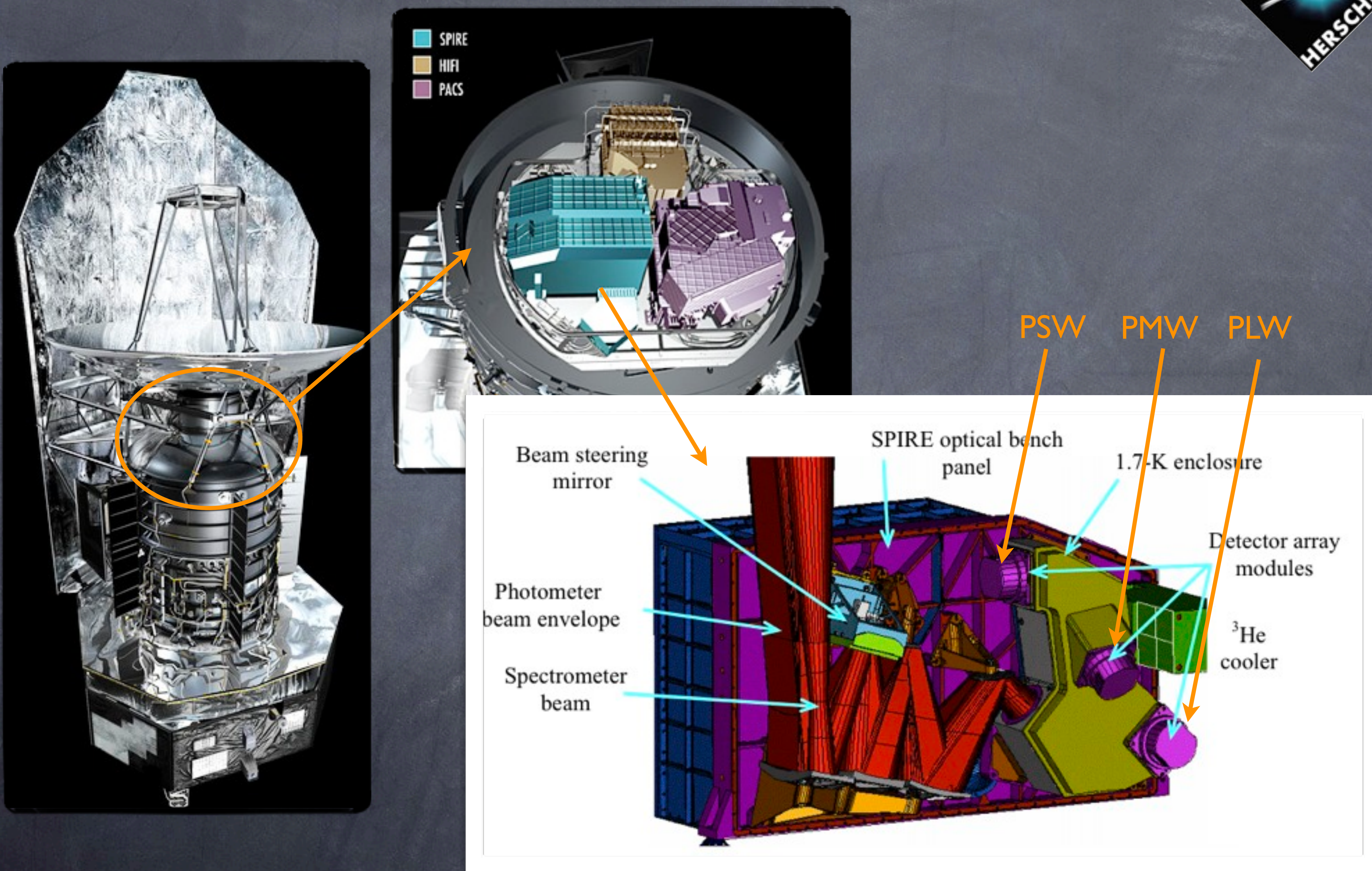
The SPIRE Photometer

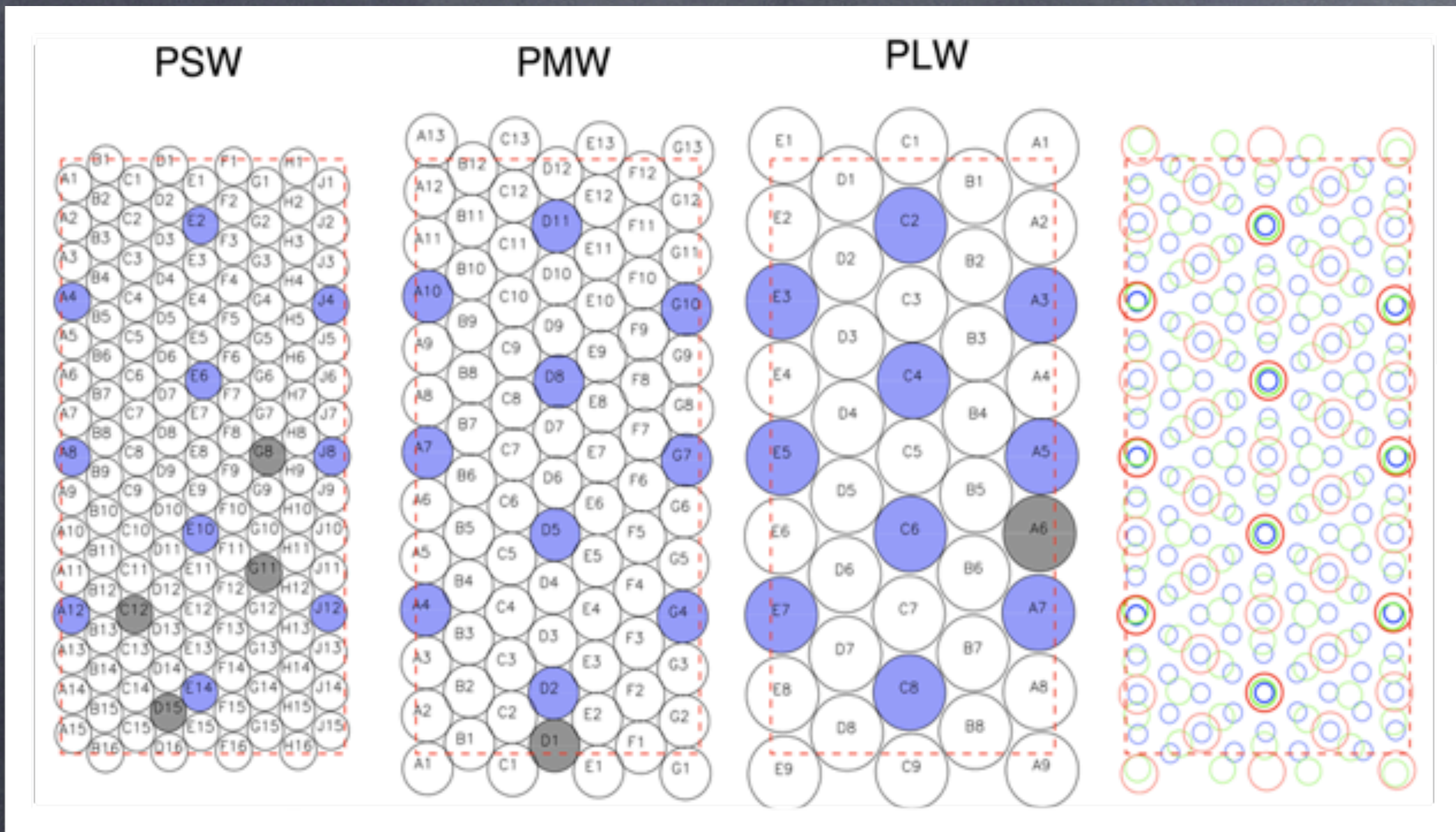
Instrument characteristics & its pipeline

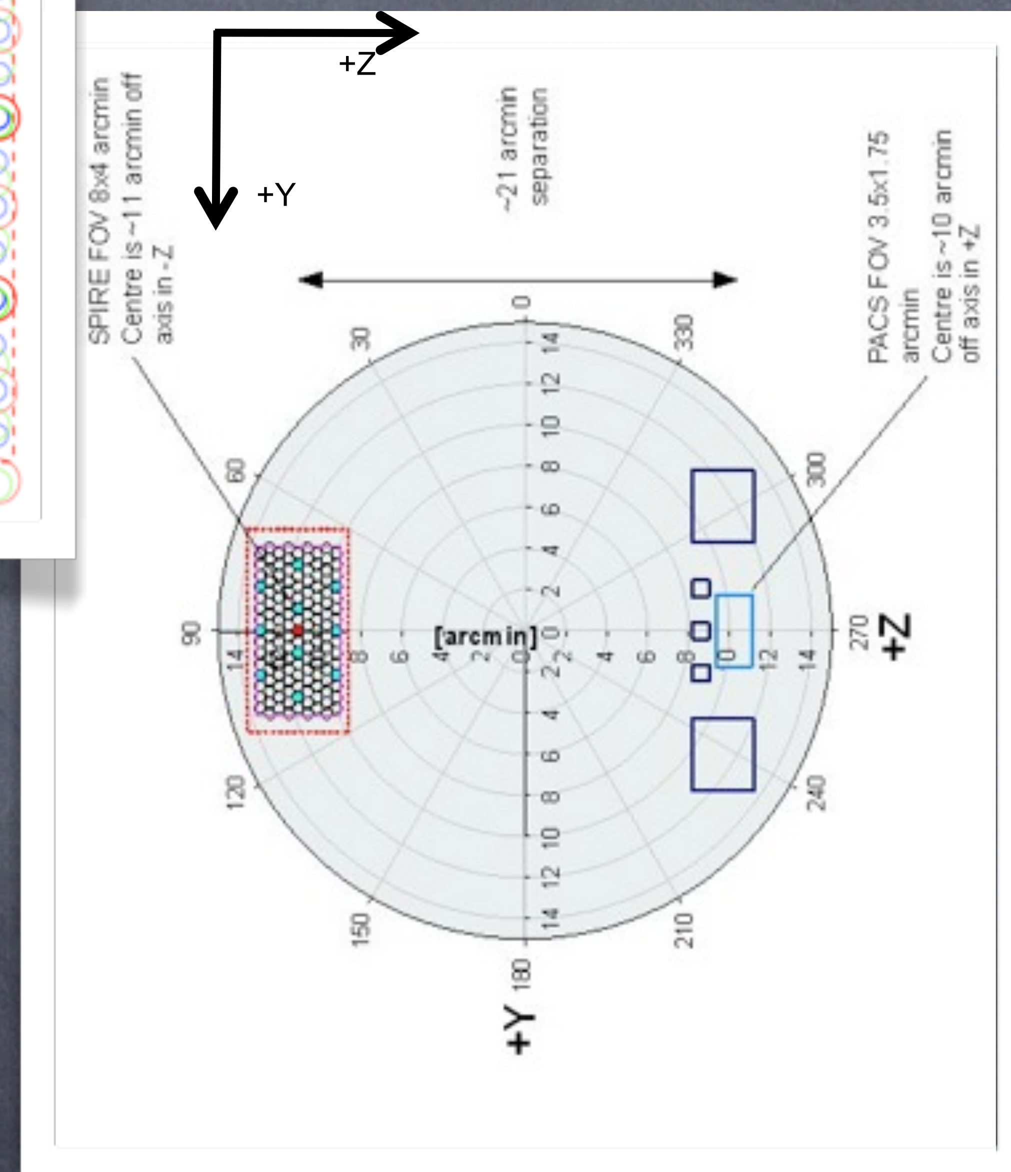
Luca Conversi
on behalf of the SPIRE ICC

The Instrument



SPIRE Detector Arrays



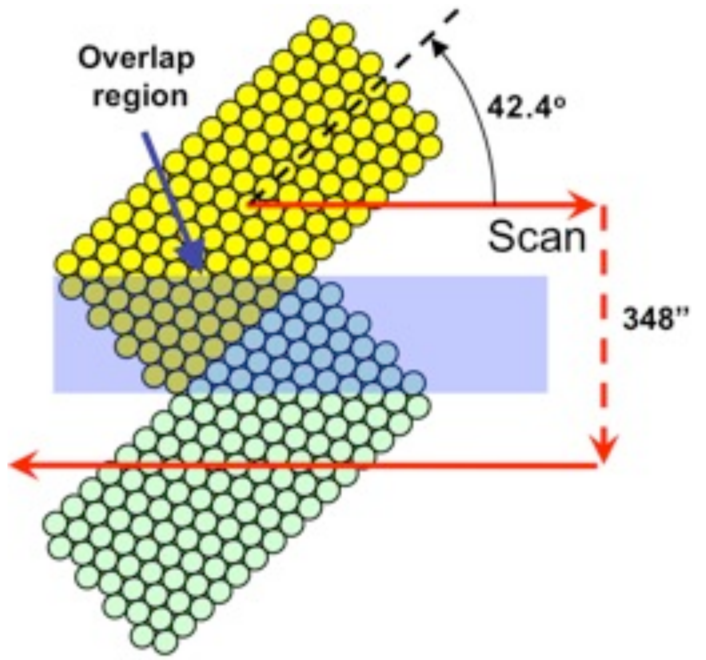


Luca Conversi
Herschel Science Centre

Herschel Map-Making Workshop
ESAC, Madrid, 28-31/01/2013

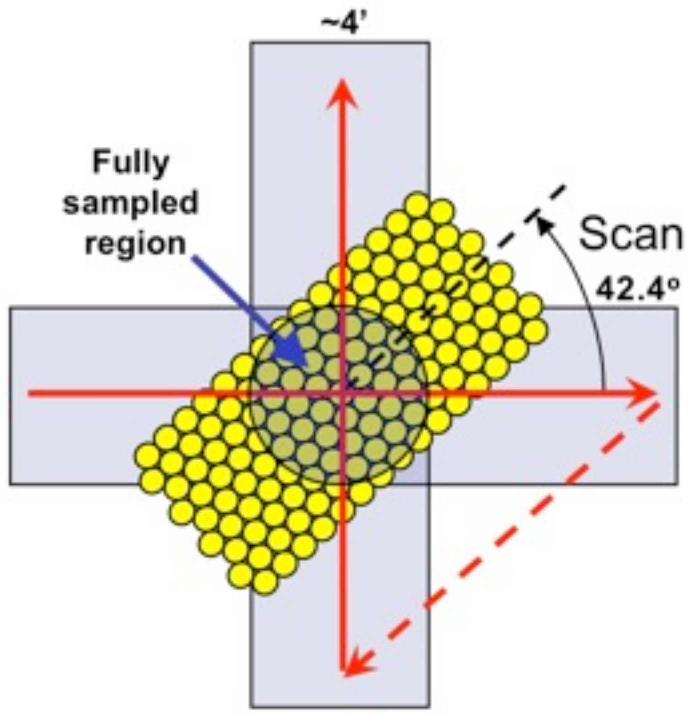
Photometer AOTs

Large (Scan) Map Mode



Scan map at speeds of 30 & 60"/sec is most efficient mode for **large-area surveys**. Parameters are optimized for full spatial sampling and uniform distribution of integration time. Cross scan capability (84.8º)

