In [ ]:
```
# %matplotlib inline
```

In [3]:
```
# Ligo.skymap
import ligo.skymap.plot
from ligo.skymap.io import read_sky_map
from ligo.skymap.postprocess import crossmatch
import pandas as pd
import pprint
import numpy as np
from pytz import timezone
import matplotlib.pyplot as plt
from matplotlib import cm

# Astropy
from astropy.io import fits
from astropy.time import Time
from astropy import units as u
from astropy.table import Table
from astropy.coordinates import SkyCoord
from astropy.coordinates import EarthLocation
from astropy.utils.data import download_file

from astroquery.ipac.ned import Ned
from astroquery.vizier import VizierClass
import astropy_healpix as ah

# TRT API
import requests
import json

from pytz import timezone

# Astroplan
from astroplan import Observer
from astroplan import FixedTarget
from astroplan.plots import plot_sky
from astroplan.plots import plot_airmass
from astroplan.plots import plot_parallactic
from astroplan import AltitudeConstraint
from astroplan import AirmassConstraint
from astroplan import AtNightConstraint
from astroplan import is_observable
from astroplan import is_always_observable
from astroplan import months_observable
from astroplan import observability_table
from astroplan import time_grid_from_range
from astroplan.plots import plot_airmass
```

```
/lustre/GOTOML/kanthanakorn/conda/envs/gws/lib/python3.10/site-packages/lig
o/lw/lsctables.py:89: UserWarning: Wswiglal-redir-stdio:

SWIGLAL standard output/error redirection is enabled in IPython.
This may lead to performance penalties. To disable locally, use:

with lal.no_swig_redirect_standard_output_error():
    ...

To disable globally, use:
```

```
lal.swig_redirect_standard_output_error(True)

Note however that this will likely lead to error messages from
LAL functions being either misdirected or lost when called from
Jupyter notebooks.

To suppress this warning, use:

import warnings
warnings.filterwarnings("ignore", "Wswiglal-redir-stdio")
import lal

  import lal
/lustre/GOTOML/kanthanakorn/conda/envs/gws/lib/python3.10/site-packages/tqd
m/auto.py:21: TqdmWarning: IProgress not found. Please update jupyter and i
pywidgets. See https://ipywidgets.readthedocs.io/en/stable/user_install.htm
l
  from .autonotebook import tqdm as notebook_tqdm
/lustre/GOTOML/kanthanakorn/conda/envs/gws/lib/python3.10/site-packages/h5p
y/__init__.py:36: UserWarning: h5py is running against HDF5 1.14.3 when it
was built against 1.14.2, this may cause problems
```

In [2]:

```
# GW EVETN: S231020ba
url = 'https://gracedb.ligo.org/api/superevents/S230731an/files/Bilby.mult
# Download file
filename = download_file(url, cache=True)
```

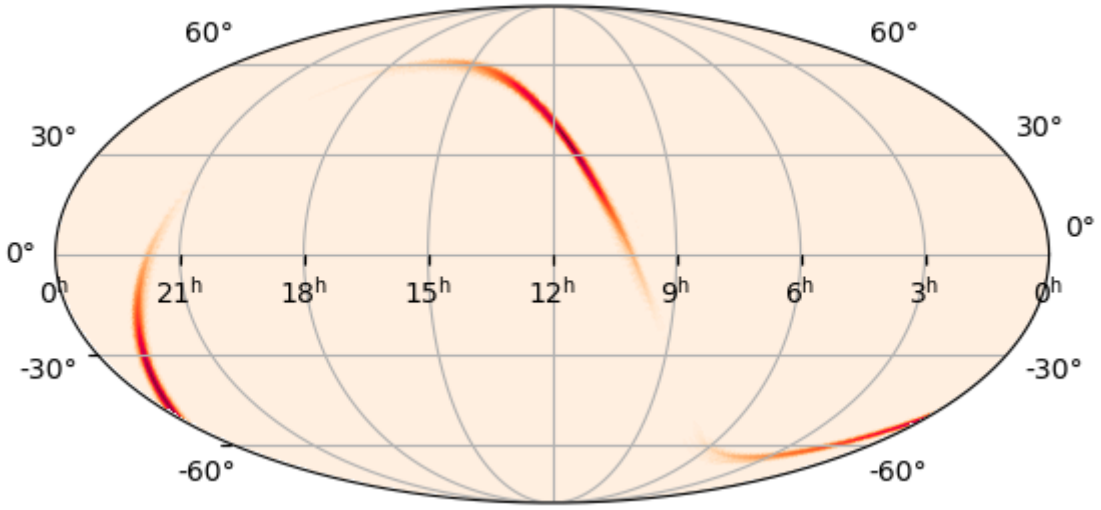
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[2], line 5
      2 url = 'https://gracedb.ligo.org/api/superevents/S230731an/files/Bil
by.multiorder.fits'
      4 # Download file
----> 5 filename = download_file(url, cache=True)

NameError: name 'download_file' is not defined
```

In [3]:

```
# read skymap by LIGO and SKymap
skymap, metadata = read_sky_map(filename, nest=None)
nside = ah.npix_to_nside(len(skymap))

# Convert sky map from probability to probability per square degree.
deg2perpix = ah.nside_to_pixel_area(nside).to_value(u.deg**2)
probperdeg2 = skymap / deg2perpix
```

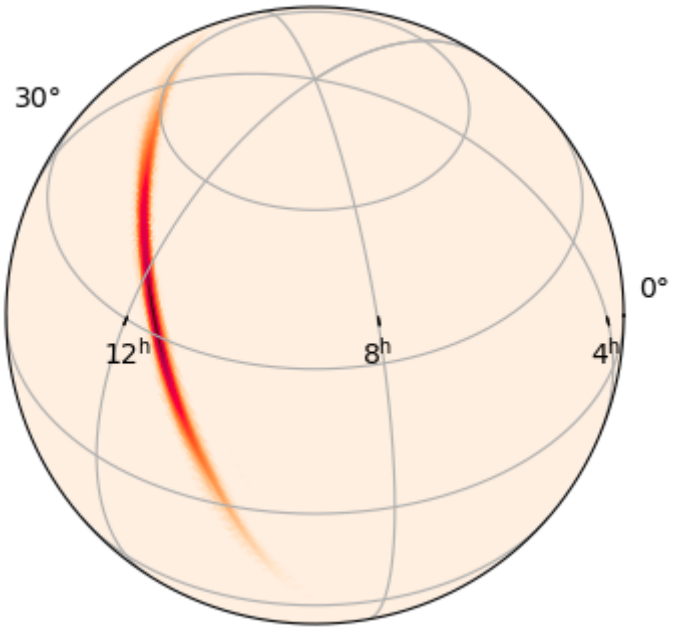
In [4]:

