What to do about huge LIGO/ Virgo sky maps

Leo Singer, NASA/GSFC

LIGO-G1800186-v4

The problem

We’re getting better at pinpointing gravitational-wave sources as more detectors come online and existing detectors become more sensitive.

Unfortunately, as position accuracy improves, the size of the sky maps that we send to observing partners is going to blow up.

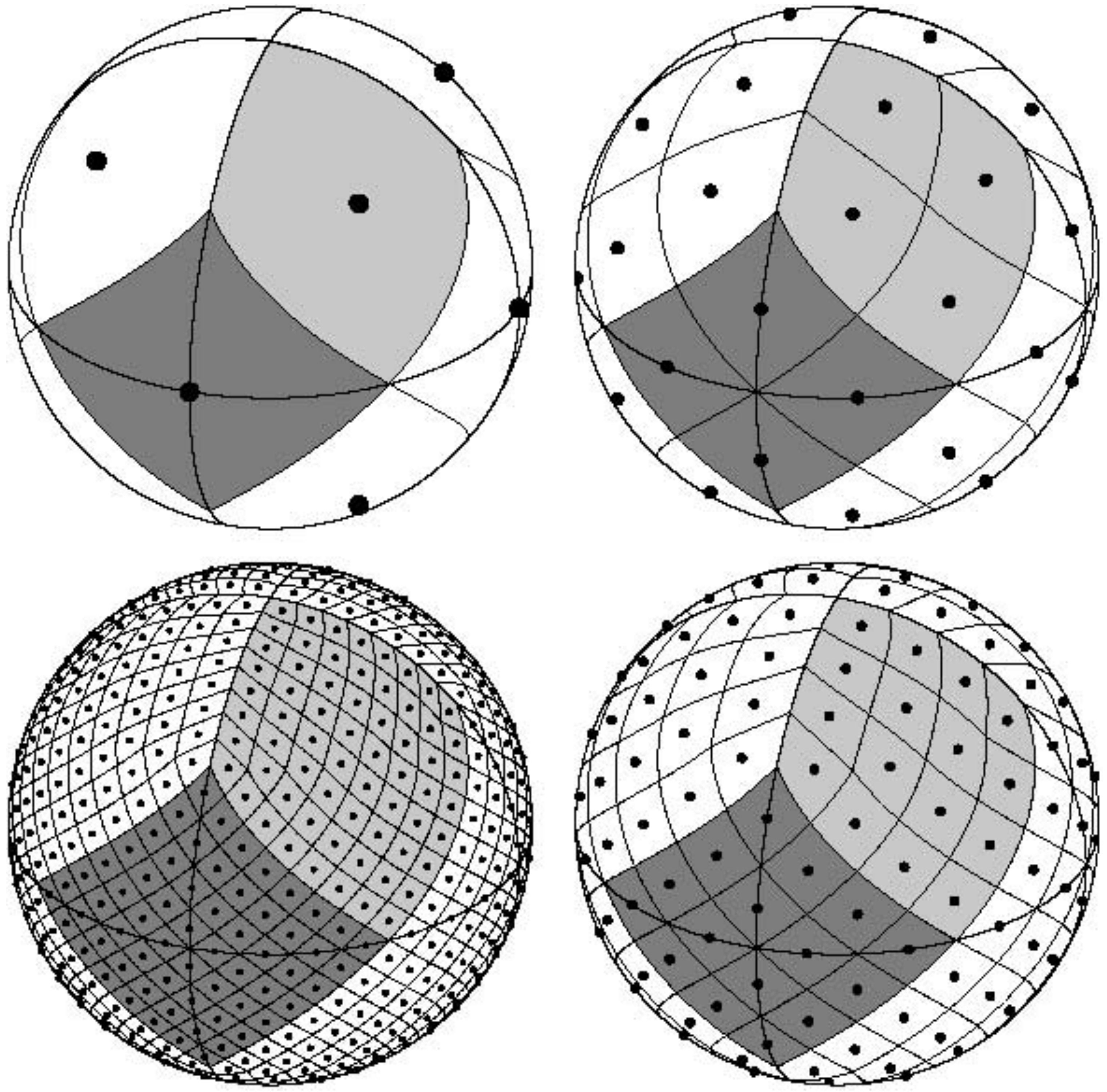
This started being a minor inconvenience in O2 with GW170817. It will get slowly worse as we approach design sensitivity. It’s already a major pain if you are studying future detector networks with simulations.

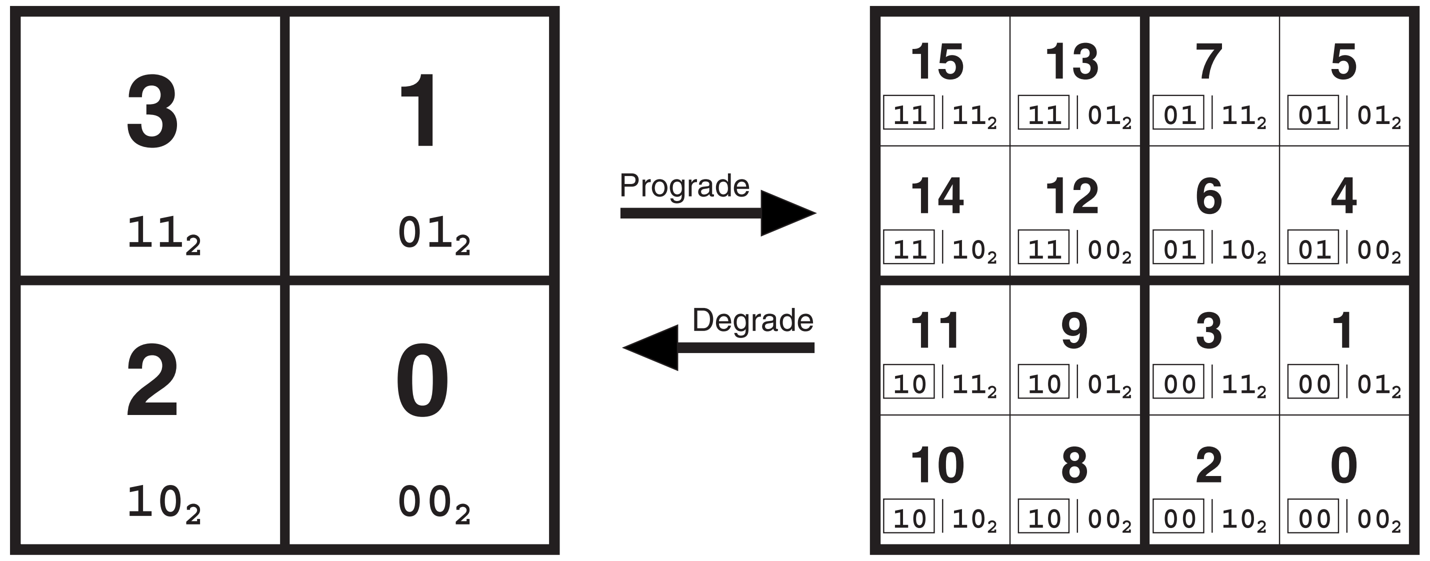
HEALPix primer

Hierarchical **E**qual **A**rea iso**L**atitude **P**ixelization: data structure for storing data that lives on the unit sphere

- Hierarchical: each tile contains four higher-resolution daughter tiles, forming a tree structure
- Equal Area: all tiles of a given resolution have the same area and approximately the same shape
- isoLatitude: tiles are arranged on rings of constant latitude to permit fast convolution using spherical harmonics
- Pixelization: good at storing images

