NASA Keck 2016B Call for Proposals

keckcfp@ipac.caltech.edu keckcfp@ipac.caltech.edu via gmail.c Feb 12

to bmmorris

The 2016B NASA Keck call for proposals is now available and proposals are due on March 17, 2016.

NASA is soliciting proposals to use the Keck Telescopes for the 2016B observing semester (Aug. 1, 2016 - Jan. 31, 2017)

NASA Keck 2016B Call for Proposals

keckcfp@ipac.caltech.edu keckcfp@ipac.caltech.edu via gmail.c Feb 12

to bmmorris

The 2016B NASA Keck call for proposals is now available and proposals are due on March 17, 2016.

NASA is soliciting proposals to use the Keck Telescopes for the 2016B observing semester (Aug. 1, 2016 - Jan. 31, 2017)



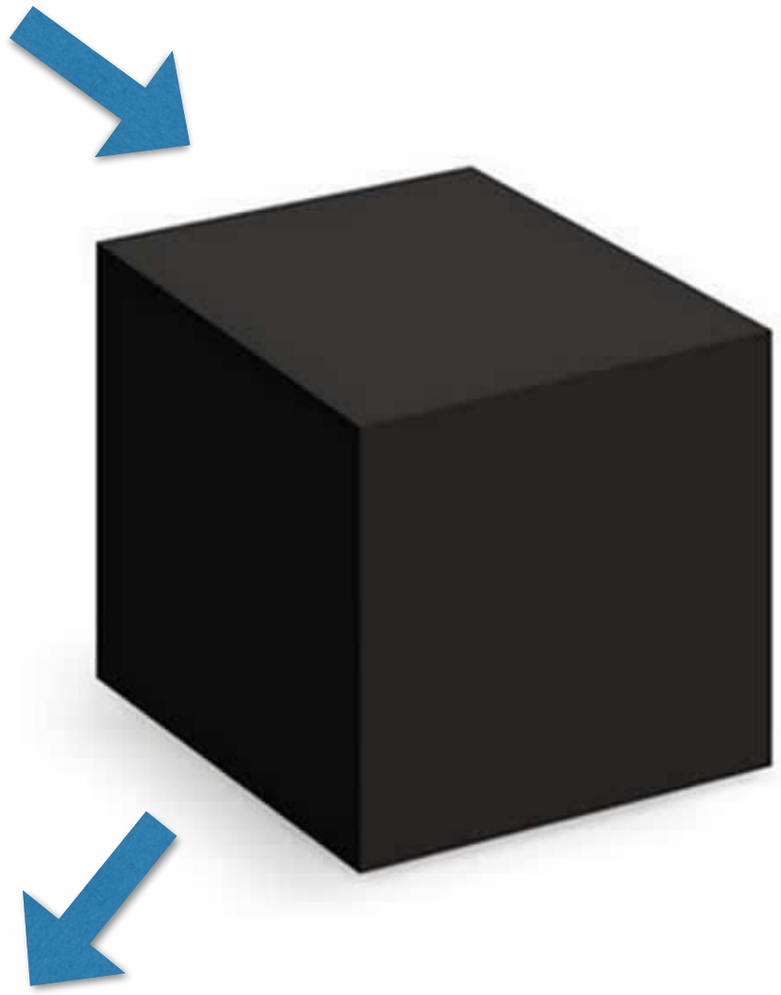
NASA Keck 2016B Call for Proposals

keckcfp@ipac.caltech.edu keckcfp@ipac.caltech.edu via gmail.c Feb 12

to bmmorris

The 2016B NASA Keck call for proposals is now available and proposals are due on March 17, 2016.

NASA is soliciting proposals to use the Keck Telescopes for the 2016B observing semester (Aug. 1, 2016 - Jan. 31, 2017)





NASA Keck 2016B Call for Proposals

keckcfp@ipac.caltech.edu keckcfp@ipac.caltech.edu via gmail.c Feb 12

to bmmorris

The 2016B NASA Keck call for proposals is now available and proposals are due on March 17, 2016.

NASA is soliciting proposals to use the Keck Telescopes for the 2016B observing semester (Aug. 1, 2016 - Jan. 31, 2017)





astroplan:
Observation planning for astronomers

Brett Morris
University of Washington

with Jazmin Berlanga Medina, Christoph Deil, Eric Jeschke, Adrian Price-Whelan, Erik Tollerud

@brettmor

@bmorris3

astroplan is

astropy-affiliated

astroplan is
Google Summer of Code project

astroplan is

in v0.1 (halp?)

