Fig. 1

1. HEALpix, MOC, and ADQL

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

- Reminder: Virtual Observatory, TAP, ADQL
- Reminder: HEALPix
- Writing ADQL queries with HEALPixes
- What are MOCs?
- Writing ADQL queries with MOCs
- Plotting MOCs

Note: I'm using desktop tools here – you could just as well use Python. Contact me for details

(cf. Fig. 1)

2. Reminder: VO

The Virtual Observatory (VO) is a set of standards to facilitate finding, retrieving and using astronomical data – and a large number of data centres and clients adopting them.

TOPCAT is one of these clients. And really nifty beyond that.

TAP is a VO standard to run database queries on remote tables. If you have queried the Gaia database, you have probably used it.

ADQL is the language TAP queries are written in.

If you have not touched any of that before, what I'll show you today may mystify you a bit. In that case, I really should be giving a VO day here, introducing some of the basic concepts. For today, just sit back and watch.

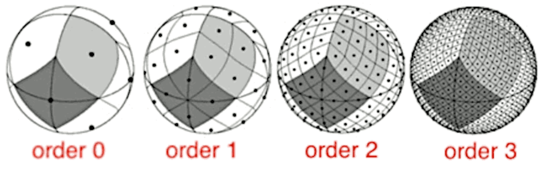

Fig. 2

3. HEALPix: Definition

HEALPix is

- Hierarchical – there are 12 pixels at level 0, and $12 \cdot 4^n$ pixels at level n, where pixels at higher levels are always true subsets of pixels in lower levels: All HEALPixes make up a tree
- Equal Area – at a given level, each pixel has the same area: pure magic!
- isoLatitude – distinct latitudes of pixel centers go with $O(n)$ rather than $O(n^2)$ with the order
- Pixelization – mapping $(\alpha, \delta) \to [0, \ldots, N]$, greatly simplifying just about *any* manipulation.

4. HEALPix: Intuition

(cf. Fig. 2)

The linear dimension of a HEALPix is $\sim 1^\circ$ at level 6; it changes by a factor of two on each level.

Extra trick: NEST numbering of the pixels lets you go between levels by integer division or multiplication by 4.

5. Queries with HEALPixes

The VO's query languge ADQL does not support HEALPix natively.

But there are standard extensions ("UDFs") for dealing with them:
`ivo_healpix_center(hpxOrder INTEGER, hpxIndex BIGINT) -> POINT`

and
```
ivo_healpix_index(order INTEGER, ra DOUBLE PRECISION, dec DOUBLE
 PRECISION) -> BIGINT
```
available on many TAP services.

To find out whether your TAP service has them, inspect the TAP capabilities; in TOPCAT, you will find the list of UDFs in the *Service* tab.

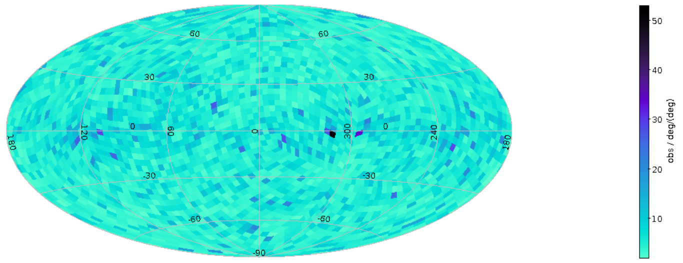

Fig. 3

6. Application: HEALPix Maps

A prime application of these functions is the creation of HEALpix maps, which allow one to get a quick idea of distributions of all kinds of things in catalogues.

On the GAVO DC TAP at http://dc.g-vo.org/tap service, there is a service-provided example for producing HEALPix maps:

```
SELECT
        MAX(parallax)/AVG(parallax) AS obs,
        ivo_healpix_index(4, ra, dec) AS hpx
FROM hipparcos.main
GROUP BY hpx
```

In case you're wondering: No, I don't think there is any physical significance to the ratio of the parallax of the closest star to the average parallax in a neighbourhood. This is just to show that you can use interesting expressions here, not just `COUNT(*)`.

In TOPCAT, you can plot this by using *Sky Plot* and then *Layers* → *HEALPix control*. With a few extra adjustments, this will yield something like this:

(cf. Fig. 3)

7. In Gaia

HEALPixes are so nifty that Gaia uses them for its source ids.To get the HEALPix of a Gaia object at level n, compute

$$\text{hpx} = \frac{\text{source\_id}}{4^{12-n} \cdot 2^{35}}.$$

This only works down to level 12, which is the healpix level used to determine the Gaia source ids. According to our rule of thumb on the linear dimension at level 6, the linear dimension of that would be a degree divided by 2^6 or about an arcminute.

Use that to compute a rough dust map:

```
SELECT source_id/8796093022208 AS pix,
  AVG(phot_bp_mean_mag-phot_rp_mean_mag) AS avgcol
FROM gaia.edr3lite
WHERE DISTANCE(ra, dec, 246.7, -24.5)<2
GROUP BY pix
```