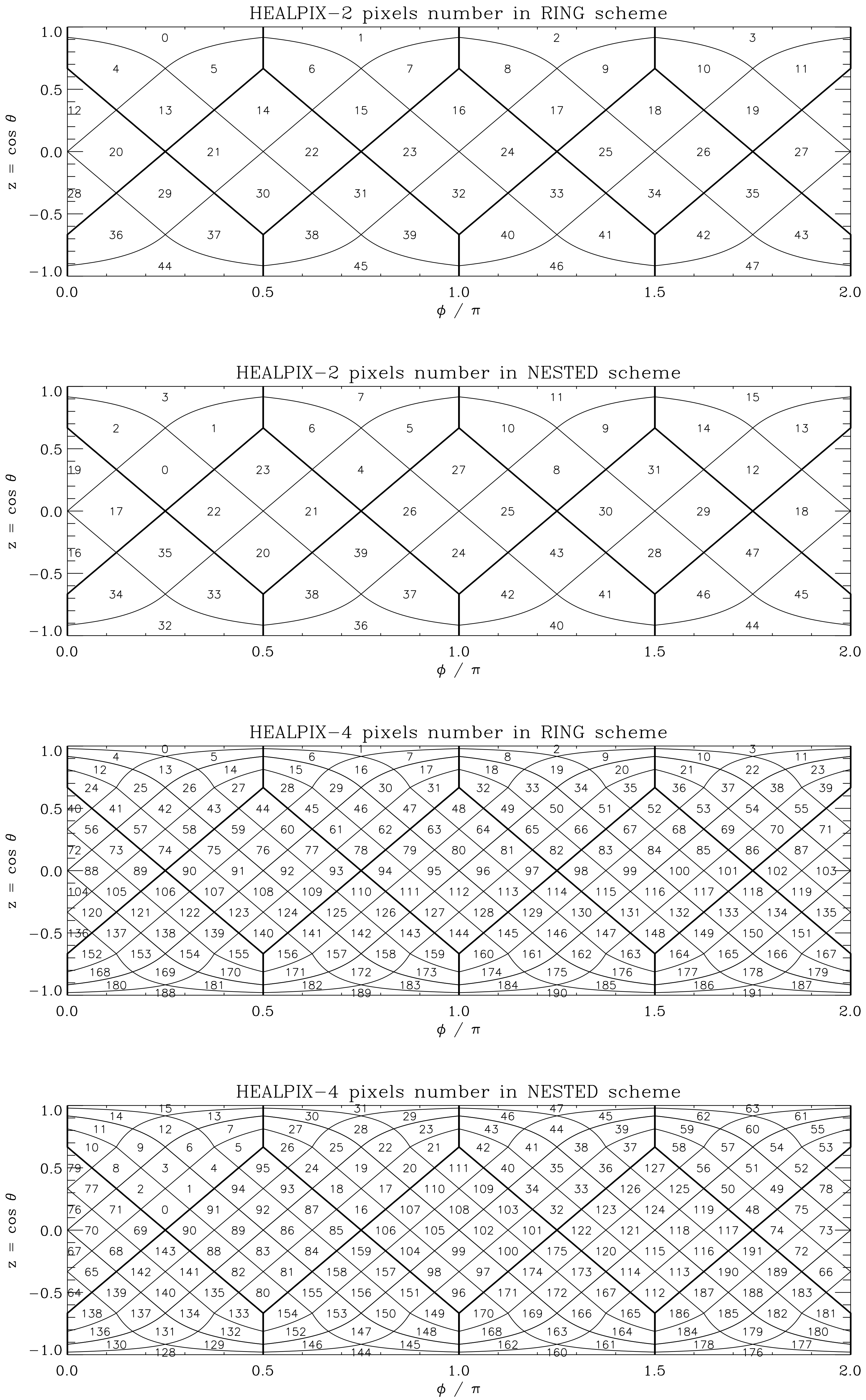
Górski+ 2005





3 11_2	**1** 01_2
2 10_2	**0** 00_2

15 $\boxed{11}\,11_2$	**13** $\boxed{11}\,01_2$	**7** $\boxed{01}\,11_2$	**5** $\boxed{01}\,01_2$
14 $\boxed{11}\,10_2$	**12** $\boxed{11}\,00_2$	**6** $\boxed{01}\,10_2$	**4** $\boxed{01}\,00_2$
11 $\boxed{10}\,11_2$	**9** $\boxed{10}\,01_2$	**3** $\boxed{00}\,11_2$	**1** $\boxed{00}\,01_2$
10 $\boxed{10}\,10_2$	**8** $\boxed{10}\,00_2$	**2** $\boxed{00}\,10_2$	**0** $\boxed{00}\,00_2$


HEALPIX−2 pixels number in RING scheme
HEALPIX−2 pixels number in NESTED scheme
HEALPIX−4 pixels number in RING scheme
HEALPIX−4 pixels number in NESTED scheme
z = cos θ
φ / π
1.0
0.5
0.0
−0.5
−1.0
0.0
0.5
1.0
1.5
2.0

Górski+ 2005

HEALPix pixel indexing

The index or address of a HEALPix tile consists of three pieces of information:

- Resolution (*nside*): lateral number of subdivisions along the twelve base-level tiles. At any resolution, there are a total of *npix* = (12 *nside* ^{2}) pixels.
- Pixel index (*ipix*): pixel number from 0 to (*npix* - 1).
- Ordering scheme (*order*): **RING** or **NESTED**. In the **RING** scheme, pixel numbers advance in right ascension and then declination. In the **NESTED** scheme, pixel indices follow the hierarchical structure described on the previous slide.

Why do we use HEALPix for LIGO/Virgo probability maps?

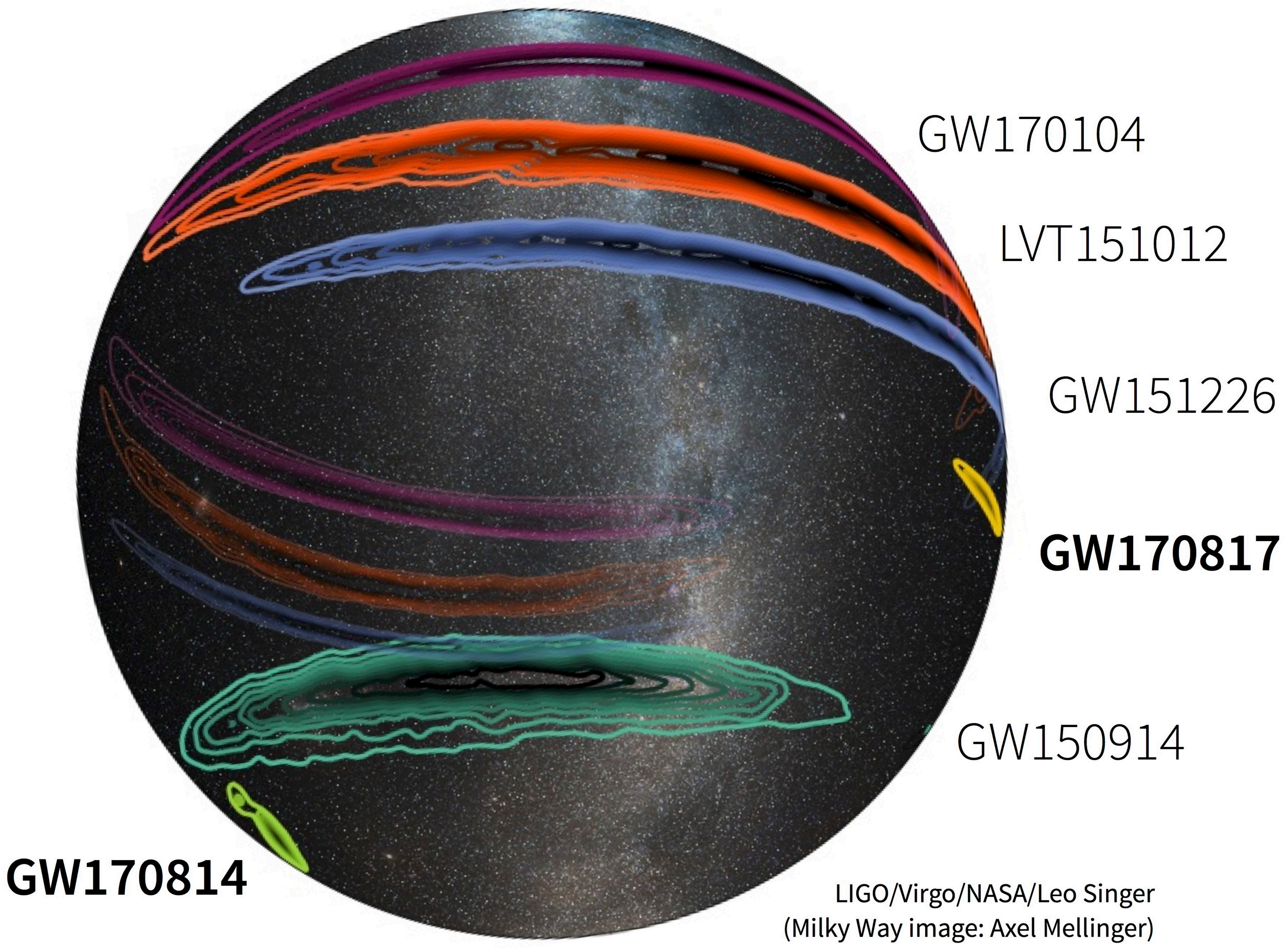
- LIGO/Virgo localizations can
  - subtend large angles
  - wrap around the whole sky
  - have multiple widely separated modes
  - have irregular shapes, fringes

- Difficult to pick a good partial-sky projection (e.g. gnomonic, orthographic) in the general case

- Traditional all-sky projections have wild variations in pixel size (e.g. *plate carée*) or shape (e.g. Mollweide, Aitoff) and as well as seams

- HEALPix was already well-established for specialized uses in astronomy (CMB, full-sky mosaics)

