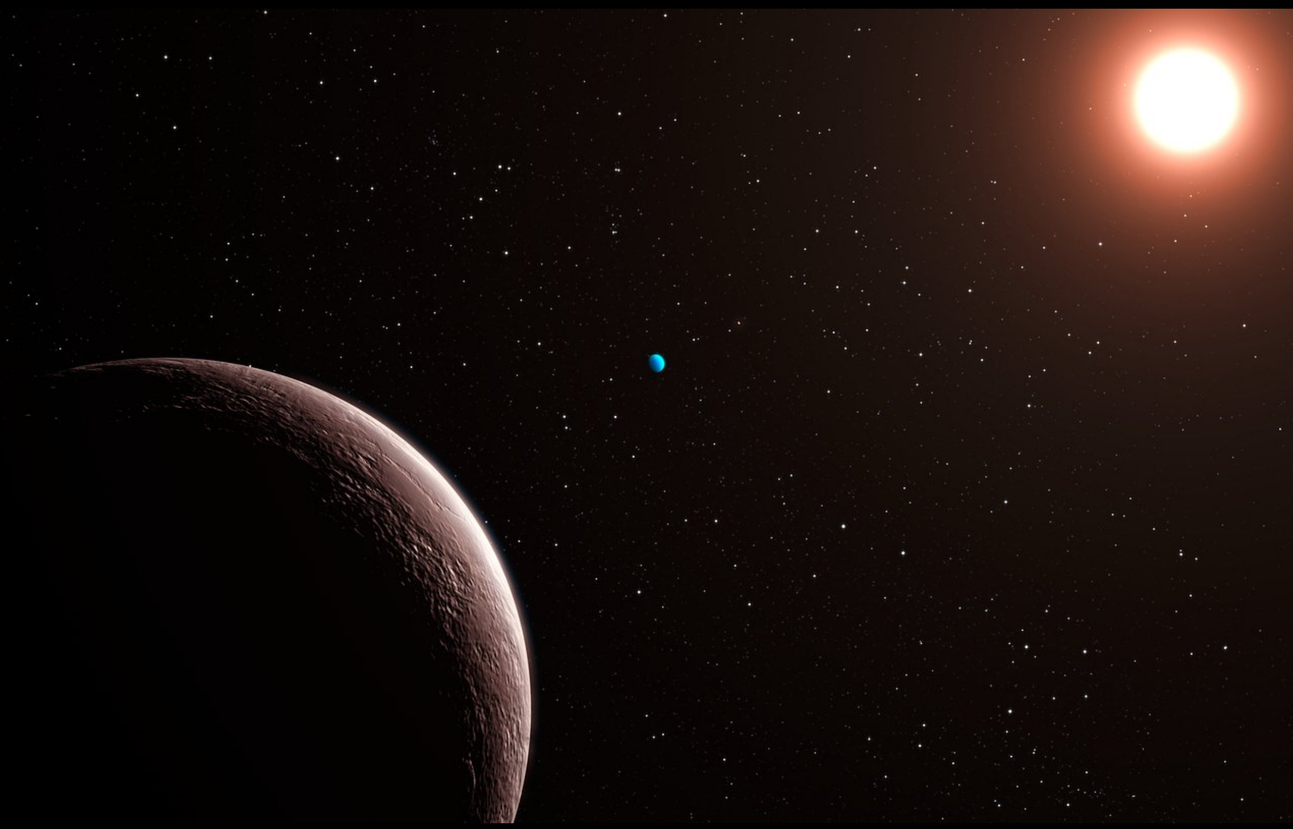
# Scheduler Optimisation for NIRPS Observations devoted to search for exoplanets around M dwarfs

