Diffuse Source Matching

Using MOC & JSON format

Marco Molinaro

INAF – OATs / INAF – IA2

IVOA *Northern Fall* Interoperability Meeting 2016
Stazione Marittima, Trieste, Italy
Apps Session
Saturday 22/10/2016

ASTERICS 653477

VIALACTEA 607380

Outline

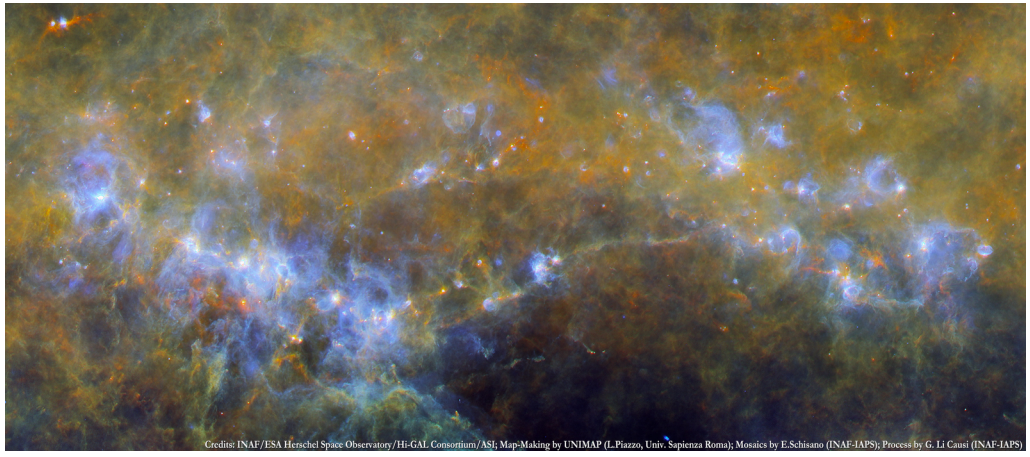
- **Compact sources & Diffuse structures**
- **Positional matching requirement**
- **MOC FITS&JSON solution**

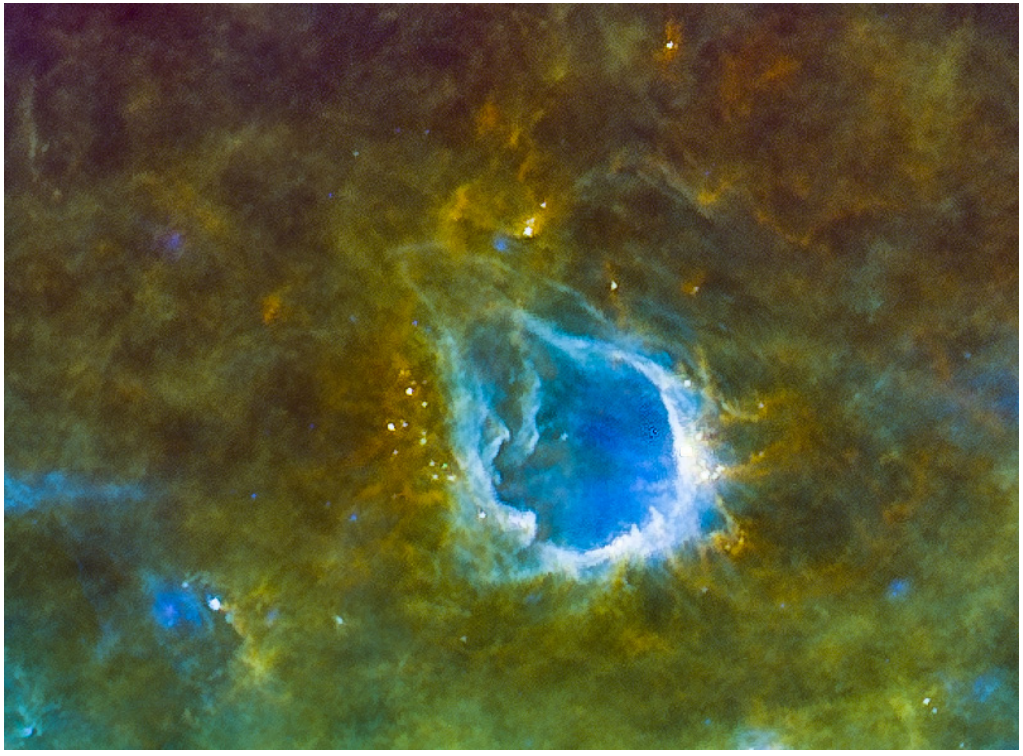
Resources in Catalogue

Catalogue	# records
MSX (21 μm)	432 K
WISE (22 μm)	29 M
MIPSGAL (24 μm)	2.3 M
Hi-Gal (70 μm)	158 K
Hi-Gal (160 μm)	580 K
Hi-Gal (250 μm)	468 K
Hi-Gal (350 μm)	252 K
Hi-Gal (500 μm)	130 K
ATLASGAL (870 μm)	11 K
BGPS (1100 μm)	8.6 K
Band-merged	1.9 M
SED models	20 M