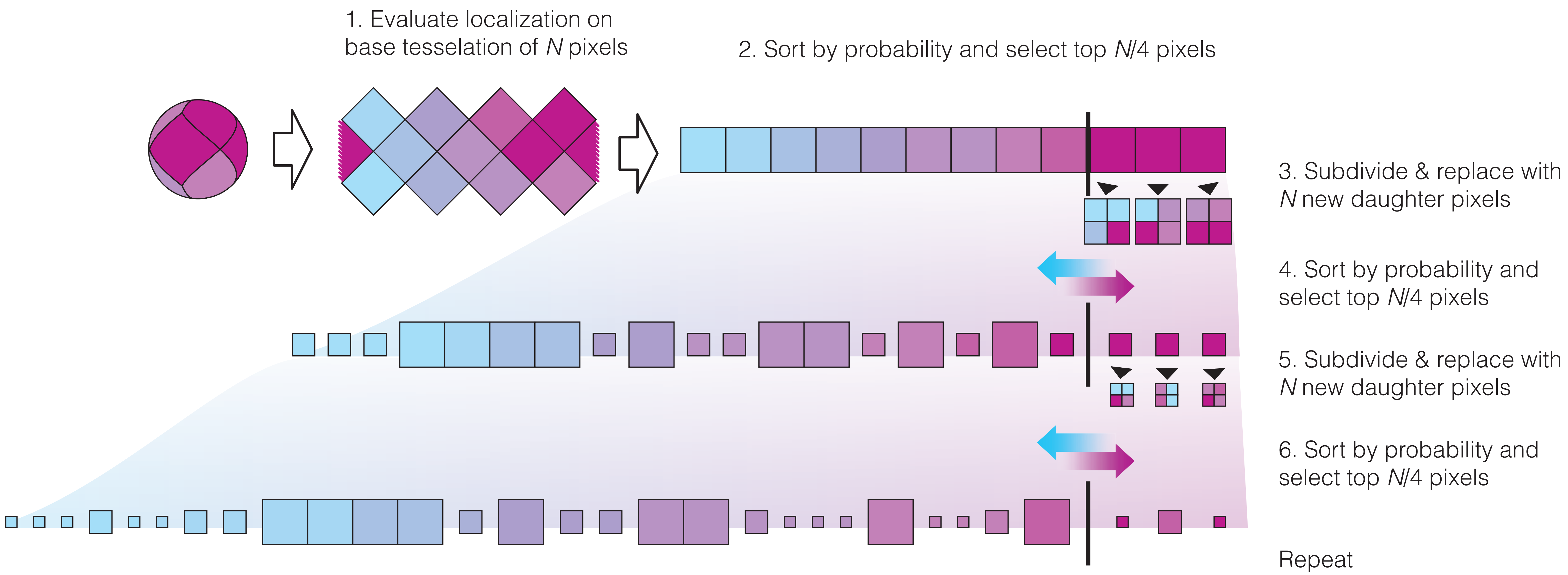
- Good support in software (e.g. DS9, Aladin) and libraries (C, C++, Python, Java, MATLAB, IDL, etc.)



LIGO/Virgo/NASA/Leo Singer
(Milky Way image: Axel Mellinger)

Adaptive subdivision

- Most LIGO/Virgo sky maps (BAYESTAR, LALInference, LIB) are generated using this adaptive subdivision sampling scheme.
- First, the probability is calculated at all sky positions at a resolution of *nside*=32 (≈13 deg^2 / tile, total of 3072 tiles).
- Then, we take the 768 highest probability tiles, and subdivide them into 3072 new tiles.
- The last step is repeated 7 times.
- This is a simple feedback control system that drives the pixels to have comparable probability: e.g. smaller pixels in regions of higher probability density.



1. Evaluate localization on base tesselation of *N* pixels

2. Sort by probability and select top *N*/4 pixels

3. Subdivide & replace with *N* new daughter pixels

4. Sort by probability and select top *N*/4 pixels

5. Subdivide & replace with *N* new daughter pixels

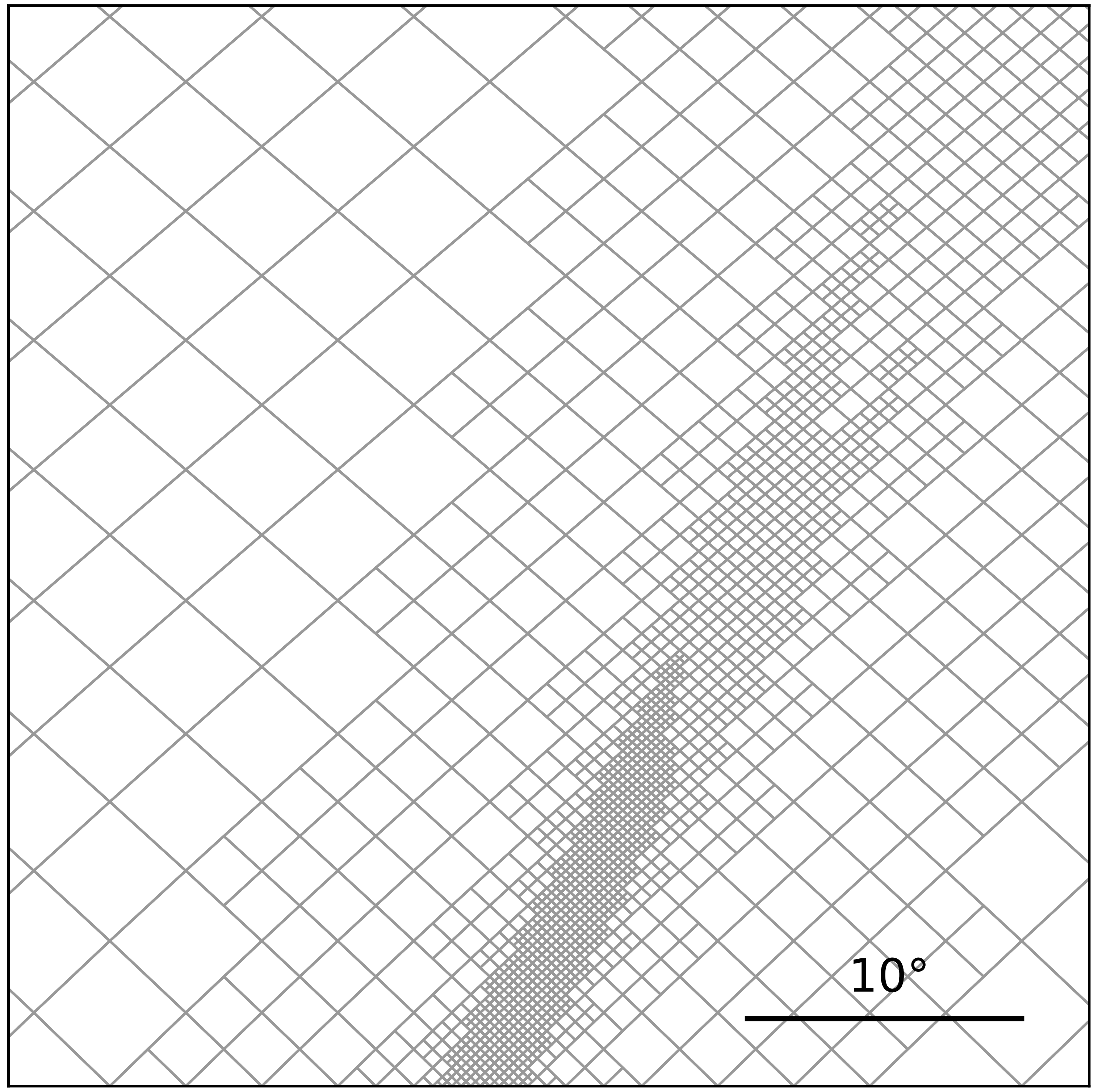
6. Sort by probability and select top *N*/4 pixels

Repeat

Singer + Price 2016

The resulting HEALPix tree contains exactly 19200 tiles.
This gets flattened out to a HEALPix image with the resolution of the smallest tile.
The lowest possible resolution is *nside*=128, ≈0.2 deg^2 / pixel, ≈200k pixels.
The highest possible resolution is *nside*=2048, ≈3 arcmin^2 / pixel, ≈50M pixels.

Adaptive subdivision: benefits



Singer + Price 2016

- Expensive per-pixel calculations are quickly focused toward regions where extra detail is needed
- Work can be done in large parallel batches (of 3072 pixels)
- Resulting sky maps are detailed but can be gzip-compressed with high compression ratios because of the long runs of identical pixel values
- Codes that are aware of the adaptive subdivision scheme can accelerate other expensive calculations massively (galaxy completeness integrals, volume rendering)

Issues for well-localized events

- For well-localized events like GW170817, the adaptive subdivision can proceed to the highest possible resolution, *nside*=2048.

- The gzip-compressed FITS files are still very small on disk (~1 MB) because the HEALPix trees still have only ~20k tiles.

- However, the uncompressed FITS files can get very large in memory, ~1.5 GB, because they are stored at high resolution.