Planning Observations

Well-defined problem

- Observatory database
- Rise/set times
- “Observability”

Planning Observations

Well-defined problem

- Observatory database
- Rise/set times
- “Observability”

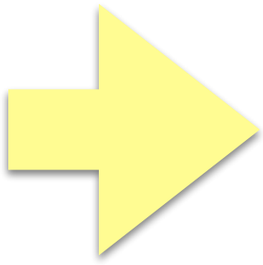
problem^n

- $N_{\text{use cases}} \approx$ infinite
- Arbitrarily complicated “constraints”

astropy -> astroplan

astropy

- coordinates.EarthLocation
- coordinates.SkyCoord
- coordinates.AltAz
- time.Time



astroplan

- FixedTarget
- Observer

Where is Vega?

…with astroplan

```python
from astroplan import Observer

# Construct an astroplan.Observer at Apache Point Observatory
apache_point = Observer.at_site("Apache Point")
```

Where is Vega?

…with astroplan

```
from astroplan import Observer

# Construct an astroplan.Observer at Apache Point Observatory
apache_point = Observer.at_site("Apache Point")

from astroplan import FixedTarget

# Construct an astroplan.FixedTarget for Vega
vega = FixedTarget.from_name("Vega")

vega_altaz = apache_point.altaz(time, vega)

vega_altaz.alt.degree, vega_altaz.az.degree
```

```
(69.47999708651103, 293.92168741017167)
```

```
In [2]: apache_point.
apache_point.altaz                          apache_point.relative_humidity
apache_point.astropy_time_to_datetime       apache_point.sun_rise_time
apache_point.at_site                        apache_point.sun_set_time
apache_point.datetime_to_astropy_time       apache_point.target_hour_angle
apache_point.is_night                       apache_point.target_is_up
apache_point.local_sidereal_time            apache_point.target_meridian_antitransit_time
apache_point.location                       apache_point.target_meridian_transit_time
apache_point.midnight                       apache_point.target_rise_time
apache_point.moon_altaz                     apache_point.target_set_time
apache_point.moon_illumination              apache_point.temperature
apache_point.moon_phase                     apache_point.timezone
apache_point.moon_rise_time                 apache_point.twilight_evening_astronomical
apache_point.moon_set_time                  apache_point.twilight_evening_civil
apache_point.name                           apache_point.twilight_evening_nautical
apache_point.noon                           apache_point.twilight_morning_astronomical
apache_point.parallactic_angle              apache_point.twilight_morning_civil
apache_point.pressure                       apache_point.twilight_morning_nautical
```

Observing "Constraints"

Target List

Observatory

Time

$30^\circ < \text{Altitude} < 85^\circ$

$20^\circ < \text{Moon Separation}$

Between civil twilights

Observable Targets

Observing "Constraints"

astroplan.FixedTarget

astroplan.Observer

astropy.time.Time

astroplan.AltitudeConstraint

astroplan.MoonSeparationConstraint

astroplan.AtNightConstraint.twilight_astronomical

Observable Targets

Observing “Constraints”

```
from astroplan import (AltitudeConstraint, MoonSeparationConstraint,
                       AtNightConstraint, observability_table)

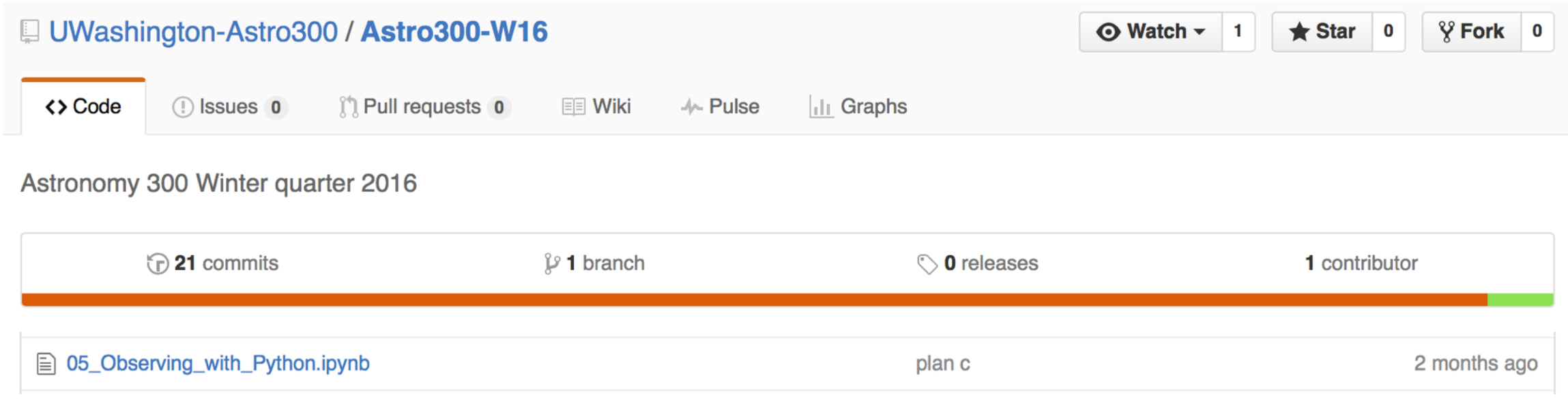
# Define constraints:
constraints = [AltitudeConstraint(min=30*u.deg, max=85*u.deg),
               MoonSeparationConstraint(min=20*u.deg),
               AtNightConstraint.twilight_civil()]

table = observability_table(constraints, subaru, targets,
                            time_range=time_range)
```

