Aladin Lite meets MOCWasm:

online calculator and interactive viewer of credible areas of GW sky localizations

Giuseppe Greco, Mateusz Bawaj, Roberto De Pietri, François-Xavier Pineau and many others

MOCWasm: A WebAssembly Library for MOC

MOCLibRust: a common library for MOCPy, MOCCli and MOCWasm

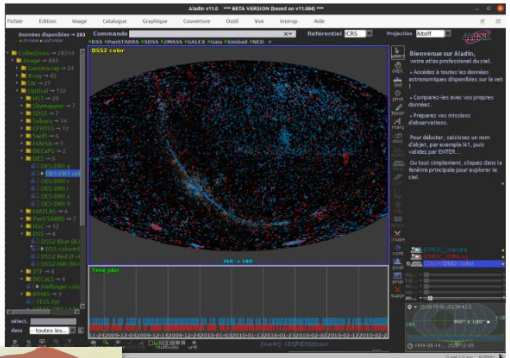
F.-X. Pineau, M. Baumann, M. Allen, T. Boch, P. Fernique, G. Greco and A. Nebot

francois-xavier.pineau@astro.unistra.fr

Centre de Données astronomiques de Strasbourg

Multi-Order Coverage map

Multi-Order Coverage map (MOC) is an IVOA standard and a powerful tool to create and manipulate discretized space, time and space-time (ST) coverages. For example, one can retrieve the pre-built ST-MOCs of XMM and Chandra and easily find the sky areas observed at the same time by both instruments.



T-MOCs of Chandra and XMM logs in Aladin. Aladin uses a Java MOC library.

MOC standard deals exclusively with spatial coverages. The vesion 2.0 introduces ime MOCs. The last version of MOC 2.0 has been implemented both in a Java e) and in a Rust library currently used by MOCPy, MOCCli and MOCWasm. Check https://github.com/cds-astro/cds-moc-rust.

MOCCli to manipulate MOCs from the command line

MOCCli is a tool to load, create, manipulate and save MOCs from the command line. It consits in a single binary file pre-compiled for various architectures: MacOS; Windows; both 32 and 64 bits Linux. A *.deb* file is also available for Debian and derivatives such as Ubuntu.

```
pineau@cds-dev-fxp1:~/tmp$ moc from cone --help
moc-from-cone 0.1.0
Create a Spatial MOC from the given cone

USAGE:
    moc from cone <depth> <lon-deg> <lat-deg> <r-deg> <SUBCOMMAND>

FLAGS:
    -h, --help       Prints help information
    -V, --version    Prints version information

ARGS:
    <depth>      Depth of the created MOC, in `[0, 29]`
    <lon-deg>    Longitude of the cone center (in degrees)
    <lat-deg>    Latitude of the cone center (in degrees)
    <r-deg>      Radius of the cone (in degrees)

SUBCOMMANDS:
    ascii    Output an ASCII MOC (VO compatible)
    fits     Output a FITS MOC (VO compatible)
    help     Prints this message or the help of the given subcommand(s)
    json     Output a JSON MOC (Aladin compatible)
    stream   Output a streamed MOC
pineau@cds-dev-fxp:~/tmp$ moc from cone 11 083.63308 +22.01458 0.25 ascii --fold 80
9/386969
9/1547853-1547855 1547864 1547866 1547888-1547890
10/6191405 6191407 6191409-6191411 6191462 6191468 6191470-6191471 6191493
 6191495 6191540-6191541 6191564 6191566 6191584
11/24765562-24765563 24765566-24765567 24765597 24765599 24765627
 24765634-24765635 24765738 24765842 24765854 24765876 24765878-24765879
 24765969 24765971 24766004-24766005 24766007 24766013 24766148-24766149
 24766200-24766202 24766340-24766341
pineau@cds-dev-fxp:~/tmp$
```

Figure 3: Create and display the ASCII serialization of the MOC of a given cone.

