

# Feedback on generating very large HiPS for PanSTARRS and HiPS from Gaia DR2 catalogue



Thomas Boch

*IVOA Interop, Victoria, BC, Apps1*



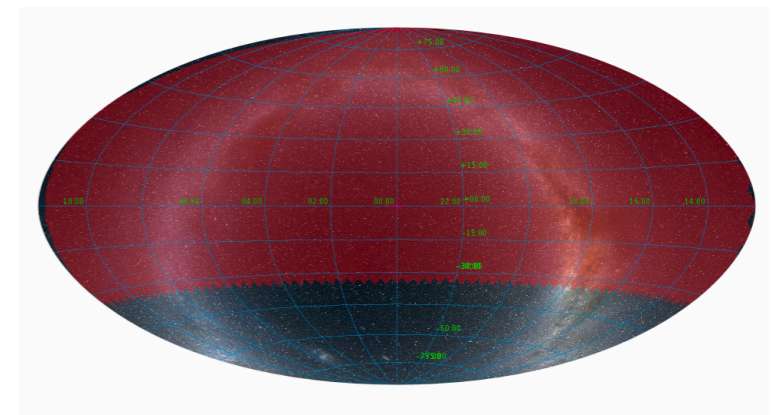
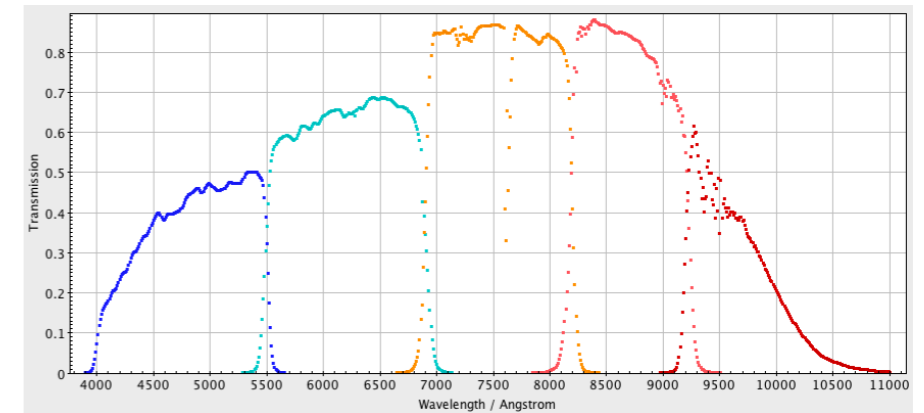
# □ PanSTARRS HiPS (1/4)

- Pan-STARRS PS1 images

- 5 bands: g, r, i, z, y
- coverage: 3/4 of the sky
- *Rice* compressed
- resolution: 0.25"/pixel
- 15 TB per band

- HiPS generation

- resolution: 200 mas (HEALPix order 20)
- with 512x512 tiles: 47 million tiles to be generated
- 10 trillion pixels per band



# □ PanSTARRS HiPS (2/4)

- Pan-STARRS *band g* HiPS creation process
  - download from STScI (Thanks to C. Brasseur et T. Donaldson)
  - uncompress (in parallel)
  - FITS tiles generation (with Hipsgen)
  - JPEG tiles generation (with Hipsgen)
  - transfer on production machine
    - slow: 12 MB/s → 1 TB/day
  - HiPS total size: 25TB
    - splitted across two 15TB disks
    - many symbolic links
      - not convenient
      - error prone

# □ PanSTARRS HiPS (3/4)

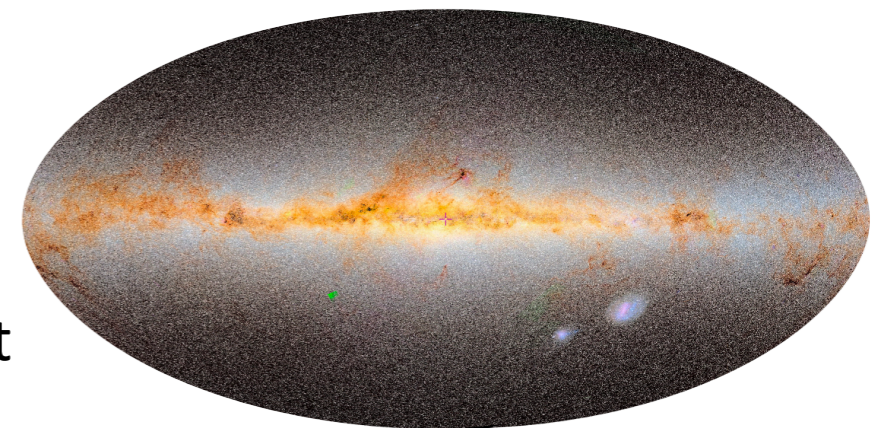
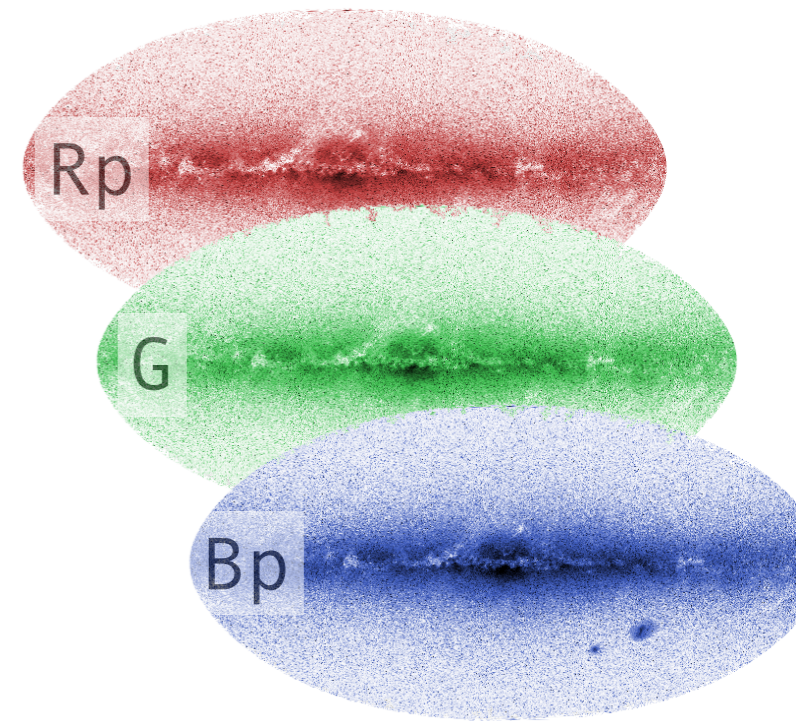
- Improvement for Pan-STARRS *band z* HiPS generation
  - download from STScI
  - direct generation of FITS tiles from *Hipsgen*
    - no longer need to uncompress Rice FITS images
  - **parsync** for file transfer - parallel rsync wrapper
    - 8 to 12-fold improvement: 80 to 120 MB/
  - transferred on one 25TB disk
  - JPEG tiles created on the fly
    - Python service
    - Apache rewrite rule
    - under test and evaluation

# □ PanSTARRS HiPS (4/4)

- PanSTARRS color HiPS
  - $z$   $0.5(z+g)$   $g$
  - tiles generated by Python script
    - Lupton-like *arcsinh* stretch to maximize contrast
    - different parameters at small and large scales
  - no control on JPEG quality from Python script
    - convert command-line tool to reprocess and reduce JPEG size
  - demonstration
    - <http://aladin.unistra.fr/AladinLite/showcase/PanSTARRS-DR1/>

# □ Gaia DR2 flux map

- HEALPix map for each band (integrated flux, weighted density map)
  - generated with *healpy* at NSIDE=8192
    - initial tests with Java unsuccessful:  
unable to create a 800 million items float array
  - map converted to HiPS with Hipsgen (order: 4)
  - alternative approach: generate individual G from TAP request
    - ```
SELECT HEALPIX(ra, dec, 5+9) AS ipix,  
SUM(phot_g_mean_flux) as g, SUM(phot_bp_mean_flux)  
as b, SUM(phot_rp_mean_flux) as r  
FROM "I/345/gaia2"  
WHERE HEALPIX(ra, dec, 5)= 42  
GROUP BY ipix
```
  - Color tiles
    - G, Rp, Bp bands map to green, red and blue channels
    - Python script with Lupton-like stretch to maximize contrast
  - Demonstration



# □ Gaia DR2 velocity map

- Median velocity computed for each HEALPix pixel
- Hipsgen to generate HiPS from HEALPix map
- Python script to generate JPEG tiles
  - matplotlib *RdBu* (red to blue) color map