target name	ever observable	always observable	fraction of time observable
Polaris	True	True	1.0
Vega	True	True	1.0
Albireo	True	False	0.833333333333
Algol	True	False	0.166666666667
Rigel	False	False	0.0
Regulus	False	False	0.0

In the classroom

Dr. Toby Smith's ASTR300



https://github.com/UWashington-Astro300/Astro300-W16

The Manastash Ridge Observatory (MRO) is operated by the Astronomy Department of the University of Washington for the training of graduate and undergraduate students as well as for astronomical research.

In [12]:
```
mro = Observer.at_site('mro')

sunset_mro = mro.sun_set_time(now, which='nearest')

print("Sunset at MRO will be at {0.iso} local time".format(sunset_mro + offset))
```

```
Sunset at MRO will be at 2016-03-21 18:18:20.580 local time
```

In [13]:
```
(sunset_here - sunset_mro).to(u.min)
```

Out[13]: 6.4643048 min

In [14]:
```
midnight_mro = mro.midnight(now, which='next')

astrolab.local_sidereal_time(midnight_mro)
```

Out[14]: $12^{h}04^{m}53.4532^{s}$

In [15]:
```
# Astronomical twilight is when the Sun is 18 degrees below the horizon

astro_set = mro.twilight_evening_astronomical(now, which='nearest')
astro_rise = mro.twilight_morning_astronomical(now, which='next')

print("Astronomical Evening Twilight starts at {0.iso} UTC".format(astro_set))
print("Astronomical Midnight is at {0.iso} UTC".format(midnight_mro))
print("Astronomical Morning Twilight starts at {0.iso} UTC".format(astro_rise))
```

```
Astronomical Evening Twilight starts at 2016-03-22 04:06:48.077 UTC
Astronomical Midnight is at 2016-03-22 08:13:04.605 UTC
Astronomical Morning Twilight starts at 2016-03-22 12:19:14.323 UTC
```

https://github.com/UWashington-Astro300/Astro300-W16

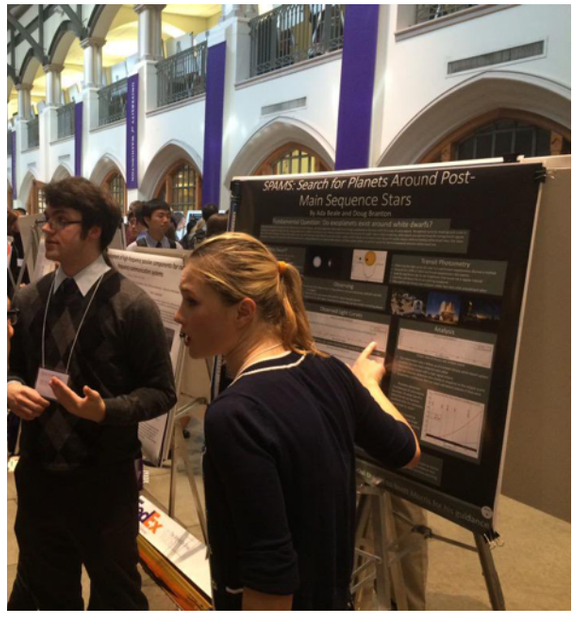
astroplan + Pre-MAP





UW Pre-MAP students:

Ada Beale
Doug Branton
Aisylnn Wallach
Andrew Yetter



- I lead **SPAMS**: Search for Planets Around post-Main sequence Stars

- Freshman undergraduates plan our observations, designing MRO's TUI

Which SDSS white dwarfs are observable for >60% of each night, mostly during A-halves?