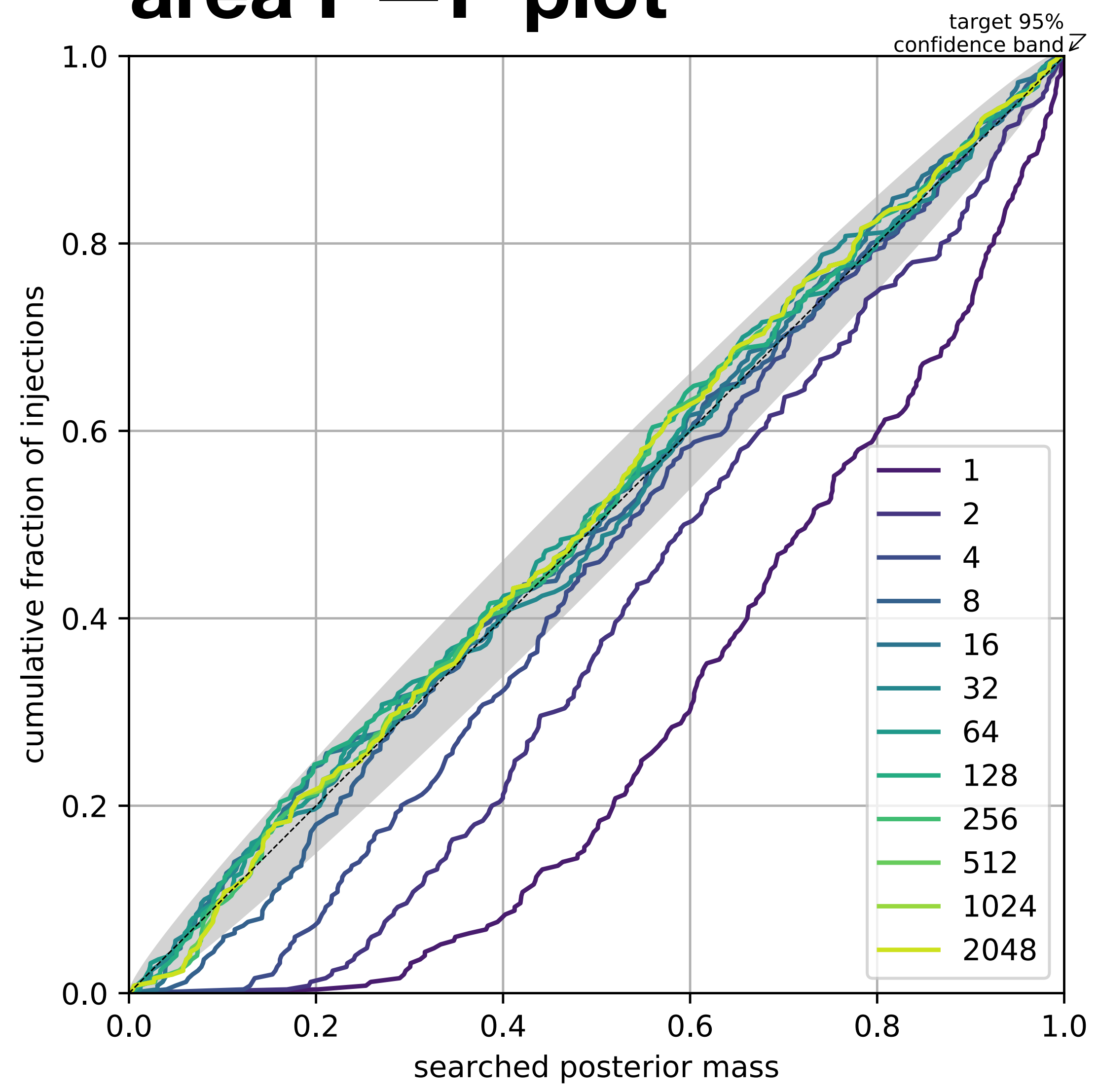
- Just decompressing the sky maps to read them in can take a few seconds, and they can be cumbersome to deal with in memory.

Proposal 1: reduce maximum resolution

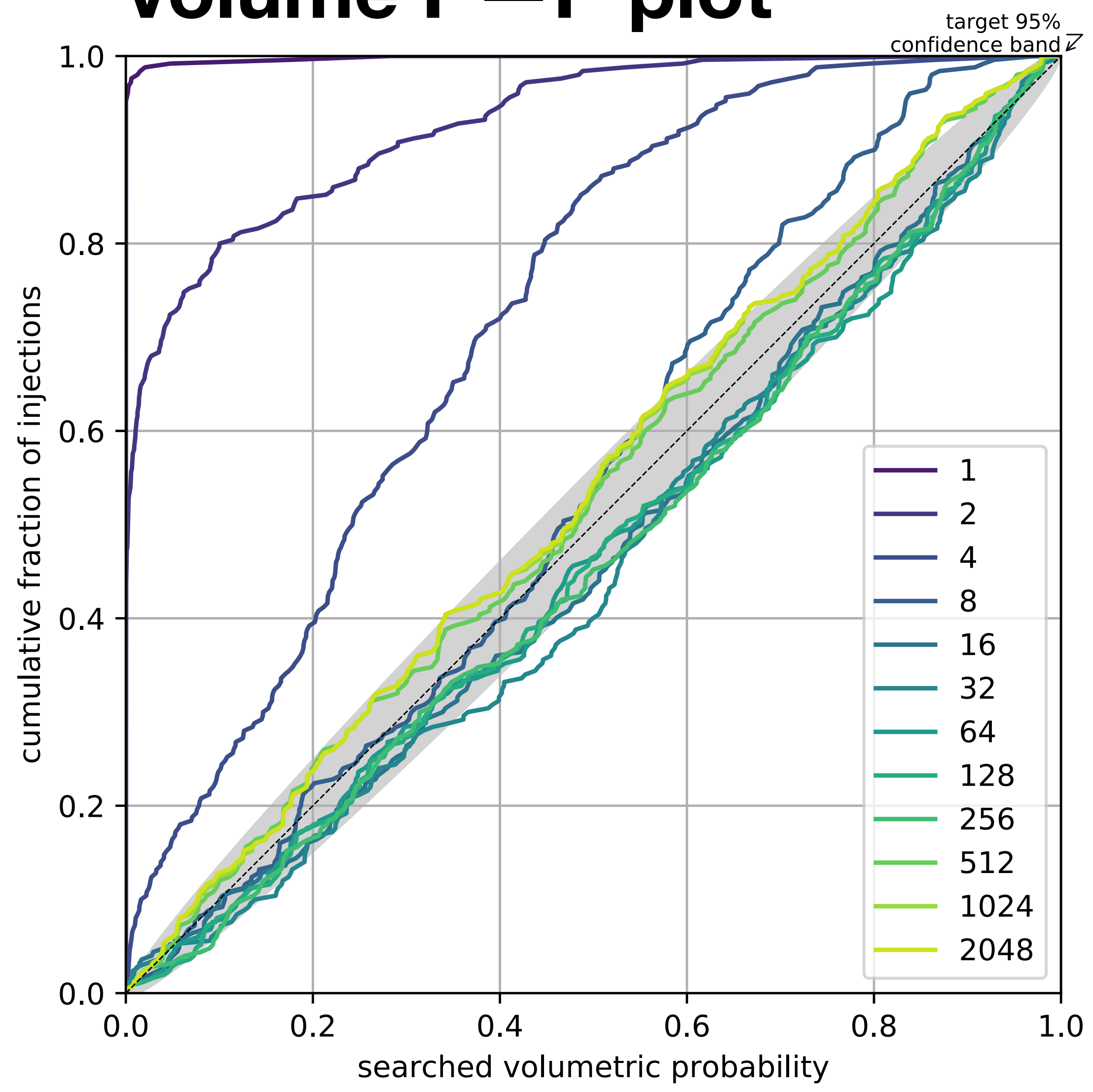
- Proposal: reduce maximum resolution by decreasing the maximum number of subdivisions by 2 steps.
- Reduces the maximum resolution to *nside*=512, 0.01 deg^2 / pixel.
- Reduces the maximum in-memory size of sky maps to ~100 MB.
- Impact on area and volume shown on next slide.