```
# Plot sky map
ax = plt.axes(projection='astro hours mollweide')
vmax = probperdeg2.max()
img = ax.imshow_hpx((probperdeg2, 'ICRS'),cmap='cylon', nested=metadata['n
ax.grid()
plt.show()
```



In [5]:

```
center='135d +40d'

fig = plt.figure(figsize=(4, 4), dpi=100)
ax = plt.axes(projection='astro globe', center=center)
ax.imshow_hpx(probperdeg2, cmap='cylon', nested=metadata['nest'], vmin=0.,
ax.grid()
plt.show()
```



Crossmatch with NED

In [6]:
```
# NED API
# https://ned.ipac.caltech.edu/Documents/Guides/Interface/GWF

# Take first 20 galaxies
GW_event = "S230731an"
ned_galaxy = "https://ned.ipac.caltech.edu/uri/NED::GWFglist/csv/{}/latest/
print(ned_galaxy)

filename = download_file(ned_galaxy, cache=True)
```

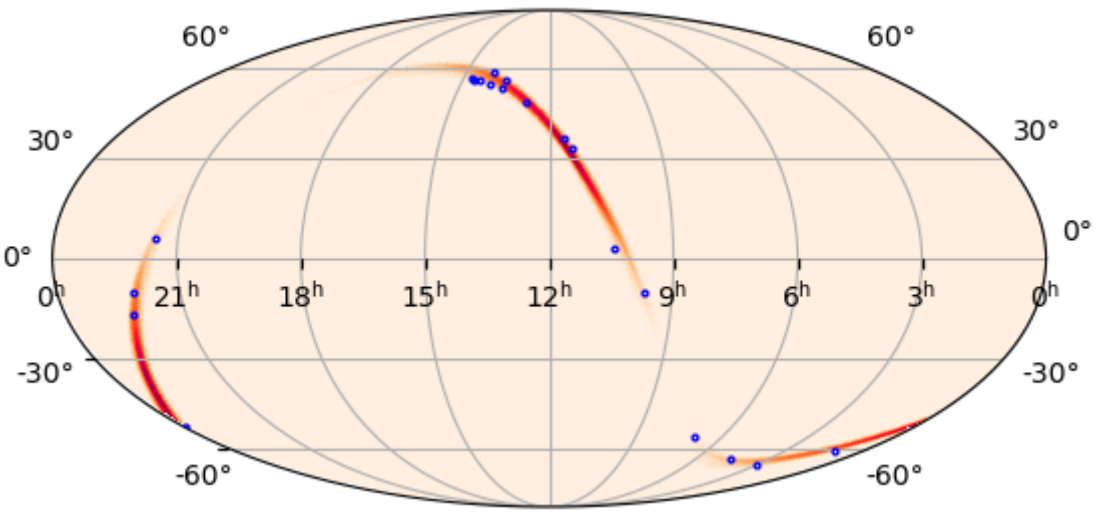
```
https://ned.ipac.caltech.edu/uri/NED::GWFglist/csv/S230731an/latest/20
```

In [7]:
```
neabyGal = Table.read(filename, format='csv')
```

In [8]:
```
print(len(neabyGal))
print(neabyGal.colnames)
```

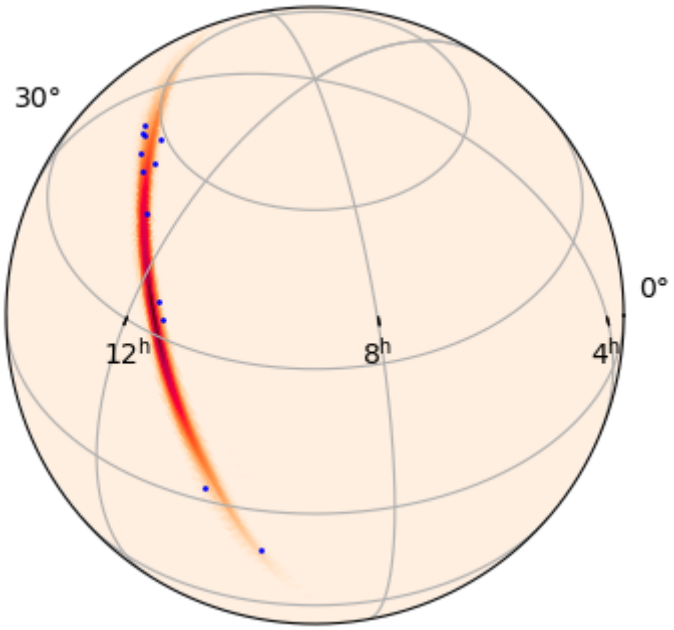
```
20
['objname', 'ra', 'dec', 'objtype', 'z', 'z_unc', 'z_tech', 'z_qual', 'z_qu
al_flag', 'z_refcode', 'ziDist', 'ziDist_unc', 'ziDist_method', 'ziDist_ind
icator', 'ziDist_refcode', 'DistMpc', 'DistMpc_unc', 'DistMpc_method', 'ebv
', 'A_FUV_MWext', 'A_NUV_MWext', 'A_J_MWext', 'A_H_MWext', 'A_Ks_MWext', 'A
_W1_MWext', 'A_W2_MWext', 'A_W3_MWext', 'A_W4_MWext', 'm_FUV', 'm_FUV_unc',
'm_NUV', 'm_NUV_unc', 'Lum_FUV', 'Lum_FUV_unc', 'Lum_NUV', 'Lum_NUV_unc', '
GALEXphot', 'm_J', 'm_J_unc', 'm_H', 'm_H_unc', 'm_Ks', 'm_Ks_unc', 'Lum_J
', 'Lum_J_unc', 'Lum_H', 'Lum_H_unc', 'Lum_Ks', 'Lum_Ks_unc', 'tMASSphot',
'm_W1', 'm_W1_unc', 'm_W2', 'm_W2_unc', 'm_W3', 'm_W3_unc', 'm_W4', 'm_W4_u
nc', 'Lum_W1', 'Lum_W1_unc', 'Lum_W2', 'Lum_W2_unc', 'Lum_W3', 'Lum_W3_unc
', 'Lum_W4', 'Lum_W4_unc', 'WISEphot', 'SFR_W4', 'SFR_W4_unc', 'SFR_hybrid
', 'SFR_hybrid_unc', 'ET_flag', 'Mstar', 'Mstar_unc', 'MLratio', 'dP_dA', '
dP_dV', 'P_2D', 'P_3D', 'P_Mstar', 'P_SFR', 'P_sSFR', 'P_LumW1', 'P_3D_Msta
r', 'P_3D_SFR', 'P_3D_sSFR', 'P_3D_LumW1']
```

In [9]:
```
ax = plt.axes(projection='astro hours mollweide')
ax.imshow_hpx(probperdeg2, cmap='cylon', nested=metadata['nest'], vmin=0.,