ARPITA GANGULY
ASTROPHYSICS LAB II
UNIVERSITY OF GENEVA



Credit:ESO/L. Calçada

# NIRPS
# (Near InfraRed Planet Searcher)

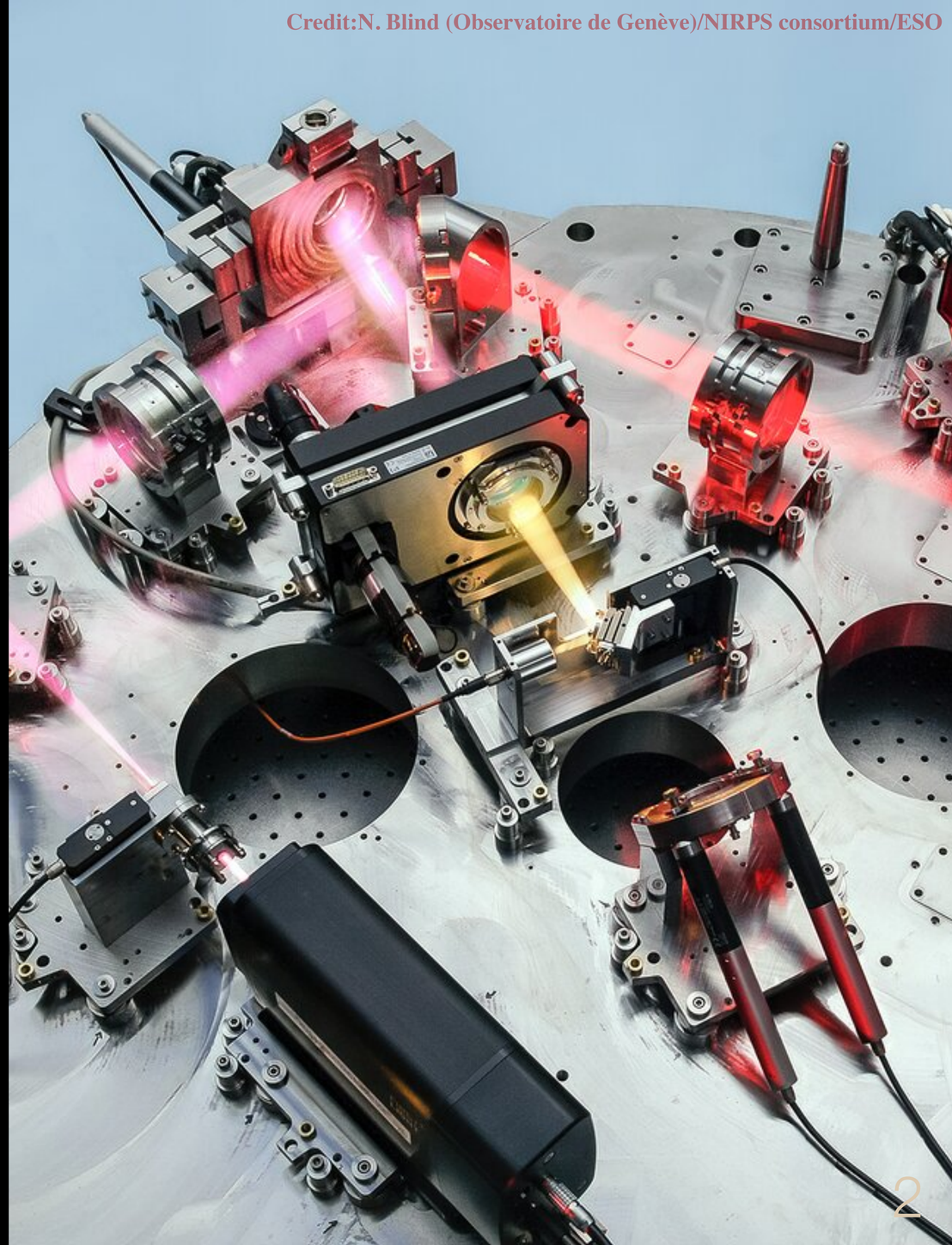
- NIRPS : an infrared precision Radial Velocity spectrograph

- Location : 3.6-m Telescope in La Silla Observatory, Chile

- Complement to HARPS , a well known Planet Searcher :

| |
|---|
| HARPS : Visible |
| NIRPS : Near Infra-Red |

- Started operation : 1st April, 2023



Credit:N. Blind (Observatoire de Genève)/NIRPS consortium/ESO

# Goals for my work ( till now )

**Sharing observing time with other programs : Earnest need for scheduling**

★Gathering Information about the targets ( Properties are essential )

★Explore the essential functionalities of Astroplan

★ Create a priority based schedule of a given list of targets with single observations for a given night.

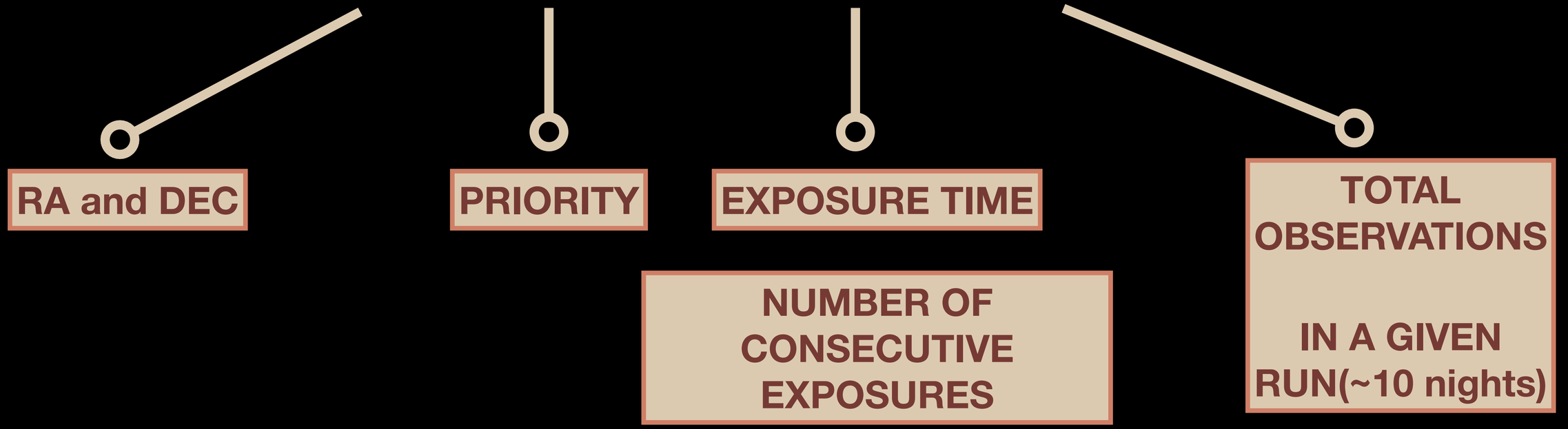
★ Add a time critical target in the observations and create a scheduling sequence around it

# Target Sample

**Over 200 SELECTED  TARGETS :**

**We are interested in the P111 CYCLE (1st April - 1st October)**

**REQUIRED ATTRIBUTES FOR MY WORK**



# REQUIRED FUNCTIONALITIES :

- ➡ TIME ATTRIBUTES
- ➡ CONSTRAINT DEFINITION
- ➡ OBSERVABILITY
- ➡ CREATION OF OBSERVATION BLOCKS
- ➡ TRANSITIONER
- ➡ SEQUENTIAL AND PRIORITY SCHEDULER


An Open Source Python Package for Scheduling Astronomical Observations

# TIME ATTRIBUTES

Define period of observation :

```
start_time = Time('2023-04-30 18:00:00')
end_time = Time('2023-05-01 18:00:00')
```