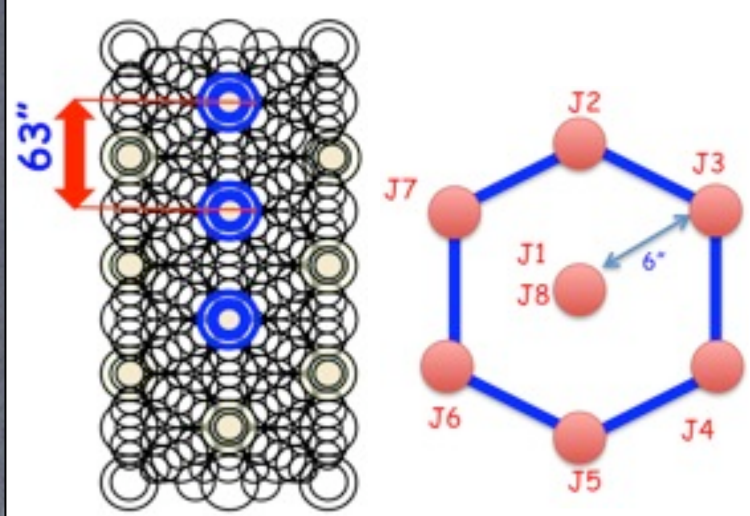
Small Map Mode



Single cross scan at 84.8º replaces 64 point Jiggle map. Scan map at speeds of 30"/sec. Full spatial sampling in center of scans. **Guarenteed coverage is a 5 arcmin circle**

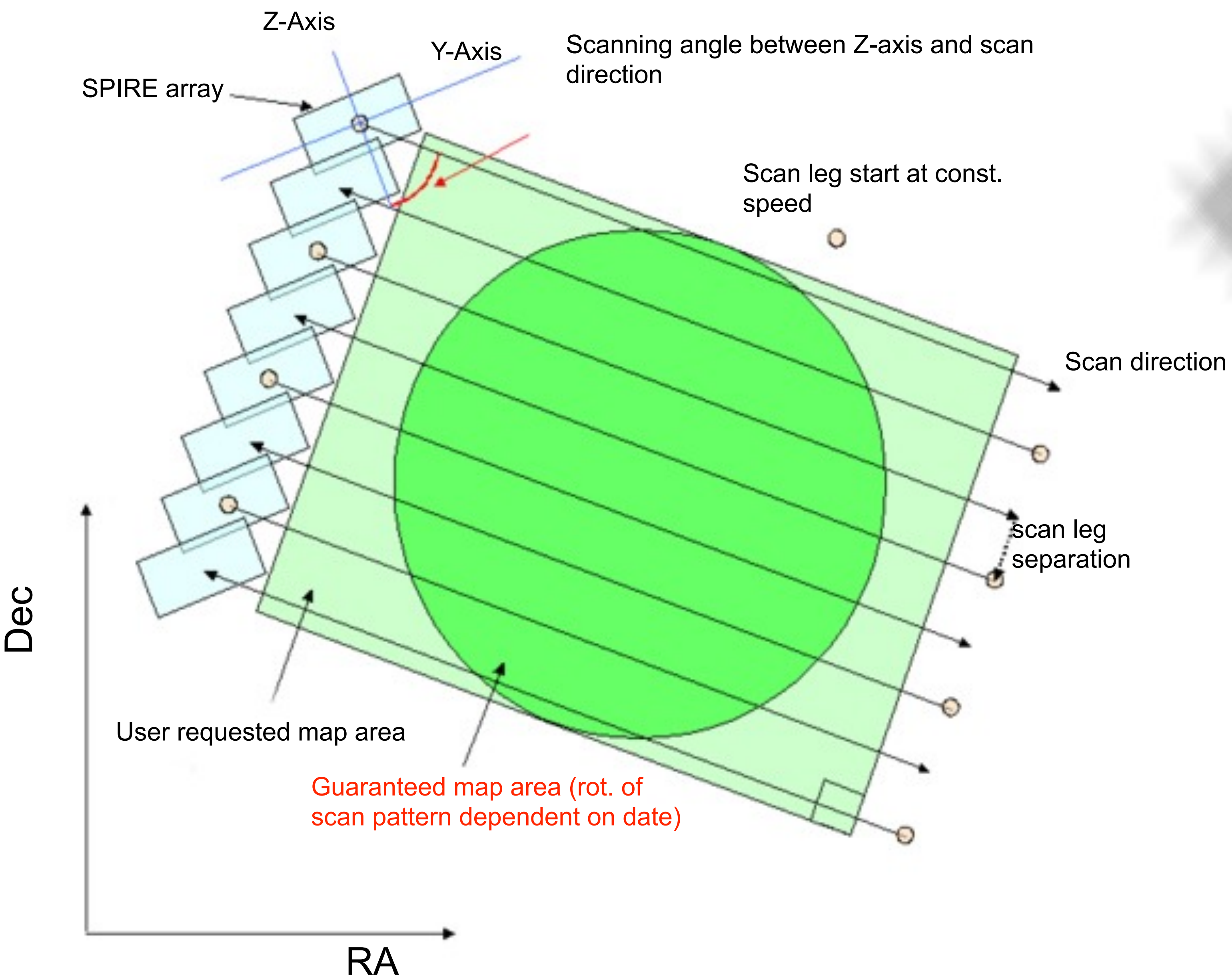
Point Source Mode

63" chop & nod

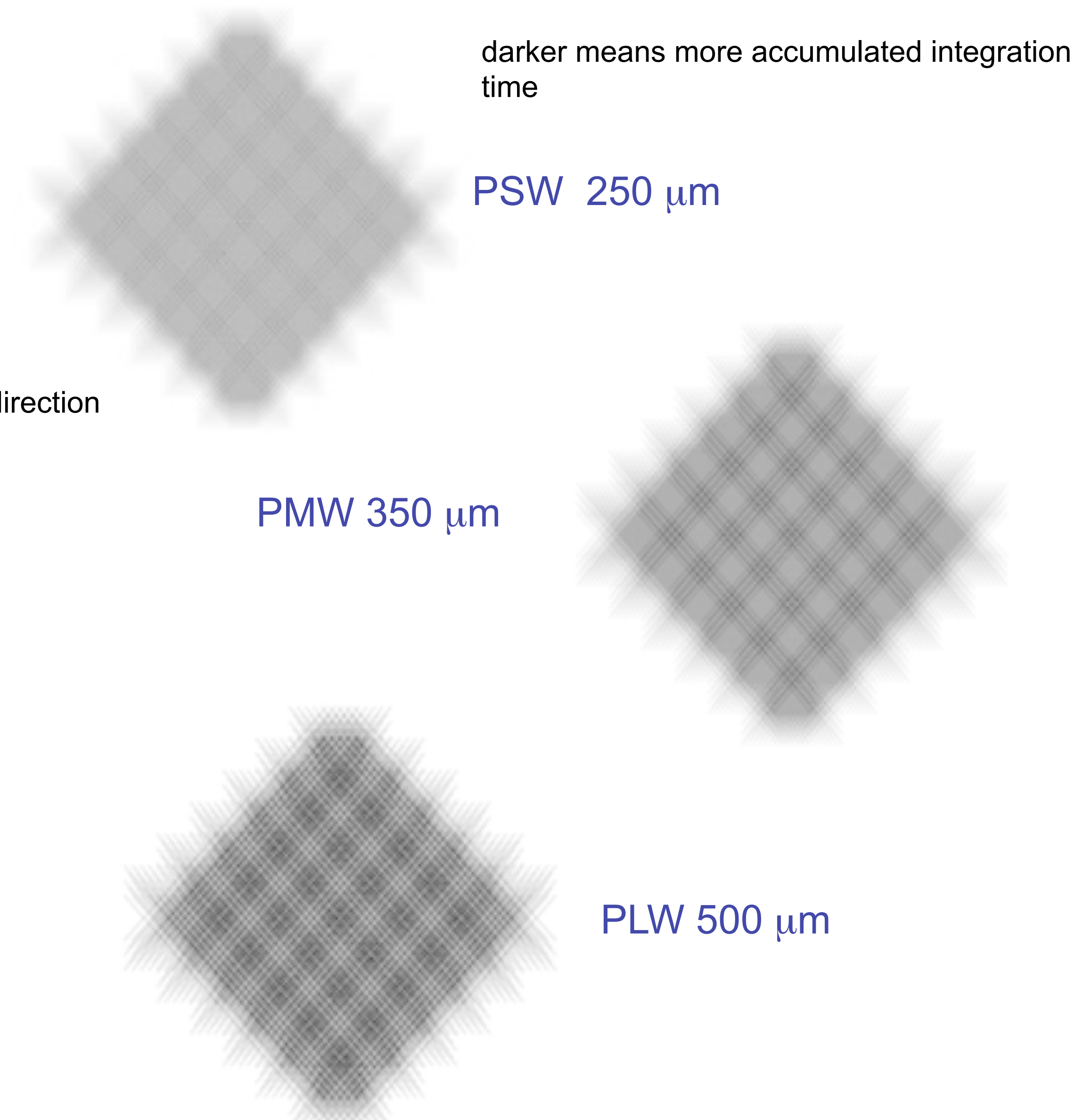


7-point jiggle for **point source photometry**, to compensate pointing error and under-sampling. Chopping and nodding at each jiggle position to remove backgrounds.

Large Map Sky Coverage



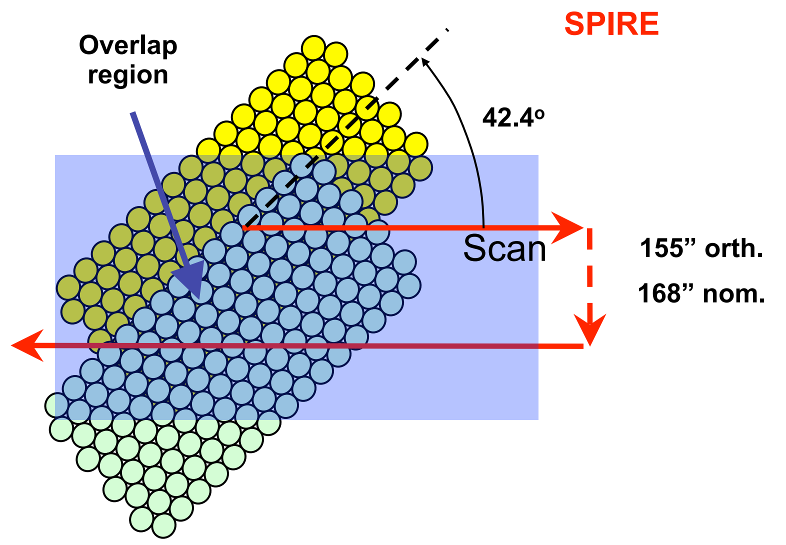
Sky Coverage for Cross Scanned Maps



Parallel Mode

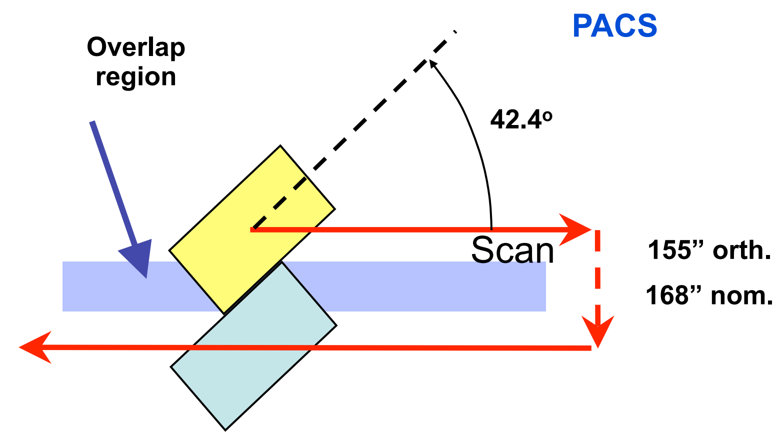
- Scan maps at speeds of 20 and 60"/sec with PACS and SPIRE in parallel are useful for large-area surveys.
- The distance between PACS and SPIRE apertures is 21 arcmin.
- Two almost orthogonal (84.80°) directions for cross scanning are available.