for ra, dec in zip(neabyGal['ra'].data, neabyGal['dec'].data):
    ax.plot_coord(SkyCoord(ra, dec, unit='deg'), "o", markerfacecolor='Non

ax.grid()
plt.show()
```



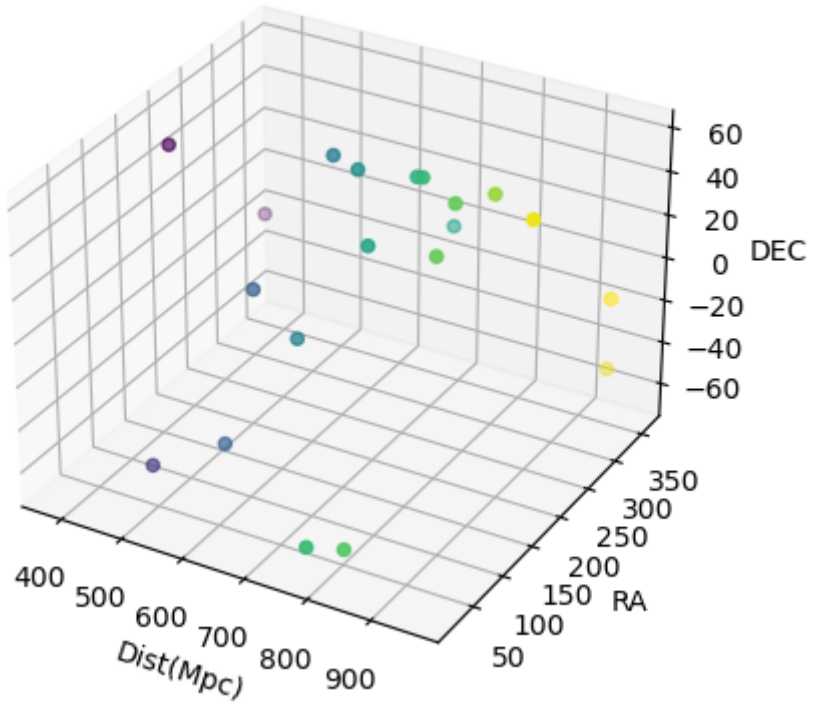
In [10]:

```
fig = plt.figure(figsize=(4, 4), dpi=100)
ax = plt.axes(projection='astro globe', center=center)

# Take marker plot from https://git.ligo.org/lscsoft/ligo.skymap/-/blob/ma
ax.imshow_hpx(probperdeg2, cmap='cylon', nested=metadata['nest'], vmin=0.,

for ra, dec in zip(neabyGal['ra'].data, neabyGal['dec'].data):
    ax.plot_coord(SkyCoord(ra, dec, unit='deg'), ".", markerfacecolor='Non

ax.grid()
plt.show()
```



In [11]:

```
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(neabyGal['DistMpc'].data, neabyGal['ra'].data, neabyGal['dec'].
ax.set_xlabel('Dist(Mpc)')
ax.set_ylabel('RA')
ax.set_zlabel('DEC')
plt.show()
```



In [12]:

```
#if less galaxy use
Gal_for_obs = neabyGal.copy()

#if lots of galaxy use
# # choose range to observe
if False:
    limit_range = np.where((neabyGal['Dist'] > 600) & (neabyGal['Dist'] < 8
    Gal_for_obs = neabyGal[limit_range]
    print(len(Gal_for_obs))
    print(Gal_for_obs)
```

In [13]:

```
Gal_for_obs
```

Out[13]: *Table length=20*

objname	ra	dec	objtype	z	z_unc	z_tech	z_qual	z_qual_
str25	float64	float64	str1	float64	float64	str4	str5	
WISEA J011010.81-605341.2	17.54542	-60.89506	G	0.1608	0.00041	None	False	F
WISEA J220828.11-101851.7	332.11717	-10.31433	G	0.1955	--	PHOT	False	F
WISEA J112419.29+331558.1	171.08041	33.26619	G	0.1505	2.3444e-05	SPEC	False	F
WISEA J113453.58+361435.0	173.72327	36.24297	G	0.1704	3.5189e-05	SPEC	False	F
WISEA J144433.53+563636.8	221.13991	56.61032	G	0.179	3.6827e-05	SPEC	False	F
WISEA J102432.58+031012.8	156.13596	3.17028	G	0.1173	--	PHOT	False	F

objname	ra	dec	objtype	z	z_unc	z_tech	z_qual	z_qual_
WISEA J235616.98-522146.6	359.07083	-52.36311	G	0.191	--	PHOT	False	F
WISEA J094032.30-100702.1	145.13458	-10.11722	G	0.1336	--	PHOT	False	F
WISEA J133007.05+553312.9	202.52971	55.55367	G	0.193	3.2915e-05	SPEC	False	F
WISEA J143432.90+554426.2	218.63724	55.74049	G	0.1565	4.6381e-05	SPEC	False	F
WISEA J030018.16-663856.0	45.0755	-66.64878	G	0.1669	0.00015	None	False	F
WISEA J043546.79-640627.5	68.94571	-64.10753	G	0.1015	--	PHOT	False	F
WISEA J142538.65+555839.1	216.41091	55.97757	G	0.1312	1.0107e-05	SPEC	False	F
WISEA J140119.34+541856.8	210.33045	54.31595	G	0.14	1.7884e-05	SPEC	False	F
WISEA J213302.40+060847.1	323.26	6.14644	G	0.1494	1.9948e-05	SPEC	False	F
WISEA J133412.28+530109.5	203.55123	53.01933	G	0.1705	4.1186e-05	SPEC	False	F
WISEA J140026.59+584328.9	210.111	58.72472	G	0.1597	2.3303e-05	SPEC	False	F
WISEA J065717.77-554614.6	104.32417	-55.77075	G	0.1182	0.00015	None	False	F
WISEA J221909.00-163051.4	334.78758	-16.51422	G	0.08378	--	PHOT	False	F

TRT Plan Observation

In [14]:

```
# Gao Mei Gu Observatory
# Latitude : 26°41'43.8" N
# Longitude : 100°01'51.6" E
# UTC+8
latitude_GAO = '+26d41m43.8s'
longitude_GAO = '+100d01m51.6s'
elevation_GAO = 3200 * u.m
location_GAO = EarthLocation.from_geodetic(longitude_GAO, latitude_GAO, ele

observer_GAO = Observer(name='GAO',
                location=location_GAO,
                pressure=0.996 * u.bar,
                relative_humidity=0.8,
                temperature= 16.67 * u.deg_C,
                timezone=timezone('Etc/GMT+8'),
                description="Gao Mei Gu Observatory")
print(observer_GAO)
```

```
<Observer: name='GAO',
    location (lon, lat, el)=(100.031 deg, 26.695499999999996 deg, 3199.9999
999995257 m),
    timezone=<StaticTzInfo 'Etc/GMT+8'>,
    pressure=<Quantity 0.996 bar>,
    temperature=<Quantity 16.67 deg_C>,
```

```
    relative_humidity=0.8>
```

In [15]:

```
# Springbrook Observatory, Australia
# Latitude : 31°16'46.9" S
# Longitude : 149°5'8.2" E
# UTC + 10
latitude_SBO = '-31d16m46.9s'
longitude_SBO = '+149d05m8.2s'
elevation_SBO = 1020 * u.m
location_SBO = EarthLocation.from_geodetic(longitude_SBO, latitude_SBO, el

observer_SBO = Observer(name='SBO',
                location=location_SBO,
                pressure=1.042 * u.bar,
                relative_humidity=0.5,
                temperature= 22 * u.deg_C,
                timezone=timezone('Australia/Brisbane'),
                description="Springbrook Observatory")
print(observer_SBO)
```

```
<Observer: name='SBO',
    location (lon, lat, el)=(149.08561111111112 deg, -31.279694444444438 de
g, 1020.0000000011602 m),
    timezone=<DstTzInfo 'Australia/Brisbane' LMT+10:12:00 STD>,
    pressure=<Quantity 1.042 bar>,
    temperature=<Quantity 22. deg_C>,
    relative_humidity=0.5>
```

In [16]:

```
# Sierra remote observatories, California, United States
# UTC-7
latitude_SRO = '+37d04m12.0s'
longitude_SRO = '-119d24m0.0s'
elevation_SRO = 1405 * u.m
location_SRO = EarthLocation.from_geodetic(longitude_SRO, latitude_SRO, el

observer_SRO = Observer(name='SRO',
                location=location_SRO,
                pressure=1.011 * u.bar,
                relative_humidity=0.6,
                temperature= 5.78 * u.deg_C,
                timezone=timezone('America/Los_Angeles'),
                description="Sierra remote observatories")
print(observer_SRO)
```

```
<Observer: name='SRO',
    location (lon, lat, el)=(-119.4 deg, 37.07 deg, 1405.000000000719 m),
    timezone=<DstTzInfo 'America/Los_Angeles' LMT-1 day, 16:07:00 STD>,
    pressure=<Quantity 1.011 bar>,
    temperature=<Quantity 5.78 deg_C>,
    relative_humidity=0.6>
```

In [18]:

```
# Check for observing
```

In [17]:
```
# Define object & obs time
start =Time.now()
stop = Time.now() + 1
time_range = [start, stop]
print("Start:{}\nStop: {}".format(start, stop))

Observatory = observer_SRO

targets = []
for i in neabyGal:
    gal = SkyCoord(i['ra'], i['dec'], unit=('deg', 'deg'), frame='icrs')
    obj = FixedTarget(name="{}".format(i['objname']), coord=gal)
    targets.append(obj)
```

```
Start:2024-01-30 12:51:04.694866
Stop: 2024-01-31 12:51:04.695269
WARNING: TimeDeltaMissingUnitWarning: Numerical value without unit or expli
cit format passed to TimeDelta, assuming days [astropy.time.core]
```

In [21]:
```
targets
```

Out[21]:
```
[<FixedTarget "WKK 3062" at SkyCoord (ICRS): (ra, dec) in deg (213.62734, -
70.43829)>,
 <FixedTarget "WKK 7923" at SkyCoord (ICRS): (ra, dec) in deg (256.69612, -
55.08547)>,
 <FixedTarget "2MFGC 02289" at SkyCoord (ICRS): (ra, dec) in deg (42.81658,
45.05714)>,
 <FixedTarget "WISEA J164708.47-173959.9" at SkyCoord (ICRS): (ra, dec) in
deg (251.78533, -17.66644)>,
 <FixedTarget "WISEA J143804.14-411453.6" at SkyCoord (ICRS): (ra, dec) in
deg (219.51721, -41.24836)>,
 <FixedTarget "WISEA J174223.49-090132.6" at SkyCoord (ICRS): (ra, dec) in
deg (265.59804, -9.02569)>,
 <FixedTarget "WISEA J153710.66-185448.0" at SkyCoord (ICRS): (ra, dec) in
deg (234.29421, -18.91372)>,
 <FixedTarget "WISEA J163135.05-225048.4" at SkyCoord (ICRS): (ra, dec) in
deg (247.89613, -22.84686)>,
 <FixedTarget "WISEA J192729.92-100453.9" at SkyCoord (ICRS): (ra, dec) in
deg (291.8745, -10.08164)>,
 <FixedTarget "WISEA J150816.26-433109.1" at SkyCoord (ICRS): (ra, dec) in
deg (227.06783, -43.51919)>,
 <FixedTarget "WISEA J050736.93+175112.4" at SkyCoord (ICRS): (ra, dec) in
deg (76.90371, 17.85314)>,
 <FixedTarget "WISEA J035657.01+425540.4" at SkyCoord (ICRS): (ra, dec) in
deg (59.23763, 42.92794)>,
 <FixedTarget "WISEA J062210.55+101845.9" at SkyCoord (ICRS): (ra, dec) in
deg (95.54367, 10.3125)>,
 <FixedTarget "WISEA J190032.60-372917.8" at SkyCoord (ICRS): (ra, dec) in
deg (285.13575, -37.48853)>,
 <FixedTarget "WISEA J165635.42-183734.4" at SkyCoord (ICRS): (ra, dec) in
deg (254.14812, -18.6265)>,
 <FixedTarget "WISEA J044148.10+125325.4" at SkyCoord (ICRS): (ra, dec) in
deg (70.45042, 12.89025)>,
 <FixedTarget "WISEA J045503.89+335030.7" at SkyCoord (ICRS): (ra, dec) in
deg (73.76637, 33.8415)>,
 <FixedTarget "WISEA J022200.27+503737.1" at SkyCoord (ICRS): (ra, dec) in
deg (35.50117, 50.62697)>,
 <FixedTarget "WISEA J193324.51+214917.1" at SkyCoord (ICRS): (ra, dec) in
deg (293.35195, 21.82165)>,
 <FixedTarget "WISEA J144124.55-444152.4" at SkyCoord (ICRS): (ra, dec) in
deg (220.35237, -44.69794)>]
```

In [18]:
```
# observation constraints

constraints = [AltitudeConstraint(15*u.deg, 85*u.deg), AirmassConstraint(5

#Chang STATION
# Are targets *ever* observable in the time range?
ever_observable_SRO = is_observable(constraints, observer_SRO, targets, tir
ever_observable_SBO = is_observable(constraints, observer_SBO, targets, tir
ever_observable_GAO = is_observable(constraints, observer_GAO, targets, tir
# Are targets *always* observable in the time range?
always_observable = is_always_observable(constraints, Observatory, targets