Find optimal observation time :

```
observe_start =  Obs.twilight_evening_civil(start_time, which='next')
observe_end = Obs.twilight_morning_civil(end_time, which='nearest')
```

# OBSERVABILITY

| | target name | ever observable | always observable | fraction of time observable | DEC | RA |
|---|---|---|---|---|---|---|
| **0** | WASP-178 | True | False | 0.64 | -42.704967 | 227.270327 |
| **1** | HD 225213 | True | False | 0.04 | -37.367744 | 1.383284 |
| **2** | G 158-27 | False | False | 0.00 | -7.546475 | 1.676350 |
| **3** | LP 938-71 | False | False | 0.00 | -37.627709 | 15.720962 |
| **4** | V TZ Ari | False | False | 0.00 | 13.044071 | 30.058987 |
| **...** | ... | ... | ... | ... | ... | ... |
| **106** | TIC 389040826 | True | False | 0.24 | 4.213186 | 141.997143 |
| **107** | TIC 277833995 | True | False | 0.36 | 2.513868 | 156.405506 |
| **108** | K2-239 | True | False | 0.36 | 4.441376 | 160.594121 |
| **109** | K2-33 | True | False | 0.56 | -19.319386 | 242.561362 |
| **110** | UCAC4 511-050629 | True | False | 0.24 | 12.191111 | 160.324950 |

111 rows × 6 columns

# CONSTRAINT DEFINITION

| | |
|---|---|
| AirmassConstraint | For each program the PI sets the limit for their target list |
| AtNightConstraint.twilight_civil() | For optimal observation time |
| MoonSeparationConstraint | 30 degrees(solar type targets)<br>10 degrees ( M-Dwarfs) |

# CREATION OF OBSERVATION BLOCKS & TRANSITIONER

```
blocks.append(ObservingBlock.from_exposures(targets[i], priority[i],
                                            target_exposure[i]*u.second,
                                            n, read_out,configuration = {'mode': mode[i]},
                                            constraints = [night_tc,
                                                           AirmassConstraint(max = 1.9, boolean_constraint = False)]))
```

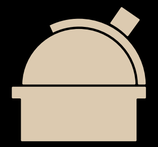
**BLOCK**

- Fixed target Object
- Priority
- Exposure time
- Mode
- Time Constraint
- AirMass constraints

**TRANSITIONER**

- Slew rate of telescope
- Changing between modes:
  HA( 0.4” bright targets)
  HE(0.9”faint targets)
- ~4 min transition time

```
slew_rate = .8*u.deg/u.second
transitioner = Transitioner(slew_rate,{'mode':{('HA','HE'): 240*u.second,
                                               ('HE','HA'): 240*u.second,'default': 30*u.second}})
```



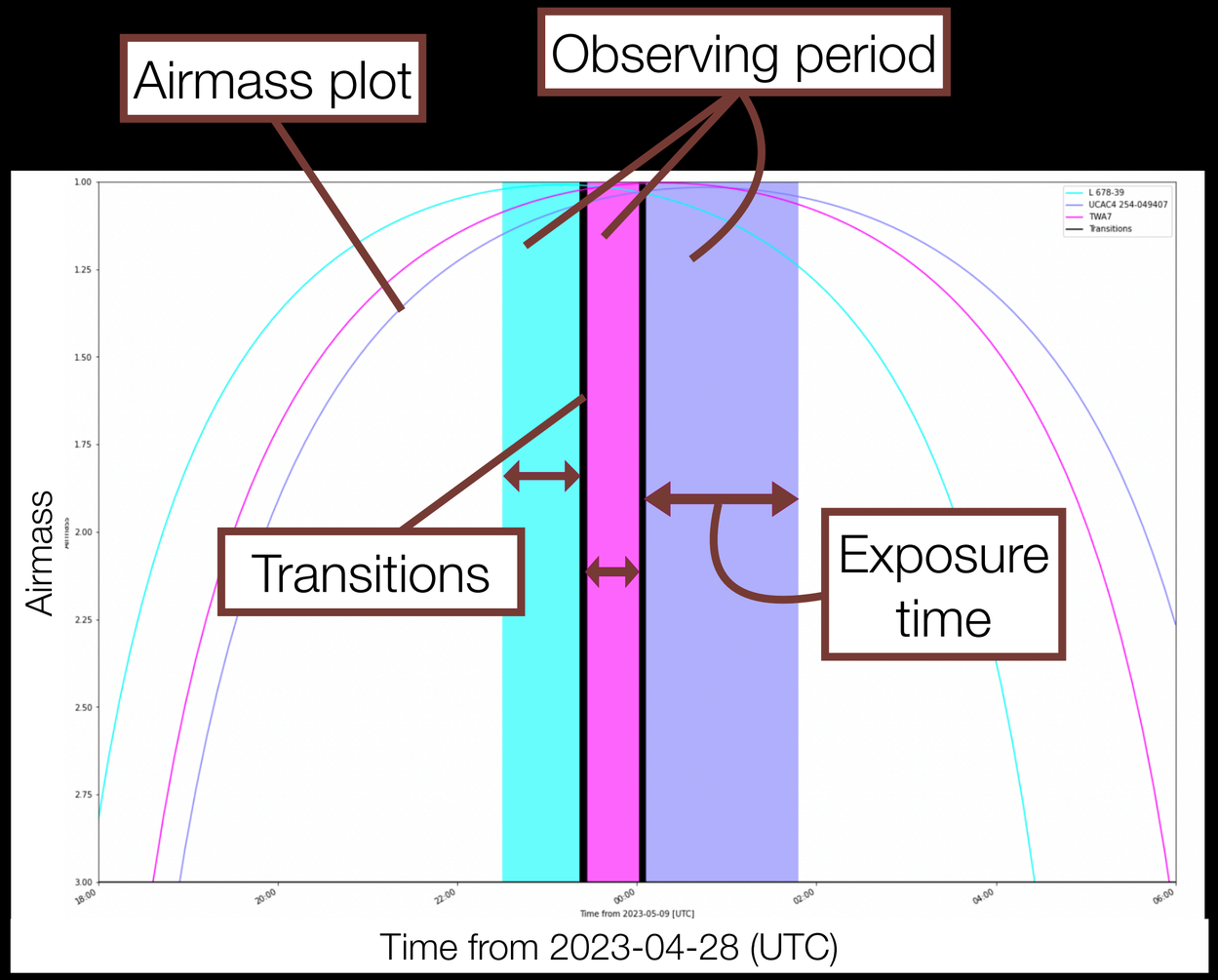
# SEQUENTIAL AND PRIORITY SCHEDULER

```
seq_scheduler = SequentialScheduler(constraints = global_constraints_1,observer = Obs,transitioner = transitioner)
sequential_schedule = Schedule(observe_1_start, observe_2_end)
seq_scheduler(blocks, sequential_schedule)
```

```
prior_scheduler = PriorityScheduler(constraints = global_constraints_1,observer = Obs,transitioner = transitioner)
priority_schedule = Schedule(observe_1_start, observe_2_end)
prior_scheduler(blocks, priority_schedule)
```

