

# Detecting relativistic SZ using multi matched filter

Aditya Rotti

Jodrell Bank Center for Astrophysics, University of Manchester

## Abstract

The inverse Compton scattering of CMB photons by the hot electrons gas in galaxy clusters alters the black body spectrum by up-scattering photons at low frequencies ( $< 217$  GHz) to higher frequencies ( $> 217$  GHz). This effect is proportional to the Compton  $y$  parameter which characterizes the electron gas pressure along the line of sight. The electron gas pressure profile is well fit by a generalized NFW profile. We have developed a multi-matched filter for detecting galaxy cluster in microwave maps. This filter uses both the spectral as well as the spatial profile of galaxy clusters to detect signatures of clusters in multi frequency microwave maps.

## 1 Outline of the algorithm

The preprocessing steps help in avoiding redundant computations which helps with speeding up the evaluations.

- Preprocessing on Planck maps
  - We first read in the Planck maps at the 6 HFI frequencies. The maps at 545 and 857 GHz are converted from units of MJy/sr to  $K_{\text{CMB}}$
  - We read in the point source catalogue from PCCS at the HFI channels and create a point source mask for sources detected above  $10\sigma$ . The point source mask for each frequency is combined to make the final point source mask.
  - We read in the MMF3 cluster catalogue and extract information on clusters used in the Planck cluster cosmology analysis. For clusters in the cosmology sample, we extract a tangent plane from each of the six HFI Planck maps centered on the sky coordinate of the clusters and the corresponding point source mask. These 7 maps are written to a fits file which will be used in the MMF analysis. For the clusters we also evaluate  $T_{500}$  and  $\theta_{500}$  using standard relations.

- Preprocessing for MMF evaluations
  - We evaluate the Planck band passed SZ spectrum at clusters temperatures ranging from 0 – 15 keV and store them in a python dictionary - a SZ spectrum bank.
  - We also precompute the spatial clusters templates by performing line on sight integrals on the 3D pressure profile for which we use the GNFW profile. We evaluate these templates with the cluster core radius varying in the range:  $\theta_c = 0 - 54$  arcminutes. We read in these templates and precompute the FFT of each of these templates and store these computations in a python dictionary - a spatial template FT bank.
  - We also precompute the smoothing filters for each of the Planck channels which are constructed using the beam FWHM for each channels and the pixel window corrections. These filters at each of the Planck channels are multiplied with the cluster template FT which is finally used in the evaluation of the MMF.
- Evaluation of the MMF
  - We initiate an MMF object in which we inform the filter about the various characteristics of the data like the patch size, the resolution, the instrument smoothing kernels etc.
  - We read in the data slices one of the clusters (here we can choose if we want to mask the data slices for the point sources), compute the FFT on the data slices and evaluate the auto and cross channel spectra to evaluate the covariance matrix and its inverse. These are now stored as attributes of the MMF object. For each new data slice the attributes get updated.
  - The FFT of the data, its inverse covariance, and the FFT of the template along with the instrument filters are combined to evaluate the MMF filter to return a filtered data slice which is a measurement of  $y_c$  and a semi-analytical estimate of the error on  $y_c$ .
    - \* At this stage we can choose the spatio-spectral template bank grid over which we want to run the filter and tabulate the findings.
    - \* For fields which have multiple clusters or a cluster detected at a very high significance, the error estimates are skewed and consequently also the estimates of detection significance. One strategy (adopted currently) to counter this is to mask the filtered field using some SNR thresholding to re-evaluate the standard deviation of the

background fluctuations and make a revised estimate of the cluster detection SNR. An alternate strategy would be to subtract from the data the cluster signal characterized by measured  $(y_c, \theta_c)$  and rerun the matched filter to get a revised estimate of the SNR. *[This alternate strategy is not yet implemented]*