# During what months are the targets ever observable?
best_months = months_observable(constraints, Observatory, targets, time_rar
```

In [19]:
```
obs_table = Table()
# obs_table['targets'] = [target.name for target in targets]
obs_table['ever_observable_SRO'] = ever_observable_SRO
obs_table['ever_observable_SBO'] = ever_observable_SBO
obs_table['ever_observable_GAO'] = ever_observable_GAO
# obs_table['always_observable'] = always_observable
print(obs_table)
```

```
ever_observable_SRO ever_observable_SBO ever_observable_GAO
------------------- ------------------- -------------------
              False                True               False
              False               False               False
               True                True                True
               True                True                True
               True               False                True
               True                True                True
              False                True               False
               True                True                True
               True               False                True
               True               False                True
              False                True               False
              False                True               False
               True               False                True
               True               False                True
               True               False               False
               True               False                True
               True               False                True
              False                True               False
              False               False               False
               True               False                True
```

In [20]:
```
# from astroplan import observability_table
obs_table_ability_SRO = observability_table(constraints, observer_SRO, tar
obs_table_ability_SBO = observability_table(constraints, observer_SBO, tar
obs_table_ability_GAO = observability_table(constraints, observer_GAO, tar
obs_table_ability = observability_table(constraints, Observatory, targets,

able_SRO = obs_table_ability_SRO["ever observable"]
able_SBO = obs_table_ability_SBO["ever observable"]
able_GAO = obs_table_ability_GAO["ever observable"]

frac_SRO = obs_table_ability_SRO['fraction of time observable']
frac_SBO = obs_table_ability_SBO['fraction of time observable']
frac_GAO = obs_table_ability_GAO['fraction of time observable']


print(len(obs_table_ability["target name"]))
```

```
20
```

In [21]:
```
# distinguis each telescope able to observe by fraction of time
observe_SRO = []
observe_SBO = []
observe_GAO = []
for name, fsro, fsbo, fgao in zip(obs_table_ability["target name"],frac_SR
#     print(name, fsro, fsbo, fgao)  # Add this line for debugging
    if fsro >= fsbo and fsro >= fgao:
#         print (fsro, fsbo, fgao)
        observe_SRO.append(name)
    elif fsbo >= fsro and fsbo >= fgao:
        observe_SBO.append(name)
    elif fgao >= fsro and fgao >= fsbo:
        observe_GAO.append(name)
# pprint.pprint(observe_SRO)
print(len(observe_GAO))
print(len(observe_SBO))
print(len(observe_SRO))
```

```
0
6
14
```

In [22]:
```
observe_SRO
```

Out[22]:
```
['WISEA J220828.11-101851.7',
 'WISEA J112419.29+331558.1',
 'WISEA J113453.58+361435.0',
 'WISEA J144433.53+563636.8',
 'WISEA J102432.58+031012.8',
 'WISEA J133007.05+553312.9',
 'WISEA J143432.90+554426.2',
 'WISEA J142538.65+555839.1',
 'WISEA J140119.34+541856.8',
 'WISEA J213302.40+060847.1',
 'WISEA J133412.28+530109.5',
 'WISEA J140026.59+584328.9',
 'WISEA J221909.00-163051.4',
 'WISEA J124140.38+481409.4']
```

In [23]:
```
# contain coordinate to plot SRO
obsable_SRO = []
for x in (observe_SRO):
    for i,y in enumerate(Gal_for_obs["objname"]):
        if x == y :
            obsable_SRO.append((Gal_for_obs["ra"][i], Gal_for_obs["dec"][i
```

In [24]:
```
obsable_SRO
```

Out[24]:
```
[(332.11717, -10.31433),
 (171.08041, 33.26619),
 (173.72327, 36.24297),
 (221.13991, 56.61032),
 (156.13596, 3.17028),
 (202.52971, 55.55367),
 (218.63724, 55.74049),
 (216.41091, 55.97757),
 (210.33045, 54.31595),
 (323.26, 6.14644),
 (203.55123, 53.01933),
 (210.111, 58.72472),
 (334.78758, -16.51422),
 (190.41829, 48.23594)]
```

In [25]:
```
# contain coordinate to plot SBO
obsable_SBO = []
for x in (observe_SBO):
    for i,y in enumerate(Gal_for_obs["objname"]):
        if x == y :
            obsable_SBO.append((Gal_for_obs["ra"][i], Gal_for_obs["dec"][i
```

In [26]:
```
# contain coordinate to plot GAO
obsable_GAO = []
for x in (observe_GAO):
    for i,y in enumerate(Gal_for_obs["objname"]):
        if x == y :
            obsable_GAO.append((Gal_for_obs["ra"][i], Gal_for_obs["dec"][i
```

In [28]:
```
# ontain data to do API

targets_SRO = []
targets_SBO = []
targets_GAO = []

for n,m in zip(observe_SRO, obsable_SRO) :
    gal = SkyCoord(m[0],m[1], unit=('deg', 'deg'), frame='icrs')
    obj = FixedTarget(name="{}".format(n), coord=gal)
    targets_SRO.append(obj)

for n,m in zip(observe_SBO, obsable_SBO) :
    gal = SkyCoord(m[0],m[1], unit=('deg', 'deg'), frame='icrs')
    obj = FixedTarget(name="{}".format(n), coord=gal)
    targets_SBO.append(obj)

for n,m in zip(observe_GAO, obsable_GAO) :
    gal = SkyCoord(m[0],m[1], unit=('deg', 'deg'), frame='icrs')
    obj = FixedTarget(name="{}".format(n), coord=gal)
    targets_GAO.append(obj)