Note : Global Constraints : Apply to all targets ,
Thus must be relaxed

# Quick guide :




## Standard Schedule Output

Target

Start and End time(UTC)

Mode

| target | start time (UTC) | end time (UTC) | duration (minutes) | ra | dec | configuration |
|---|---|---|---|---|---|---|
| | | | | deg | deg | |
| str30 | str23 | str23 | float64 | float64 | float64 | object |
| L 678-39 | 2023-04-30 22:39:19.629 | 2023-04-30 22:46:19.629 | 6.999999999999975 | 144.0074851 | -21.6652009 | {'mode': 'HE'} |
| L 320-124 | 2023-04-30 22:50:52.965 | 2023-04-30 22:56:12.965 | 5.333333333333261 | 153.7088543 | -47.1548781 | {'mode': 'HA'} |
| TWA7 | 2023-04-30 22:56:31.042 | 2023-04-30 23:01:51.042 | 5.333333333333261 | 160.6247904 | -33.67126227 | {'mode': 'HA'} |
| DENIS J104814.6-395606 | 2023-04-30 23:05:59.007 | 2023-04-30 23:17:59.007 | 11.999999999999957 | 162.0538877 | -39.93962595 | {'mode': 'HE'} |
| UCAC4 254-049407 | 2023-04-30 23:18:04.772 | 2023-04-30 23:25:04.772 | 6.999999999999975 | 167.9908779 | -39.3281254 | {'mode': 'HE'} |
| L 143-23 | 2023-04-30 23:29:32.631 | 2023-04-30 23:36:32.631 | 6.999999999999975 | 161.0852755 | -61.2026383 | {'mode': 'HA'} |

