

MOC2.0 status

IVOA interop, 25 May 2021

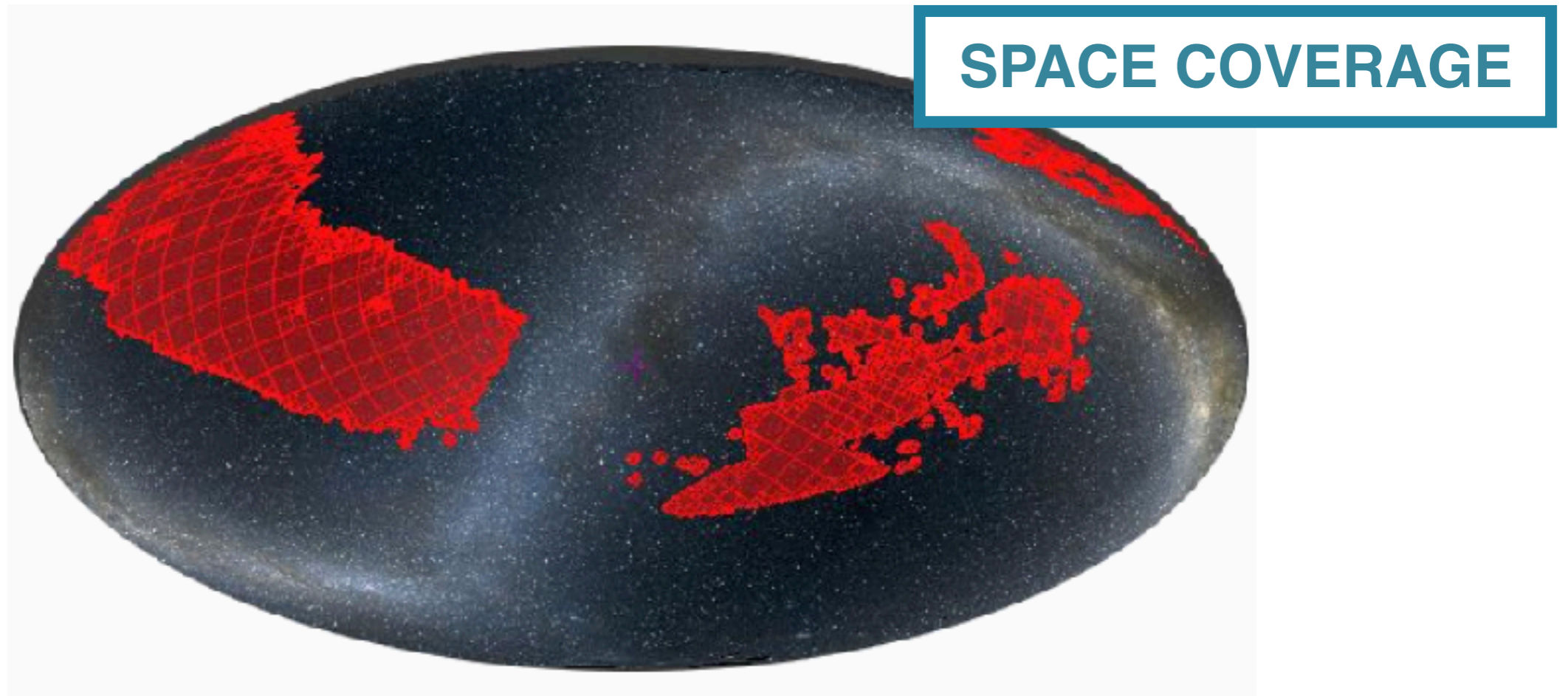
Ada Nebot
A team effort



Pierre Fernique (CDS), Ada Nebot (CDS), Daniel Durand (CADC),
Matthieu Baumann (CDS), Thomas Boch (CDS), Giuseppe Greco
(EGO-Virgo), Tom Donaldson (STScI/NASA), Francois-Xavier
Pineau (CDS), Mark Taylor (University of Bristol), Wil O'Mullane
(Vera C. Rubin Observatory), Martin Reinecke (Max Plank),
Sébastien Derrière (CDS)

