This file documents the details of the simulation.

List of functions:

a. 1D Compton-y profile

```
def battaglia_profile(r, Mvir, z)
```

Input: radius (Mpc), virial mass (m_{solar}), redshift, ect.
Output: Electron pressure profile (KeV/cm^3)
Library: colossus
Paper: https://arxiv.org/abs/1109.3711 (Battaglia, 2012)
https://arxiv.org/abs/1608.04160 (Vikram, 2016)
https://ui.adsabs.harvard.edu/abs/1986MNRAS.222..323K/abstract (Kaiser, 1986)
https://arxiv.org/abs/astro-ph/0410173 (Voit, 2005)
Steps:

1. Calculate ρ_{cri} using Battaglia Eq.1
2. Convert M_{vir} to M_{200} & R_{200} using colossus (Concentration: Ishiyama, 2021)
3. Calculate P_{200} (Kaiser, 1986; Voit, 2005)
4. Calculate p_{th} using Battaglia Eq.10
5. Calculate p_e from p_{th} assuming primordial helium mass fraction of 0.24 (Vikram, 2016)

```
def epp_to_y(profile)
```

Input: Electron pressure profile (p_e)
Output: Compton-y profile
Paper: https://arxiv.org/abs/1006.1945 (Shaw, 2010)
Steps:

1. Convert from p_e to y using Shaw Eq. 4

b. 2D Compton-y submap

```
def make_proj_image_new(radius, profile,maxRadius=0.5,pixel_scale=0.005,extrapolate=False)
```

Input: radius, profile (numpy arrays)
Output: image (default to range of 0.5 Mpc)
Steps:

1. Meshgrid and calculate distance from center
2. Assign amplitude using profile

```
def Mpc_to_arcmin(r, z)
```

Input: Distance, redshift
Output: angular scale ($arcmin$)
Library: astropy

Steps:

1. Set cosmology and use conversion from astropy

c. Convolution with gaussian kernel

```
def gaussian_kernal(pix_size,beam_size_fwhp)
```

Input: pixel_size, beam_size, submap
Output: convolved submap
Paper: https://arxiv.org/abs/1106.5065 (Aniano, 2011)
Source: CMB summer school
Steps:

1. Meshgrid and calculate distance from center
2. Construct Gaussian using Aniano Eq.13

```
def convolve_map_with_gaussian_beam(pix_size, beam_size_fwhp, Map)
```

Input: pixel_size, beam_size, submap
Output: convolved submap
Library: scipy.signal
Steps:

1. Make gaussian beam
2. Convolution using scipy (FT & IFT)

d. Conversion between Compton-y and ΔT submap

```
def f_sz(f, T_CMB)
```

Input: Electron pressure profile (p_e)
Output: Compton-y profile
Paper: https://arxiv.org/abs/2101.08373 (Vavagiakis, 2021)
Steps:

1. Calculate radiation frequency using Vavagiakis Eq.2
2. Calculate temperature decrement using Vavagiakis Eq.1

e. Image output

```
def generate_img(radius, profile, f, noise_level, beam_size, z, option, s = False, p = None):
```

Options:

1. 1D Compton-y Profile
2. 2D Compton-y Submap (unconvolved)
3. Gaussian beam
4. 2D Compton-y Submap (convolved)
5. 2D ΔT submap
6. Gaussian Noise
7. 2D ΔT submap with noise
8. 2D Compton-y Submap with noise

The simulation contains the following 3 files:

1. generator.ipynb: imports function from the script and data from the yaml file, stores the images in a subdirectory
2. simulation.py: critical functions used to construct the profile and output image
3. config.yaml: specifies the cosmology, telescope

Main inputs: Mass (unit of $10^{13} M_{solar}$), Redshift

Other parameters:

a. Cosmology (options of Battaglia, Planck 2015, Planck 2018)
b. Telescope (options of ACT_DR4, ACT_DR5, SPT)
c. Observation frequency (unit of GHz)

Output:

a. Final submap: ΔT with noise (option 7)
b. Other intermediate images (provide sanity check)