- **VIALACTEA Knowledge Base hosts various datasets**
- **Among them catalogues**
  - **Compact sources**
  - **Diffuse objects**
- **Figures are not too big**

Catalogue	# records
Filaments	30 K
F - branches	132 K
F - nodes	191 K
Bubbles	5 K

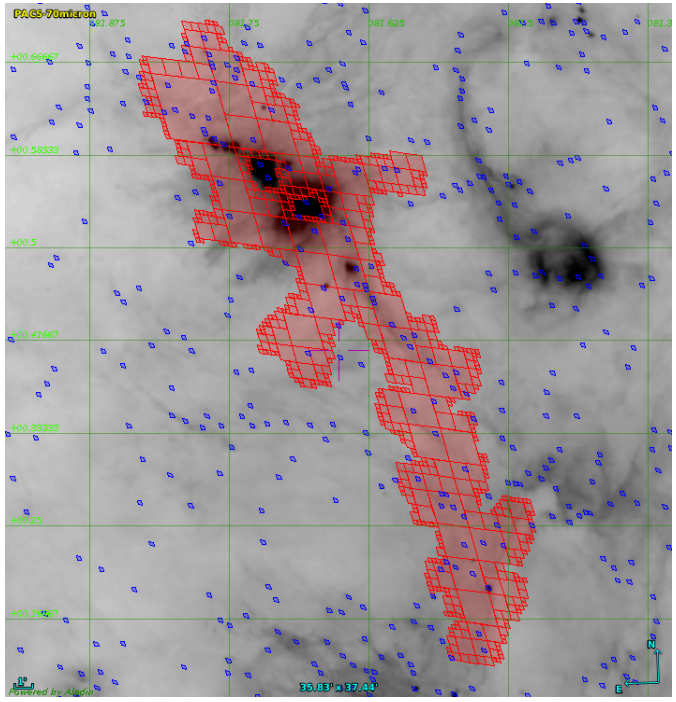


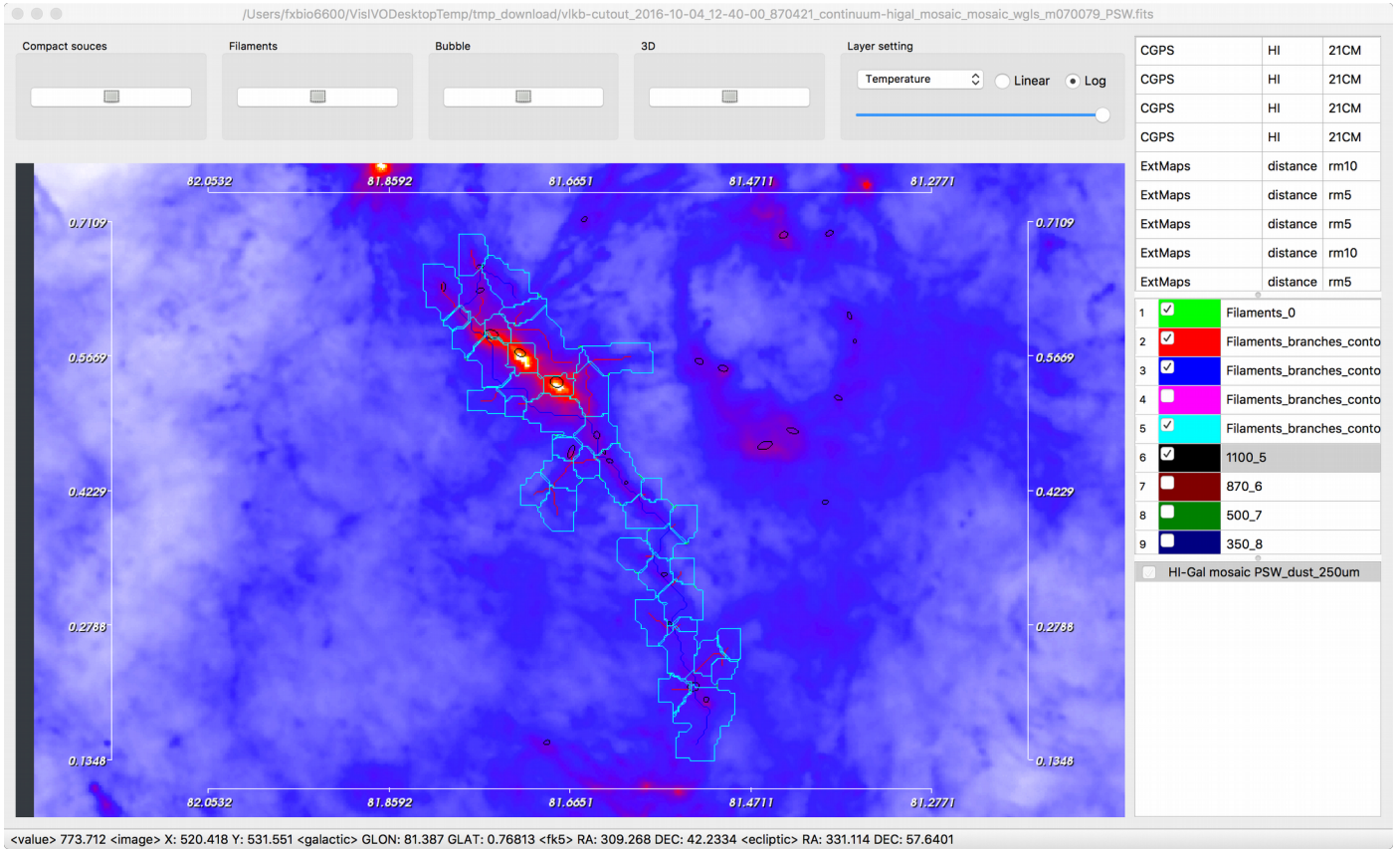


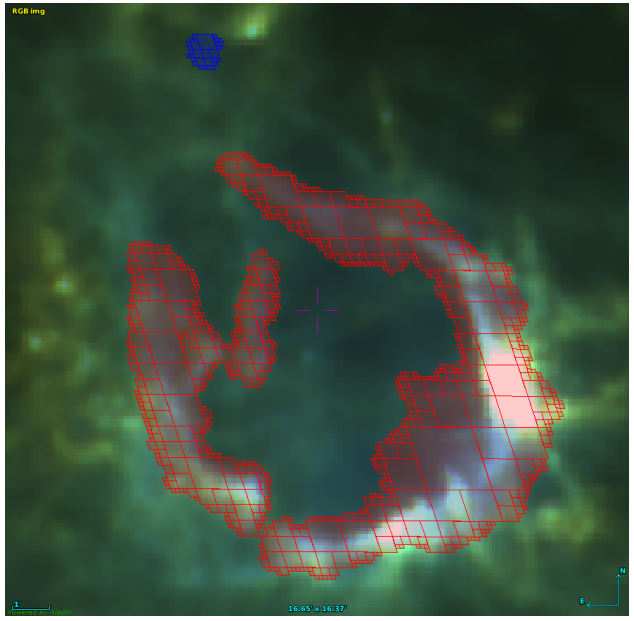
Access & Match requirements

- **Standard access granted through a TAP service**
  - **not yet in registry**
  - **consumed by project-developed visual tool**
- **Overlap of compact vs. diffuse sources and vice-versa required**
  - **Complex morphology in place**
- **...and a pre-existing constraints:**
  - **MySQL**
    - **No arrays**
      - **How do I serialize a polygon in it?**
    - **Difficult to handle geometry**
  - **Galactic coordinates**
    - **Usually VO protocols speak ICRS**

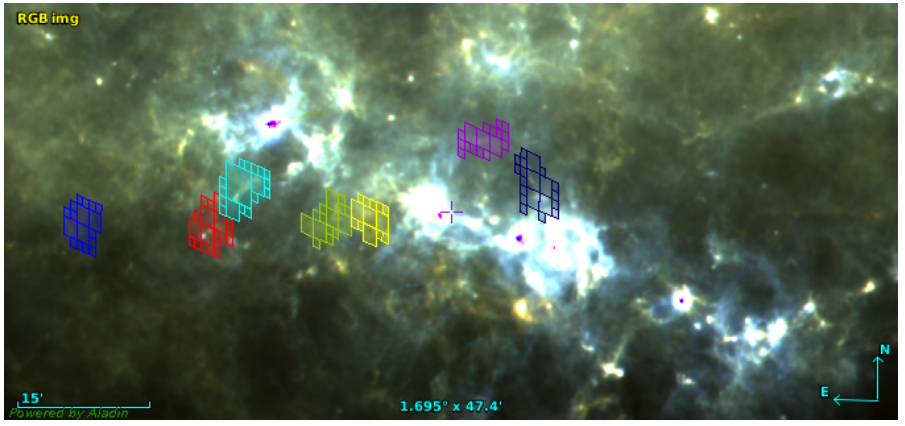
MOC in place







Filaments and bubbles tessellation examples



MOCs in use

- **Formats**
  - **FITS**
    - **For full catalogues**
  - **JSON**
    - **For each diffuse object**
      - **Stored as MySQL MEDIUMTEXT**
  - **NPIX**
    - **For each catalogue entry**
    - **For diffuse helper tables**
      - **“prefiltering”**
- **Order**
  - **19**
    - **Full catalogues**
  - **17**
    - **Bubbles**
  - **16**
    - **Filaments**
  - **9**
    - **Helper table prefilter**
- **Prototype solution**
  - **Multi-order needed to better fit the bandmerged sources**

Usage

- **Search interface the simplest possible**
  - **4 parameters**
    - **Search for: what you'd like to retrieve**
    - **Starting from**
      - **Objects identifier & type (2 params)**
      - **Region (polygon of vertexes)**
      - **Mutually exclusive**
- **...thus something like**
  - **{server-endpoint}/search=bubble&id=1&type=compact**
  - **{server-endpoint}/search=compact&region=a_polygon**
    - **a_polygon: array of {l,b}**

Summary

- MOCs are useful
  - For coverage estimation/visualization
  - For allowing quick overlap when backend has no support for it
- Different serializations help a lot
  - But don’t make this a “pro” point in standardizing whatever hack you may think about

Thank you for your attention!