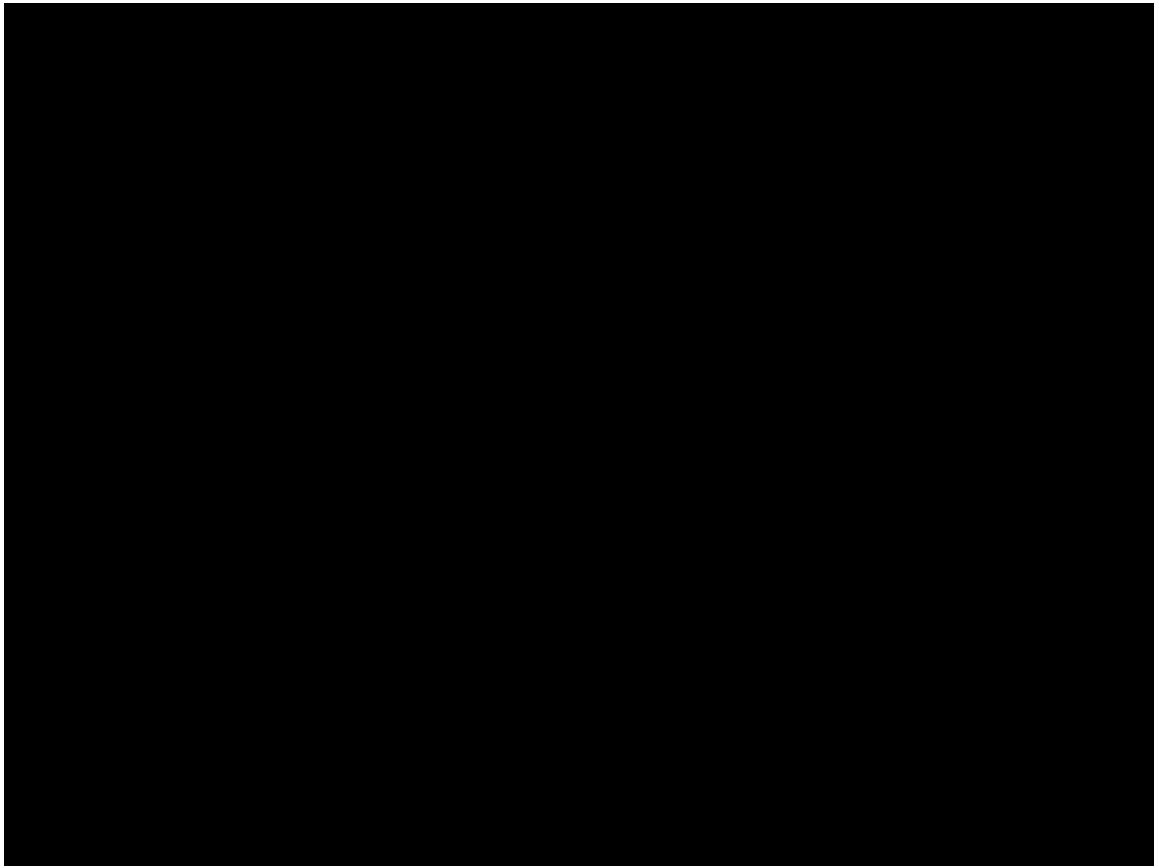
Check MOCCli releases at https://github.com/cds-astro/cds-moc-rust/releases and the source code at https://github.com/cds-astro/cds-moc-rust/tree/main/crates/cli.

MOCWasm to manipulate MOCs from Web Browsers

MOCWasm is a WebAssembly library to load, create, manipulate and save MOCs from JavaScript. It can be used direclty from the console of your favorite Web browser, or could be included in an user interface such as Aladin Lite. Check the project at https://github.com/cds-astro/cds-moc-rust/tree/main/crates/wasm and the

LIVE DEMO

https://virgo.pg.infn.it/maps/index.html



Video recording

Next Steps

- Tutorial and documentation (A&C paper?)
- Aladin Lite MOC transparency
- ST-MOC support by adding a time range
- Testing and improving css and javascript codes
- Query 2D/3D galaxy catalogs

Testing/Idea/Suggestions are very welcome!