky Atlas
using the SAMPIntegratedClient class.

http://docs.astropy.org/en/stable/vo/samp/example_table_image.html
    """

    from astropy.vo.samp import SAMPIntegratedClient

    client = SAMPIntegratedClient()
    client.connect()

    params = {}
    import urlparse
    import os.path
    params[ "url" ] = urlparse.urljoin( 'file:',
                      os.path.abspath( infile ) )

    message = {}
    message[ "samp.mtype" ] = "image.load.fits"
    message[ "samp.params" ] = params

    client.notify_all( message )

    client.disconnect()
```

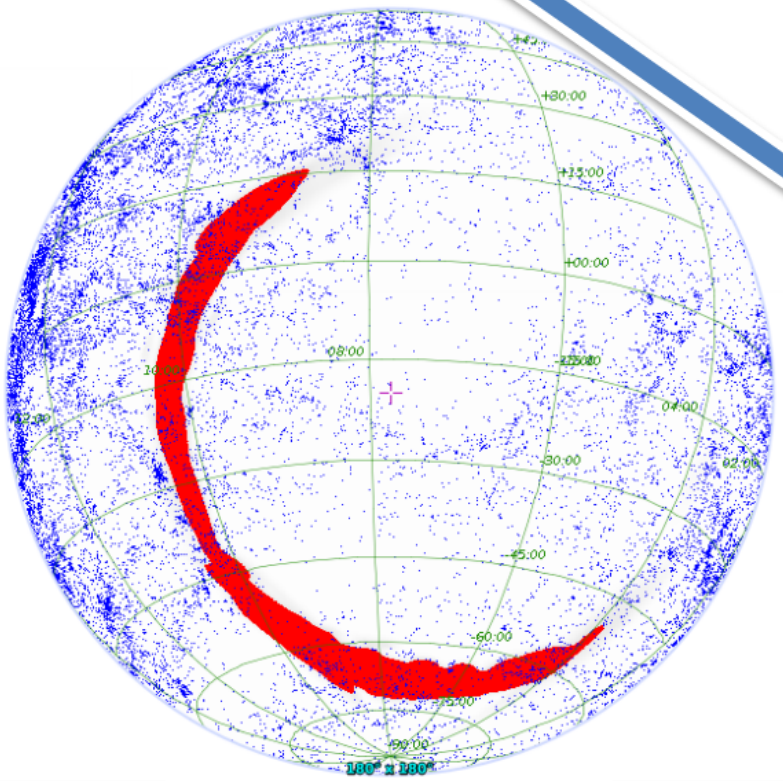


Selected Catalog

Specifying the ID of a catalog, a query to the Vizier database is sent and the relative values are listed in each FoV. The entire catalog is displayed in Aladin plane.

```
def get_VizieR( catalog ):

    '''
building command script for Aladin console:
        "get VizieR(catalog,allsky)".
It is sent via SAMP to Aladin console.
    '''
    import aladinSAMP

    get_vizier = 'get VizieR('+catalog+','+'allsky'+')'

    aladinSAMP.send_script(get_vizier)
```

```
def send_script( script ):

    """
Sending a script to Aladin Sky Atlas using the
SAMPIntegratedClient class.

http://docs.astropy.org/en/stable/vo/samp/example_table_
image.html
    """

    from astropy.vo.samp import SAMPIntegratedClient

    client = SAMPIntegratedClient()
    client.connect()

    params = {}
    message = {}
    message[ "samp.mtype" ] = "script.aladin.send"
    message[ "samp.params" ] = { "script" : script }

    client.notify_all( message )