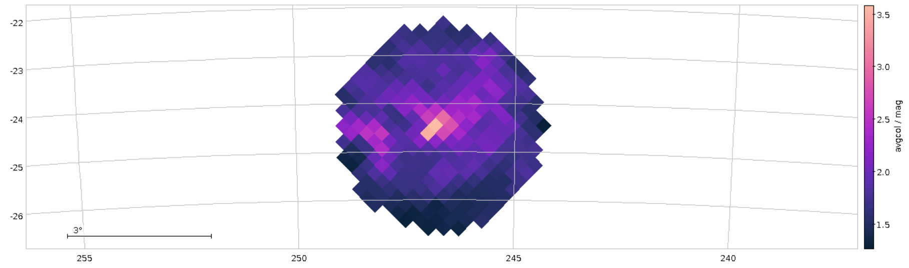

Fig. 4

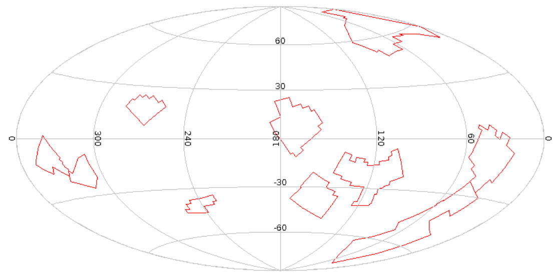

Fig. 5

That's level 8 two degrees around the center of a really close dust cloud in Monoceros and plots as:

(cf. Fig. 4)

8. Polygon union Polygon

Have you ever tried to compute the union or intersection of two spherical polygons?

It's a nightmare. Not to mention the result is not a polygon any more:

(cf. Fig. 5)

In case you are curious: something like this is what `select top 10 * from cstl.geo` produces.

MOCs to the rescue!

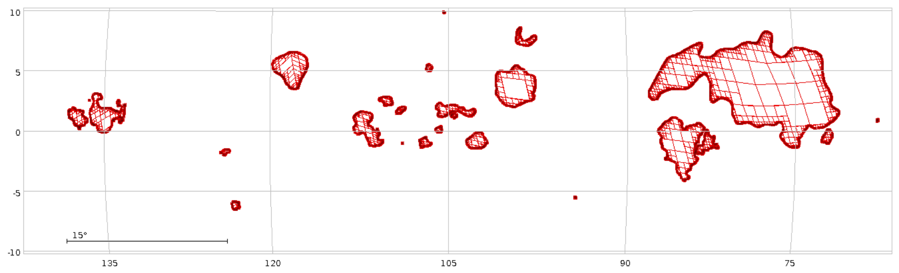

Fig. 6

9. MOC?

You can represent arbitrary shapes to high precision (order 29 is 0.4 mas) as lists of HEALPix indexes.

Alas, you need about 10 million such pixels for a shape of 1 deg^2.

Solution: Abbreviate ranges and use lower-order indexes when the pixels are full.

That's a Multi Order Coverage map, or MOC in short.

10. MOC examples

`select * from openngc.shapes`

When plotting this, consider that the HEALpix lists this time are in columns. Hence, you will need an Area control.

(cf. Fig. 6)

Such a shape may be written like
`11/34094023 12/136376116-136376117`

– all the shapes together are less than half an MB.

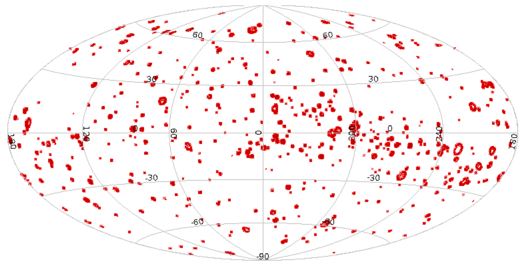

Fig. 7

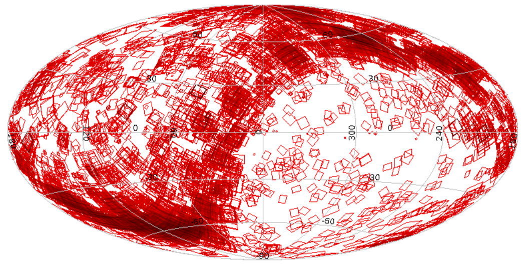

Fig. 8

11. Math with MOCs

Most operations really become simple with MOCs. For instance, the area on the sky within 30 arcminutes of Hipparcos stars brighter than 4 mag:

(cf. Fig. 7)

That's the result of the following ADQL query; note, however, that MOC support in ADQL is highly experimental at this time and unavailable in most TAP servers.

```
SELECT SUM(MOC(8, CIRCLE(ra, dec, 0.5*(4-vmag)))) AS contaminated
FROM hipparcos.main
WHERE vmag<4
```

That's *one* shape you can manipulate as such.

12. Find Uncontaminated Images

Once you have such a MOC, you can use it to look for uncontaminated images, for instance like this:

```
SELECT accref, coverage
FROM lsw.plates
JOIN tap_upload.t14
ON 0=INTERSECTS(MOC(9, coverage), contaminated)
```

(cf. Fig. 8)

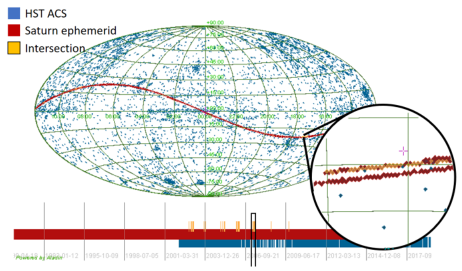

Fig. 9

13. TMOCs, STMOCs

Recently, people have extended the scheme to time and correlated space-time. That's cool if you want to find data on fast-moving objects:

(cf. Fig. 9)

You can use and create these with Aladin, for instance. Outside of this, tools largely still need updates. In particular, there is no ADQL support for them at all, and one would need to think quite a bit how that would look like.

14. Parting Words

If you now wish your ADQL was better: Talk to me about ADQL courses on all levels.

If you have data to publish: Talk to me about getting it into the VO
http://docs.g-vo.org/talks/2022-hq-moc.pdf . . . Thanks!