Map-based component separation survey

Edgar Jaber

QUBIC - Data analysis Teleconf

21/10/2020

Parametric component separation

Broadly speaking, it consists of finding spectral parameters of the foreground components with a pixel-by-pixel optimization procedure.
(ref. Stompor et. al 2009)

We model the data as

$$\mathbf{d}_p = \mathbf{A}_p \mathbf{s}_p + \mathbf{n}_p$$

where $\mathbf{A}_p = \mathbf{A}_p(\beta)$ is the mixing matrix of the form number of freqs $\times$ number of components.
A log-likelihood can be written as:

$$-2 \log \mathcal{L}_{\text{data}}(\mathbf{s}, \beta) = \text{const} + (\mathbf{d} - \mathbf{A}\mathbf{s})^t \mathbf{N}^{-1} (\mathbf{d} - \mathbf{A}\mathbf{s})$$

The log-likelihood reaches a maximum for the values of $(\mathbf{s}, \beta)$ fulfilling the relations:

$$-(\partial_\beta(\mathbf{A})\mathbf{s})^t \mathbf{N}^{-1}(\mathbf{d} - \mathbf{A}\mathbf{s}) = 0 \tag{1}$$

$$\mathbf{s} = (\mathbf{A}^t \mathbf{N}^{-1} \mathbf{A})^{-1} \mathbf{A}^t \mathbf{N}^{-1} \mathbf{d} \tag{2}$$

We can constrain the spectral parameters by eliminating the sky signals from the full log-likelihood using (1):

$$-2 \log \mathcal{L}_{\text{spec}}(\beta) = \text{const} - (\mathbf{A}^t \mathbf{N}^{-1} \mathbf{d})^t (\mathbf{A}^t \mathbf{N}^{-1} \mathbf{A})^{-1} (\mathbf{A}^t \mathbf{N}^{-1} \mathbf{d}) \tag{3}$$

using it, we can find the maximum value of the likelihood attained for any chosen value of the β.
Finally we uncover the signals s using (2).

FGBuster

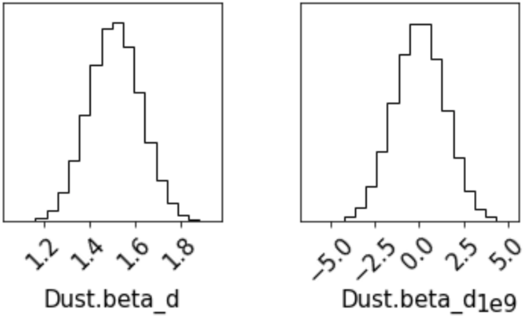
ref: API

- Developed by Josquin Errard (from APC) et. al
- Is a Python module that includes a procedure with the parametric component separation method described above. Also has ILC maybe for future studies.
- Works fine with partial maps
- Has a PySM toolbox providing most functionalities

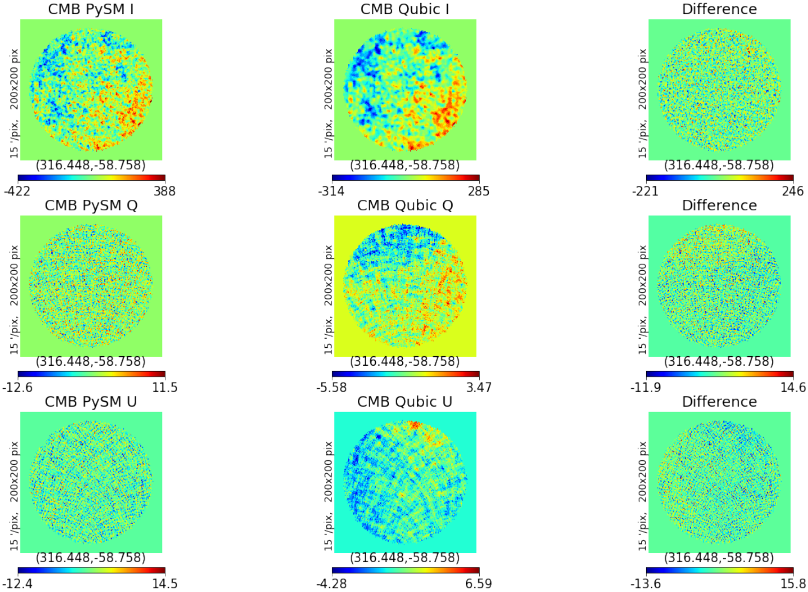
Results

Results for the following

- 3 bands at 150GHz
- skyconfig $= \{$'dust':'d0', 'CMB':'c1'$\}$
- qss.get_partial_sky_maps_withnoise(spatial_noise $=$ *False*, nunu_correlation $=$ *False*)

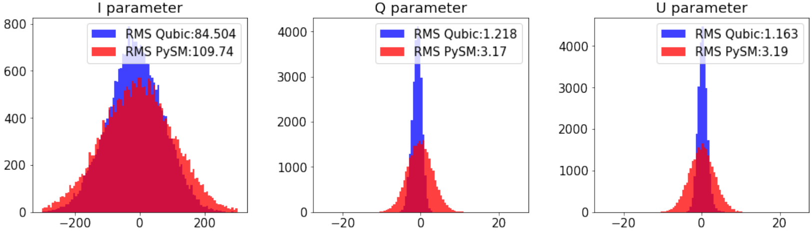


Results



CMB PySM I
15 '/pix, 200x200 pix
(316.448,-58.758)
-422
388
CMB Qubic I
15 '/pix, 200x200 pix
(316.448,-58.758)
-314
285
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-221
246
CMB PySM Q
15 '/pix, 200x200 pix
(316.448,-58.758)
-12.6
11.5
CMB Qubic Q
15 '/pix, 200x200 pix
(316.448,-58.758)
-5.58
3.47
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-11.9
14.6
CMB PySM U
15 '/pix, 200x200 pix
(316.448,-58.758)
-12.4
14.5
CMB Qubic U
15 '/pix, 200x200 pix
(316.448,-58.758)
-4.28
6.59
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-13.6
15.8

Results

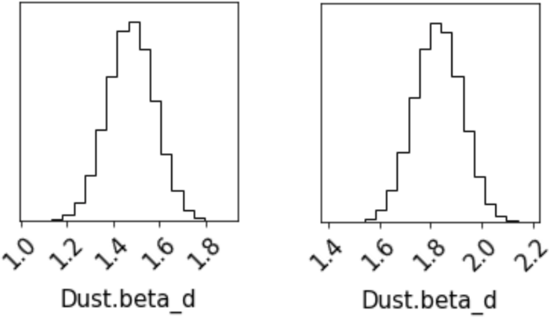

I parameter
RMS Qubic:84.504
RMS PySM:109.74
800
600
400
200
0
−200
0
200
Q parameter
RMS Qubic:1.218
RMS PySM:3.17
4000
3000
2000
1000
0
−20
0
20
U parameter
RMS Qubic:1.163
RMS PySM:3.19
4000
3000
2000
1000
0
−20
0
20

- $\beta = 1.5$ for PySM maps which is coherent
- $\beta = 0.437$ for QSS maps, which is problematic, error is huge ($> 1e9$) as well
- The difference of the CMB maps is non zero

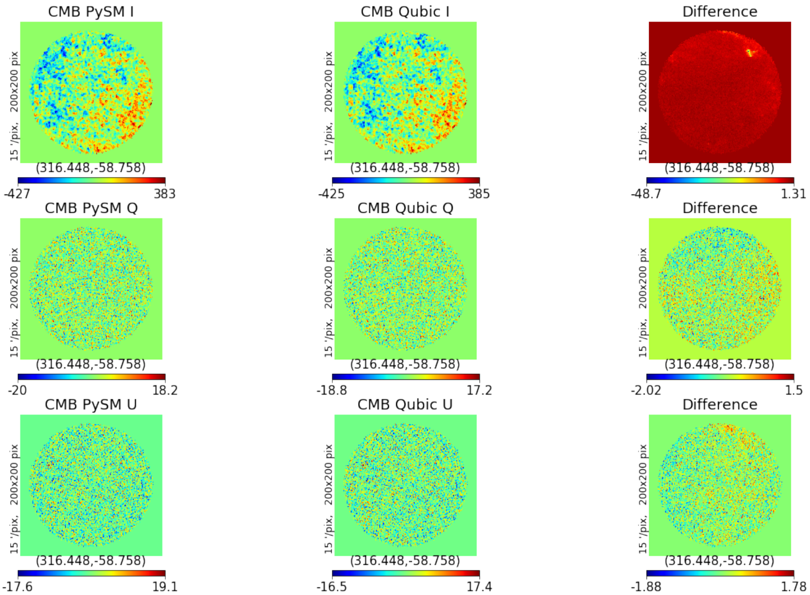
The problem should be related to the noise structure of the Qubic simulated maps...

Results

- 3 bands at 150GHz
- skyconfig = {'dust':'d0', 'CMB':'c1'}
- qss.get_partial_sky_maps_withnoise(spatial_noise = *False*, nunu_correlation = *False*)
- White noise and no Qubic noise

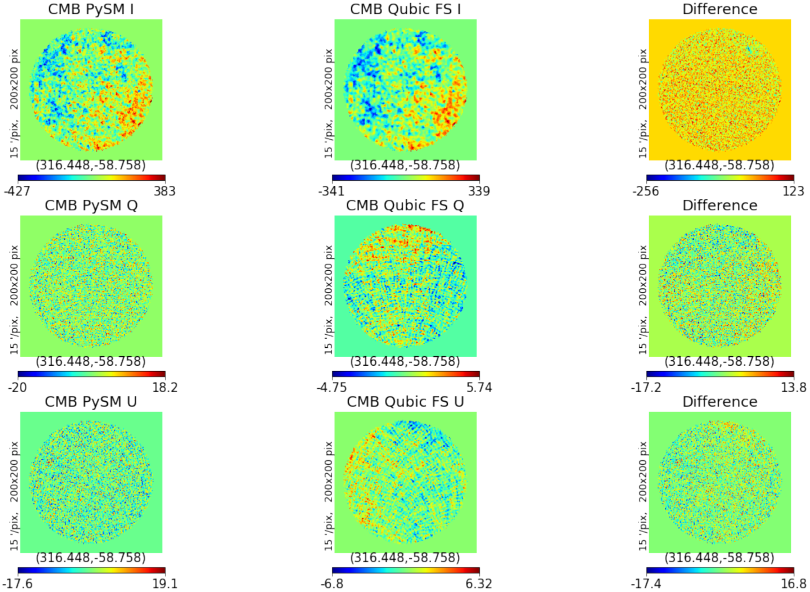


Results

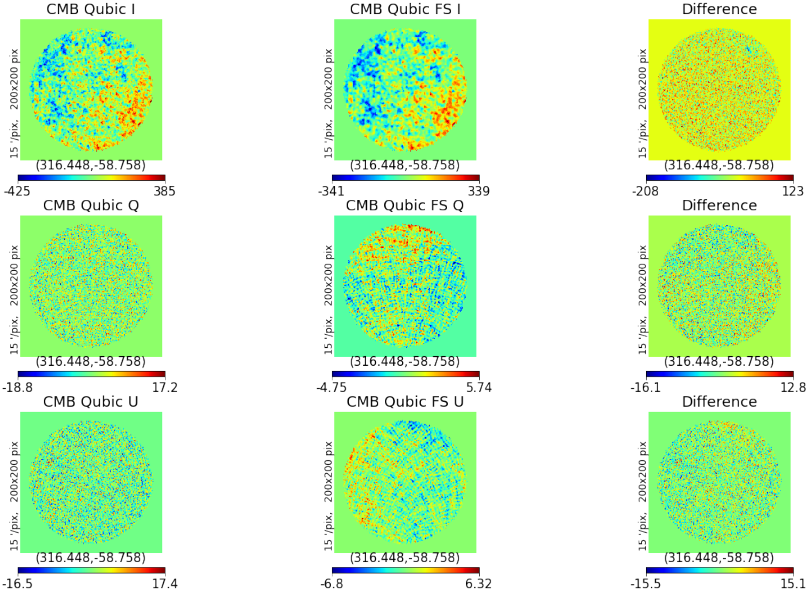


CMB PySM I
CMB Qubic I
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-427
383
-425
385
-48.7
1.31
CMB PySM Q
CMB Qubic Q
Difference
-20
18.2
-18.8
17.2
-2.02
1.5
CMB PySM U
CMB Qubic U
Difference
-17.6
19.1
-16.5
17.4
-1.88
1.78

Results


CMB PySM I
15 '/pix, 200x200 pix
(316.448,-58.758)
-427
383
CMB Qubic FS I
15 '/pix, 200x200 pix
(316.448,-58.758)
-341
339
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-256
123
CMB PySM Q
15 '/pix, 200x200 pix
(316.448,-58.758)
-20
18.2
CMB Qubic FS Q
15 '/pix, 200x200 pix
(316.448,-58.758)
-4.75
5.74
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-17.2
13.8
CMB PySM U
15 '/pix, 200x200 pix
(316.448,-58.758)
-17.6
19.1
CMB Qubic FS U
15 '/pix, 200x200 pix
(316.448,-58.758)
-6.8
6.32
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-17.4
16.8

Results



CMB Qubic I
15 '/pix, 200x200 pix
(316.448,-58.758)
-425
385
CMB Qubic FS I
15 '/pix, 200x200 pix
(316.448,-58.758)
-341
339
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-208
123
CMB Qubic Q
15 '/pix, 200x200 pix
(316.448,-58.758)
-18.8
17.2
CMB Qubic FS Q
15 '/pix, 200x200 pix
(316.448,-58.758)
-4.75
5.74
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-16.1
12.8
CMB Qubic U
15 '/pix, 200x200 pix
(316.448,-58.758)
-16.5
17.4
CMB Qubic FS U
15 '/pix, 200x200 pix
(316.448,-58.758)
-6.8
6.32
Difference
15 '/pix, 200x200 pix
(316.448,-58.758)
-15.5
15.1

- $\beta = 1.476$ for PySM maps which is coherent
- $\beta = 1.827$ for QSS maps, which is okay
- Some statistics with the white noise realisations are in the making
- The difference of the CMB maps is non zero and the difference between the FS maps and the results of FGBuster.
- Maybe apply the procedure to the full pipeline and not to the sky simulator.