first2years / 2015

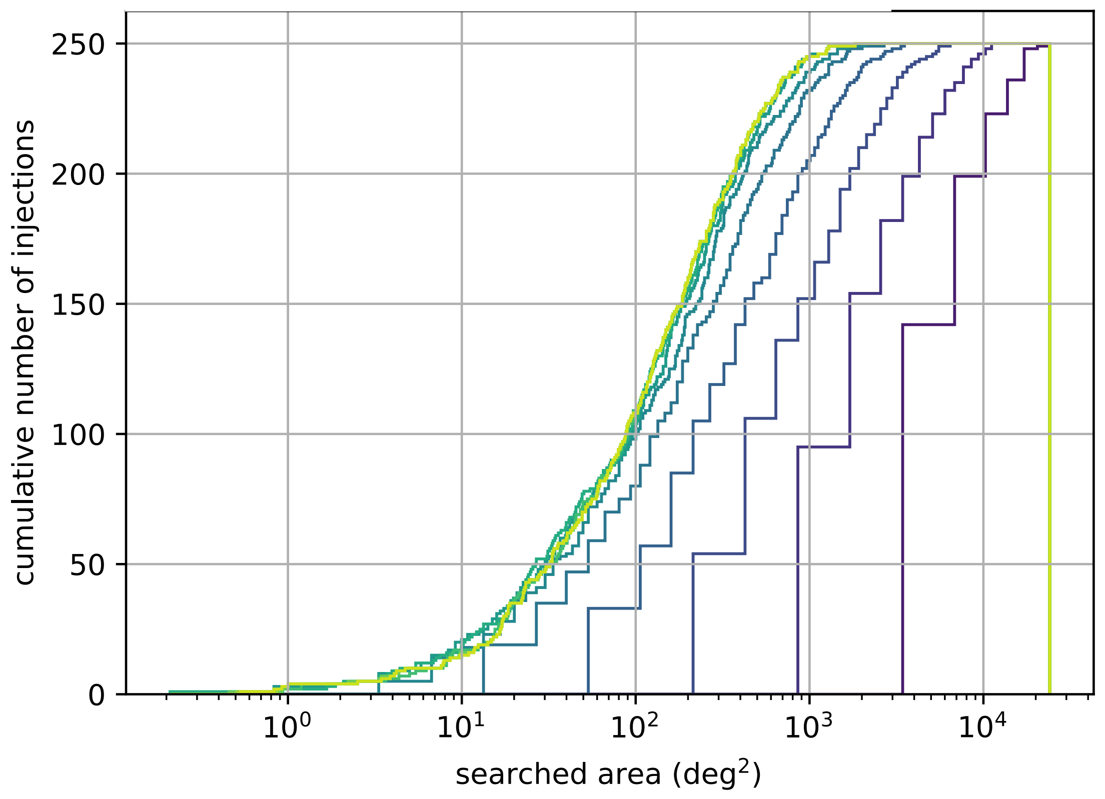
area P−P plot



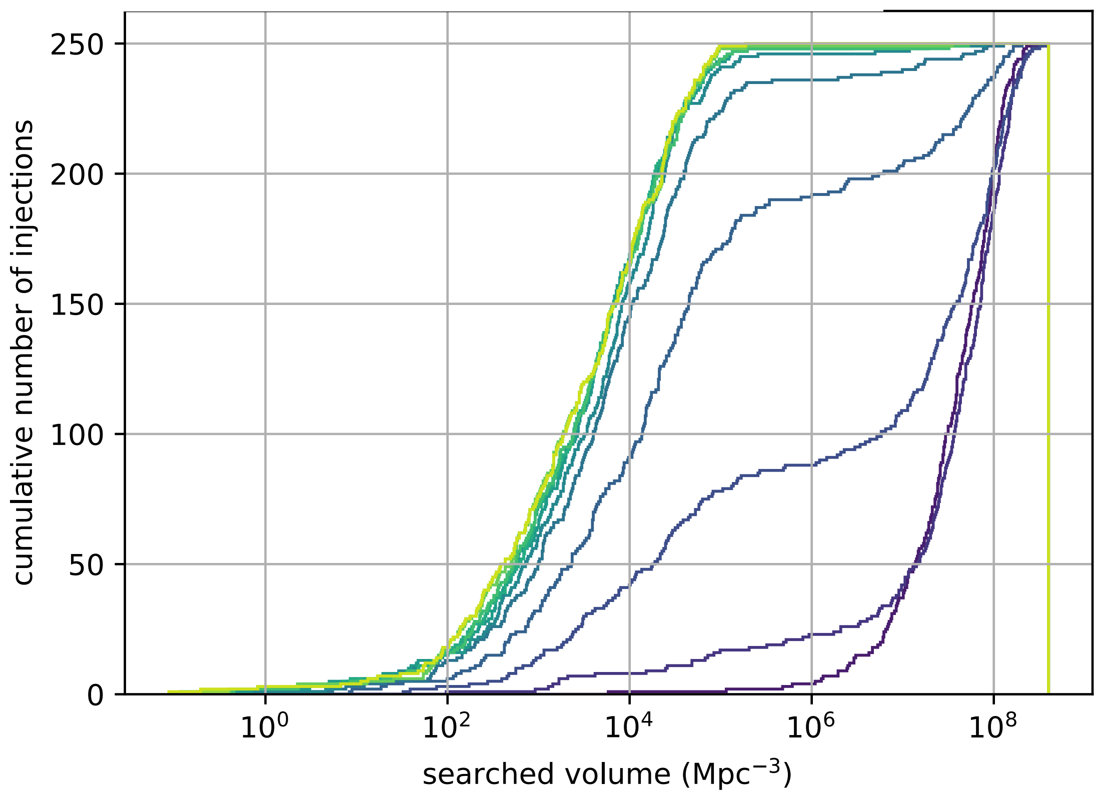
volume P−P plot



area histogram



volume histogram



- Based on “First Two Years” mock data challenge sample (Singer+ 2014, Berry+ 2015, Farr+ 2016, Singer+ 2016)

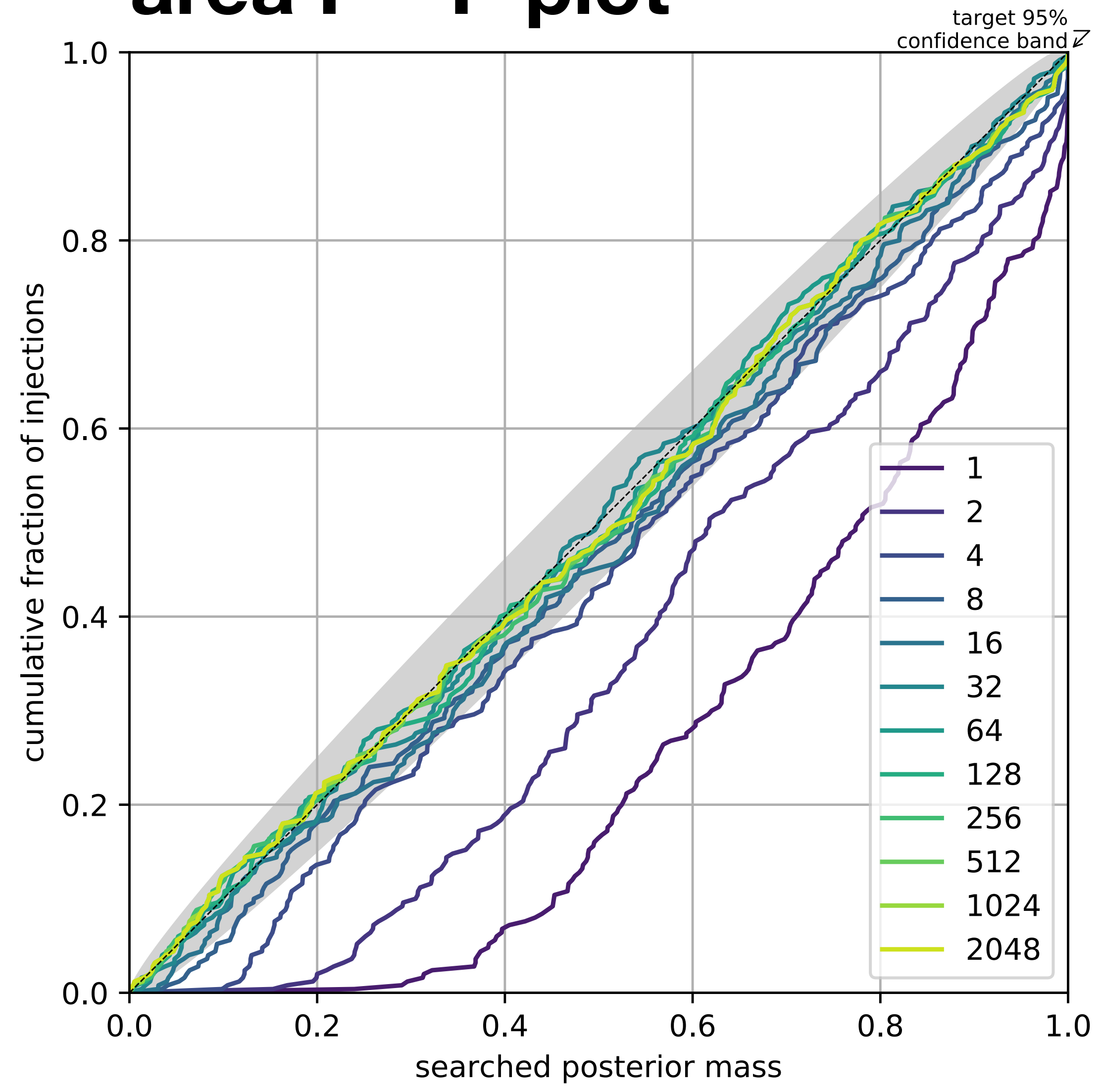
- Downsampled events and to a maximum resolution of *nside*=1 to 1024 (maximum original resolution is 2048)

- Downsampling approximates the effect that reducing the number of adaptive subdivision steps would have