obs_table_ability_SRO = observability_table(constraints, observer_SRO, tar
obs_table_ability_SBO = observability_table(constraints, observer_SBO, tar
# obs_table_ability_GAO = observability_table(constraints, observer_GAO, t
```

In [29]:
```
obs_table_ability_SRO
```

Out[29]: *Table length=14*

target name	ever observable	always observable	fraction of time observable
str25	bool	bool	float64
WISEA J220828.11-101851.7	False	False	0.0
WISEA J112419.29+331558.1	True	False	0.42857142857142855
WISEA J113453.58+361435.0	True	False	0.42857142857142855
WISEA J144433.53+563636.8	True	False	0.3877551020408163
WISEA J102432.58+031012.8	True	False	0.42857142857142855
WISEA J133007.05+553312.9	True	False	0.42857142857142855
WISEA J143432.90+554426.2	True	False	0.3877551020408163
WISEA J142538.65+555839.1	True	False	0.3877551020408163
WISEA J140119.34+541856.8	True	False	0.3877551020408163
WISEA J213302.40+060847.1	True	False	0.02040816326530612
WISEA J133412.28+530109.5	True	False	0.40816326530612246
WISEA J140026.59+584328.9	True	False	0.42857142857142855
WISEA J221909.00-163051.4	False	False	0.0
WISEA J124140.38+481409.4	True	False	0.42857142857142855

targets_true = [] for x,y in zip(targets, obs_table['ever_observable']): if y == True:
targets_true.append(x)len(targets_true)

In [60]:

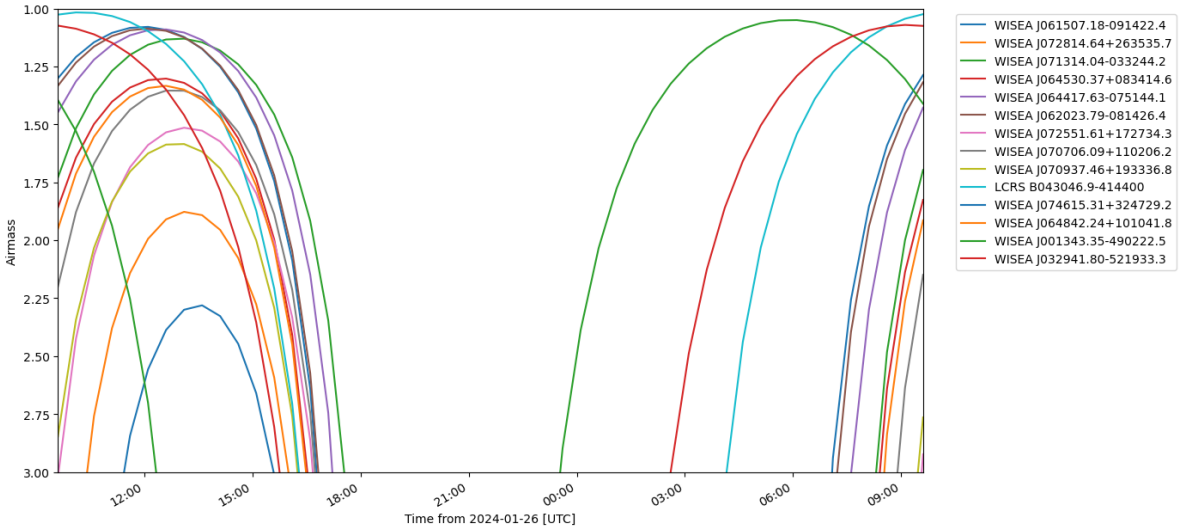
```
# define time grid from our observing time range
time_grid = time_grid_from_range(time_range)

plt.figure(figsize=(10,10))
cmap = cm.Set1              # Cycle through this colormap

for i, target in enumerate(targets_true):
    ax = plot_sky(target, Observatory, time_grid,
                  style_kwargs=dict(color=cmap(float(i)/len(targets_true))
                                    label=target.name))

    legend = ax.legend(loc='upper right', bbox_to_anchor=(2.0, 1.0))

    legend.get_frame().set_facecolor('w')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[60], line 7
      4 plt.figure(figsize=(10,10))
      5 cmap = cm.Set1              # Cycle through this colormap
----> 7 for i, target in enumerate(targets_true):
      8     ax = plot_sky(target, Observatory, time_grid,
      9                   style_kwargs=dict(color=cmap(float(i)/len(targets
_true)),
     10                                     label=target.name))
     12     legend = ax.legend(loc='upper right', bbox_to_anchor=(2.0, 1.
0))

NameError: name 'targets_true' is not defined
<Figure size 1000x1000 with 0 Axes>
```

In [70]:

```
# from astroplan.plots import plot_airmass
plt.figure(figsize=(13,7))
plot_airmass(targets_true, Observatory, time_grid)
plt.legend(loc='upper right', bbox_to_anchor=(1.3, 1.0))
plt.show()
```

```
WARNING: TimeDeltaMissingUnitWarning: Numerical value without unit or expli
cit format passed to TimeDelta, assuming days [astropy.time.core]
```



In [11]:

```
# Sun
print(start, '\n')
sunset_tonight = Observatory.sun_set_time(start, which='nearest')
sunrise_tonight = Observatory.sun_rise_time(start, which='nearest')

print("Sun sets  (UTC): {}".format(sunset_tonight.iso))
print("Sun rises (UTC): {}".format(sunrise_tonight.iso))
# set end observation
end_obs = stop + 5
exp_obs = stop + 10
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[11], line 2
      1 # Sun
----> 2 print(start, '\n')
      3 sunset_tonight = Observatory.sun_set_time(start, which='nearest')
      4 sunrise_tonight = Observatory.sun_rise_time(start, which='nearest')

NameError: name 'start' is not defined
```

NED Observation on TRT API

In [31]:

```
# API address
url = "http://192.168.2.40:2800/hub/api/newobservation"
keep_id = []

