Plan for Sky Model Code

Code structure

As a starting point for the sky model, my plan is to replicate most of the calculations in the ESO sky model http://www.eso.org/observing/etc/bin/gen/form?INS.MODE=swspectr+INS.NAME=SKYCALC+SKYCALC.POSTFILE.FLAG.SET=1. For a quick start, I'll just download a grid of spectra from ESO and use those to interpolate to arbitrary conditions. When the interpolation isn't sufficient, we can upgrade that method to calculate things from first principles.

SkyModel class

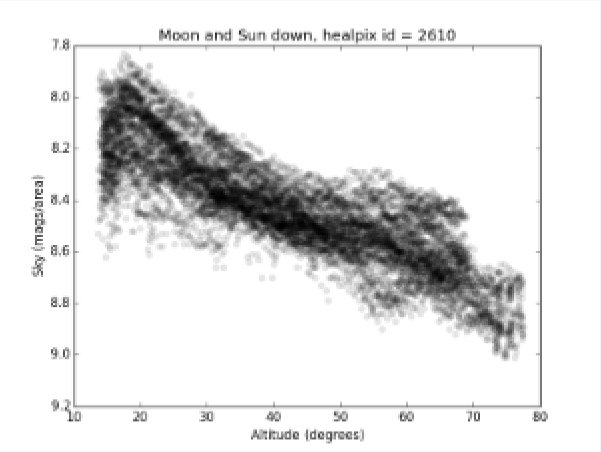
```
class SkyModel(object)
        def __init__(self, params, kwargs):
                #Set observatory parameters, filter throughput curves, which sky components to compute
                #use the kwargs to set any "magic numbers" that are needed to tune the results to the site.
                #load up template spectra
        def _loadMoonSED(self):
                #load up a grid of scattered moon light SEDs
        def interpolateMoon(self, moonPhase, moonAlt, moon2targetDist):
                #Interpolate the moon SEDs to the requested paramter values.
        #similar load/interpolate methods for the other sky components--zodiacal light, lower atmosphere, upper
atmosphere, airglow continuum, twilight,

        def setRaDecMjd(self, ra, dec, mjd, kwargs):
                set the MJD of the observations and the locations to calculate sky brightnesses at.
                additional kwargs for things like PWV, solar activity, etc.
                Calculate things like airmass for each sky point, distance to moon for each sky point
        def calcSky(self, mags=True):
                use the template spectra to interpolate a best guess for the sky SED at each RA,Dec point.
                Return SEDs or magnitudes
```

The idea here is that one can start up a SkyModel object, and then loop over a number of MJD values to calculate the expected sky brightness.

Testing the Results

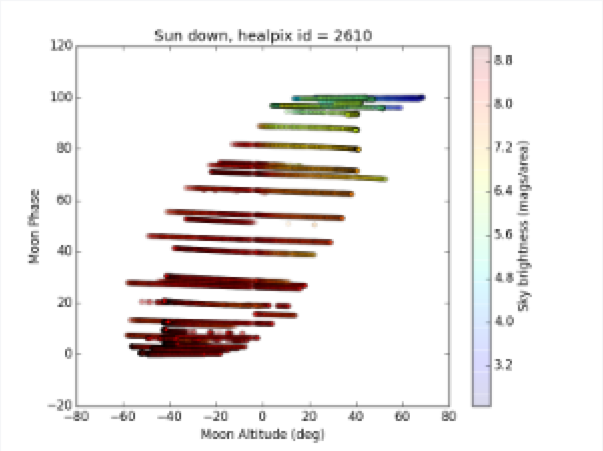
We can compare the results of code to the Canon all-sky camera. For example, looking at one patch of sky in dark time:



Note that much of the scatter in the plot is from cloudy nights.

Once the cloudy nights are clipped, we can use this (and the rest of the sky) to validate the atmospheric emission and zodiacal light models.

With the dark sky is matched, we can use the all sky data to test the scattered moonlight model:



Sky brightness for a single healpixel as a function

of moon altitude and phase

The all-sky camera has now been running for a full year, and we have several million photometry measurements, so there should be plenty of data to test the sky model.

Flux Zeropointing

We'll use Chuck's photodiode data to check the overall flux zeropointing of the spectra.

Fitting Feedback

It would be good to map out the sky model residuals and include that information back in the model so that the model can return both the expected sky brightness and the expected variation.

Possible Issues

- We will need to add a twilight sky model from scratch since there is not one in the ESO model
- The Canon data does not extend into the IR, so we will need to rely on the photodiode data (pointed at zenith), to quantify the sky brightness variation in y-band.
- Does anyone have a favourite way to interpolate template spectra? I'd obviously like to avoid python loops.