SPIRE Geometry



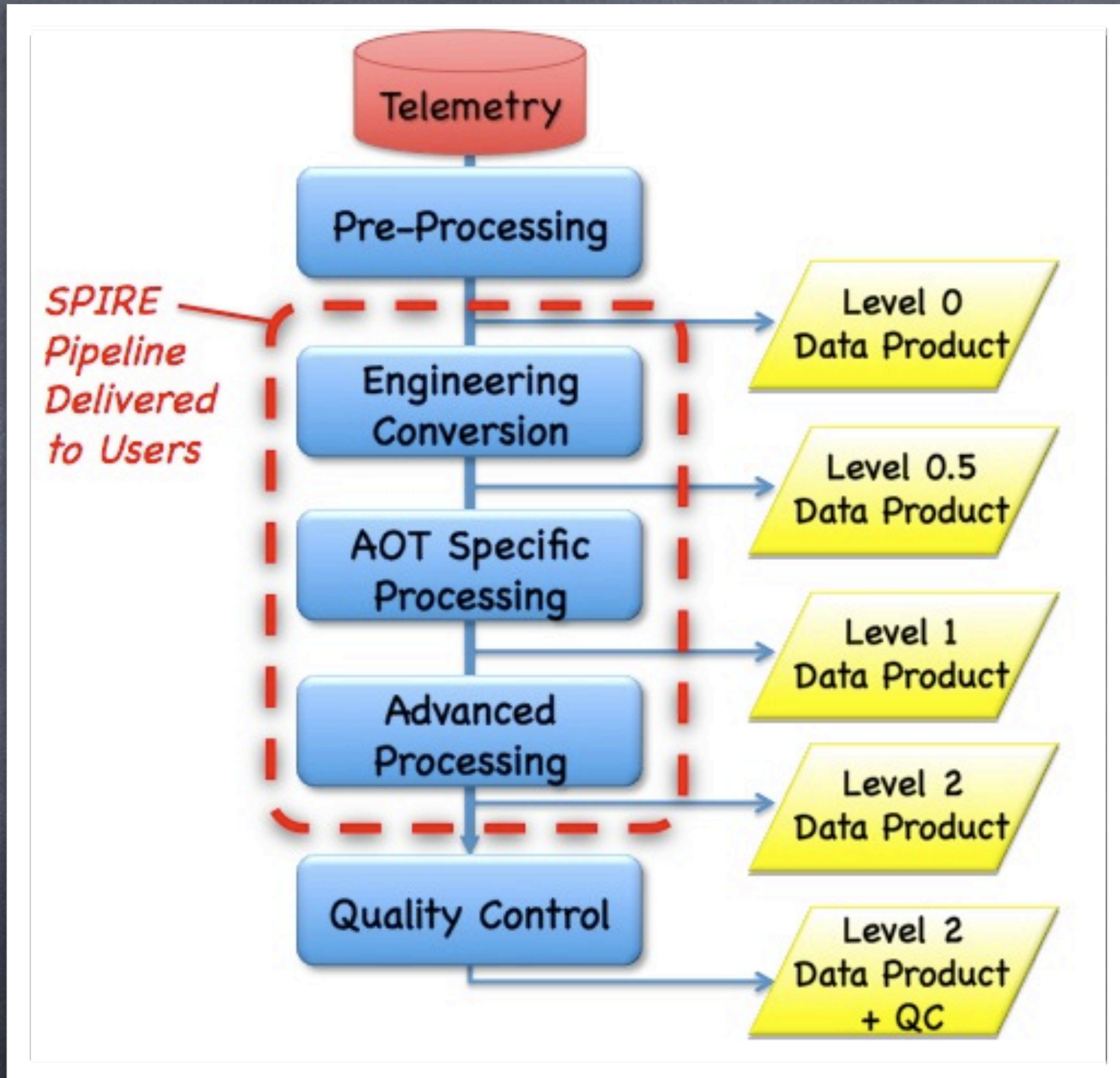
Considerable redundancy for SPIRE due to smaller scan distance needed by PACS arrays. Sample rate lowered from 18.6 to 10Hz.

PACS Geometry

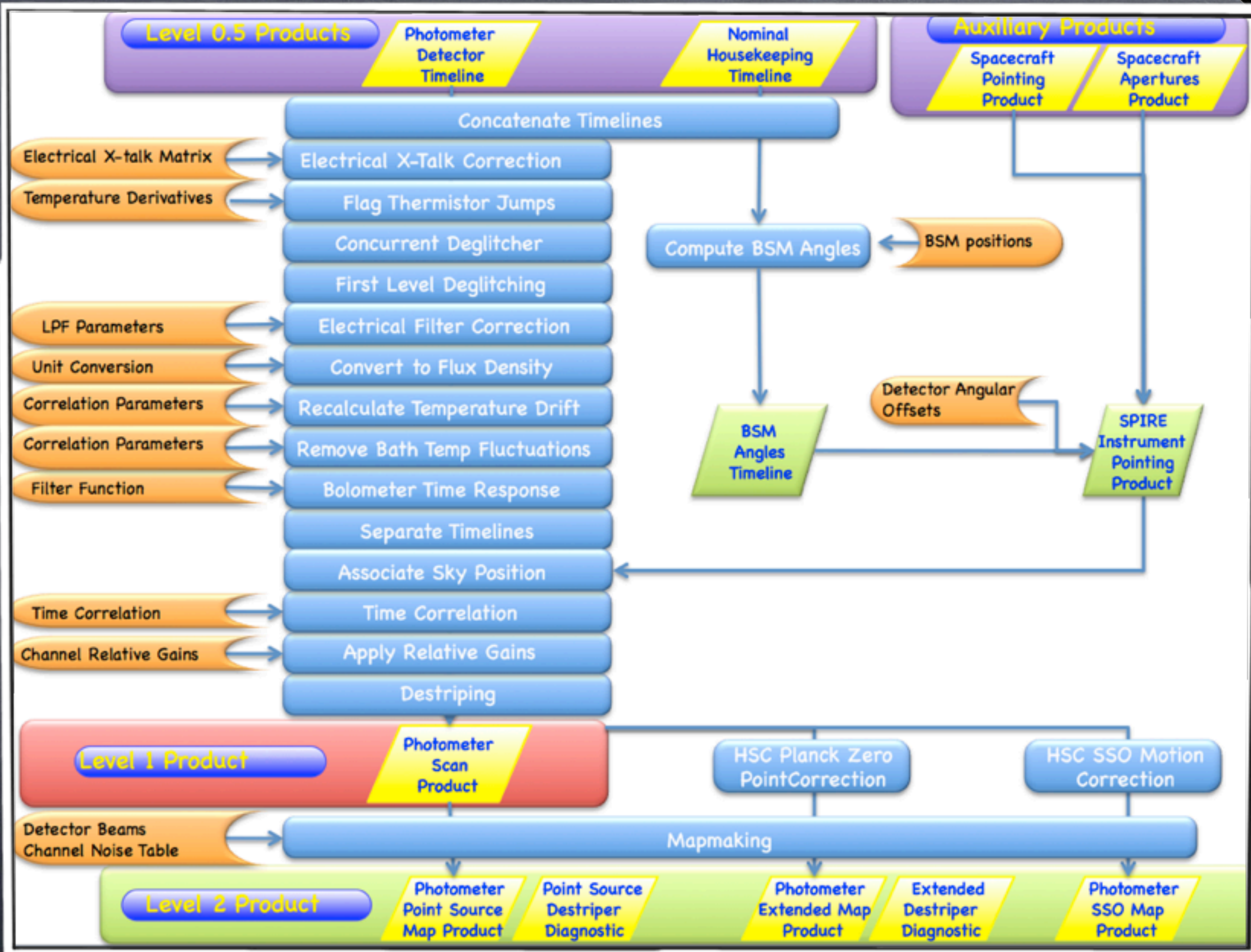


Even field coverage for PACS. **Additional frame averaging needed to keep data rate down.** Blue array frames are averaged by additional factor two compared to PACS only mode.

SPIRE Obs. Structure



Scan Map Pipeline



Monday, 28 January 13

HIPE 10 Level2

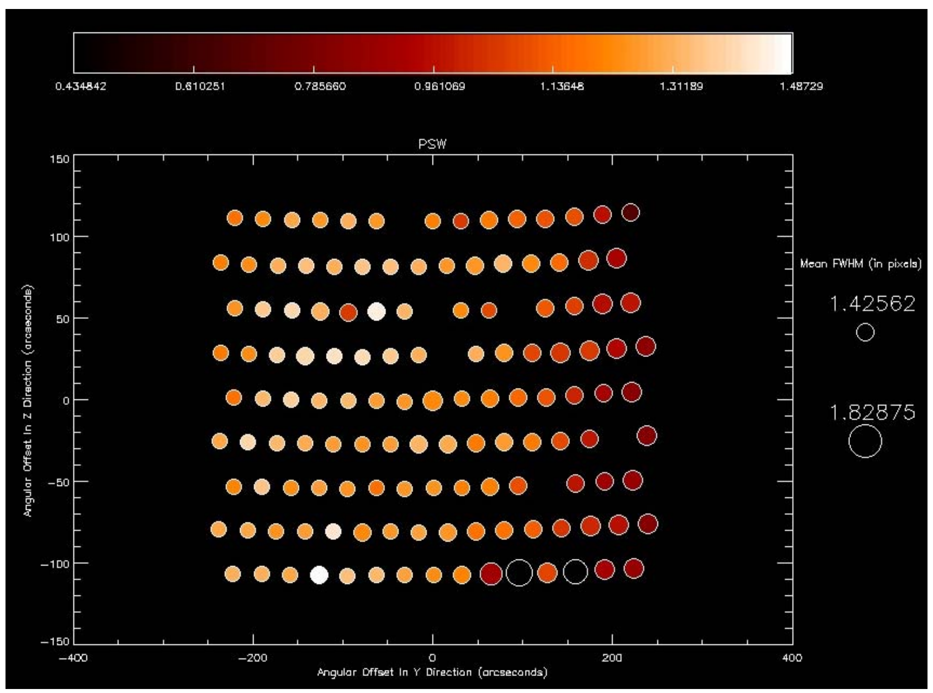
- Standard (*original*) SPIRE maps are calibrated for point source and νF_ν=const
    - maps now named psrcPxW
    - Units are still in Jy/beam, hence no direct aperture photometry possible (instead use e.g. Gaussian/PSF fitting)

- HIPE 10 includes maps specifically calibrated for extended emission:
    1. starting from level1, the relative gain correction is applied
    2. maps are divided by the new beam area and converted in MJy/sr, always under the νF_ν=const assumption
    3. absolute offset is computed from cross-calibration with Planck maps

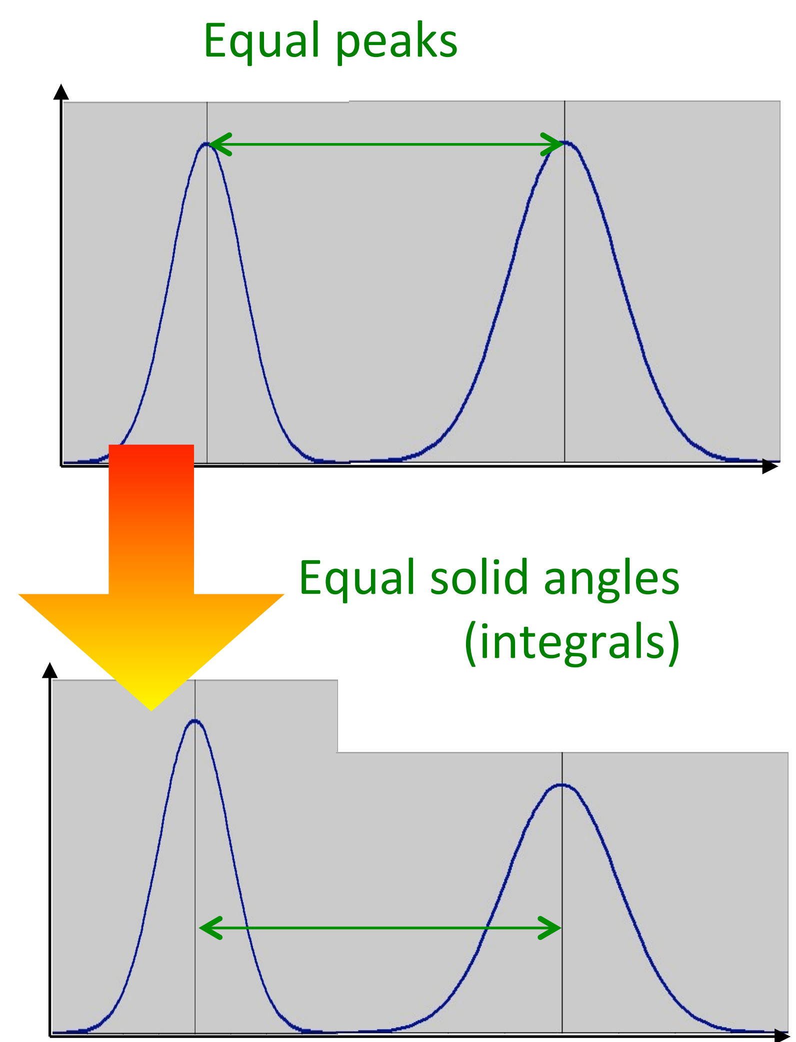
- For SSO observations, 3 maps with proper motion correction are available too. SSO will be *on focus*, background will be *blurred*

Relative gain correction - 1

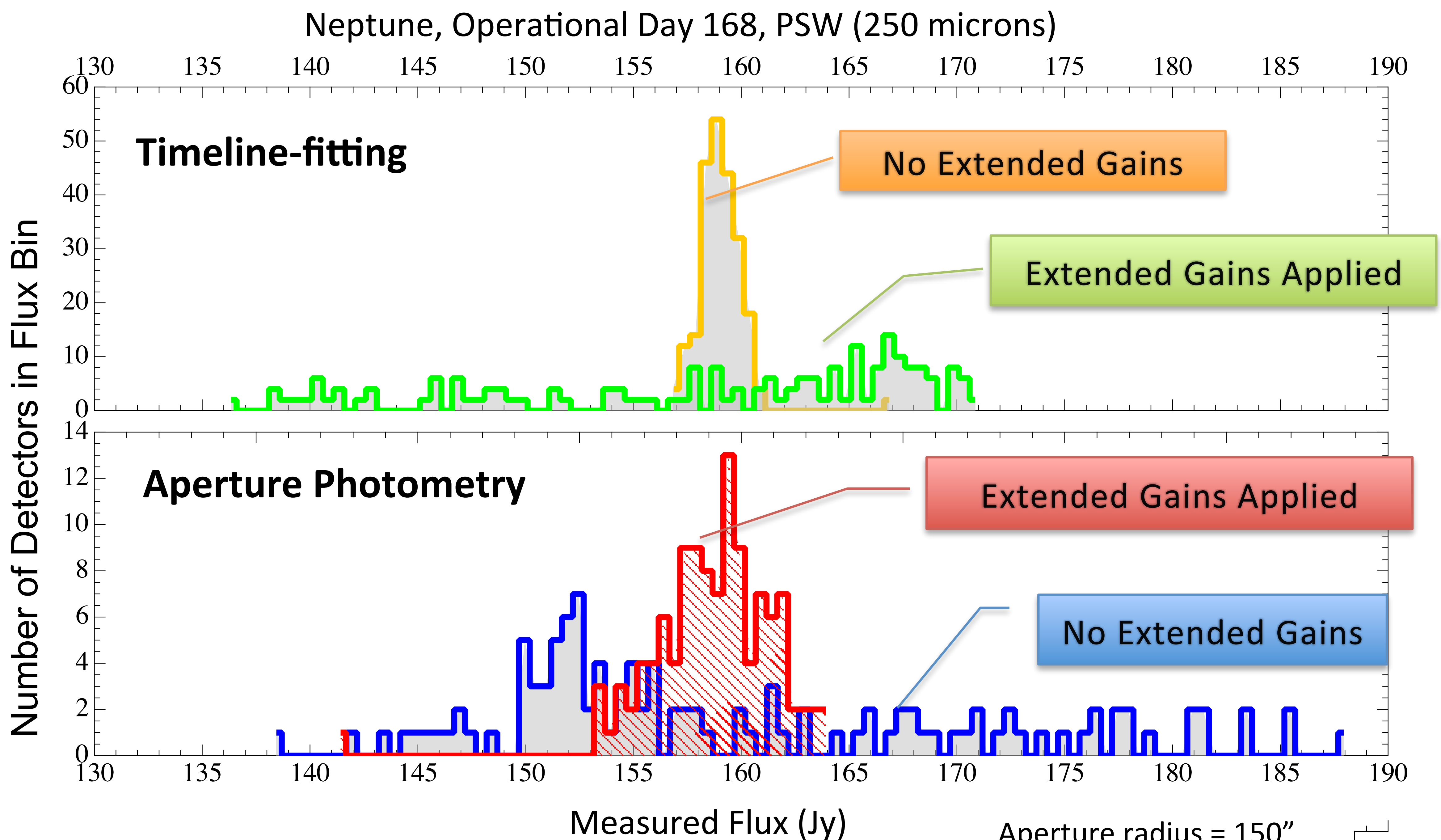
- Not all detector beam-profiles have the same width
- Applying the Extended Gains equalizes the detector areas (instead of the peaks)
- The numbers are provided in the SPIRE calibration tree



