


maps & maps based analysis

gammapy coding sprint,
Oct 2018,
UCM, Madrid








Image and cube analysis with maps

- GitHub project: maps
- <https://github.com/gammapy/gammapy/projects/2>

Map analysis  Updated 3 days ago




Filter cards + Add cards

26 To do + ...

-  **Background computation in MapMaker** ...
#1830 opened by AtreyeeS
bug 
-  **Change the convolution function used in `Map.convolve`?** ...
#1813 opened by AtreyeeS
question 
-  **Introduce `RegionGeom` and `RegionNDMap`?** ...
#1805 opened by adonath
feature **question** 
-  **Fit background in MapFit** ...






Automated as **To do** Manage

1 In progress + ...

-  **Add a method to create 2D PSF from a PSF Kernel** ...
#1837 opened by AtreyeeS
feature 
 Changes requested

Automated as **In progress** Manage

139 Done + ...

-  **Changed image_fitting_with_sherpa tutorial to use DC1 runs** ...
#1829 opened by AtreyeeS
cleanup 
-  **Radius of "disk" kernel in WcsNDMap.smooth** ...
#1803 opened by Imohrmann
question 
-  **Add test for KernelBackgroundEstimator without defaults** ...
#1704 opened by joleroi

Automated as **Done** Manage

Image and cube analysis using `gammapy.maps`

- All dependencies on `SkyImage` and `SkyCube` have been removed:
 - 1D spectral analysis and 2D ring background classes have been updated
- Image and cube analysis is performed by `MapMaker` and `MapMakerObs`
 - Configurable to produce any map type in `counts`, `exposure`, `background`
 - Users need to provide a `WcsGeom`, a FoV angle cut and optionally a mask in the form of a `WcsNDMap`.
 - Background is integrated over energy bins but not normalized

gammapy.maps

- Outstanding issues: *astropy.units* handling
 - #1792: MapAxis/MapCoord do not deal properly with units
 - If you don't use the right unit, you will get wrong results but no failure!
 - Do we need some specialized MapAxis?
 - e.g. do we need some EnergyMapAxis?
 - Could replace `gammapy.utils.energy`
 - Could expose nature of axis i.e. true or reco energy
- Longer term issues:
 - #1805: Extend map concept to other type of data

Image and cube analysis using `gammapy.maps`

- Outstanding issues: *background integration*
 - #1830: Background integration is incorrect for too large energy bins (e.g. 2D analysis).
 - `irf.background.integrate_on_energy_range` is performing log trapezoidal integration
 - the oversampling factor needs to be adapted to the specific energy binning. Options:
 - expose parameter to the user.
 - impose minimal number of evaluation per decade
 - perform exact integration. `Background3D` performs only linear or nearest neighbor interpolation.
 - the latter requires more work but is likely more accurate and faster

Image and cube analysis using `gammapy.maps`

- Outstanding issues: *background normalization*
 - So far background normalization is only based on IRFs
 - This won't work in most real cases. Need to add background normalization schemes used in classical Cherenkov analysis:
 - `RingBackgroundEstimator` and `AdaptiveRingBackgroundEstimator` objects exist
 - Need to add a `FoVBackgroundEstimator`
- How to pass this to `MapMaker`
 - pass normalization object directly
 - Create new distinct map maker class?

Image and cube analysis using `gammapy.maps`

- Outstanding issues: *handling true and reco energies*
 - #1583 : `e_true` and `e_reco` geometries are assumed to be identical in `MapMaker`
 - Not adapted to 2D analysis and likely error prone in general
 - Users should provide the `e_reco MapGeom`
 - true energy bins could be provided separately
 - e.g. `Emin`, `Emax`, `nbins`
 - Note there are typical default values per instrument/configuration
 - exposure map & PSF kernels in reco energy should be computed through `apply_edisp` function

Image and cube analysis using `gammapy.maps`

- Longer term issues: *IRF reprojection efficiency*
 - IRF reprojection is the most time-consuming task for maps creation
 - Computing reprojected IRFs on fine binning (e.g. 0.01°) is useless and inefficient when IRF are stored on $\sim 0.5^\circ$ theta grids.
 - Better interpolate and mosaic on large bins and interpolate final image on required binning
 - Define minimal spatial binning for IRF maps and interpolate by `Map.interp_by_coord` as final step.
-

Image and cube analysis using `gammapy.maps`

- Longer term issues: *IRF reprojection*
 - Finish `PSFMap` class
 - Include `PSFMap` summation function
 - Prepare proper energy dispersion treatment:
 - Create `EdispMap` class
 - How to deal with reco energy mask?
- Longer term issues: *Improving parameter exposure on MapMaker*
 - How many parameters should we expose in MapMaker? How much complex can it really get?
 - Config-file driven version?