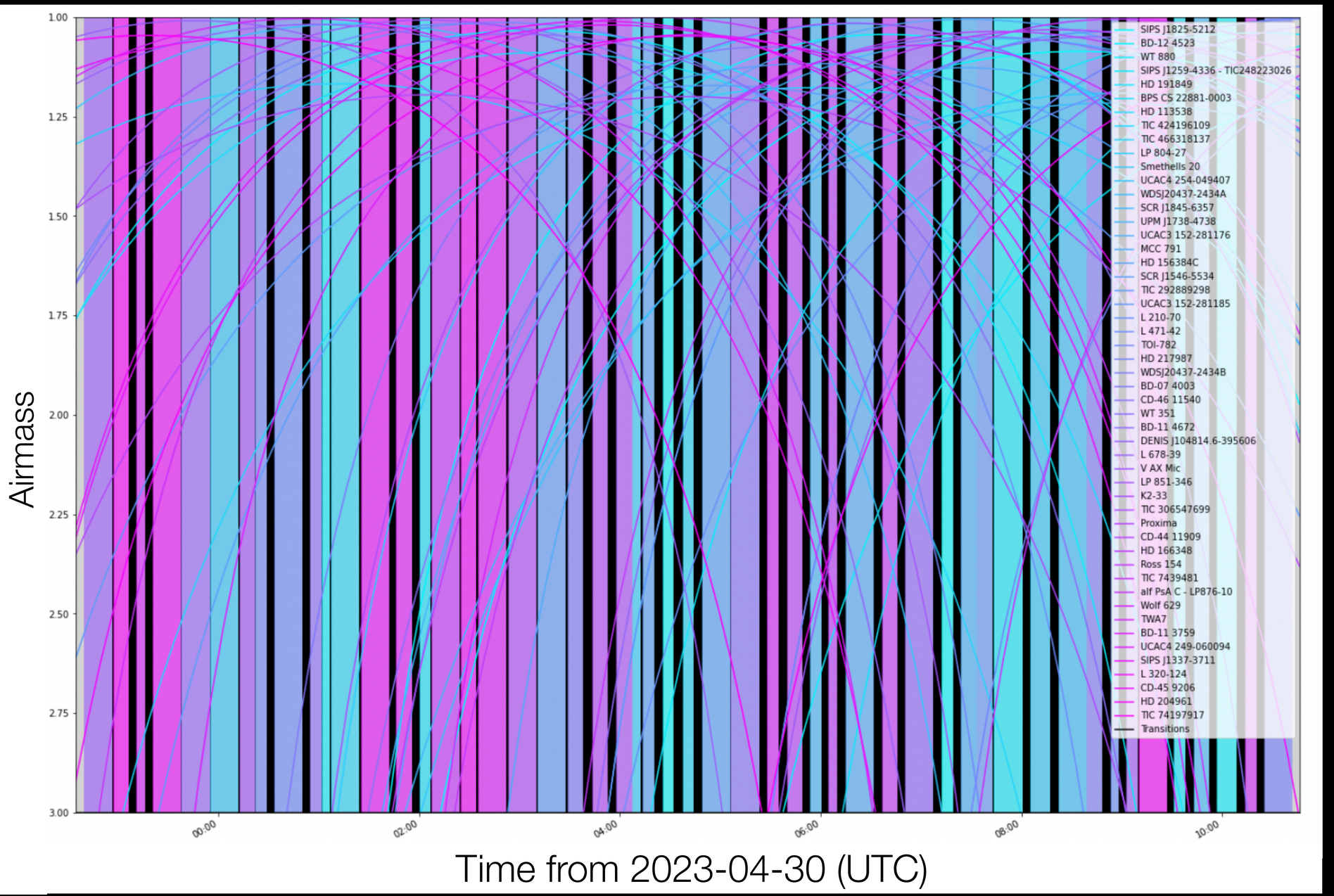
Transitions

RA,DEC (degrees)

# Time to explore :

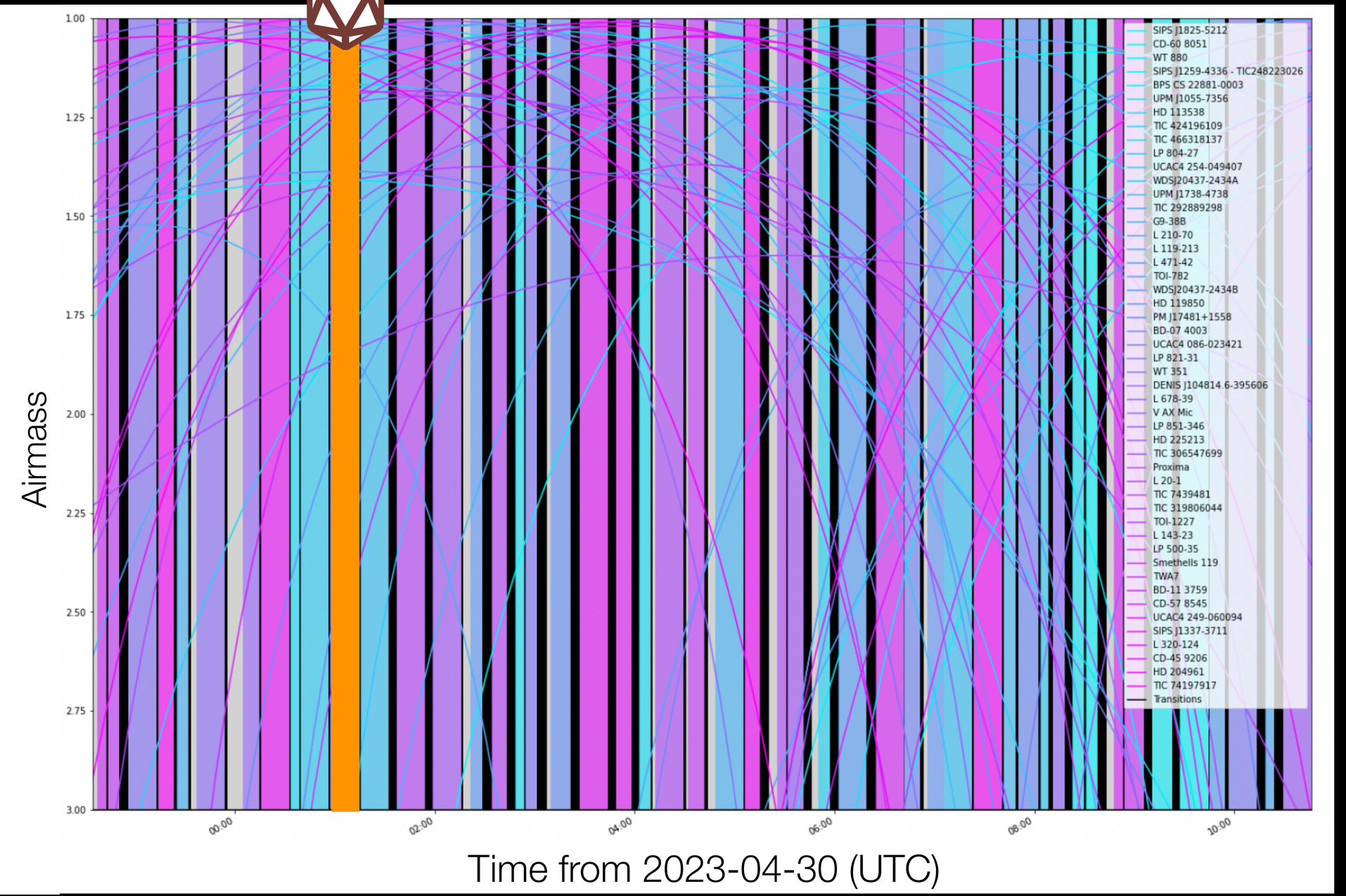
- ✓ Using a random target list with Equal Priorities
- ✓ Add two priority levels to the targets
- ✓ Try adding more priorities
- ✓ Add a time critical target

