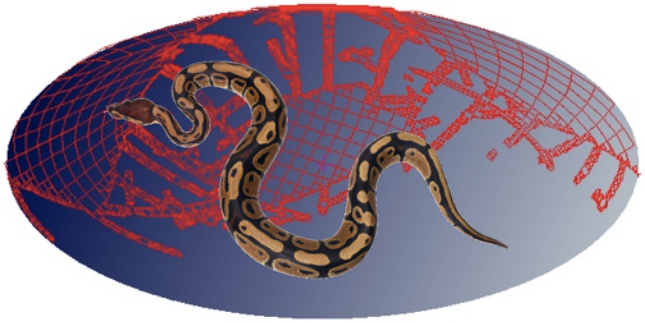
MOCPy



A Python library to handle MOCs

Thomas BOCH

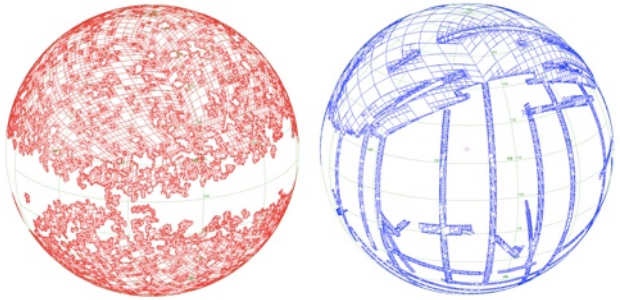
CENTRE DE DONNÉES
ASTRONOMIQUES DE STRASBOURG

Outline

1. Context
2. Main features
3. Demonstration
4. Requirements/installation
5. Links

Context

- MOC *(Multi-Order Coverage map)*
  - IVOA standard to describe arbitrary sky regions
  - Based on HEALPix tessellation
  - Serialized in a FITS file



- Available MOCs
  - 14,000+ VizieR tables with positions
  - 200+ CDS HiPS *(Hierarchical Progressive Surveys)*
  - Spanish VO resources
  - WFAU-hosted surveys (UKIDSS, VISTA, OmegaCAM) @Edinburgh

MOC usage

- Goals
  - Visualization
  - Fast comparison of coverages
  - Data access methods (query a service by a complex region)

- Tools supporting MOCs
  - Topcat
    - cross-match and multi-cone search
  - Aladin
    - visualization of dataset coverages

  + MOCPy

MOCPy features (1/2)

- Read a MOC
  - from local file or URL

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / array data type
NAXIS   =                    2 / number of array dimensions
NAXIS1  =                    4 / length of dimension 1
NAXIS2  =                71002 / length of dimension 2
PCOUNT  =                    0 / number of group parameters
GCOUNT  =                    1 / number of groups
TFIELDS =                    1 / number of table fields
TTYPE1  = 'UNIQ    '
TFORM1  = '1J      '
PIXTYPE = 'HEALPIX '
ORDERING= 'NUNIQ   '
COORDSYS= 'C       '
MOCORDER=                    8
```

- Retrieve a MOC
  - for a VizieR table
  - for a given HiPS

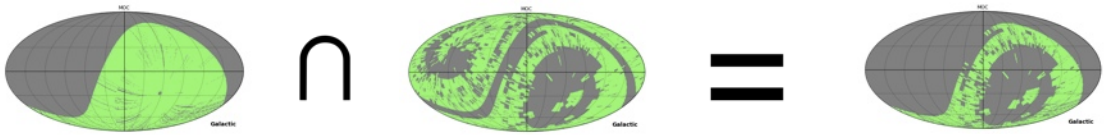
  query CDS MOCServer

  → (cf. Pierre Fernique's presentation in Registry 2 tomorrow)

- Create a MOC
  - from scratch
  - from a table with positions

MOCPy features (2/2)

- Operations
  - intersection
  - union



- Plot

- Filter a list of astronomical sources
  - keep sources inside the coverage described by a MOC

- Query by MOC
  - any VizieR table having position
  - a view of SIMBAD data

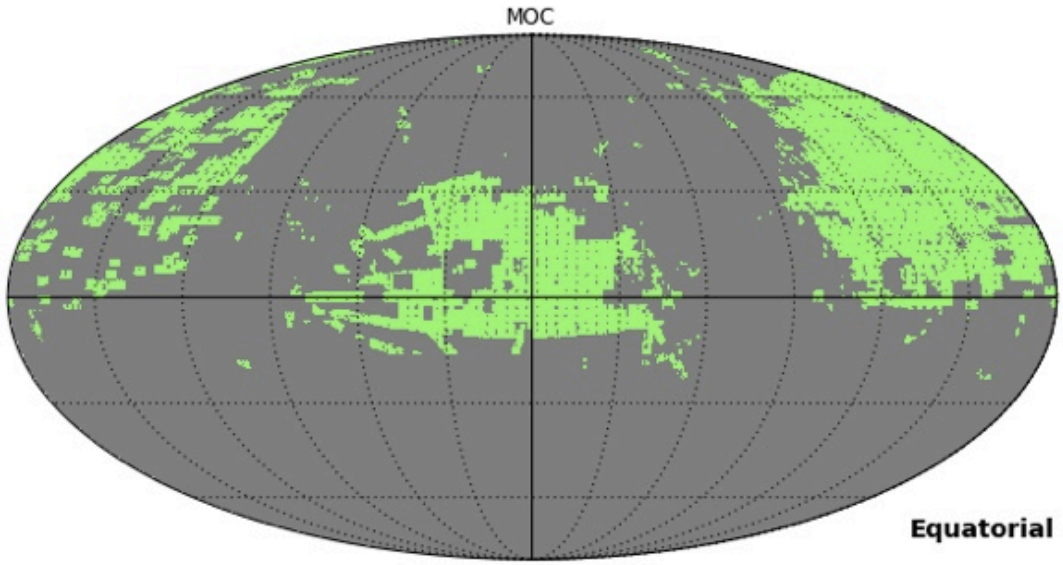
Demonstration

```
In [1]: from mocpy import MOC

In [2]: m1 = MOC.from_vizier_table('II/313/table3', nside=512)

In [3]: m2 = MOC.from_vizier_table('V/139/sdss9', nside=512)

In [4]: m1.intersection(m2).plot()
0.0 180.0 -180.0 180.0
The interval between parallels is 30 deg -0.00'.
The interval between meridians is 30 deg -0.00'.
```



Installation

- Requirements
  - Python 2 and Python 3
  - Dependencies
    - `astropy`
    - `numpy`
    - `healpy`
- Available on PyPi repository
  - `pip install mocpy`

Work in progress

- Improvements
  - Performances
    - Some operations currently slow
      - MOC creation from > 1 million positions
  - Better documentation
  - More tests

Links

- Github project
  - https://github.com/tboch/mocpy
  - GPL v3

- Notebooks
  - Examples on how to use the API: https://github.com/tboch/mocpy/tree/master/notebooks