s4cmb
Systematics For Cosmic Microwave Background

Julien Peloton

Department of Physics & Astronomy
University of Sussex, UK

November 2017
https://github.com/JulienPeloton/s4cmb

Motivations

- As instrument noise goes down, we start to probe instrument systematic dominated regimes.
- This is not the problem of a single CMB experiment, but all experiments have to face the same issue!
- We want to have a **fast** and **robust** way of estimating a large number of effects for current and future CMB experiments.
- `s4cmb` is a library which provides various tools to
  - Simulate your own CMB instrument.
  - Simulate a scanning strategy.
  - Scan sky maps to produce Time-Ordered Data (map2tod).
  - Project back the time data into sky maps (tod2map).
  - Simulate and inject various instrument effects at any of the previous steps.

- Goals of the bootcamp:
  - Learn how to use `s4cmb`.
  - Then, learn how to implement and study your own instrument systematic effects.
- The bootcamp is organised in 2 parts:
  - Part 1: installation + presentation of the code + first App.
  - Part 2: Real-life cases.
- Material: README, notebooks, plus python scripts to be executed either on a laptop or at NERSC.
- The langage used is `Python`, and the parallelisation is done using `MPI` (optional, but recommended).
- The platform used is GitHub: use the bug tracker and pull requests if you have troubles or if you want to contribute to the project!

Part 1

- Depending on your `Python` skills, this can be quite quick to do.
- Once you managed to install the library and `jupyter`, Part 1 is organised the following way:
  - Lecture 01: presentation of the library.
  - Lecture 02: how to generate an instrument.
  - Lecture 03: how to generate a scanning strategy.
  - Lecture 04: how to generate TOD.
  - Lecture 05: An example of instrument systematic (crosstalk).
- In addition, you will find an end-to-end example that can be ran on a laptop.
- I encourage you to run yourself the notebooks, and play with parameters! Also, read comments in the notebooks and README, they contain a lot of informations (and let me know if it is unclear!).

Part 2

- Here starts the real mess!
- Part 2 starts with 4 notebooks:
  - Simulation of gain drifts (deep patch).
  - Simulation of gain drifts (shallow patch).
  - Simulation of differential pointing.
  - On the use of HWP demodulation.
- In addition, there are ready-to-use examples to use `s4cmb` on a cluster/supercomputer (`MPI` version). These examples use SO-like configuration (order of magnitude).

Before you leave

- Feedback: use the issue tracker if you have troubles or comments/doc are not clear. Use pull requests if you want to contribute to the project!
- Do not hesitate to dig into the code!

have fun!