PSW: FWHMs are exaggerated

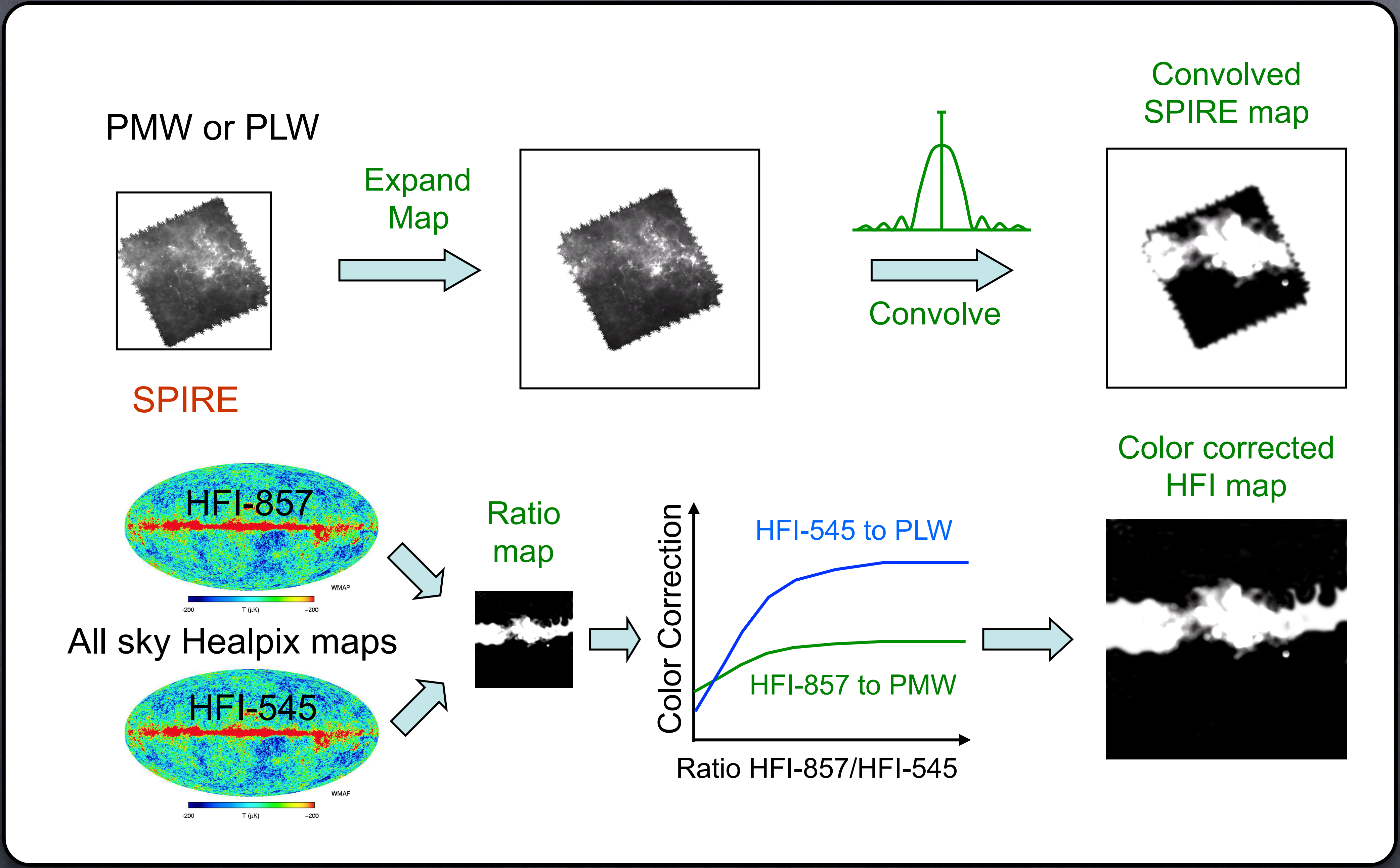


Relative gain correction - 2



Aperture radius = 150”
Aperture correction = 1.027

Planck cross-calibration - 1



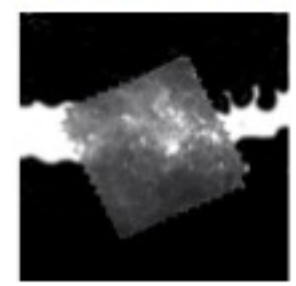
Planck cross-calibration - 2

Convolved SPIRE map

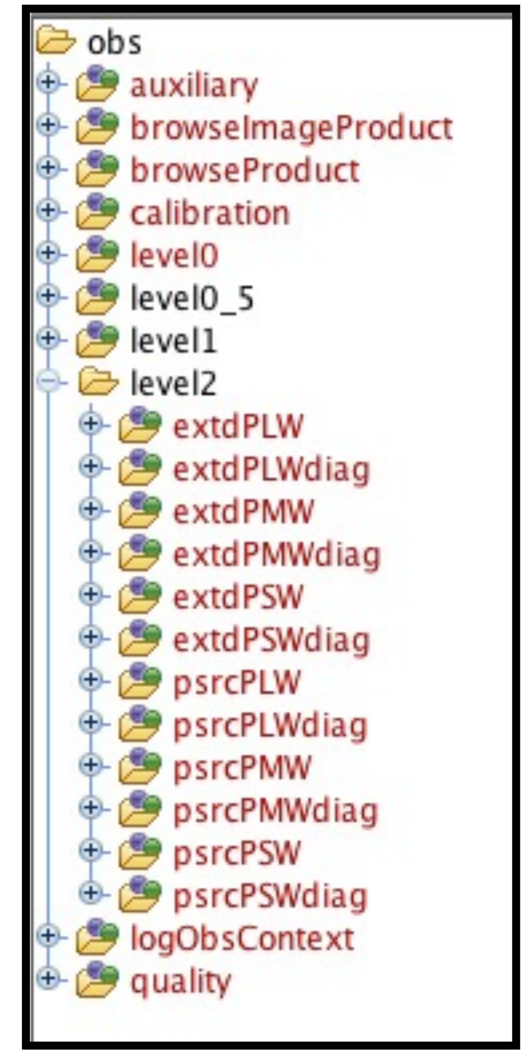
Color corrected HFI map

Offset determination - 1st iteration:
computed on difference map

Embedded SPIRE map



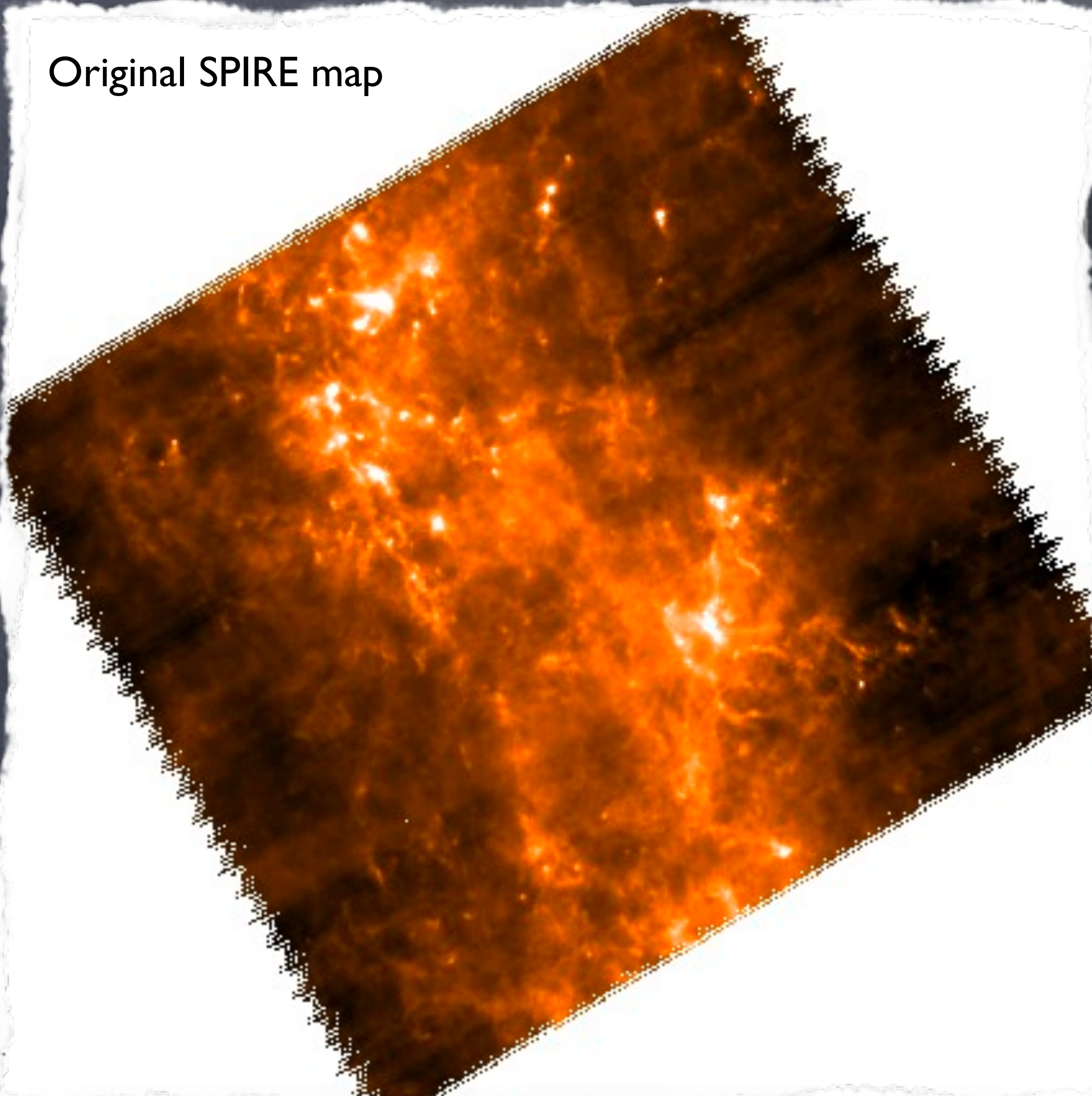
Offset determination - 2nd iteration:
- convolve embedded map with Planck beam
- compute offset on new difference map