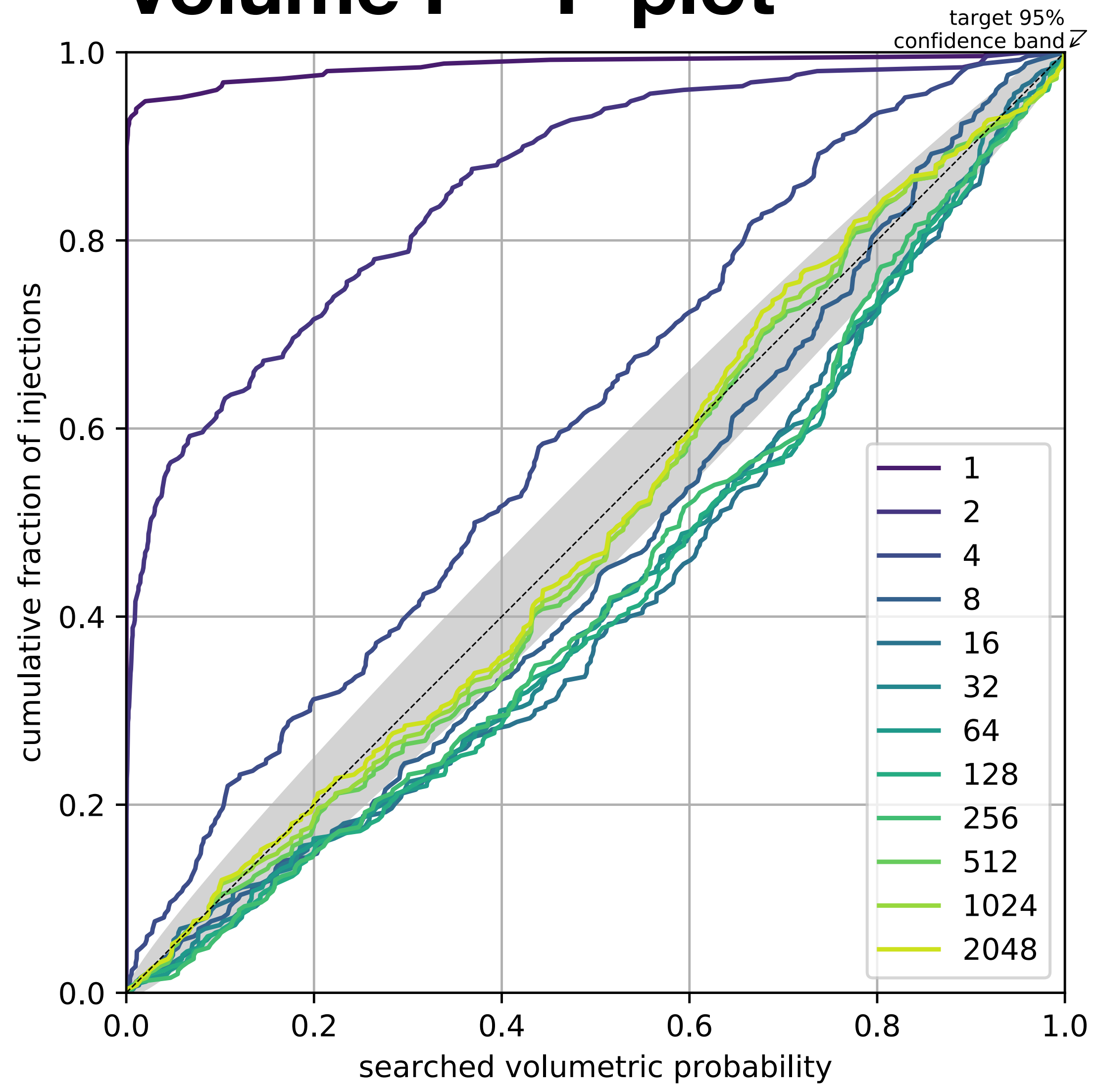
- Looks like reducing the number of adaptive subdivision steps by 1 or 2 would be safe

first2years / 2016

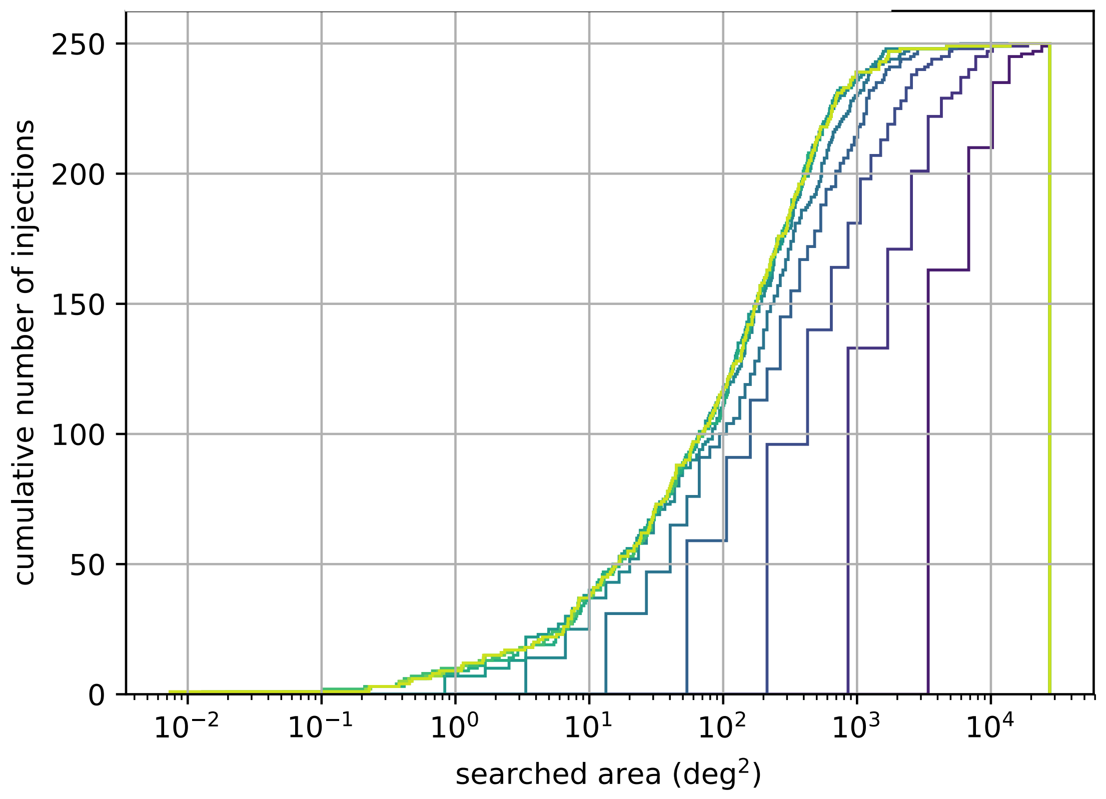
area P−P plot



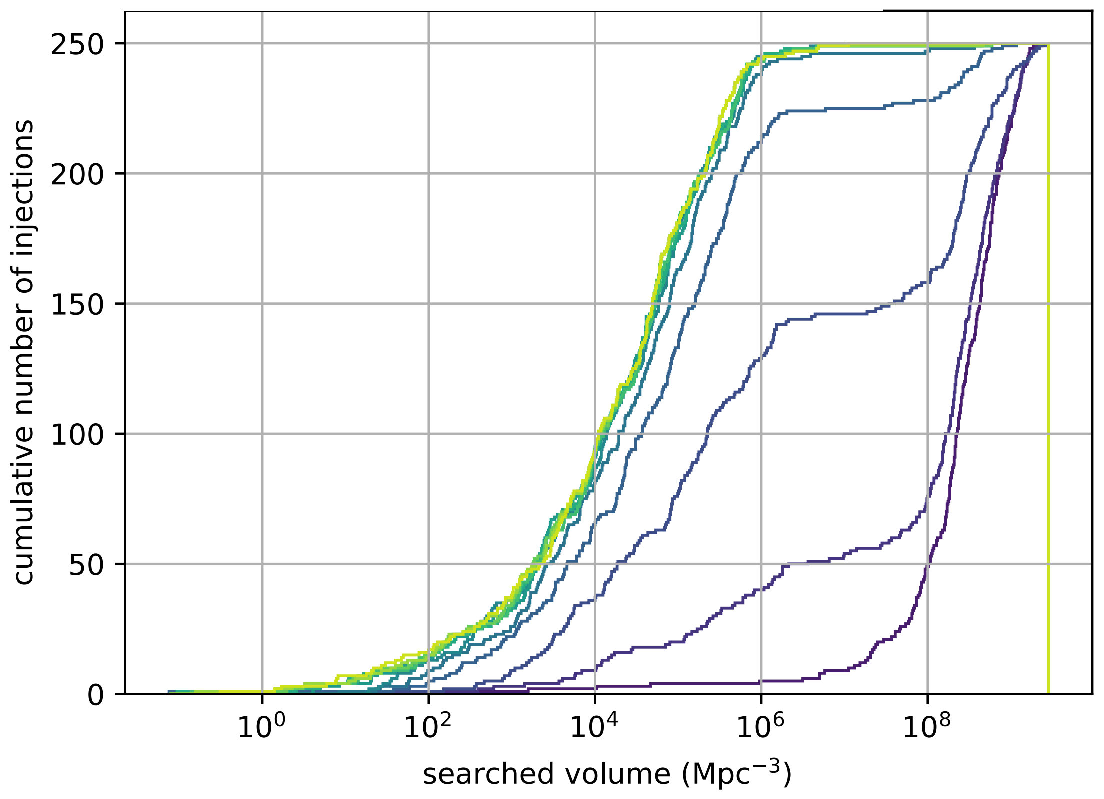
volume P−P plot



area histogram



volume histogram



- Based on “First Two Years” mock data challenge sample (Singer+ 2014, Berry+ 2015, Farr+ 2016, Singer+ 2016)