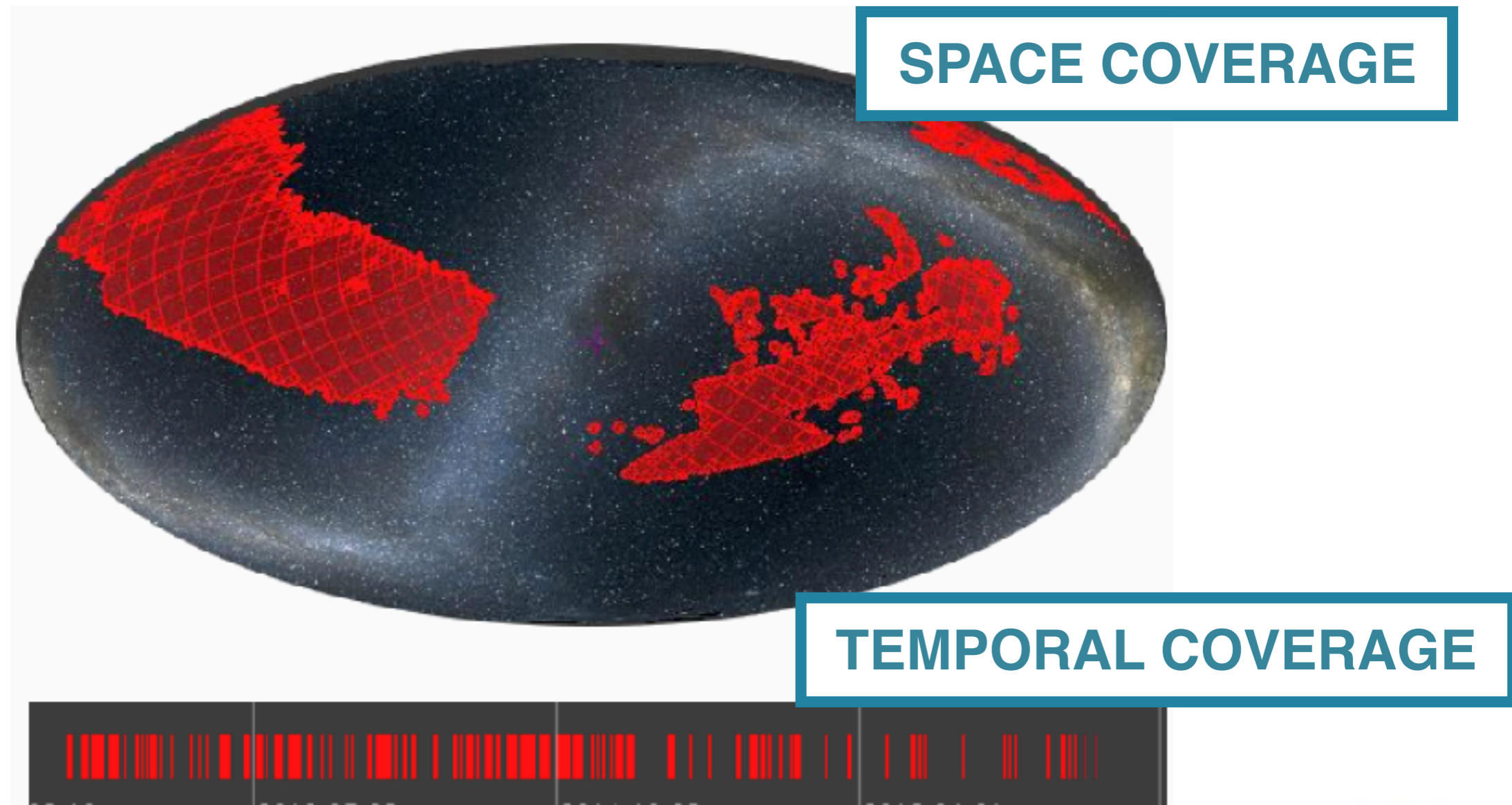
□ MOC1.1 current standard

- Coverage maps based on MOC IVOA standard are created from positions
 - ➔ We know where but we don't know when!
 - ➔ Need to add the time dimension



□ MOC2.0 Working draft

- Generalisation of MOC to integrate the time
 - Same principles for the space as for MOC1.0
 - Time discretisation based on JD



□ MOC2.0 Working draft

➔ allows to look for *simultaneous observations*

Possibility of extremely fast

➔ computations (generate from catalog, images, regions)

The only thing you need is a list of positions and times, spatial and temporal resolution)

➔ operations (unions, intersections,...)

E.g. Have these two telescopes observed the same sky region within this interval of time?

□ MOC2.0 Working draft

- Since 2018 :
 - Wrote a Note with the idea of a time MOC (TMOC)
 - Created python library for TMOC under mocpy
 - Published jupyter notebooks showing different ways of using it
 - Created TMOCs for hundreds of Vizier catalogs
 - Developed both space + time MOC in python and in java
 - Tested and evolved (iterative process)
 - Shown around the community at different conferences
 - Shown at interop meetings the status at different stages of the process

□ MOC2.0 Working draft



MOC: Multi-Order Coverage map Version 2.0

IVOA Working Draft 2021-03-24

Working group
Applications

This version
<http://www.ivoa.net/documents/moc/20210324>

Latest version
<http://www.ivoa.net/documents/moc>

Previous versions
Version1.1
Version1.0

Author(s)
Pierre Fernique (CDS), Ada Nebot (CDS), Daniel Durand (CADC),
Matthieu Baumann (CDS), Thomas Boch (CDS), Giuseppe Greco
(EGO-Virgo), Tom Donaldson (STScI/NASA), Francois-Xavier
Pineau (CDS), Mark Taylor (University of Bristol), Wil O'Mullane
(Vera C. Rubin Observatory), Martin Reinecke (Max Plank),
Sébastien Derrière (CDS)