Which SDSS white dwarfs are observable for >60% of each night, mostly during A-halves?

jupyter whitedwarfs_astroplan-sdss Last Checkpoint: a few seconds ago (unsaved changes)

File Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

New white dwarfs for SPAMS at APO

Use WDs in SDSS DR 10 from Kepler et al. 2015. Make a target list of metal-polluted white dwarfs.

In [ ]:

```
from astroplan import Observer, FixedTarget
from astroquery.vizier import Vizier
from astropy.coordinates import SkyCoord
import astropy.units as u
import numpy as np

###################################
## User inputs
catalog_name = 'J/MNRAS/446/4078/'
max_g = 18
###################################

# Get catalog of white dwarfs from Vizier
Vizier.ROW_LIMIT = -1
catalog_table = Vizier.get_catalogs([catalog_name])[0]
sc = SkyCoord(ra=catalog_table['_RAJ2000'],
              dec=catalog_table['_DEJ2000'], frame='icrs')
names = catalog_table['SDSS']
types = catalog_table['Type']
gmag = catalog_table['gmag']

# Get only the metal-polluted white dwarfs as FixedTarget objects
bright_metal_polluted = np.array([i for i, t, g in
                                  zip(range(len(types)), types.data, gmag.data)
                                  if 'z' in str(t).lower() and g < max_g])

target_list = [FixedTarget(sc[i], name="SDSS{0:s}...g={1:.2f}"
                .format(names[i].decode('ascii'), gmag[i]))
               for i in bright_metal_polluted]

print('{0} stars selected'.format(len(target_list)))
```

Future Work

Target List

Observatory

Time

$30^{o} < \text{Altitude} < 85^{o}$

$20^{o} < \text{Moon Separation}$

Between civil twilights

Observable Targets

Scheduled Observations

Future Work

Target List

Observatory

Time

$30^{o} < \text{Altitude} < 85^{o}$

$20^{o} <$ Moon Separation

Between civil twilights

Observable Targets

(Scheduled Observations)



Google Summer of Code

State of the Sprints

Stephanie Douglas

Schedule #151

Open **stephtdouglas** wants to merge 7 commits into `astropy:master` from `stephtdouglas:schedule`