- Downsampled events and to a maximum resolution of *nside*=1 to 1024 (maximum original resolution is 2048)

- Downsampling approximates the effect that reducing the number of adaptive subdivision steps would have

- Looks like reducing the number of adaptive subdivision steps by 1 or 2 would be safe

Proposal 2: publish multi-resolution HEALPix trees

- Make experimental multi-resolution HEALPix files available *along side* the traditional FITS files.

- No standard data format exists. However, there is related prior art in the Virtual Observatory:

  - Multi-Order Coverage (MOC) maps — Boch+ 2014

  - HEALPix multi-resolution data structures — Reinecke + Hivon 2015

  - Hierarchical Progressive Surveys (HiPS) — Fernique+ 2017

  - UNIQ indexing scheme — Górski+ 2017, section 3.2

UNIQ indexing scheme

- Recall from before that three pieces of information are required to specify a HEALPix tile: *nside*, *ipix*, and *order*.

- There is a third HEALPix indexing scheme called **UNIQ**. The **UNIQ** ordering assigns a single unique integer to every HEALPix tile at every resolution. If *ipix* is the pixel index in the **NESTED** ordering, then the unique pixel index *uniq* is

$$uniq = ipix + 4\ nside^{2}.$$

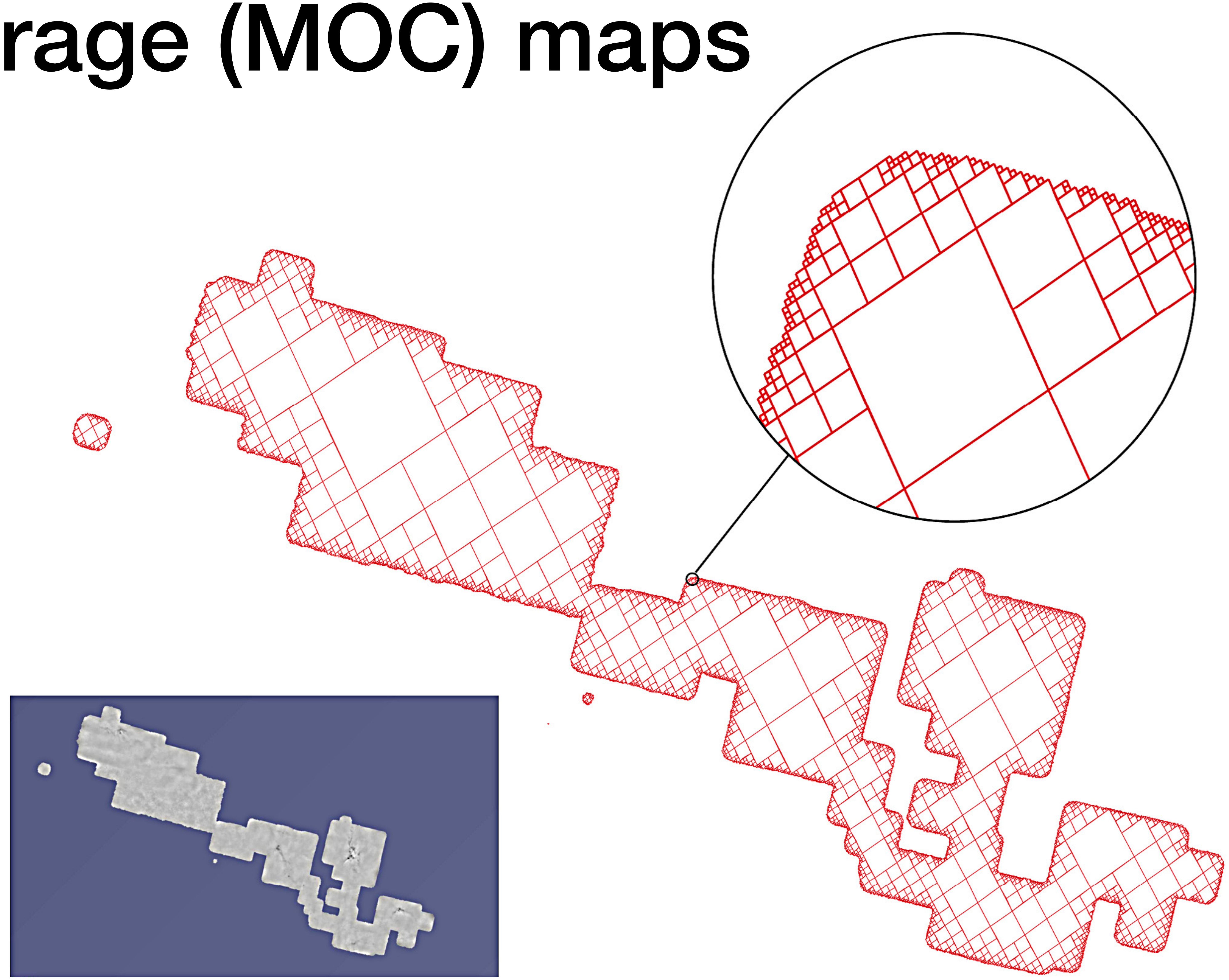
The inverse is

$$nside = 2^{\mathrm{floor}(\log_2(uniq/4)/2)}$$
$$ipix = uniq - 4\ nside^{2}.$$

Multi-Order Coverage (MOC) maps

- Used by Virtual Observatory (e.g. Aladin and related tools) to store survey footprint shapes

- Stored as a list of **UNIQ** HEALPix indices in a FITS binary table



Boch+ 2014, Fernique+ 2015

Hierarchical Progressive Surveys (HiPS)

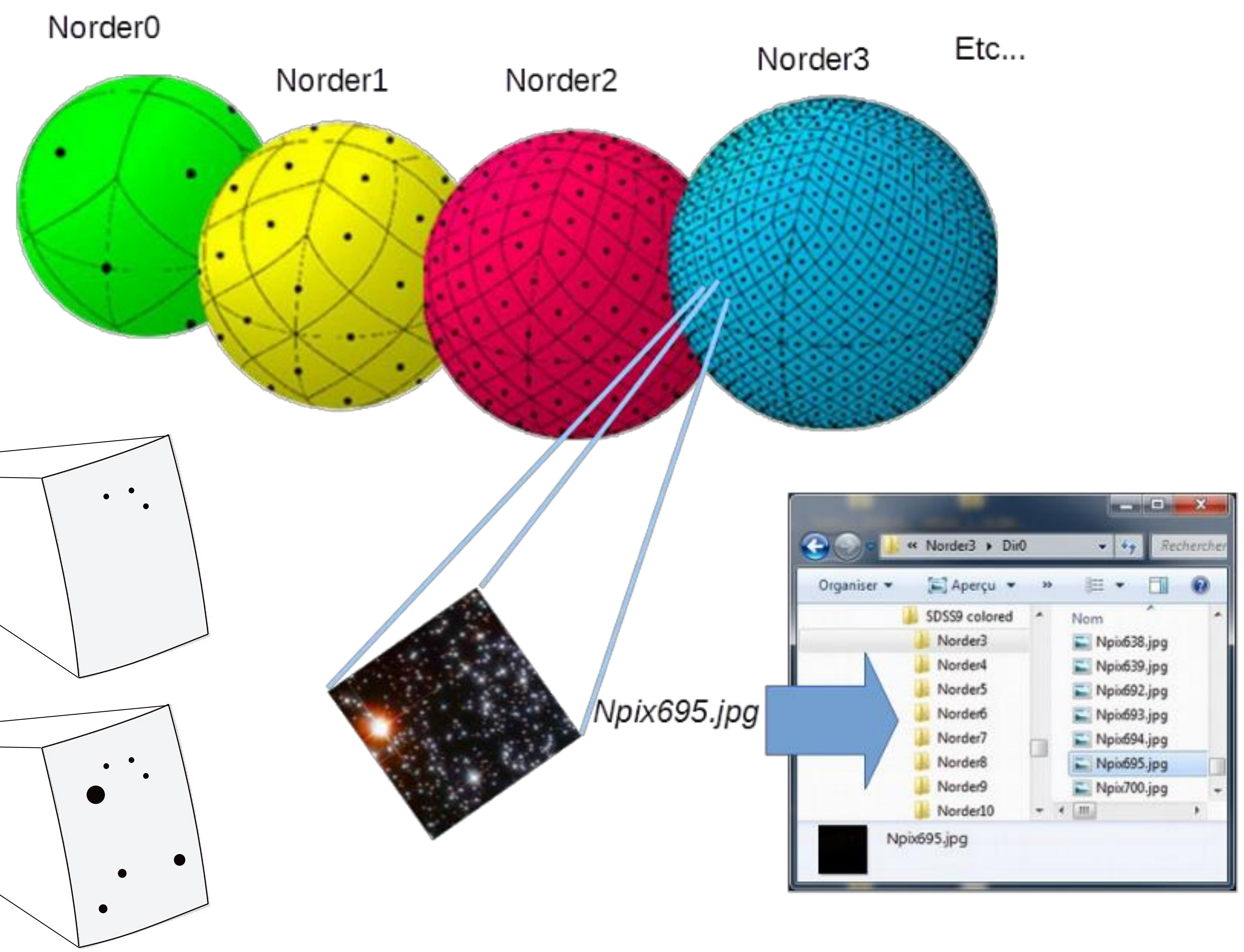
- Used by VO tools (e.g. Aladin) for storing all-sky imaging data; supporting deep zooming