Editor(s)
Pierre Fernique, Ada Nebot, Daniel Durand

25/05/2021

- Document in progress can be found under:
 - <https://www.ivoa.net/documents/MOC/20210324/index.html>
 - <https://github.com/ivoa-std/MOC>
- To become a standard **we need 2 reference implementations and one validator.**

□ MOC2.0 status

- Existing implementations:
 - java - java - mocJava lib
 - python - mocpy package
- Validator
 - In the java lib (cds.moc.misc.moclint) ; Implemented in Mocserver
 - ➡ More on Friday @ 22:00 UTC in Ops session
- In preparation for the RFC period we are collecting information under :
 - <https://wiki.ivoa.net/twiki/bin/view/IVOA/MocInfo?>

□ MOC2.0 status

- **MocJava version 6.0**

- fully MOC 2.0 compliant (last WD)
- in beta test phase (you can use it for tests)
- already integrated in:
 - Aladin Desktop beta (>v11.05) <http://aladin.cds.unistra.fr/java/AladinBeta.jar>
 - Hipsgen beta with STMOC possibility
- lib available under:
 - <http://aladin.cds.unistra.fr/JavaMoc6beta> (temporary URL)
 - <https://wiki.ivoa.net/twiki/bin/view/IVOA/MocInfo?>

□ MOC2.0 status

Page Contents
Welcome to MOCPy's documentation!

- What is a MOC ?
- References

Indices and tables

Next topic
Install

This Page
Show Source

Welcome to MOCPy's documentation!

Contents:

- [Install](#)
- [Examples](#)
 - [Spatial coverages](#)
 - [Temporal coverages](#)
 - [Space & Time coverages](#)
- [API](#)
 - [Class overview](#)
- [Contribute](#)
 - [Setting up the environment](#)
 - [Running the tests](#)
 - [Building the documentation](#)

MOCPy is a Python library allowing easy creation, parsing and manipulation of MOCs (Multi-Order Coverage maps). Its code is hosted on [GitHub](#) and distributed under the [BSD-3 license](#).

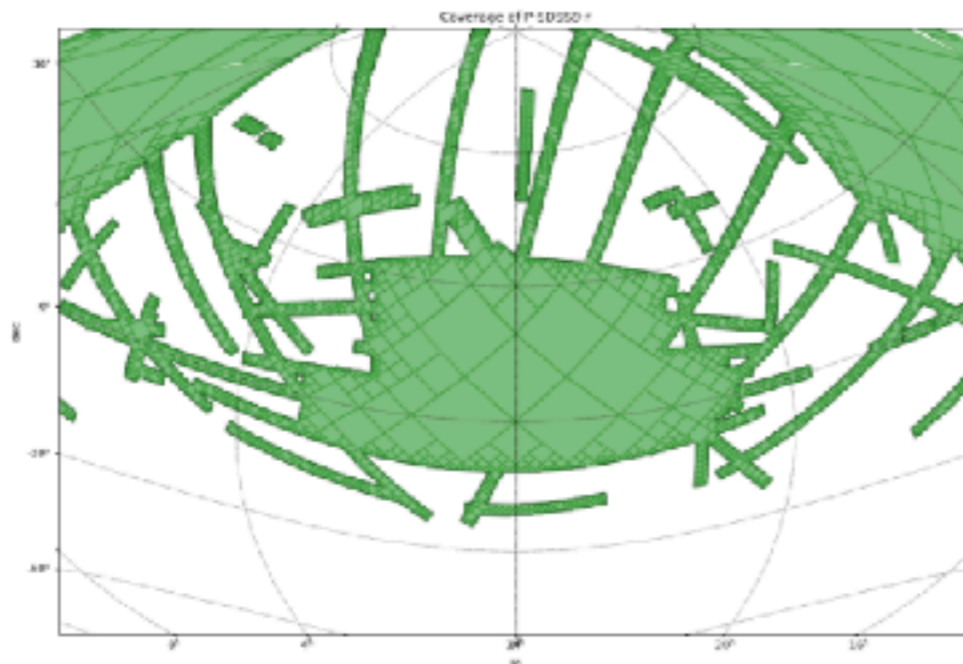
What is a MOC ?

MOC is an [IVOA standard](#) enabling description of arbitrary sky regions. Based on the HEALPix sky tessellation, it maps regions on the sky into hierarchically grouped predefined cells.

MOCPy provides the `MOC` and `TimeMOC` classes handling respectively the manipulation of spatial and temporal MOCs.

As an example, here is the sky coverage of the SDSS sky survey:

([Source code](#), [png](#), [hires.png](#), [pdf](#))



- **mocpy**
 - a Python library to handle MOCs
 - Astropy affiliated package
 - Documentation
 - Examples
- **actions**
 - need to make it fully compatible with the latest WD

□ MOC2.0 news

And now ...

...for something completely different...

DEMO time !

<https://youtu.be/lhWBSvM8Khk>