Output size: 18 $arcmin$ in diameter (also labeled with Mpc)

Pixel size: 0.5 $arcmin$ (default to Atacama Cosmology Telescope)

$M = 10^{15} M_{solar}$, $z = 0.01$, *ACT_DR4* , $90GHz$

Inputs: Distance, Virial Mass, Redshift

↓

battaglia_profile(r, Mvir, z)
epp_to_y (profile)

↓

make_proj_image_new(radius, profile)

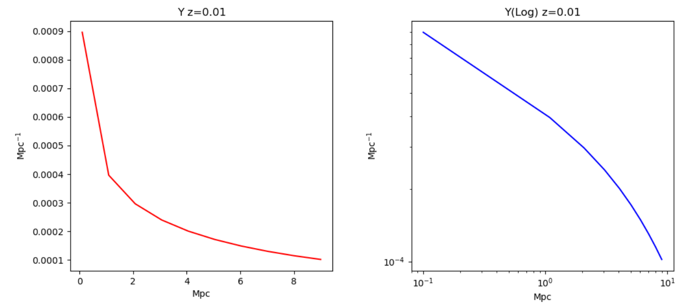
↓

convolve_map_with_gaussian_beam(pix_size, beam_size_fwhp, Map)

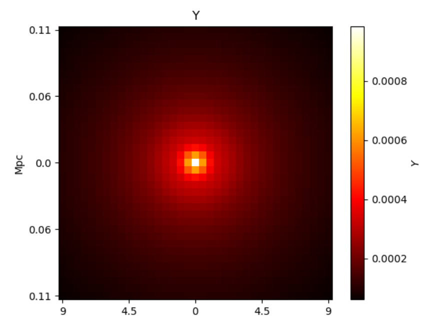
↓

f_sz(f, T_CMB)

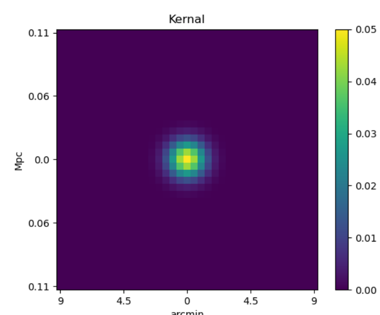
Option 1



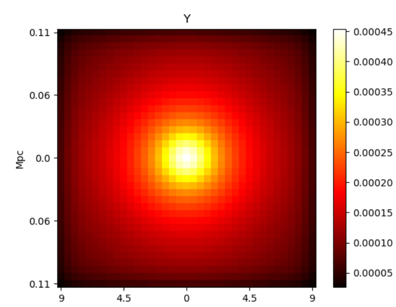
Option 2



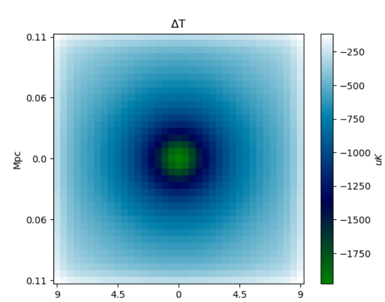
Option 3



Option 4

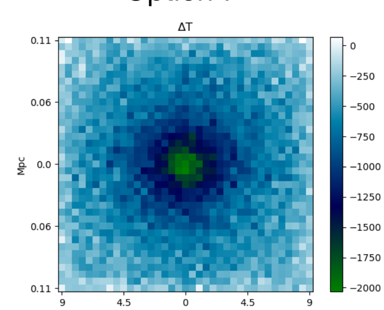


Option 5

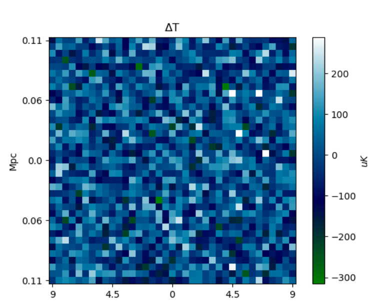


Noise →

Option 7



Option 6 (Noise)



Option 8