# SEQUENTIAL SCHEDULE : SORTED RA LIST, SAME PRIORITY




- Does not care about RA
- Puts at best Airmass

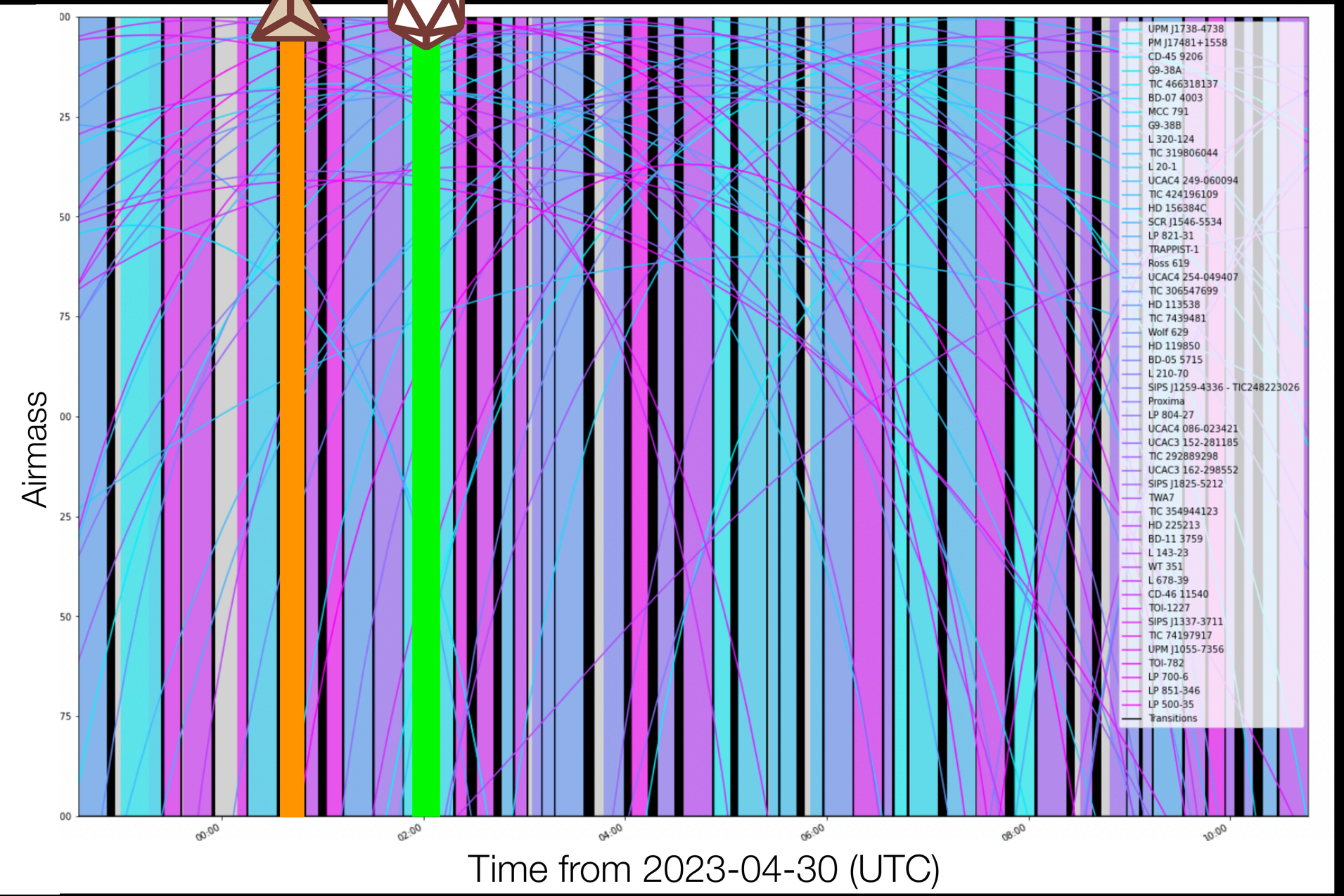
# PRIORITY SCHEDULE : SORTED RA LIST, SAME PRIORITY




**Priority : 1**
**Priority : 2**
**Priority : 3**

- Takes RA into account (Explains the gaps)
- Tries to fill the beginning and end of night
- Still at best Airmass for most targets