    client.disconnect()
```

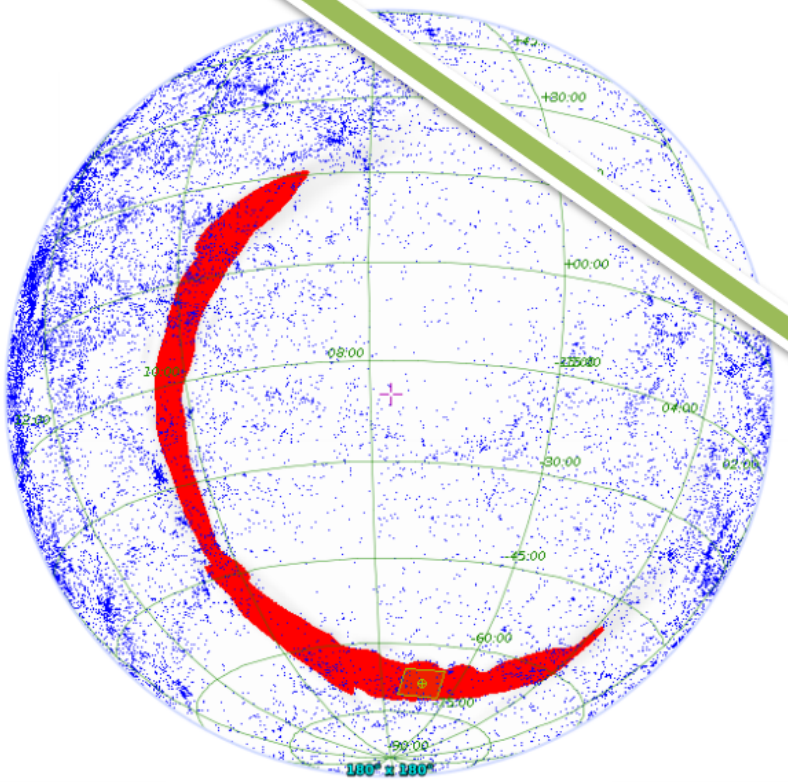


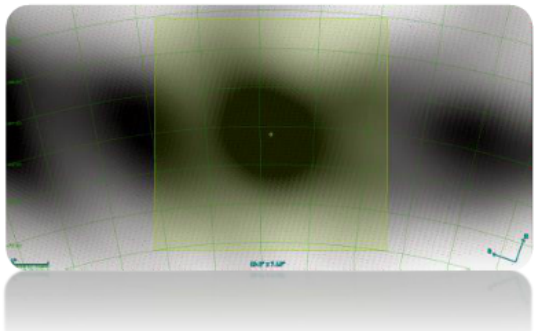
Get FoV

The FoV tile centers at the highest probability pixel. The FoV are defined using the Instrument Footprint Editor and the Votable is modified by the *instrument_FOV* function.

```
def get_FoV( x, y ):

    '''
building command script for Aladin console:
              "get FoV(pointing)".
It is sent via SAMP to Aladin console.
    '''
    import aladinSAMP

    position = [ x, y ]
    position = '  '.join(map(str, position))

    FoV_pointing = 'get FoV(pointing)'+' '+position

    aladinSAMP.send_script ( FoV_pointing )
```

```
def instrument_FOV( FOV_base, FOV_height ):

    """
    Modify the file output of Instrument Footprint Editor
    provided by Aladin with a user FOV size.

    """

    import aladinSAMP

    from astropy.io.votable import parse

    votable = parse( "footprint_GWsky2.vot" )

    table = votable.get_first_table()

    data = table.array

    FOV_base_arcsec = FOV_base*3600.0

    FOV_height_arcsec = FOV_height*3600.0

    data[0] = - FOV_base_arcsec /  2.0,   FOV_height_arcsec / 2.0
    data[1] =   FOV_base_arcsec /  2.0,   FOV_height_arcsec / 2.0
    data[2] =   FOV_base_arcsec /  2.0, - FOV_height_arcsec / 2.0
    data[3] = - FOV_base_arcsec /  2.0, - FOV_height_arcsec / 2.0

    votable.to_xml( 'instrument_FOV.vot' )