Conversation 0 · Commits 7 · Files changed 4



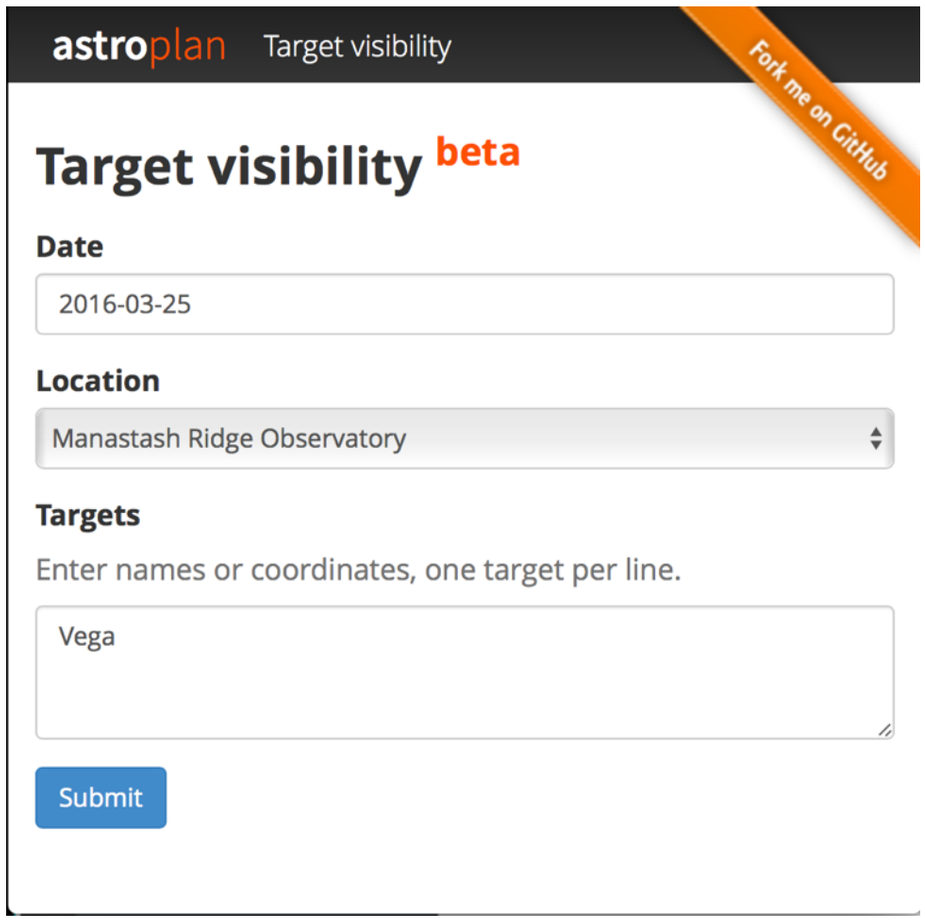
stephtdouglas commented 23 hours ago

This is an updated version of this pull request, but **@eteq** and I decided that I'd mostly take this over for now, so this transfers the responsibility to me. My branch was forked from his, so includes all of his updates from the old PR, plus mine.

I implemented a scheduler for prioritized observations, which optimizes the best schedule location for each target in order of priority.

Geert Barentsen

astroplan.herokuapp.com



State of the Sprints

Stephanie Douglas

Schedule #151

Open **stephtdouglas** wants to merge 7 commits into `astropy:master` from `stephtdouglas:schedule`

Conversation 0 · Commits 7 · Files changed 4



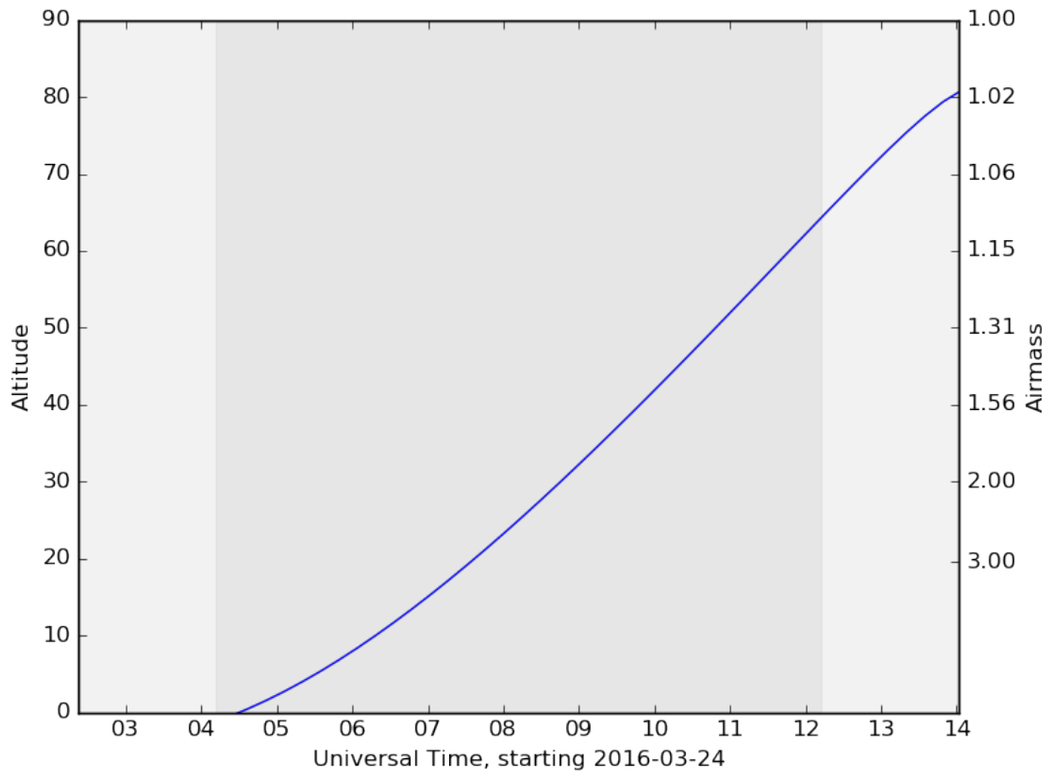
stephtdouglas commented 23 hours ago

This is an updated version of this pull request, but **@eteq** and I decided that I'd mostly take this over for now, so this transfers the responsibility to me. My branch was forked from his, so includes all of his updates from the old PR, plus mine.

I implemented a scheduler for prioritized observations, which optimizes the best schedule location for each target in order of priority.

Geert Barentsen

astroplan.herokuapp.com



astroplan

welcomes contributions!

Visit astroplan.readthedocs.org,



github.com/astropy/astroplan/

Acknowledgements

Contributors

- **Jazmin Berlanga Medina**
- Erik Tollerud
- Christoph Deil
- Adrian Price-Whelan
- Brigitta Sipocz
- Eric Jesche