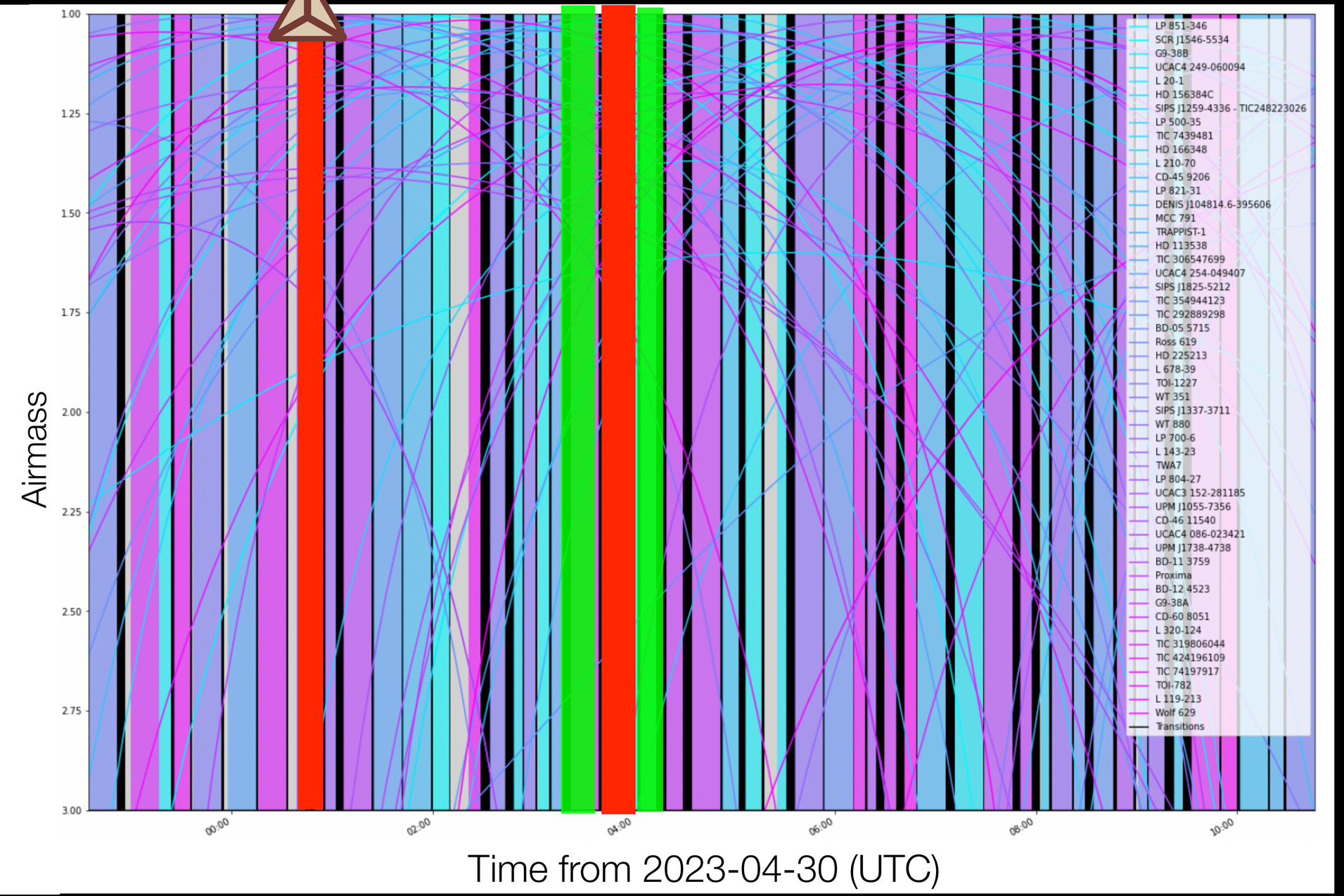
# PRIORITY SCHEDULE : SORTED RA LIST, PRIORITY :1, 2




**Priority : 1**
**Priority : 2**
**Priority : 3**

- TIC 466318137 : moves down the schedule while still satisfying Airmass PRIORITY 1 ➡ 2
- Targets with same RA not scheduled : Different nights

# PRIORITY SCHEDULE : SORTED RA LIST, >2 PRIORITIES




**Priority : 1**
**Priority : 2**
**Priority : 3**

- No more high priority targets at given RA, moves to third priority
- Similarly,TWA7 still there as there is no other target at that RA

Credit: Gabriel Pérez Díaz, SMM (IAC).

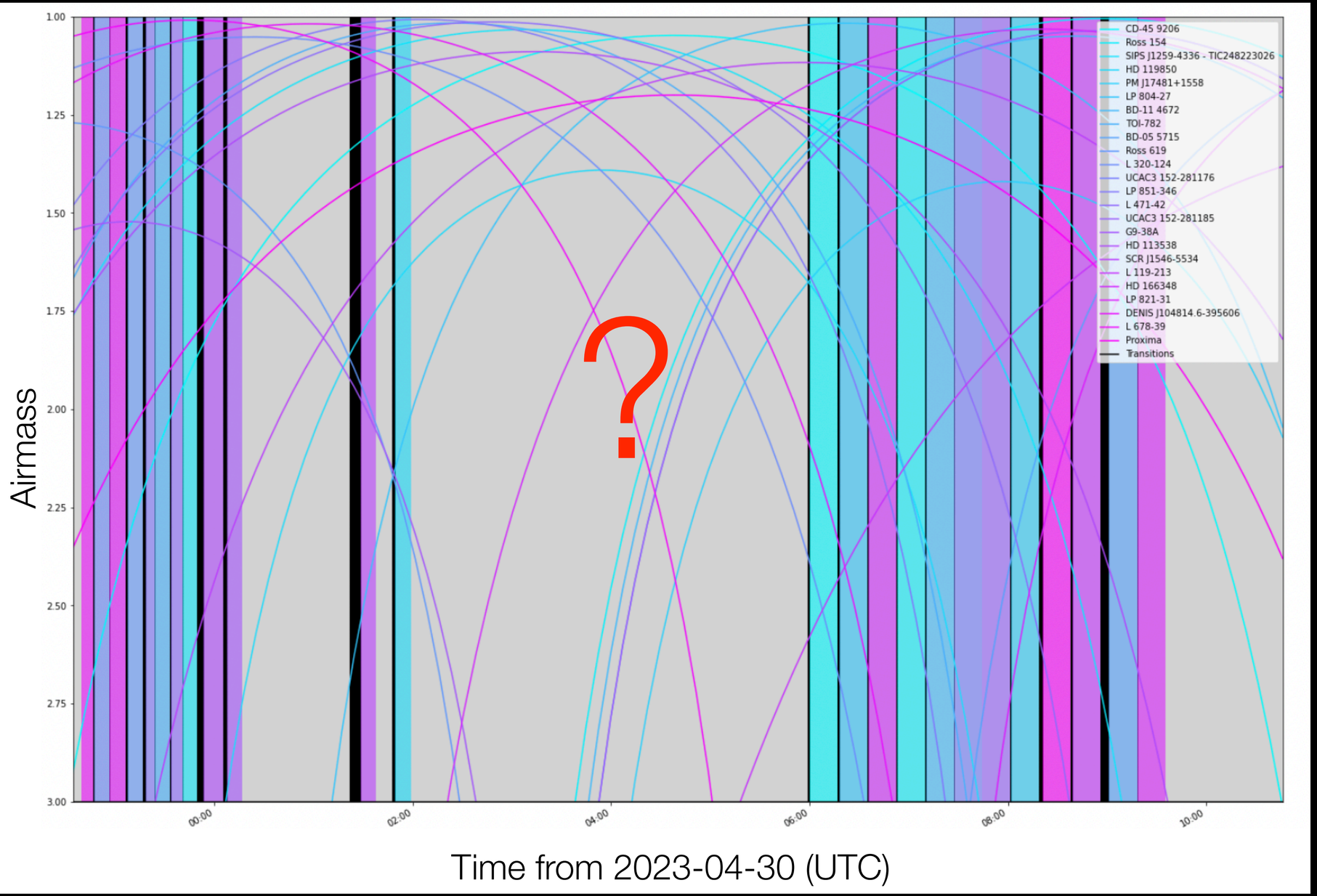
# TIME CRITICAL TARGET

➡ Urgent science objective : Transit , Eclipsing binary, etc

➡ Highest Priority

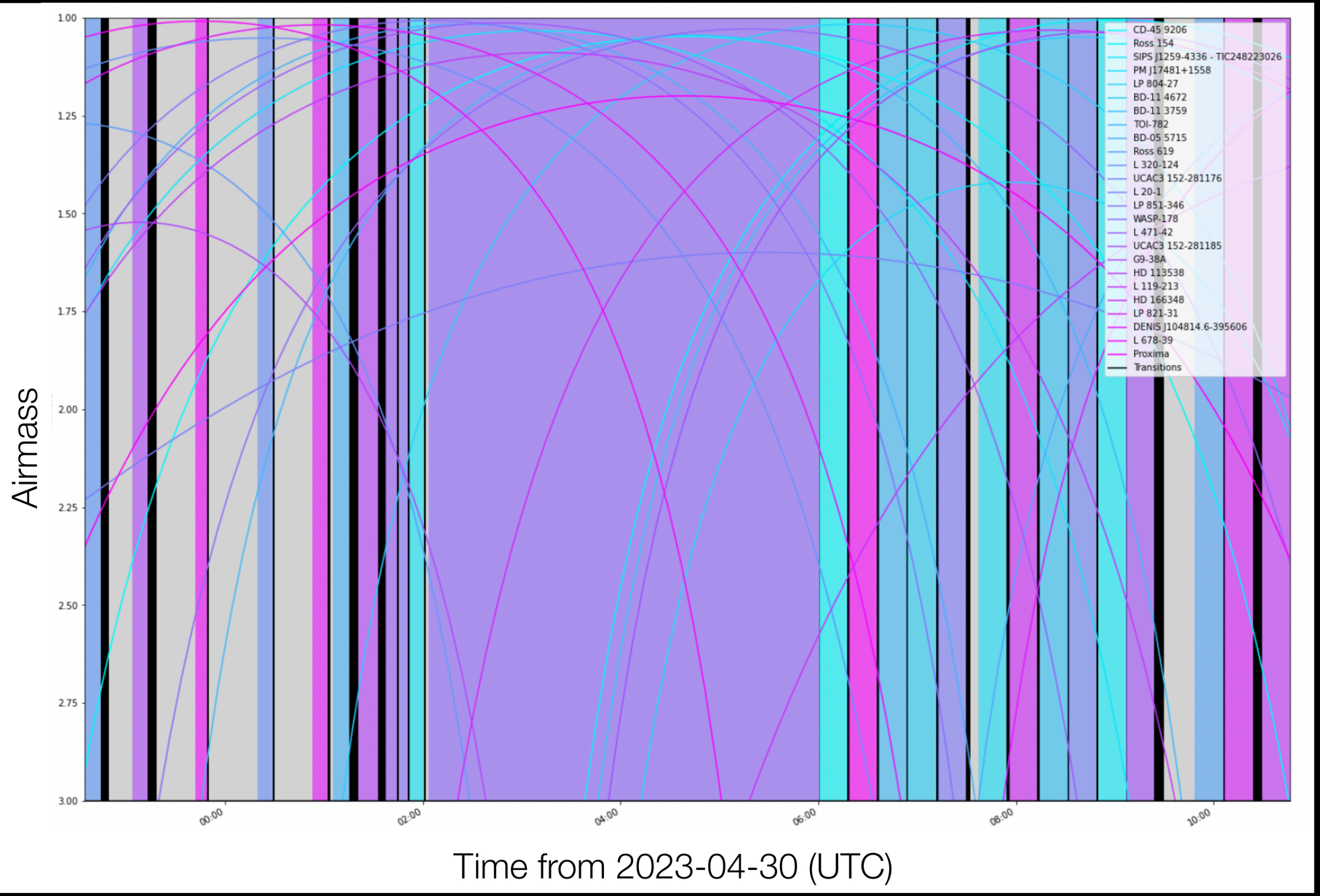
➡ Is a huge constraint , we cannot observe anything else during that period in the night (Time allotted by ESO)

# SEQUENTIAL SCHEDULE : SORTED RA LIST, >2 PRIORITIES




- No space for time critical

# PRIORITY SCHEDULE : TIME CRITICAL




- Filled the beginning and end of night
- Time critical target scheduled

# Things to keep in mind :

- The scheduling is heavily dependent on Priority, Airmass, RA
- Time critical target must be given top priority
- Rigid time constraints during block creation are essential
- The input target list must be carefully constructed ( RA, Exposure : to fill the night)

for the priority schedule to work efficiently

# NEXT STEP:

**Create a scheduling sequence with multiple observations for the given target list during a whole observing run**


# CONCLUSION

ASTROPLAN could be a beneficial tool to be able to automatise scheduling for observers