    aladinSAMP.send_file( 'instrument_FOV.vot' )
```

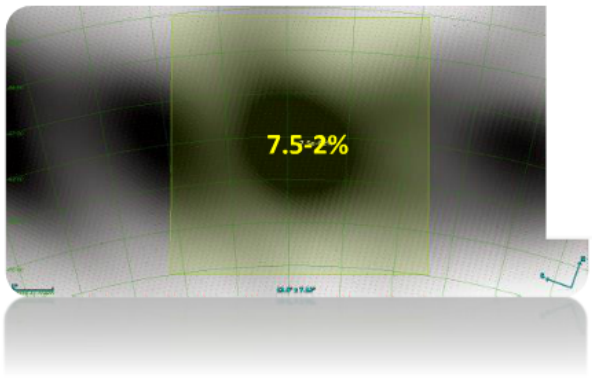


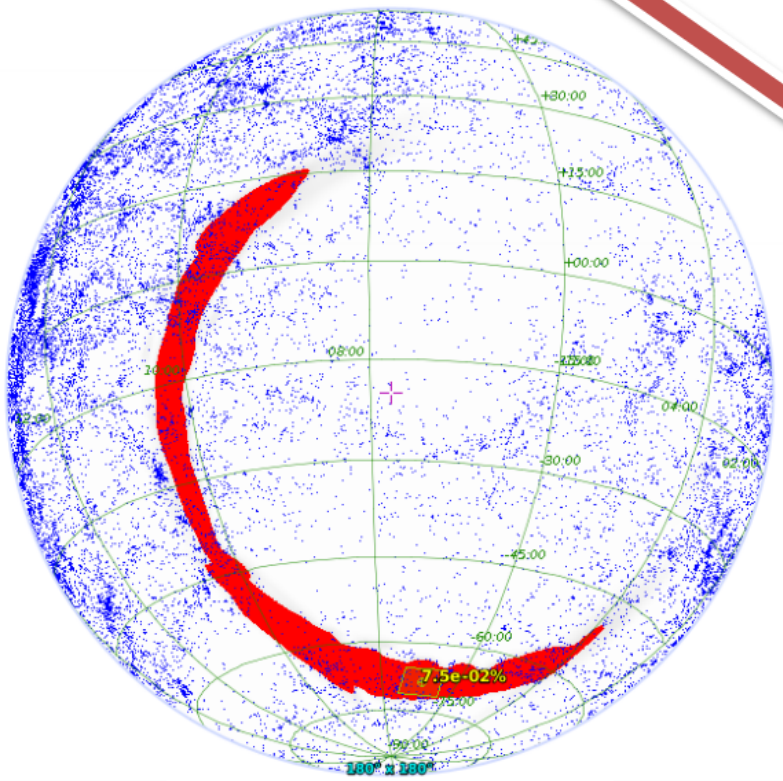


Integrated Probability

The integrated probalility confined in the selected FoV is shown in the last Aladin plane.

```
def  draw_string_float( x, y, number ):

     '''
building command script for Aladin console:
"draw string ( x, y, number )". It is sent via
SAMP to Aladin console; the parameter num is a
float.
     '''
     import aladinSAMP

     position = [ x, y ]
     position = ' , '.join(map(str,position))

     draw_string_number = 'draw string' + '( ' +
               position +','+str(('% .1e'
               % number))+'%)

