Overview of gammapy.maps

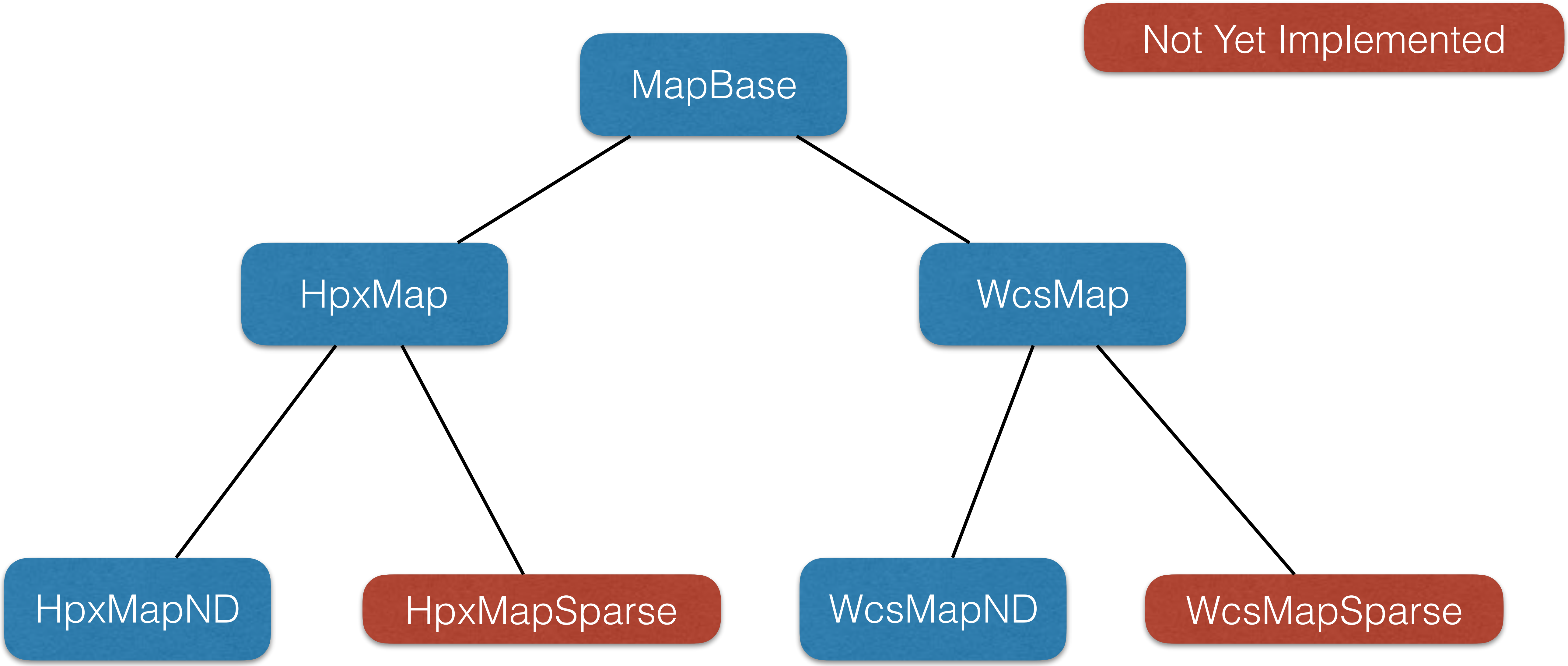
Matthew Wood

Gammapy Coding Sprint
9/21/17

Package Overview

- Documentation URL: https://gammapy.readthedocs.io/en/latest/maps/index.html
- Core Features
  - Common interface for HPX- and WCS-based maps
  - Support for arbitrary number of non-spatial dimensions (energy, time, hadronness, FoV angle, etc.)
  - Sparse FITS serialization format
  - In-memory sparse maps (not yet implemented)
  - Multi-resolution maps (different pixel size or geometry in each image plane)

Class Hierarchy



Not Yet Implemented

MapBase

HpxMap

WcsMap

HpxMapND

HpxMapSparse

WcsMapND

WcsMapSparse

FITS I/O

- All classes support serialization to FITS (see http://gamma-astro-data-formats.readthedocs.io/en/latest/skymaps/index.html for format details)
- Both HPX and WCS classes support I/O to a sparse FITS table format by calling write with **sparse=True**
  - One row per pixel with pixel # and image plane index (channel #)
  - Geometry for non-spatial dimensions written to BANDS table
- Sample All-Sky Sparse LAT counts cube: https://www.dropbox.com/sh/wzh0p8d949uikz0/AACW50nmmG8wliJ-vsg2QDg_a?dl=0
  - 100 MeV - 100 GeV (4 bins/decade)
  - Energy-dependent pixel size (~3 deg at 100 MeV and ~0.1 deg at 100 GeV)
  - Eight years (MET 239557414-492018217)
  - File Size: ~30MB

Multi-resolution Maps

- Multi-resolution maps allow for a different pixel or image geometry in each image plane
    - Allows for pixel-size to be matched to PSF (very useful for LAT where PSF is rapidly varying)
    - Model maps can be truncated at some distance that is a function of energy
- Currently implemented for both HpxMapND and WcsMapND by allocating a numpy array with image dimension commensurate with largest image plane
- Sparse maps would handle multi-resolution geometries more efficiently

Sparse Maps

- Sparse maps combine the advantages of binned and unbinned analysis
    - Speed proportional to # of photons in low counts regime (reduces to photon list)
    - Speed proportional to # of bins in high counts regime (reduces to counts map)
- For likelihood evaluation model only needs to be computed at non-zero pixels (as in unbinned analysis)
    - Likelihood evaluation loop can be a loop over pixels in the sparse map
    - Model map objects occupy much less memory (only occupied pixels need to be saved)

To Do List

- Missing API methods
  - `cutout/paste`
  - `downsample/upsample`
  - `pad/crop`
  - `convolve`
- Image Slicing (`to_slice`)
- Sparse Maps
  - Just a basic skeleton currently — lots of work to be done
  - Plan to implement with `scipy.sparse`
- Access by view vs. copy
  - `get_by` accessor methods return by value rather than reference
  - Need to implement view-based methods for high-performance applications

Open Questions and Next Steps

- Feedback on API and Design would be extremely valuable at this stage — API will be much harder to change in the future
  - Method name and API changes
  - Proposals for new methods
- Improve integration with gammapy as a whole
  - What additional features are needed to allow migration from SkyImage and SkyCube?
  - Is there functionality gammapy.maps should be using from other submodules?
  - What should be the relationship with NDData classes?
  - Is there code in gammapy.maps that should be merged with code else in gammapy?
- Migrate fermipy to use gammapy
  - Plan to replace existing fermipy maps classes with gammapy.maps
  - Would like to have this ready for v0.7 release