#Observation to input
for i, j in zip(obsable_SRO, obs_table_ability_SRO):
    if j[1]:
#         print(j['ever observable'], j['target name'], Gal_for_obs[i]['ra
        radec = SkyCoord(i[0], i[1], unit=('deg', 'deg'), frame='icrs')
        ra, dec = radec.to_string('hmsdms').split(" ")
        script= dict([("ObjectName", j['target name']),
                      ("StationName", "SRO"),
                      ("RA", "11:24:19"),
                      ("DEC", dec.replace("d", ":").replace("m", ":").replace(
                      ("Subframe", "1"),
                      ("BinningXY", "1,1"),
                      ("CadenceInterval", "00:00:00"),
                      ("MaxAirmass", "3"),
                      ("PA", "0"),
                      ("Dither", "0"),
                      ("ExpiryDate", "{}".format(exp_obs.iso)),
                      ("StartDate", "{}".format(sunset_tonight.iso)),
                      ("EndDate", "{}".format(end_obs.iso)),
                      ("ExposuresMode", "1"),
                      ("M3Port", "1"),
                      ("Filter", ["B", "V", "R", "I"]),
                      ("Suffix", ["{}_B".format(GW_event),"{}_V".format(GW_ever
                      ("Exposure", ["120","120","120", "120"]),
                      ("Repeat", ["5","5","5", "5"])
                     ])

        payload = json.dumps({"script": [ script] })
        print(payload, "\n")
        headers = {
              'Content-Type': 'application/json',
              'TRT': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1ZDk(
        }
        if True:
            response = requests.request("POST", url, headers=headers, data=
            keep_id.append(response.text)
            print(response.text)

    else:
        pass
```

{"script": [{"ObjectName": "WISEA J112419.29+331558.1", "StationName": "SR
O", "RA": "11:24:19.2984", "DEC": "+33:15:58.284", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24OPCF_1"]
{"script": [{"ObjectName": "WISEA J113453.58+361435.0", "StationName": "SR
O", "RA": "11:34:53.5848", "DEC": "+36:14:34.692", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12

0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24GE1M_1"]
{"script": [{"ObjectName": "WISEA J144433.53+563636.8", "StationName": "SR
O", "RA": "14:44:33.5784", "DEC": "+56:36:37.152", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24ESPX_1"]
{"script": [{"ObjectName": "WISEA J102432.58+031012.8", "StationName": "SR
O", "RA": "10:24:32.6304", "DEC": "+03:10:13.008", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24I1H8_1"]
{"script": [{"ObjectName": "WISEA J133007.05+553312.9", "StationName": "SR
O", "RA": "13:30:07.1304", "DEC": "+55:33:13.212", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["246N3T_1"]
{"script": [{"ObjectName": "WISEA J143432.90+554426.2", "StationName": "SR
O", "RA": "14:34:32.9376", "DEC": "+55:44:25.764", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24CK2E_1"]
{"script": [{"ObjectName": "WISEA J142538.65+555839.1", "StationName": "SR
O", "RA": "14:25:38.6184", "DEC": "+55:58:39.252", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["245CQD_1"]
{"script": [{"ObjectName": "WISEA J140119.34+541856.8", "StationName": "SR
O", "RA": "14:01:19.308", "DEC": "+54:18:57.42", "Subframe": "1", "BinningX
Y": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "Di
ther": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-01-
30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode": "
1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["243TCU_1"]
{"script": [{"ObjectName": "WISEA J213302.40+060847.1", "StationName": "SR

O", "RA": "21:33:02.4", "DEC": "+06:08:47.184", "Subframe": "1", "BinningX
Y": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "Di
ther": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-01-
30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode": "
1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["2429EC_1"]
{"script": [{"ObjectName": "WISEA J133412.28+530109.5", "StationName": "SR
O", "RA": "13:34:12.2952", "DEC": "+53:01:09.588", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24M7N7_1"]
{"script": [{"ObjectName": "WISEA J140026.59+584328.9", "StationName": "SR
O", "RA": "14:00:26.64", "DEC": "+58:43:28.992", "Subframe": "1", "BinningX
Y": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "Di
ther": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-01-
30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode": "
1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["244R6G_1"]
{"script": [{"ObjectName": "WISEA J124140.38+481409.4", "StationName": "SR
O", "RA": "12:41:40.3896", "DEC": "+48:14:09.384", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["2444P2_1"]

In [32]:

```
# API address
url = "http://192.168.2.40:2800/hub/api/newobservation"
keep_id = []

#Observation to input
for i, j in zip(obsable_SBO, obs_table_ability_SBO):
    if j[1]:
#         print(j['ever observable'], j['target name'], Gal_for_obs[i]['ra
        radec = SkyCoord(i[0], i[1], unit=('deg', 'deg'), frame='icrs')
        ra, dec = radec.to_string('hmsdms').split(" ")
        script= dict([("ObjectName", j['target name']),
                      ("StationName", "SBO"),
                      ("RA", ra.replace("h", ":").replace("m", ":").replace("s'
                      ("DEC", dec.replace("d", ":").replace("m", ":").replace('
                      ("Subframe", "1"),
                      ("BinningXY", "1,1"),
                      ("CadenceInterval", "00:00:00"),
                      ("MaxAirmass", "3"),
                      ("PA", "0"),
                      ("Dither", "0"),
                      ("ExpiryDate", "{}".format(exp_obs.iso)),
                      ("StartDate", "{}".format(sunset_tonight.iso)),
                      ("EndDate", "{}".format(end_obs.iso)),
                      ("ExposuresMode", "1"),
                      ("M3Port", "1"),
                      ("Filter", ["B", "V", "R", "I"]),
                      ("Suffix", ["{}_B".format(GW_event),"{}_V".format(GW_ever
                      ("Exposure", ["120","120","120", "120"]),
                      ("Repeat", ["5","5","5", "5"])
                     ])

        payload = json.dumps({"script": [ script] })
        print(payload, "\n")
        headers = {
              'Content-Type': 'application/json',
              'TRT': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1ZDk(
        }
        if True:
            response = requests.request("POST", url, headers=headers, data=
            keep_id.append(response.text)
            print(response.text)

    else:
        pass
```

{"script": [{"ObjectName": "WISEA J011010.81-605341.2", "StationName": "SB
O", "RA": "01:10:10.9008", "DEC": "-60:53:42.216", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24PM8N_1"]
{"script": [{"ObjectName": "WISEA J235616.98-522146.6", "StationName": "SB
O", "RA": "23:56:16.9992", "DEC": "-52:21:47.196", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12

0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24KHN9_1"]
{"script": [{"ObjectName": "WISEA J094032.30-100702.1", "StationName": "SB
O", "RA": "09:40:32.2992", "DEC": "-10:07:01.992", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24OSTT_1"]
{"script": [{"ObjectName": "WISEA J030018.16-663856.0", "StationName": "SB
O", "RA": "03:00:18.12", "DEC": "-66:38:55.608", "Subframe": "1", "BinningX
Y": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "Di
ther": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-01-
30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode": "
1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["243S2F_1"]
{"script": [{"ObjectName": "WISEA J043546.79-640627.5", "StationName": "SB
O", "RA": "04:35:46.9704", "DEC": "-64:06:27.108", "Subframe": "1", "Binnin
gXY": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "
Dither": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-0
1-30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode":
"1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24P14I_1"]
{"script": [{"ObjectName": "WISEA J065717.77-554614.6", "StationName": "SB
O", "RA": "06:57:17.8008", "DEC": "-55:46:14.7", "Subframe": "1", "BinningX
Y": "1,1", "CadenceInterval": "00:00:00", "MaxAirmass": "3", "PA": "0", "Di
ther": "0", "ExpiryDate": "2024-02-10 12:51:04.695", "StartDate": "2024-01-
30 01:15:24.457", "EndDate": "2024-02-05 12:51:04.695", "ExposuresMode": "
1", "M3Port": "1", "Filter": ["B", "V", "R", "I"], "Suffix": ["S230731an_
B", "S230731an_V", "S230731an_R", "S230731an_I"], "Exposure": ["120", "12
0", "120", "120"], "Repeat": ["5", "5", "5", "5"]}]}

["24WCZG 1"]

In [10]:

```
# API address
url = "http://192.168.2.40:2800/hub/api/newobservation"
keep_id = []

#Observation to input
radec = SkyCoord(10, 50, unit=('deg', 'deg'), frame='icrs')
ra, dec = radec.to_string('hmsdms').split(" ")
script= dict([("ObjectName", 'name'),
                  ("StationName", "GAO"),
                  ("RA", ra.replace("h", ":").replace("m", ":").replace("s
                  ("DEC", dec.replace("d", ":").replace("m", ":").replace(
                  ("Subframe", "1"),
                  ("BinningXY", "1,1"),
                  ("CadenceInterval", "00:00:00"),
                  ("MaxAirmass", "3"),
                  ("PA", "0"),
                  ("Dither", "0"),
                  ("ExpiryDate", "{}".format(exp_obs.iso)),
                  ("StartDate", "{}".format(sunset_tonight.iso)),
                  ("EndDate", "{}".format(end_obs.iso)),
                  ("ExposuresMode", "1"),
                  ("M3Port", "1"),
                  ("Filter", ["B", "V", "R", "I"]),
                  ("Suffix", ["{}_B".format(GW_event),"{}_V".format(GW_ever
                  ("Exposure", ["120","120","120", "120"]),
                  ("Repeat", ["5","5","5", "5"])
                 ])

payload = json.dumps({"script": [ script] })
print(payload, "\n")
headers = {
              'Content-Type': 'application/json',
              'TRT': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1ZDk(
}
if False:
            response = requests.request("POST", url, headers=headers, data=
            keep_id.append(response.text)
            print(response.text)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[10], line 18
      6 radec = SkyCoord(10, 50, unit=('deg', 'deg'), frame='icrs')
      7 ra, dec = radec.to_string('hmsdms').split(" ")
      8 script= dict([("ObjectName", 'name'),
      9                   ("StationName", "GAO"),
     10                   ("RA", ra.replace("h", ":").replace("m", ":").rep
lace("s", "")),
     11                   ("DEC", dec.replace("d", ":").replace("m", ":").r
eplace("s", "")),
     12                   ("Subframe", "1"),
     13                   ("BinningXY", "1,1"),
     14                   ("CadenceInterval", "00:00:00"),
     15                   ("MaxAirmass", "3"),
     16                   ("PA", "0"),
     17                   ("Dither", "0"),
---> 18                   ("ExpiryDate", "{}".format(exp_obs.iso)),
     19                   ("StartDate", "{}".format(sunset_tonight.iso)),
     20                   ("EndDate", "{}".format(end_obs.iso)),
     21                   ("ExposuresMode", "1"),
     22                   ("M3Port", "1"),
```

```
     23                   ("Filter", ["B", "V", "R", "I"]),
     24                   ("Suffix", ["{}_B".format(GW_event),"{}_V".format
(GW_event),"{}_R".format(GW_event), "{}_I".format(GW_event)]),
     25                   ("Exposure", ["120","120","120", "120"]),
     26                   ("Repeat", ["5","5","5", "5"])
     27                  ])
     29 payload = json.dumps({"script": [ script] })
     30 print(payload, "\n")
```

keep observation id in ascii text

In [48]:
```
file_to_write = "{}_obs_id.txt".format(GW_event)
file = open(file_to_write, 'w')
for i in keep_id:
    txt = i.replace('[', '').replace(']', '').replace('"', '')
    file.write(txt+"\n")
file.close()
```

In [49]:
```
rfile = open(file_to_write, 'r')
txt = rfile.read()
rfile.close()

print(txt.rsplit())
```

```
['24N4JD_1', '24TNDF_1', '24VV1Q_1']
```

Cancel Script

In [50]:
```
keep_cid = txt.rsplit()
```

In [51]:
```
url = "http://192.168.2.40:2800/hub/api/cancelobservation"

payload = json.dumps({
    "obs_id": keep_cid
})
print(payload)

headers = {
  'Content-Type': 'application/json',
  'TRT': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1ZDk0MTdhNzQzNzU
}
response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

```
{"obs_id": ["24N4JD_1", "24TNDF_1", "24VV1Q_1"]}
{"n":3,"nModified":3,"ok":1}
```

In [8]:

```
# Cancel specific ID
url = "http://192.168.2.40:2800/hub/api/cancelobservation"

payload = json.dumps({
    "obs_id": ["24ZKUG_1", "24YP9X_1", "24F6P5_1", "24WCZG_1", "24P14I_1",
               "24OSTT_1", "24PM8N_1", "241UNV_1", "24FUV7_1", "24GE6G_1",
               "24KVNU_1", "24ZQM5_1", "246UF4_1", "2451JX_1", "24EIDM_1",
               "24E9VS_1", "249S50_1", "24R90U_1", "24POJO_1", "24TUD6_1",
               "24UF3N_1", "2466KE_1", "24IB4K_1", "24W6MB_1", "24FHR7_1"]
})
print(payload)

headers = {
  'Content-Type': 'application/json',
  'TRT': 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI1ZDk0MTdhNzQzNzU
}
response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

```
{"obs_id": ["24ZKUG_1", "24YP9X_1", "24F6P5_1", "24WCZG_1", "24P14I_1", "24
3SF_1", "24OSTT_1", "24PM8N_1", "241UNV_1", "24FUV7_1", "24GE6G_1", "246LP4
_1", "24KVNU_1", "24ZQM5_1", "246UF4_1", "2451JX_1", "24EIDM_1", "24PS25_
1", "24E9VS_1", "249S50_1", "24R90U_1", "24POJO_1", "24TUD6_1", "24OZQU_1",
"24UF3N_1", "2466KE_1", "24IB4K_1", "24W6MB_1", "24FHR7_1"]}
{"n":28,"nModified":28,"ok":1}
```

In [ ]: