

---

# **plancklens Documentation**

***Release 0***

**Julien Carron**

**Jun 26, 2023**



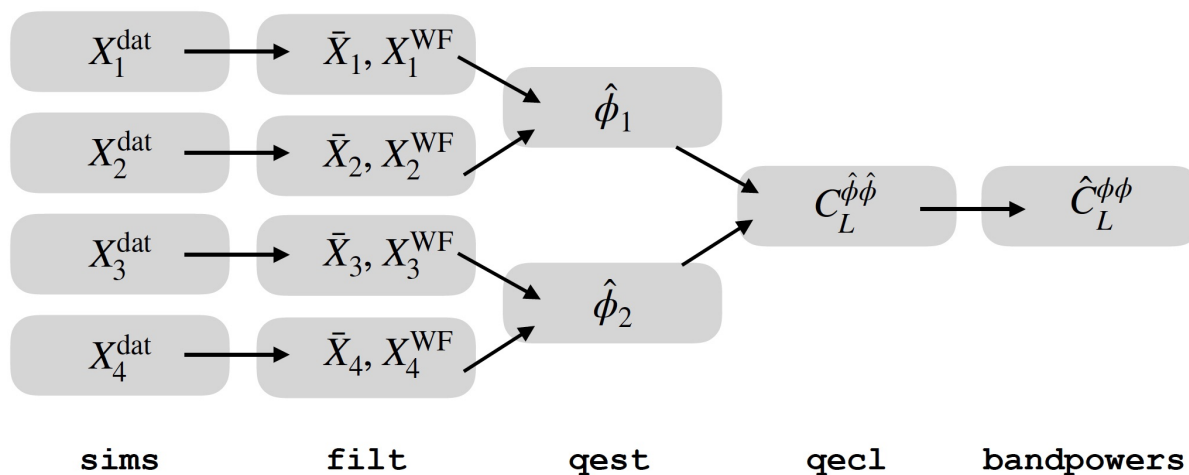
---

## Contents:

---

<b>1</b>	<b>plancklens.sims</b>	<b>3</b>
1.1	plancklens.sims.maps . . . . .	3
1.2	plancklens.sims.planck2018_sims . . . . .	3
<b>2</b>	<b>plancklens.filt</b>	<b>7</b>
2.1	plancklens.filt.filt_simple . . . . .	7
2.2	plancklens.filt.filt_cinv . . . . .	7
2.3	plancklens.filt_util . . . . .	7
<b>3</b>	<b>plancklens.qest</b>	<b>9</b>
<b>4</b>	<b>plancklens.qecl</b>	<b>11</b>
<b>5</b>	<b>plancklens.bandpowers</b>	<b>13</b>
<b>6</b>	<b>plancklens.n0s</b>	<b>15</b>
<b>7</b>	<b>plancklens.n1.n1</b>	<b>17</b>
<b>8</b>	<b>plancklens.qresp</b>	<b>19</b>
<b>9</b>	<b>plancklens.patchy</b>	<b>21</b>
<b>10</b>	<b>Indices and tables</b>	<b>23</b>
	<b>Python Module Index</b>	<b>25</b>
	<b>Index</b>	<b>27</b>







This package defines simulation libraries. A simulation library basically only defines *get\_sim\_tmap* and *get\_sim\_pmap* methods.

### 1.1 plancklens.sims.maps

### 1.2 plancklens.sims.planck2018\_sims

Planck 2018 release simulation libraries.

---

**Note:** These simulations are located on NERSC systems.

---

---

**Note:** Units of the maps stored at NERSC are  $K$  but this module returns maps in  $\mu K$

---

**class** `plancklens.sims.planck2018_sims.cmb_len_ffp10`

FFP10 input sim libraries, lensed alms.

The lensing deflections contain the L=1 aberration term (constant across all maps) due to our motion w.r.t. the CMB frame.

**static** `get_sim_blm(idx)`

**Parameters** `idx` – simulation index

**Returns** lensed B-polarization simulation healpy alm array

**static** `get_sim_elm(idx)`

**Parameters** `idx` – simulation index

**Returns** lensed E-polarization simulation healpy alm array

**static** `get_sim_tlm(idx)`

**Parameters** `idx` – simulation index

**Returns** lensed temperature simulation healpy alm array

**class** `plancklens.sims.planck2018_sims.cmb_unl_ffp10`

FFP10 input sim libraries, unlensed alms.

**static** `get_sim_blm(idx)`

**Parameters** `idx` – simulation index

**Returns** unlensed B-polarization simulation healpy alm array

**static** `get_sim_elm(idx)`

**Parameters** `idx` – simulation index

**Returns** unlensed E-polarization simulation healpy alm array

**static** `get_sim_plm(idx)`

**Parameters** `idx` – simulation index

**Returns** lensing potential  $\phi_{LM}$  simulation healpy alm array

**static** `get_sim_tlm(idx)`

**Parameters** `idx` – simulation index

**Returns** unlensed temperature simulation healpy alm array

**class** `plancklens.sims.planck2018_sims.ffp10cmb_widnoise` (*transf*, *nlevt*, *nlevp*,  
*pix\_libphas*, *nside=2048*)

Simulation library with freq-0 FFP10 lensed CMB together with idealized, homogeneous noise.

**Parameters**

- **transf** – transfer function (beam and pixel window)
- **nlevt** – temperature noise level in  $\mu K$ -arcmin.
- **nlevp** – polarization noise level in  $\mu K$ -arcmin.
- **pix\_libphas** – random phases simulation library (see `plancklens.sims.phas.py`) of the noise maps.

**class** `plancklens.sims.planck2018_sims.smica_dx12`

SMICA 2018 release simulation and data library at NERSC in uK.

---

**Note:** This now converts all maps to double precision (healpy 1.15 changed `read_map` default type behavior, breaking in a way that is not very clear as yet the behavior of the conjugate gradient inversion chain)

---

**get\_sim\_pmap** (*idx*)

Returns dx12 SMICA polarization map for a simulation

**Parameters** `idx` – simulation index

**Returns** SMICA Q and U simulation *idx*, including noise. Returns dx12 SMICA data maps for *idx* = -1

**get\_sim\_tmap** (*idx*)

Returns dx12 SMICA temperature map for a simulation

**Parameters** `idx` – simulation index



**Returns** SMICA simulation *idx*, including noise. Returns dx12 SMICA data map for *idx* = -1

**class** `plancklens.sims.planck2018_sims.smica_dx12_SZdeproj`  
tSZ-deprojected SMICA 2018 release simulation and data library at NERSC in uK

---

**Note:** This now converts all maps to double precision (healpy 1.15 changed read\_map default type behavior, breaking in a way that is not very clear as yet the behavior of the conjugate gradient inversion chain)

---

**get\_dat\_tmap**()  
Returns dx12 tSZ-deproj SMICA Planck data temperature map

**get\_sim\_tmap**(*idx*)  
Returns dx12 tSZ-deproj SMICA temperature map for a simulation

**Parameters** *idx* – simulation index

**Returns** SMICA simulation *idx*, including noise. Returns dx12 SMICA data map for *idx* = -1



This defines modules used for Wiener-filtering of CMB maps.

### 2.1 plancklens.filt.filt\_simple

### 2.2 plancklens.filt.filt\_cinv

### 2.3 plancklens.filt\_util

CMB filtering utilities module.

This module collects some convenience wrapper libraries.

**class** `filt_util.library_ftl` (*ivfs, lmax, lfilt\_t, lfilt\_e, lfilt\_b*)

Library of a-posteriori re-scaled filtered CMB maps, for separate temperature and polarization filtering

#### Parameters

- **ivfs** – inverse filtering library instance (e.g. one of those in `plancklens.filt.filt_simple`).
- **lmax** (*int*) – defines the new healpy alm array shape (identical for temperature and polarization)
- **lfilt\_t** (*1d array*) – filtered temperature alms are rescaled by `lfilt_t`
- **lfilt\_e** (*1d array*) – filtered E-polarization alms are rescaled by `lfilt_e`
- **lfilt\_b** (*1d array*) – filtered B-polarization alms are rescaled by `lfilt_b`

Wraps the input filtering instance (*ivfs*) methods to keep the same interface.

---

**Note:** `ftl` `fel` `fbl` should eventually be taken off to be replaced by `fal` in all cases

---

**class** `filt_util.library_shuffle` (*ivfs*, *idxs*)

A library of filtered sims with remapped indices.

This is useful for lensing biases calculations, such as  $\hat{N}_L^{(0)}$ .

**Parameters**

- **ivfs** – inverse-variance filtering library instance.
- **idxs** – index `idx` of this new instance points to `idxs[idx]` of the input *ivfs* instance.

Wraps the input filtering instance (*ivfs*) methods to keep the same interface.

## CHAPTER 3

---

plancklens.qest

---



## CHAPTER 4

---

plancklens.qecl

---





## CHAPTER 5

---

plancklens.bandpowers

---



## CHAPTER 6

---

plancklens.n0s

---



## CHAPTER 7

---

plancklens.n1.n1

---



## CHAPTER 8

---

plancklens.qresp

---





## CHAPTER 9

---

plancklens.patchy

---



## CHAPTER 10

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### f

`filt_util`, [7](#)

### p

`plancklens.sims.planck2018_sims`, [3](#)



## C

cmb\_len\_ffp10 (class in *planck-lens.sims.planck2018\_sims*), 3  
 cmb\_unl\_ffp10 (class in *planck-lens.sims.planck2018\_sims*), 4

## F

ffp10cmb\_widnoise (class in *planck-lens.sims.planck2018\_sims*), 4  
 filt\_util (module), 7

## G

get\_dat\_tmap() (planck-lens.sims.planck2018\_sims.smica\_dx12\_SZdeproj method), 5  
 get\_sim\_blm() (planck-lens.sims.planck2018\_sims.cmb\_len\_ffp10 static method), 3  
 get\_sim\_blm() (planck-lens.sims.planck2018\_sims.cmb\_unl\_ffp10 static method), 4  
 get\_sim\_elm() (planck-lens.sims.planck2018\_sims.cmb\_len\_ffp10 static method), 3  
 get\_sim\_elm() (planck-lens.sims.planck2018\_sims.cmb\_unl\_ffp10 static method), 4  
 get\_sim\_plm() (planck-lens.sims.planck2018\_sims.cmb\_unl\_ffp10 static method), 4  
 get\_sim\_pmap() (planck-lens.sims.planck2018\_sims.smica\_dx12 method), 4  
 get\_sim\_tlm() (planck-lens.sims.planck2018\_sims.cmb\_len\_ffp10 static method), 3  
 get\_sim\_tlm() (planck-lens.sims.planck2018\_sims.cmb\_unl\_ffp10 static method), 4

get\_sim\_tmap() (planck-lens.sims.planck2018\_sims.smica\_dx12 method), 4  
 get\_sim\_tmap() (planck-lens.sims.planck2018\_sims.smica\_dx12\_SZdeproj method), 5

## L

library\_ftl (class in *filt\_util*), 7  
 library\_shuffle (class in *filt\_util*), 7

## P

plancklens.sims.planck2018\_sims (module), 3

## S

smica\_dx12 (class in *planck-lens.sims.planck2018\_sims*), 4  
 smica\_dx12\_SZdeproj (class in *planck-lens.sims.planck2018\_sims*), 5