Finally update the observation context

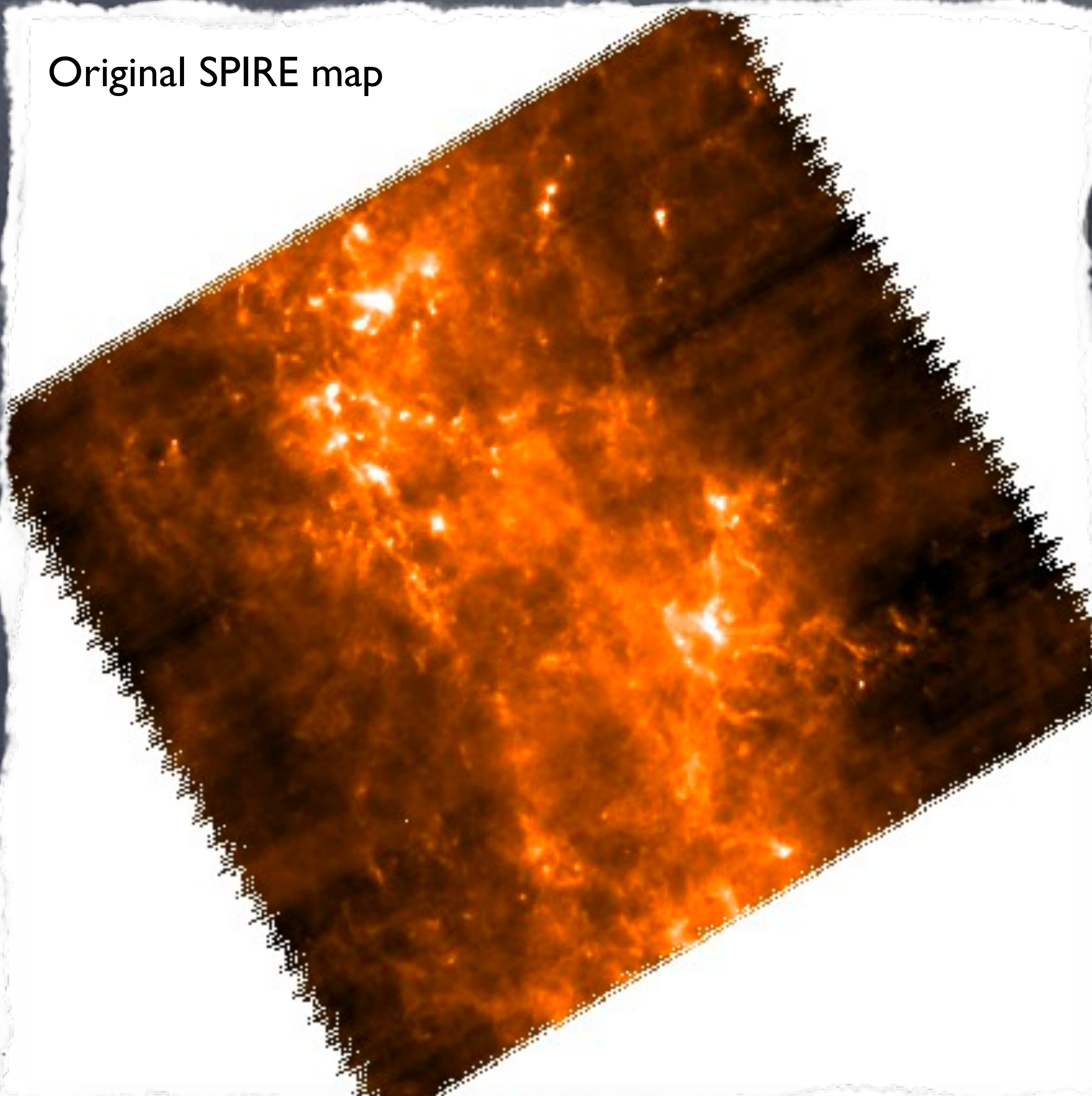
Maps comparison

Original SPIRE map

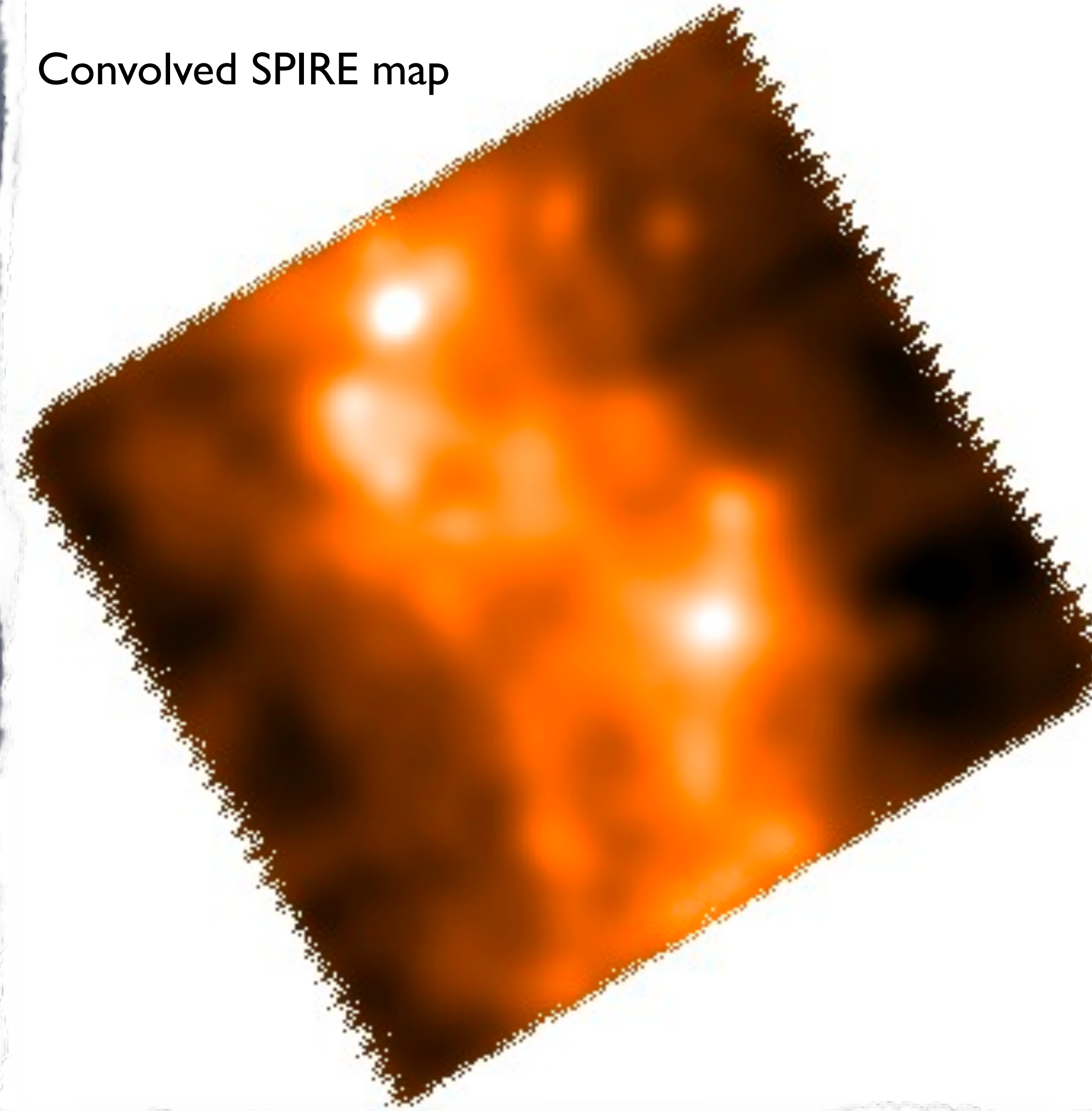


Maps comparison

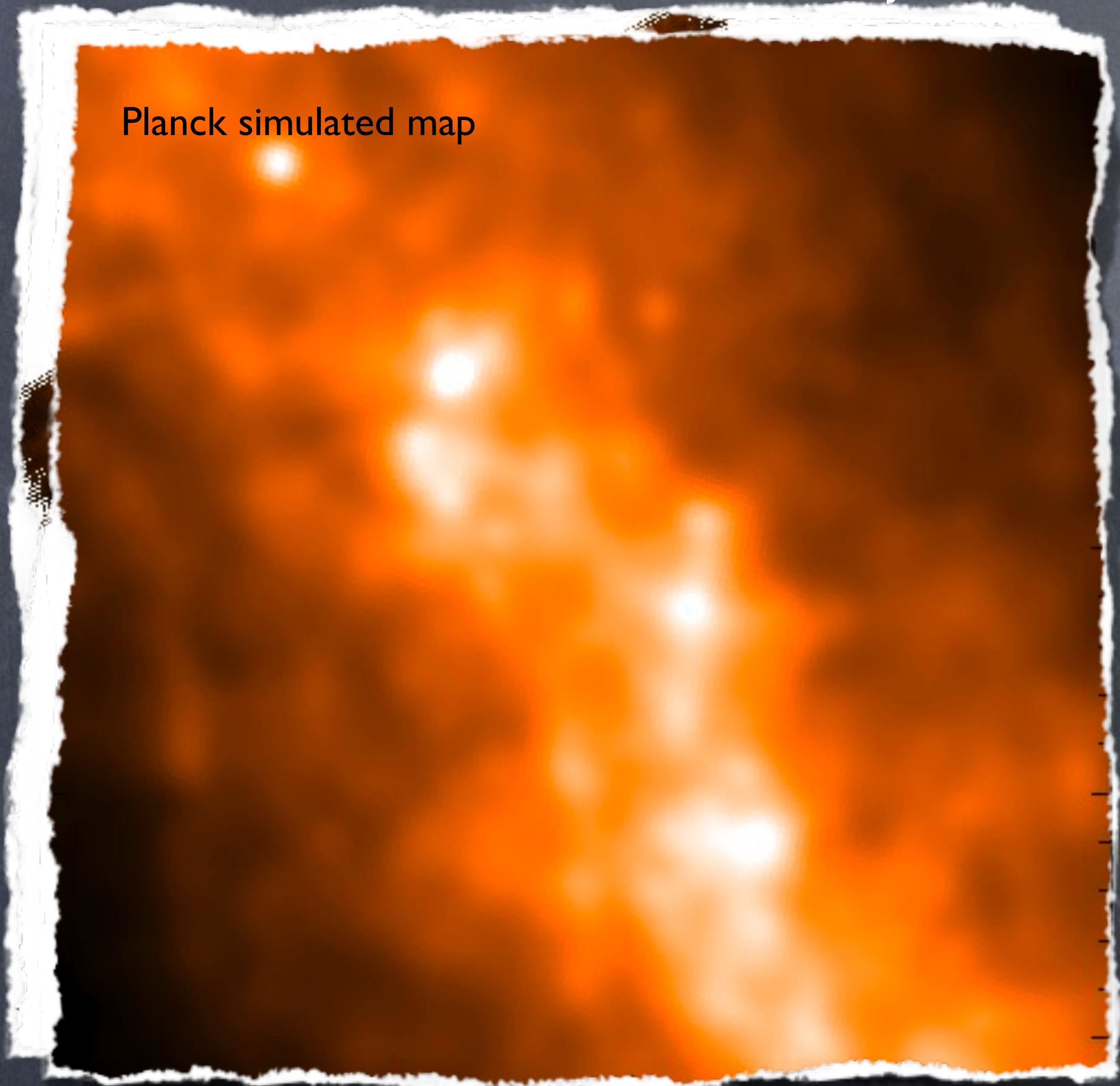
Original SPIRE map

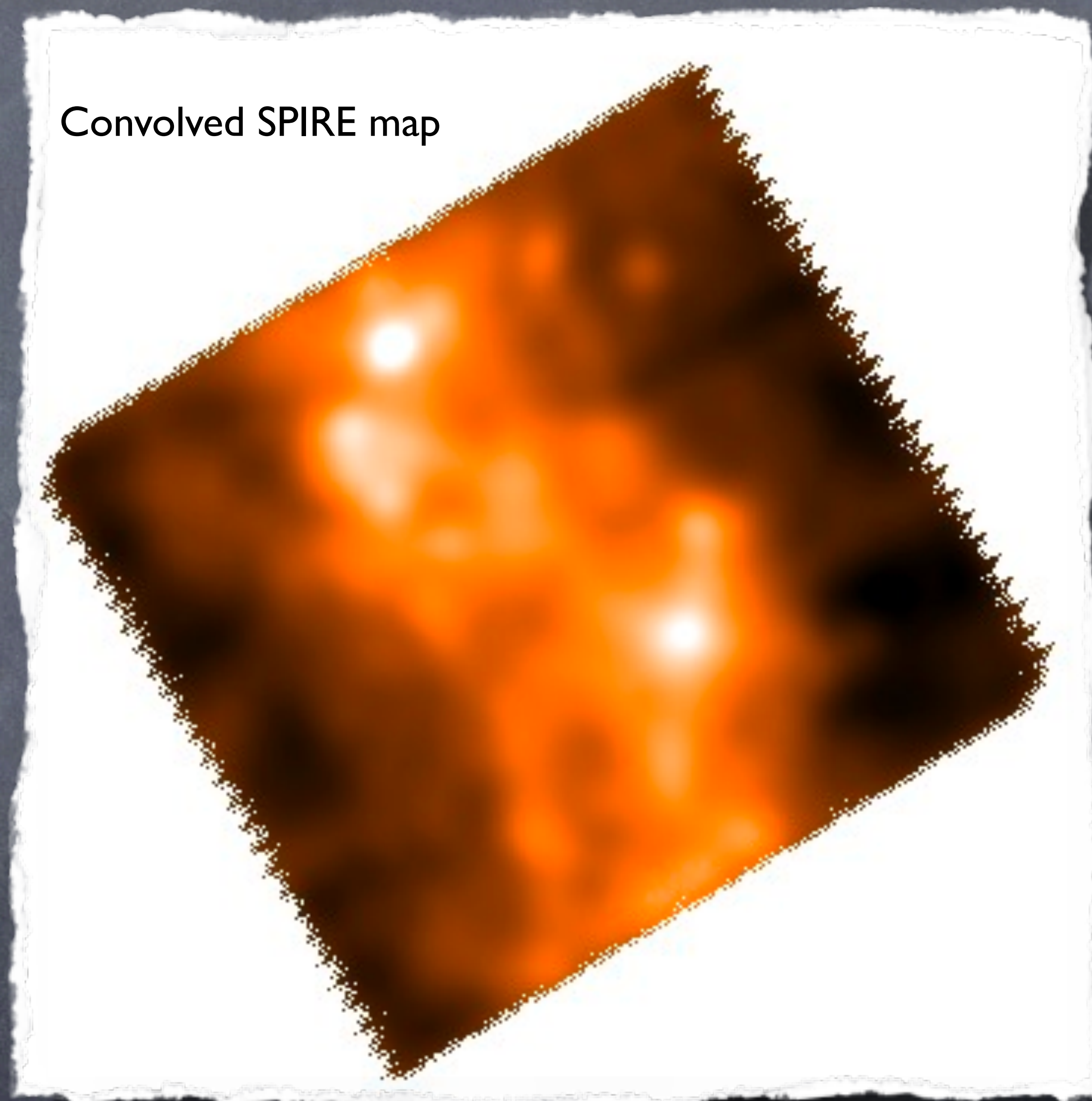


Convolved SPIRE map

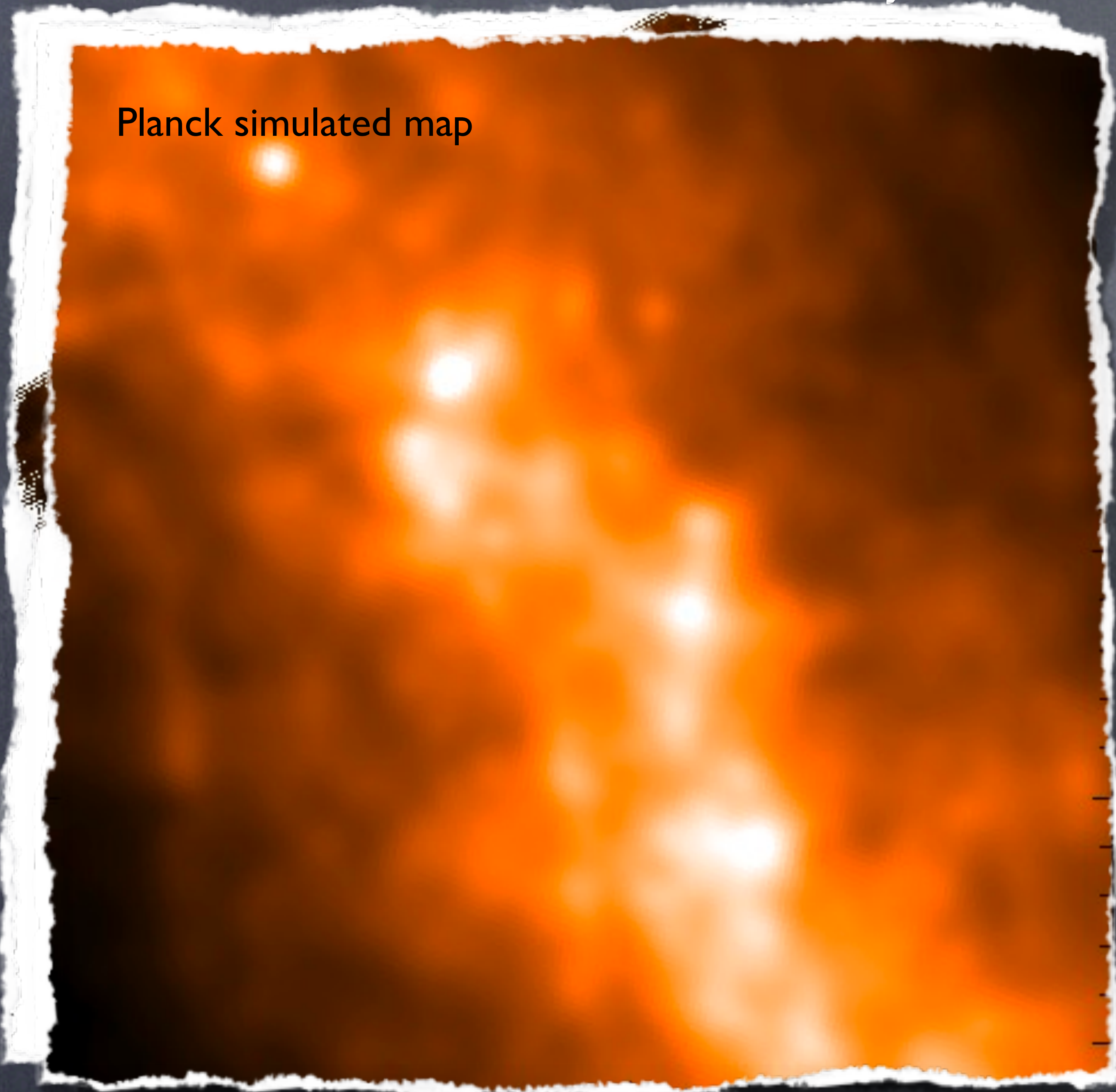


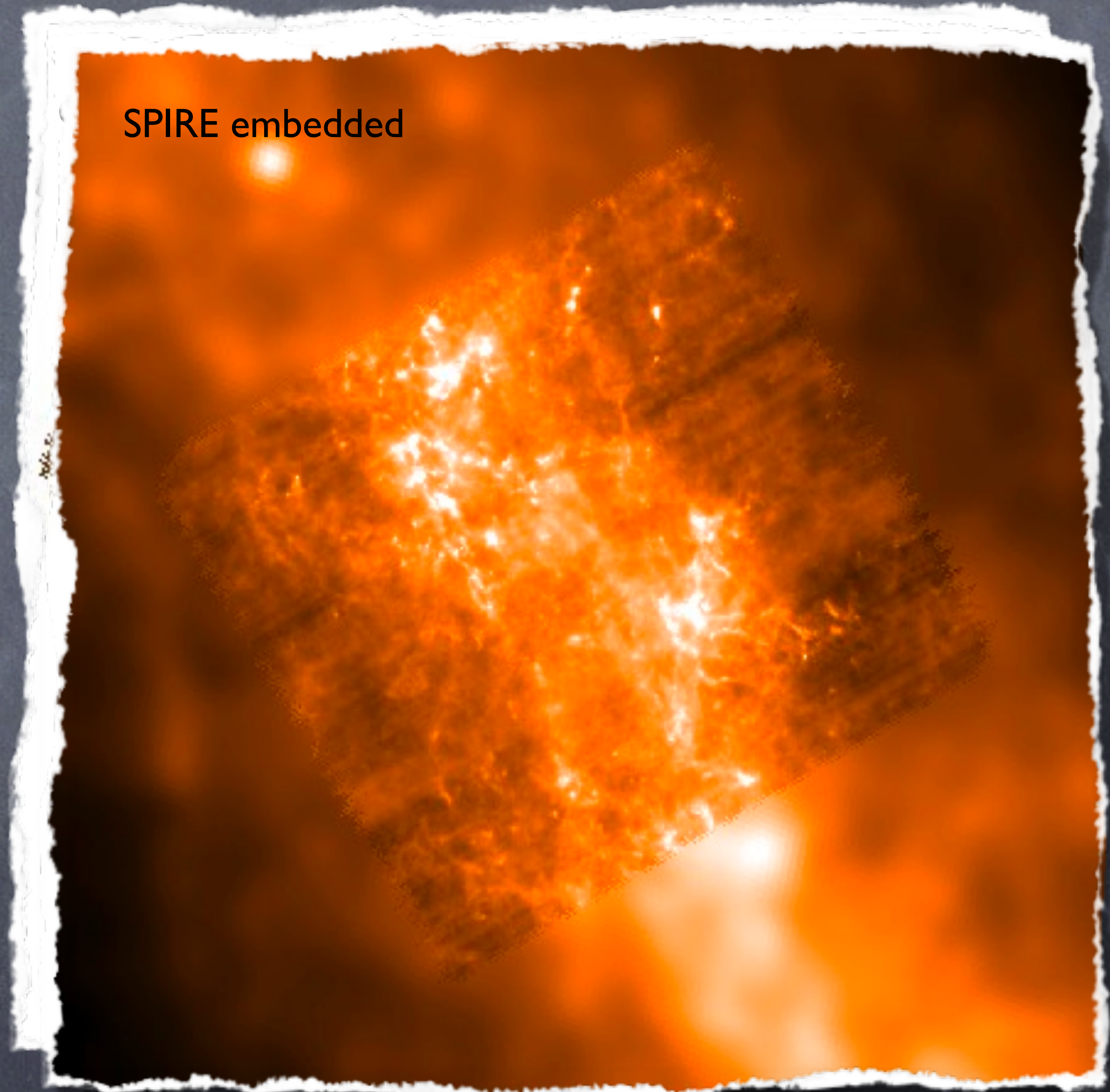
Maps comparison



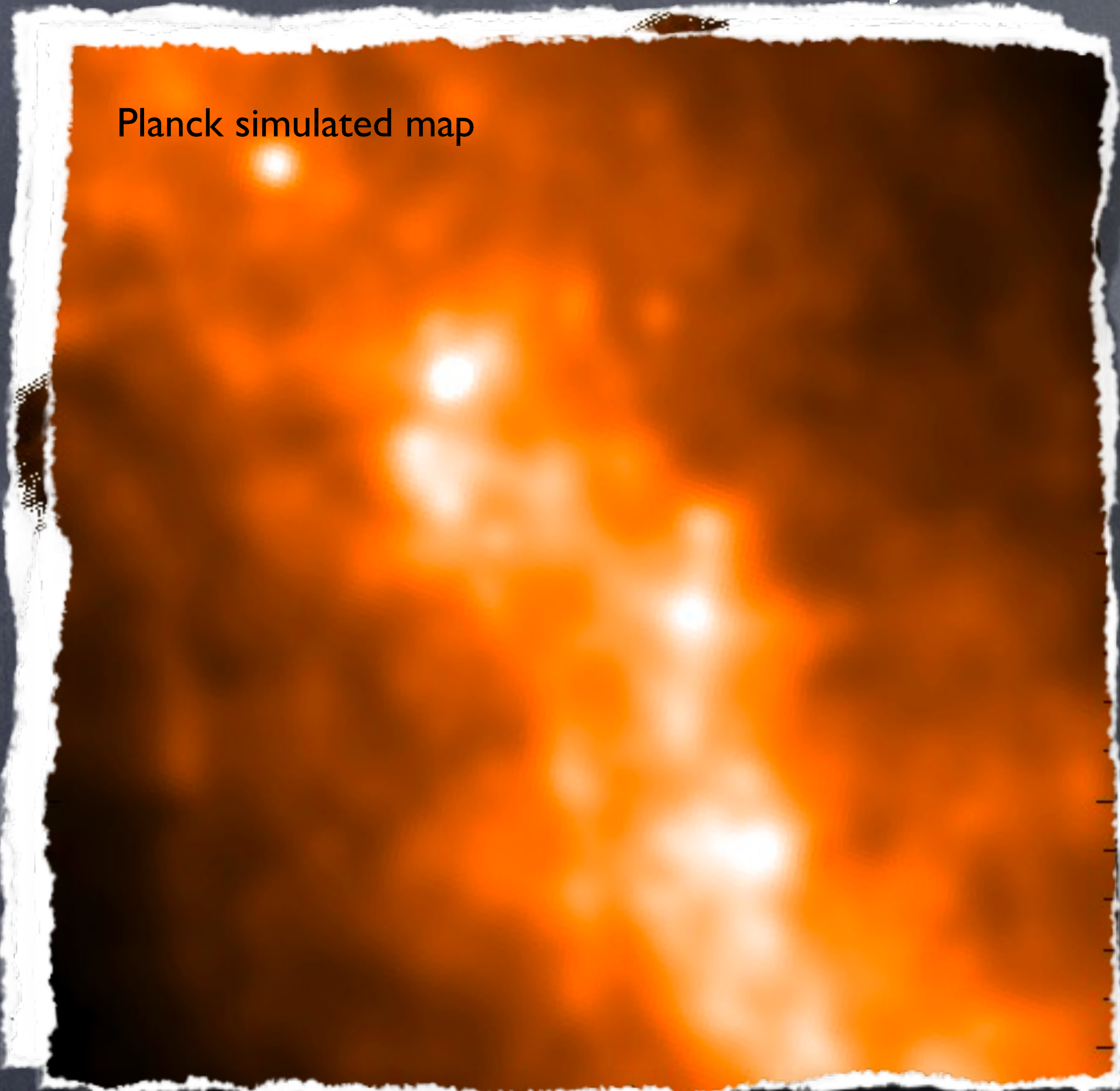


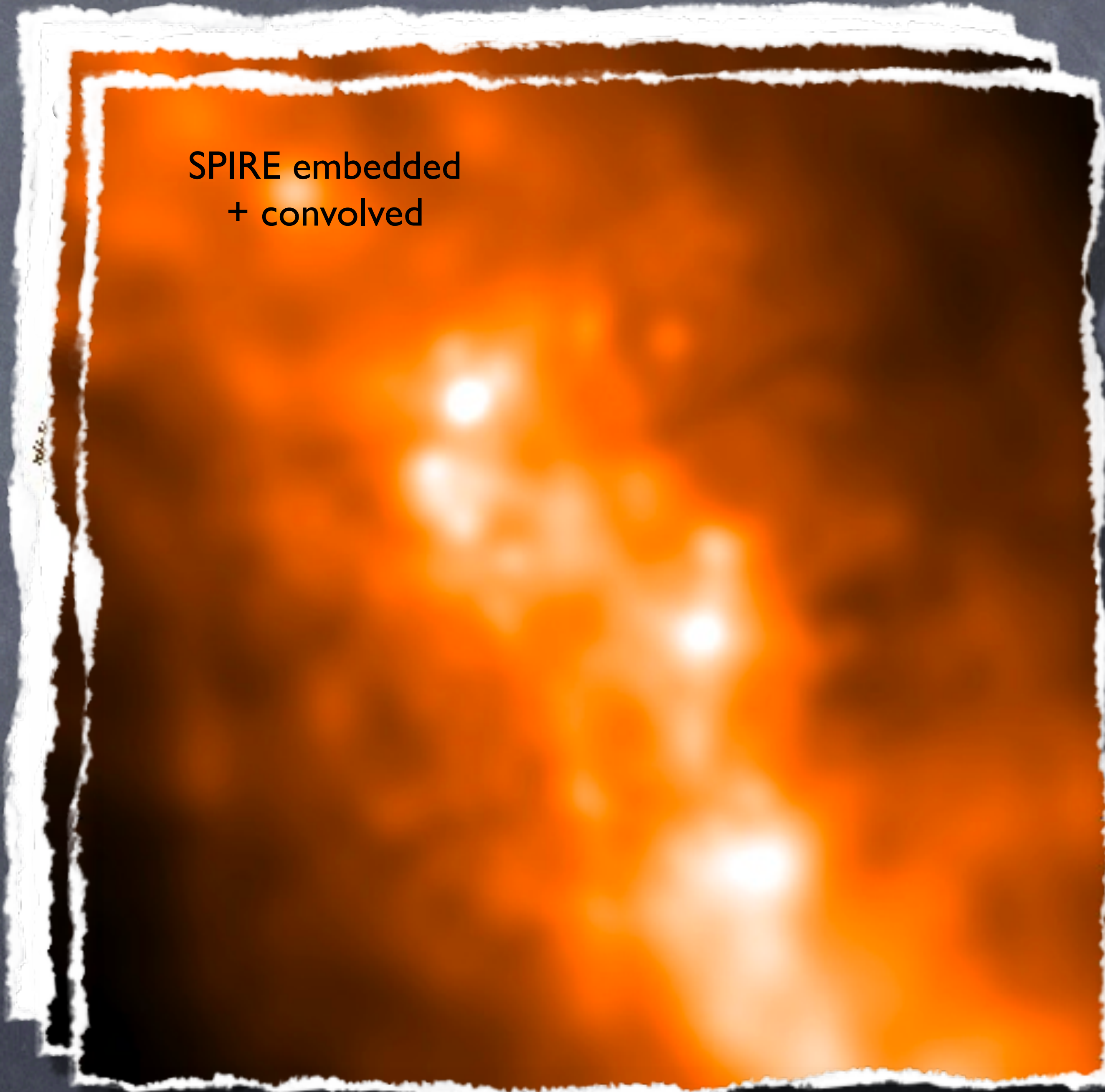
Maps comparison





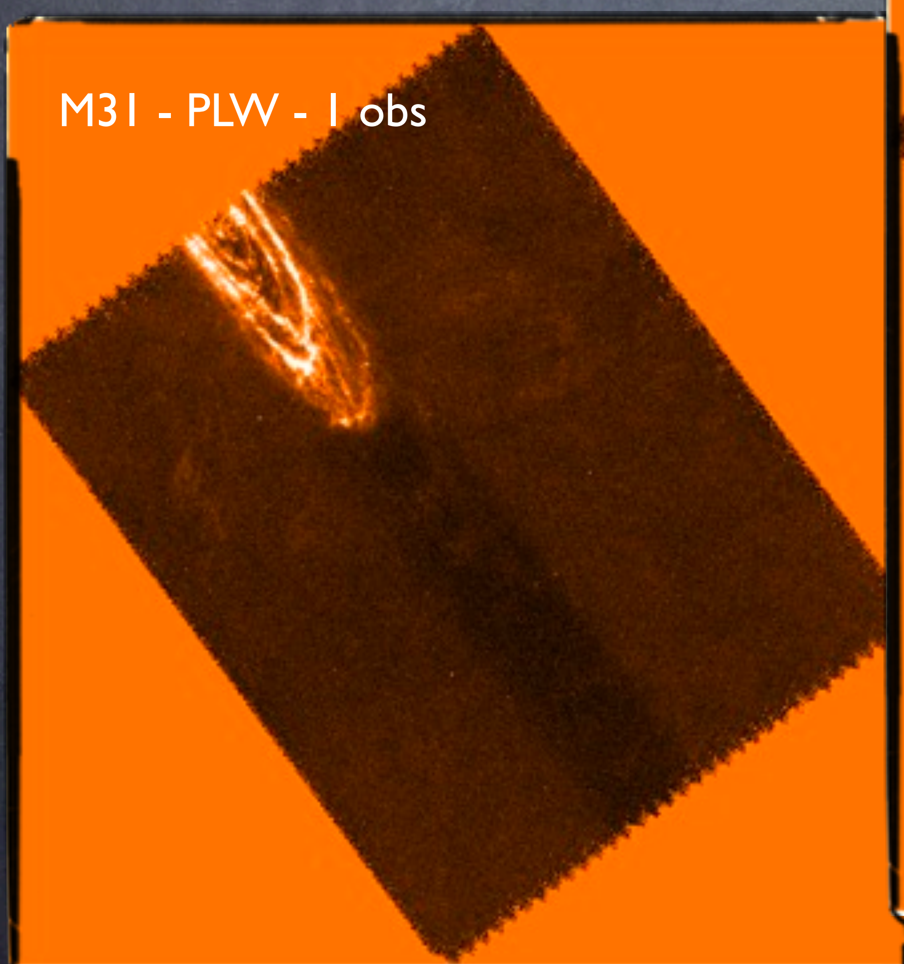
Maps comparison

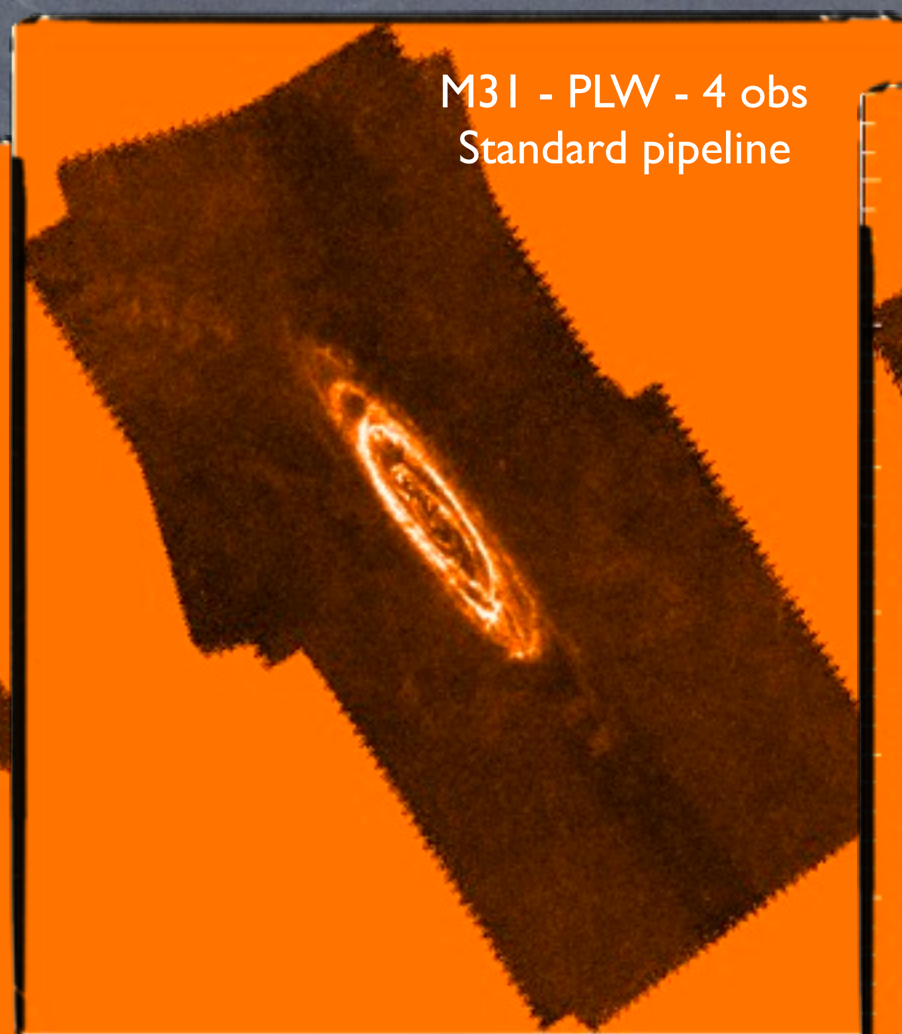


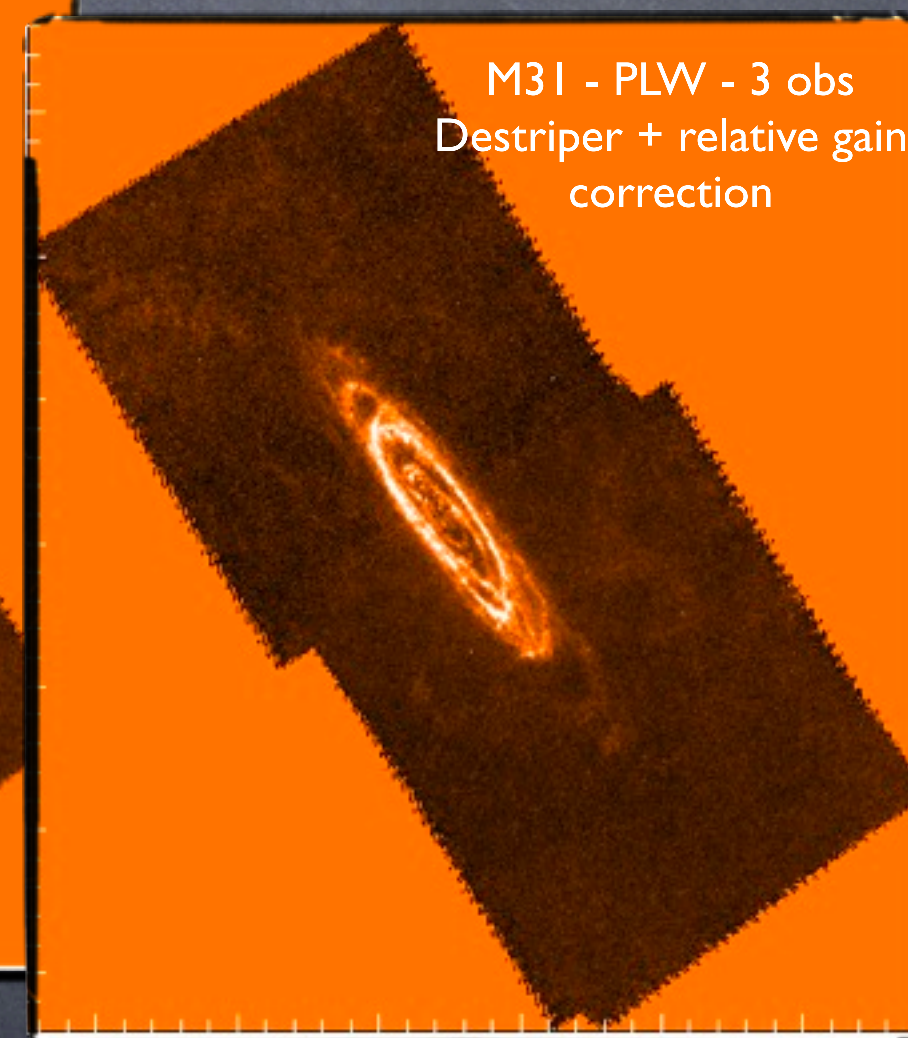


SPIRE Level 2.5

- As of HIPE 6.1.1, a new task is able to find *contiguous* observations observed in compatible modes and same program
- Level 1s of the selected observations will be merged into one product, and maps will be produced and saved into a Level 2.5
- Especially useful for P-Mode and dithered observations

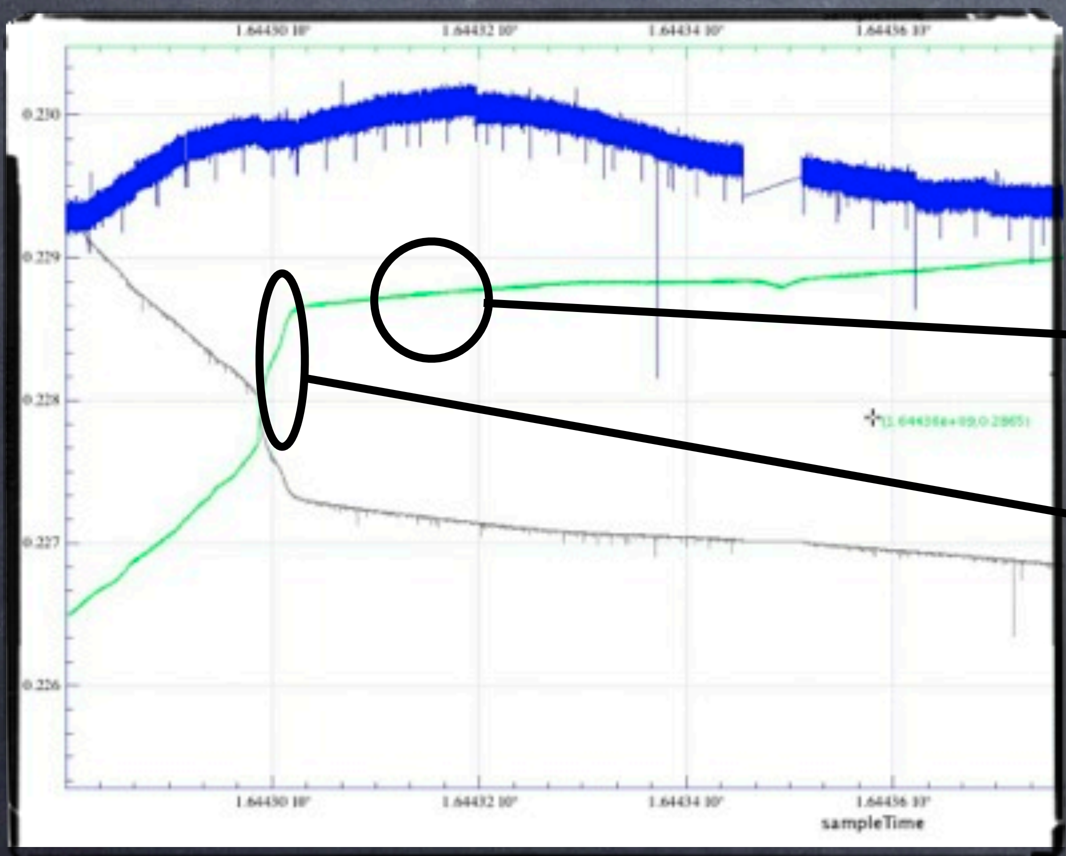


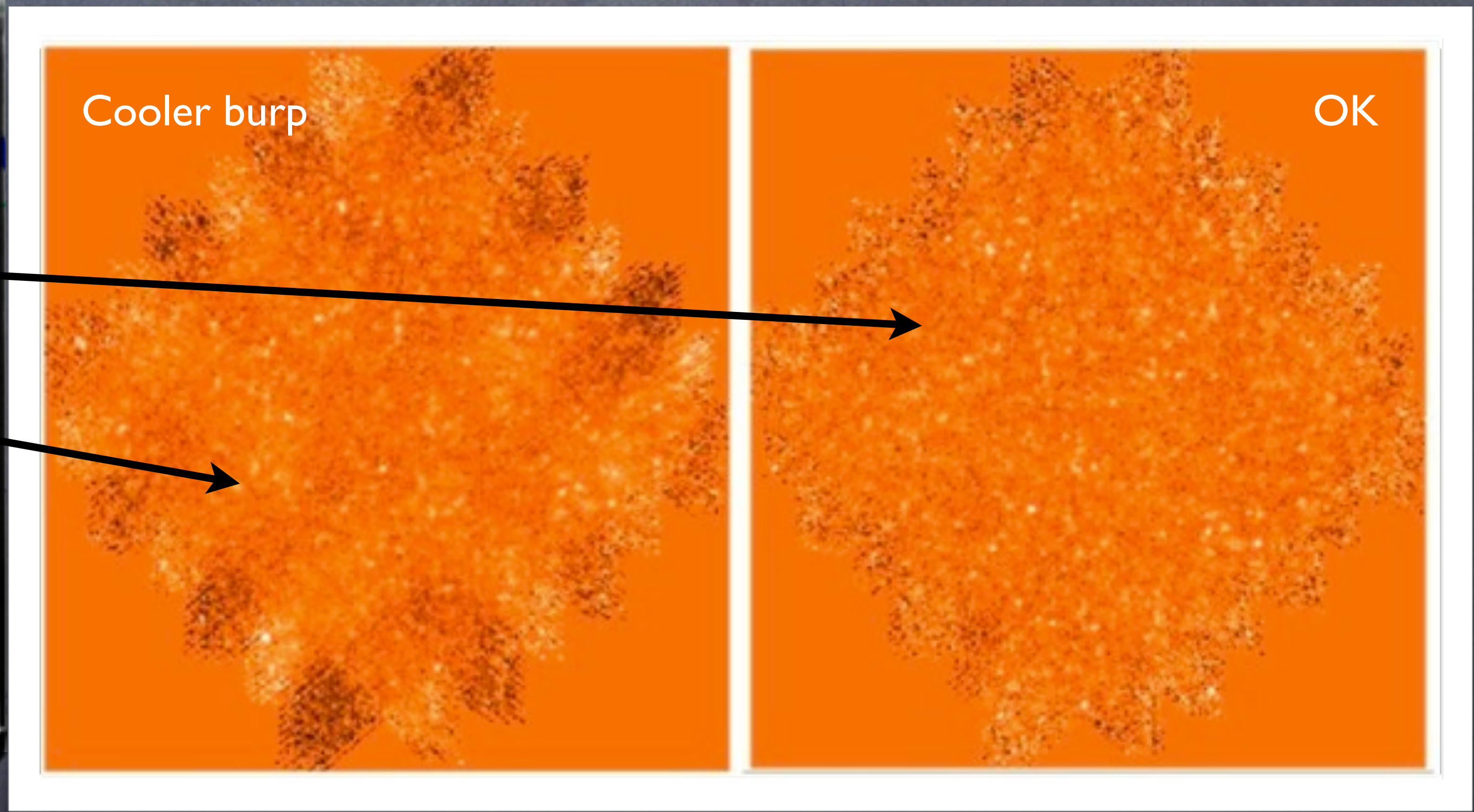




Do I Need To Reprocess?