      aladinSAMP.send_script(draw_string_number)
```





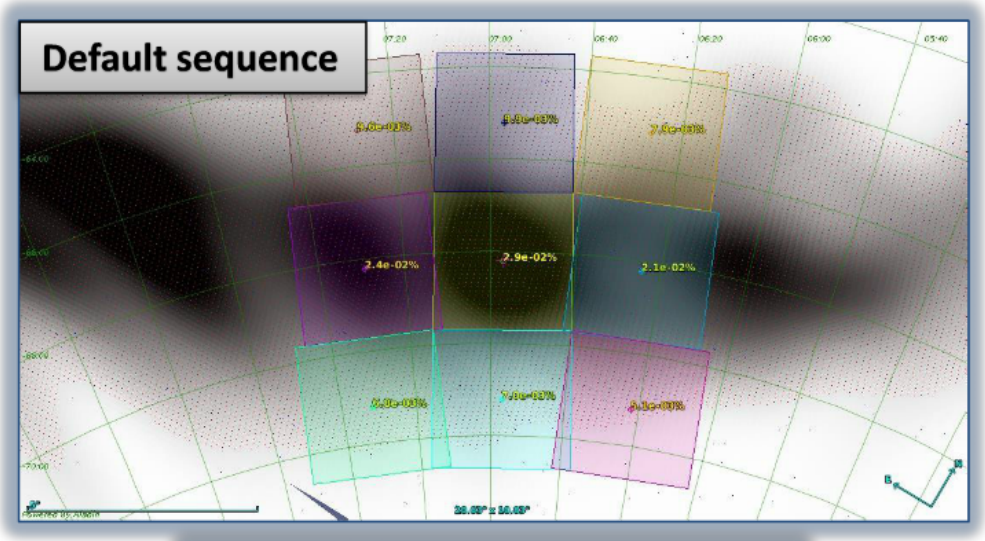
FoV Sequence

The FoV are evenly spaced assuming that the shortest angular distance between two points on the celestial sphere is measured along a great circle that passes through both them

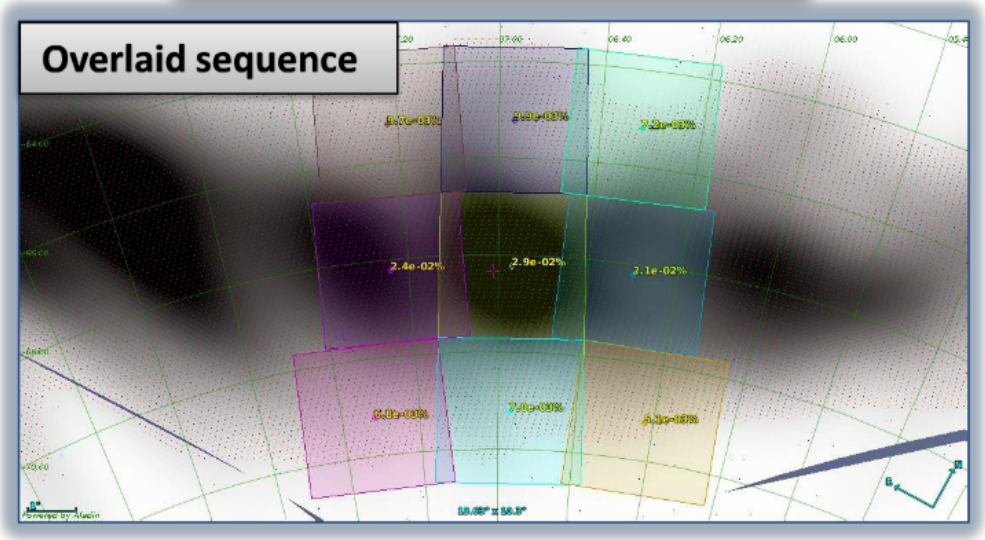
$$\cos\theta = \sin\delta_1 \sin\delta_2 + \cos\delta_1 \cos\delta_2 \cos(\alpha_1 - \alpha_2)$$

The cardinal and intercardinal directions are permitted to develop a FoV sequence from a fixed FoV center.

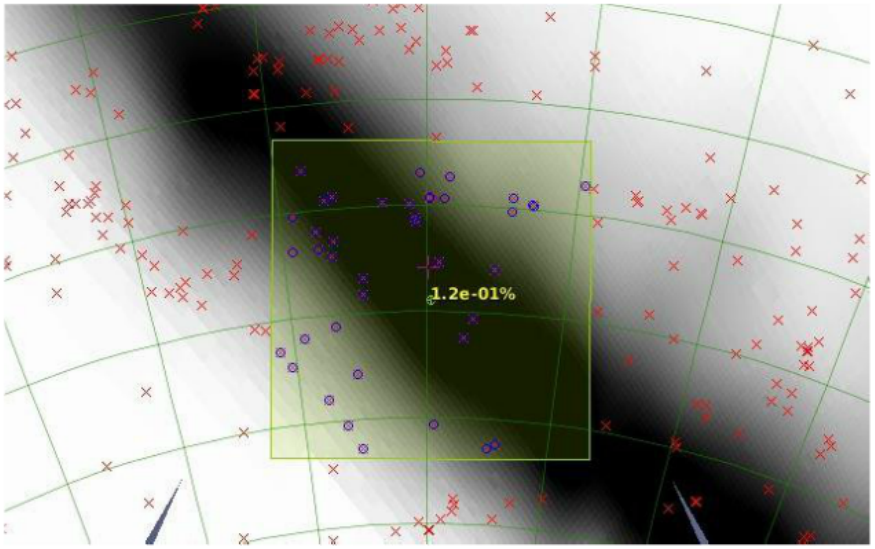
At user option, the FoVs can be overlaid or separated from their default positions.

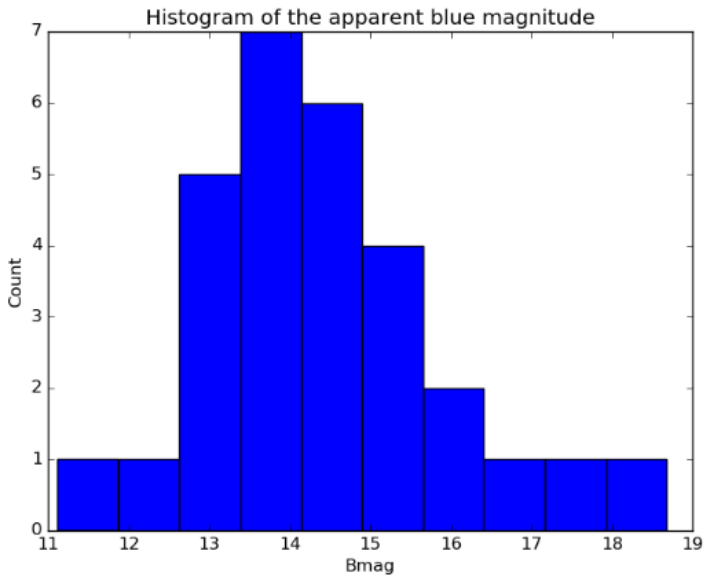


N/NW/W/SW/S/SE/E/NE



FoV airmass and catalog statistic





2015-09-18 21:00:00.000 ---
2015-09-18 22:00:00.000 ---
2015-09-18 23:00:00.000 ---> The airmass of the FOV center is 4.24.
2015-09-19 00:00:00.000 ---> The airmass of the FOV center is 2.18.
2015-09-19 01:00:00.000 ---> The airmass of the FOV center is 1.53.
2015-09-19 02:00:00.000 ---> The airmass of the FOV center is 1.25.
2015-09-19 03:00:00.000 ---> The airmass of the FOV center is 1.11.
2015-09-19 04:00:00.000 ---> The airmass of the FOV center is 1.06.
2015-09-19 05:00:00.000 ---> The airmass of the FOV center is 1.09.

- *The Phyton shell shows the airmass in step of one hours in each FoV.*
- *If the GWGC is selected, the histogram of the apparent magnitude is plotted for each FoV. This can be useful to determine the integration time of each image.*

Future Developments

- Output to an html page
- HiPS - Hierarchical Progressive Survey
- Errors and Exceptions
- Countour plot