- Consists of a multi-order coverage map (MOC) and a directory tree of data files containing HEALPix files or HEALPix file fragments

👍 Already well-supported by Aladin

👎 Complicated directory structure makes it hard to use for data analysis

Fernique+ 2015, 2017

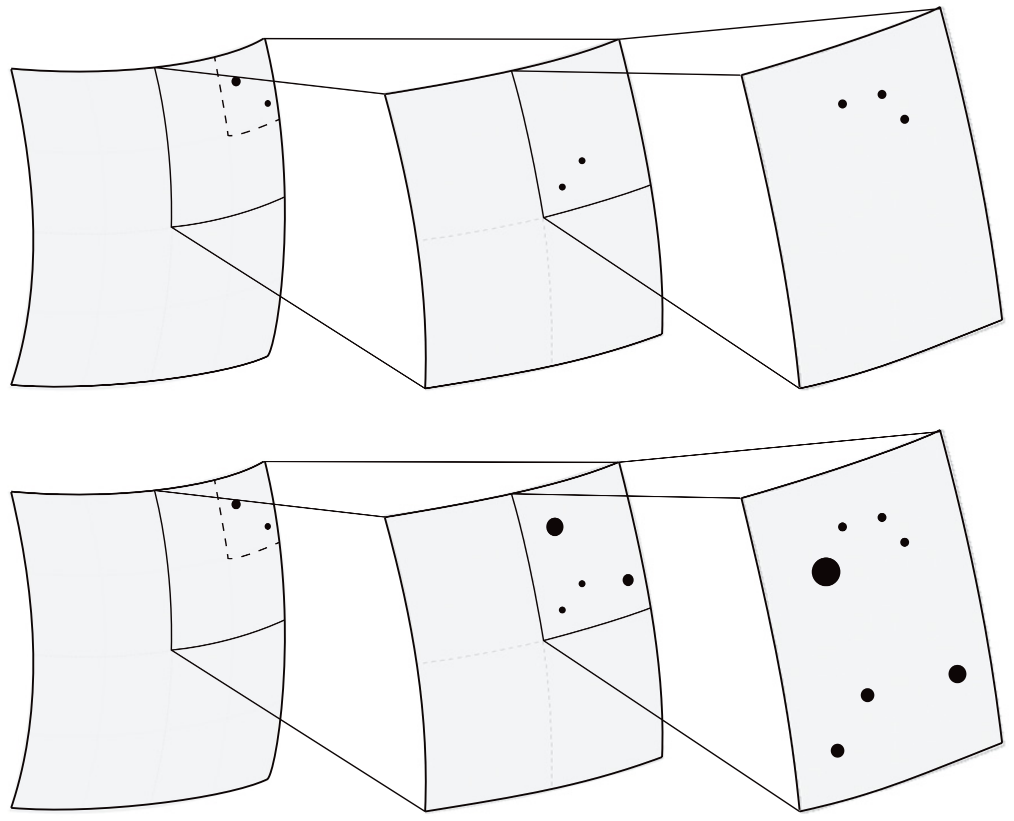


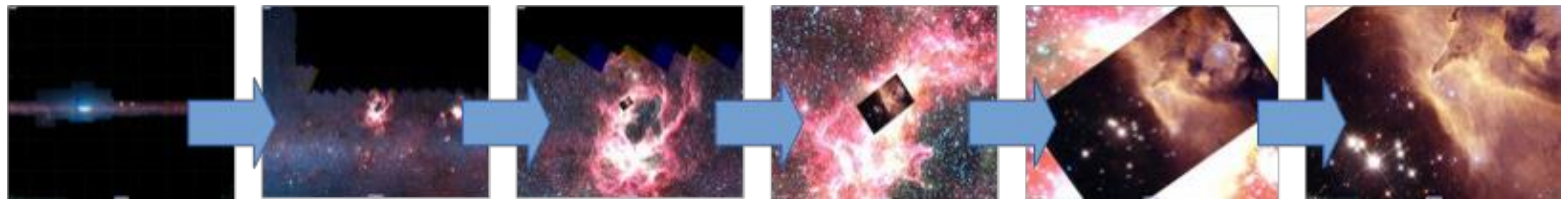
« Norder3 ▸ Dir0

- SDSS9 colored
  - Norder3
  - Norder4
  - Norder5
  - Norder6
  - Norder7
  - Norder8
  - Norder9
  - Norder10

Nom
- Npix638.jpg
- Npix639.jpg
- Npix692.jpg
- Npix693.jpg
- Npix694.jpg
- Npix695.jpg
- Npix700.jpg

Npix695.jpg





Proposal: multi-resolution HEALPix image format

- A superset of the **MOC** format: a table of **UNIQ** pixel indices, but with extra floating point columns for image data.
- In our case, the columns are:
    - **UNIQ** pixel index
    - **PROBDENSITY** probability density per steradian
    - **DISTMU** distance location parameter
    - **DISTSIGMA** distance scale parameter
    - **DISTNORM** distance normalization parameter

FITS header — HEALPix image

This is a minimal FITS header for a LIGO/Virgo sky map in the conventional fixed-resolution image format.

```
TFIELDS =                    2 / number of table fields
TTYPE1  = 'PROB    '
TFORM1  = 'D       '
TUNIT1  = 'pix-1   '
PIXTYPE = 'HEALPIX '           / HEALPix magic code
ORDERING= 'NESTED  '           / NESTED coding method
COORDSYS= 'C       '           / ICRS reference frame
NSIDE   =                 2048 / Resolution parameter of HEALPIX
INDXSCHM= 'IMPLICIT'           / Indexing: IMPLICIT or EXPLICIT
```

FITS header — HEALPix image

This is a minimal FITS header for a multi-order coverage map according to the IVOA document.

```
TFIELDS =                    1 / number of table fields
TFORM1  = 'K       '
TTYPE1  = 'UNIQ    '
PIXTYPE = 'HEALPIX '           / HEALPix magic code
ORDERING= 'NUNIQ   '           / NUNIQ coding method
COORDSYS= 'C       '           / ICRS reference frame
MOCORDER=                   11 / MOC resolution (best order)
```

FITS header — multires image

This is a minimal FITS header for a LIGO/Virgo sky map in the proposed multi-resolution image format.

```
TFIELDS =                    2 / number of table fields
TTYPE1  = 'UNIQ    '
TFORM1  = 'K       '
TTYPE2  = 'PROBDENSITY'
TFORM2  = 'D       '
TUNIT2  = 'sr-1    '
PIXTYPE = 'HEALPIX '           / HEALPix magic code
ORDERING= 'NUNIQ   '           / NUNIQ coding method
COORDSYS= 'C       '           / ICRS reference frame
INDXSCHM= 'EXPLICIT'           / Indexing: IMPLICIT or EXPLICIT
```

Software support

- This format has been used *internally* in BAYESTAR, LALInference, and LIB sky map generation for all of O3 because the **UNIQ** ordering is convenient for the implementation of the adaptive subdivision algorithm.

- The sky maps are converted from this format to NESTED ordering before writing them to disk.

- Transparent support for reading and writing in the ligo.skymap.io.fits Python module: whatever ordering is used on disk, files can be loaded into RING, NESTED, or UNIQ data structures. Likewise, whatever representation is used in memory, HEALPix array can be written in any requested ordering.

- **UNIQ** indexing functions in Python in ligo.skymap.moc Python module.

- My own sky map postprocessing codes (credible areas and 3D volumes, P—P plots) already exploit the **UNIQ** data structure for speed.

Discuss!