Map-based component separation survey

Edgar Jaber

QUBIC - Data analysis Teleconf

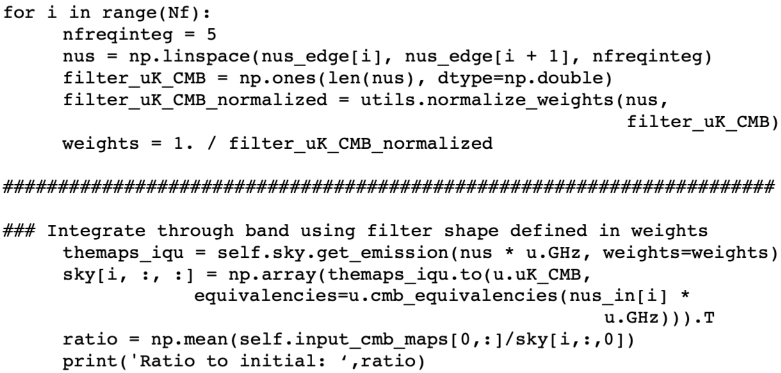
3/12/2020

Introduction

- Working on parametric based component separation with the module FGBuster developed by Josquin Errard (from APC) et. al.
- Prior to this, I was confronted with the problem of working with QubicSkySim simulated maps
- The problems have been partially resolved however the results are not yet perfectly satisfying.

Bandpass in QSS

Before:



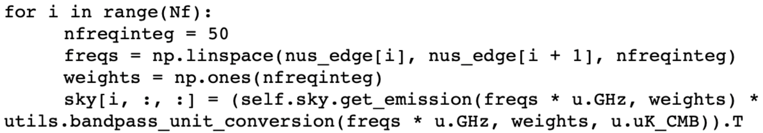
```
for i in range(Nf):
      nfreqinteg = 5
      nus = np.linspace(nus_edge[i], nus_edge[i + 1], nfreqinteg)
      filter_uK_CMB = np.ones(len(nus), dtype=np.double)
      filter_uK_CMB_normalized = utils.normalize_weights(nus,
                                                         filter_uK_CMB)
      weights = 1. / filter_uK_CMB_normalized

######################################################################

### Integrate through band using filter shape defined in weights
      themaps_iqu = self.sky.get_emission(nus * u.GHz, weights=weights)
      sky[i, :, :] = np.array(themaps_iqu.to(u.uK_CMB,
                  equivalencies=u.cmb_equivalencies(nus_in[i] *
                                                     u.GHz))).T
      ratio = np.mean(self.input_cmb_maps[0,:]/sky[i,:,0])
      print('Ratio to initial: ‘,ratio)
```

With 3 bands, Ratio to initial : ~ 1.05

After the correction ideas provided by Josquin



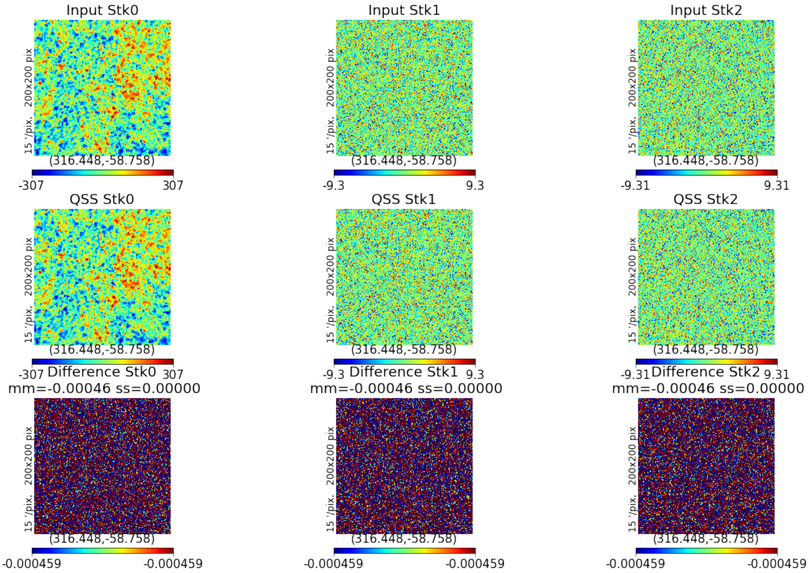
```
for i in range(Nf):
     nfreqinteg = 50
     freqs = np.linspace(nus_edge[i], nus_edge[i + 1], nfreqinteg)
     weights = np.ones(nfreqinteg)
     sky[i, :, :] = (self.sky.get_emission(freqs * u.GHz, weights) *
utils.bandpass_unit_conversion(freqs * u.GHz, weights, u.uK_CMB)).T
```

With 3 bands, Ratio to initial: ~ 0.9995

Explanations

- We suppose the filter is flat in Jy/sr, then in the get_emission() procedure it is converted to μK_{RJ} and we integrate on more subfrequencies.
- Finally the conversion to μK_{CMB} must be done with the utils.bandpass_unit_conversion() to take in account the usage of bandpass weights.

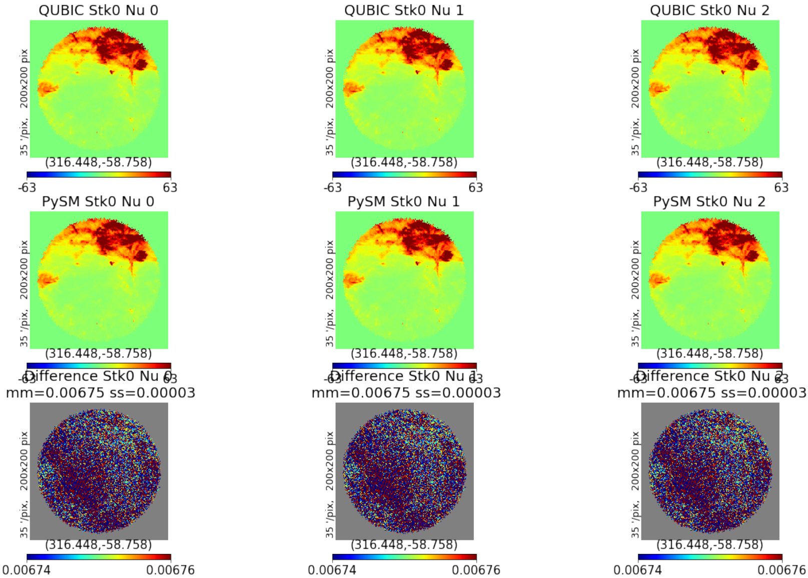
Maps


Input Stk0
15 '/pix, 200x200 pix
(316.448,-58.758)
-307
307
Input Stk1
15 '/pix, 200x200 pix
(316.448,-58.758)
-9.3
9.3
Input Stk2
15 '/pix, 200x200 pix
(316.448,-58.758)
-9.31
9.31
QSS Stk0
15 '/pix, 200x200 pix
(316.448,-58.758)
-307
307
QSS Stk1
15 '/pix, 200x200 pix
(316.448,-58.758)
-9.3
9.3
QSS Stk2
15 '/pix, 200x200 pix
(316.448,-58.758)
-9.31
9.31
Difference Stk0
mm=-0.00046 ss=0.00000
15 '/pix, 200x200 pix
(316.448,-58.758)
-0.000459
-0.000459
Difference Stk1
mm=-0.00046 ss=0.00000
15 '/pix, 200x200 pix
(316.448,-58.758)
-0.000459
-0.000459
Difference Stk2
mm=-0.00046 ss=0.00000
15 '/pix, 200x200 pix
(316.448,-58.758)
-0.000459
-0.000459

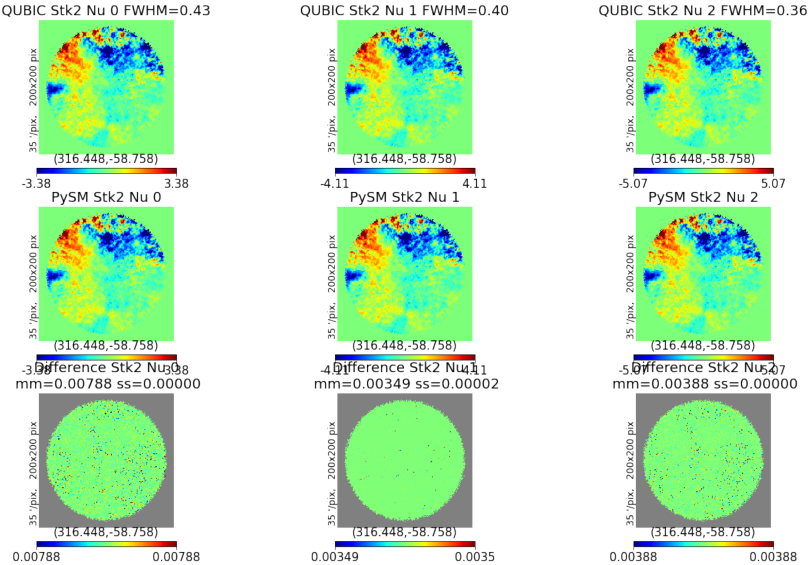
Back to FGBuster

- We have a procedure to simulate PySM maps, however this one does not integrate yet the bandpass, this has yet to come and understand
- To perform the CS, maps must be put at the same resolution, that of the lowest band
- It doesn't work with CMB yet, we don't get the expected β_{dust}
- But there are other aspects that I have not yet explored

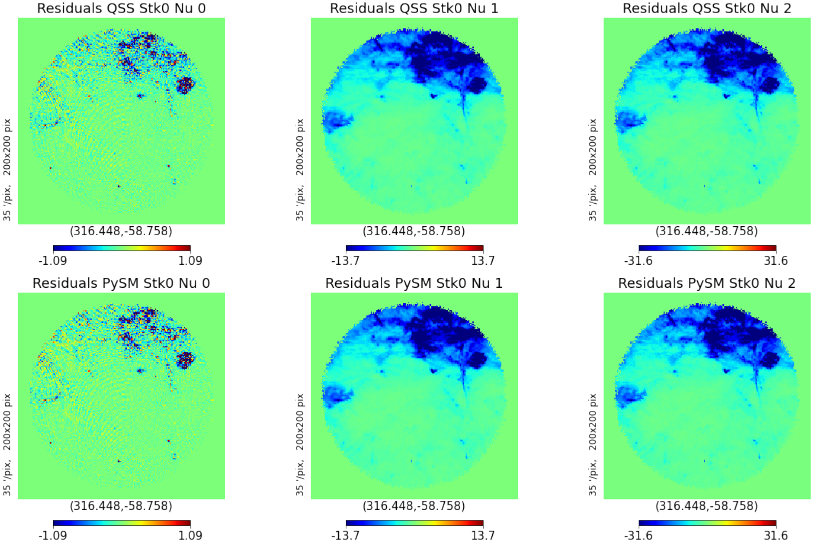
QSS / PySM


QUBIC Stk0 Nu 0
QUBIC Stk0 Nu 1
QUBIC Stk0 Nu 2
35 '/pix, 200x200 pix
(316.448,-58.758)
-63
63
PySM Stk0 Nu 0
PySM Stk0 Nu 1
PySM Stk0 Nu 2
Difference Stk0 Nu 0
Difference Stk0 Nu 1
Difference Stk0 Nu 2
mm=0.00675 ss=0.00003
0.00674
0.00676

Component separation



Residuals


Residuals QSS Stk0 Nu 0
35 '/pix, 200x200 pix
(316.448,-58.758)
-1.09
1.09
Residuals QSS Stk0 Nu 1
35 '/pix, 200x200 pix
(316.448,-58.758)
-13.7
13.7
Residuals QSS Stk0 Nu 2
35 '/pix, 200x200 pix
(316.448,-58.758)
-31.6
31.6
Residuals PySM Stk0 Nu 0
35 '/pix, 200x200 pix
(316.448,-58.758)
-1.09
1.09
Residuals PySM Stk0 Nu 1
35 '/pix, 200x200 pix
(316.448,-58.758)
-13.7
13.7
Residuals PySM Stk0 Nu 2
35 '/pix, 200x200 pix
(316.448,-58.758)
-31.6
31.6