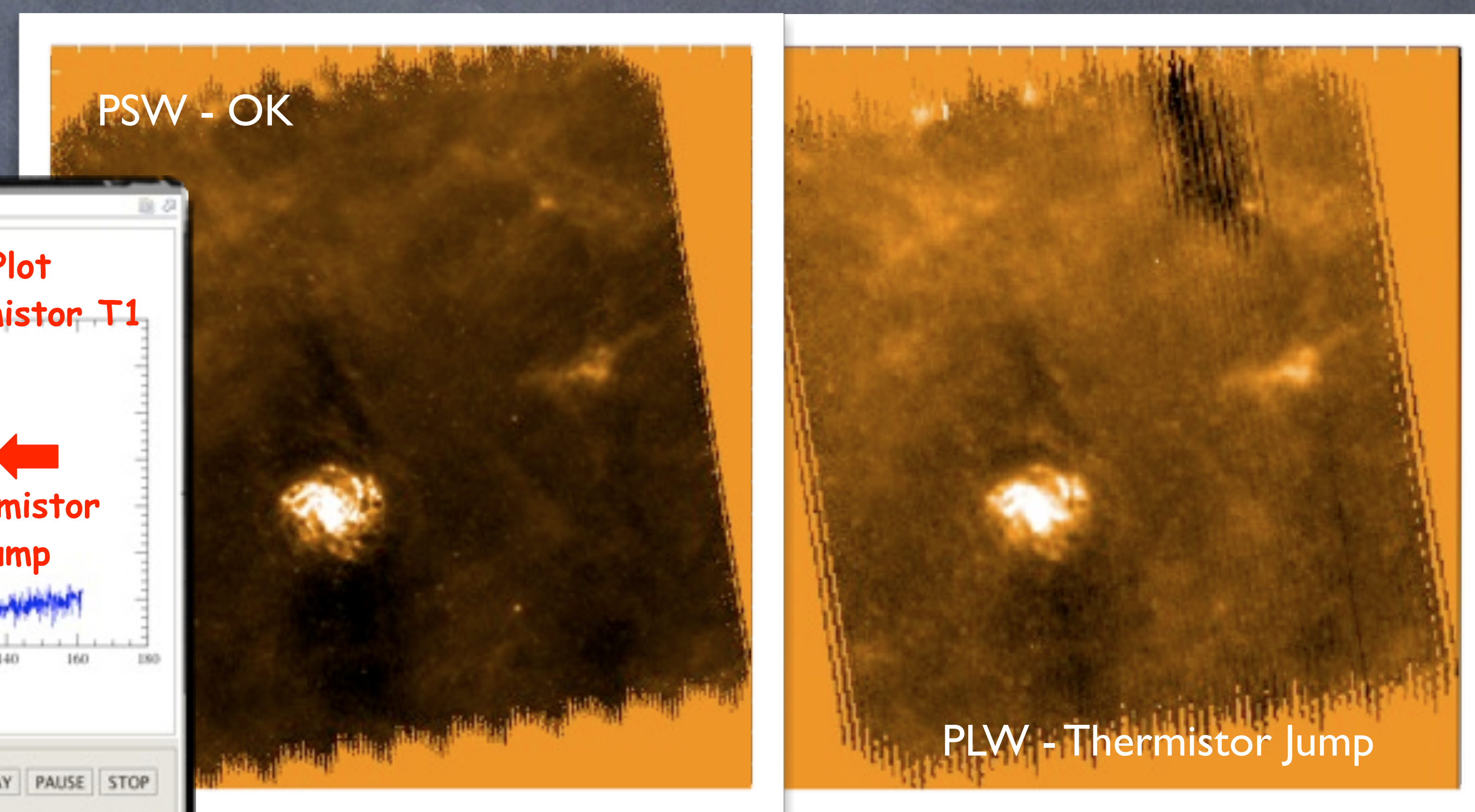
- Often SPIRE maps look great straight out of the box, but problems can occur and artifacts can be seen
- Map artifacts may be visible in the final maps, often in the form of stripes or bands
- Reasons for striping, and their solutions:
  - Bias voltage drift during “burp”: better baseline removal / destriper

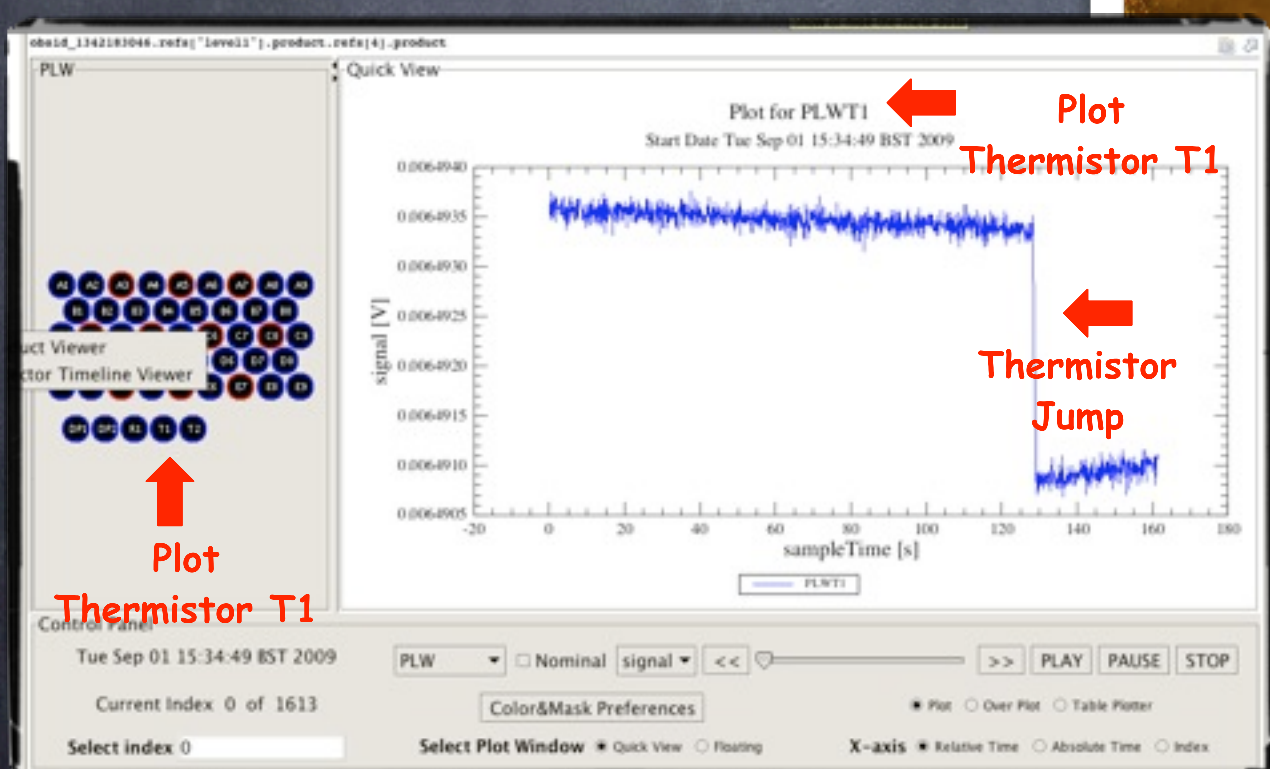




Do I Need To Reprocess?

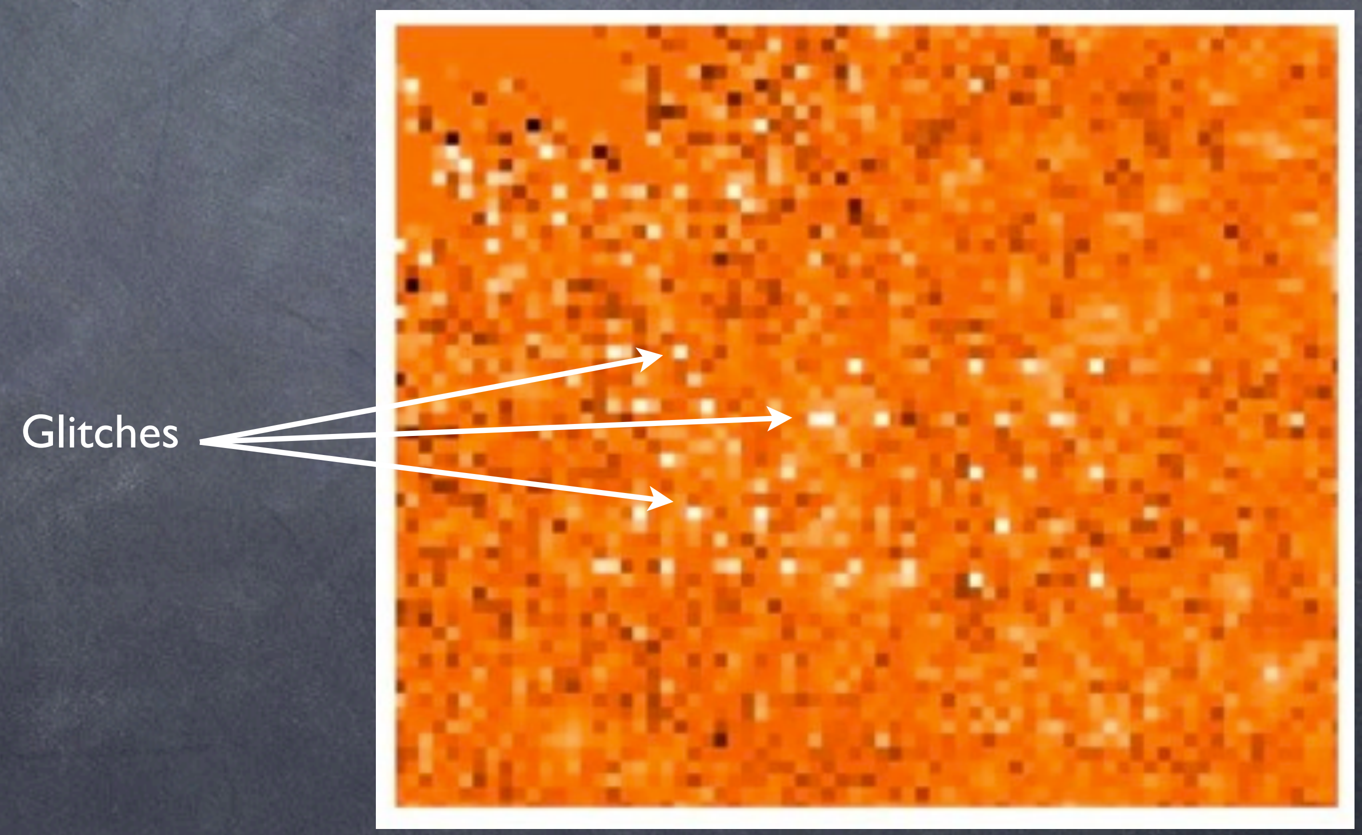
- Often SPIRE maps look great straight out of the box, but problems can occur and artifacts can be seen
- Map artifacts may be visible in the final maps, often in the form of stripes or bands
- Reasons for striping, and their solutions:
  - Bias voltage drift during "burp": better baseline removal / destriper
  - Thermistor jump: run jumpDetector / mask bad thermistor





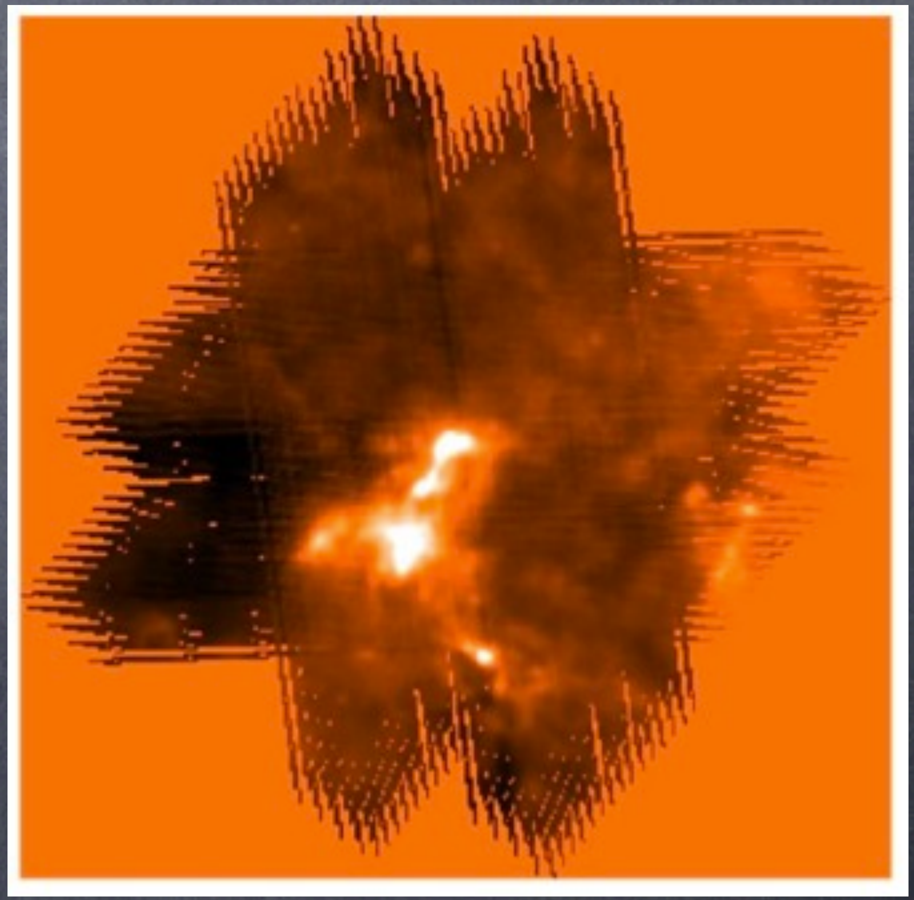
Do I Need To Reprocess?

- Often SPIRE maps look great straight out of the box, but problems can occur and artifacts can be seen
- Map artifacts may be visible in the final maps, often in the form of stripes or bands
- Reasons for striping, and their solutions:
  - Bias voltage drift during "burp": better baseline removal / destriper
  - Thermistor jump: run jumpDetector / mask bad thermistor
  - Missed bright (broad) glitches: alternative deglitchers / interactive deglitching



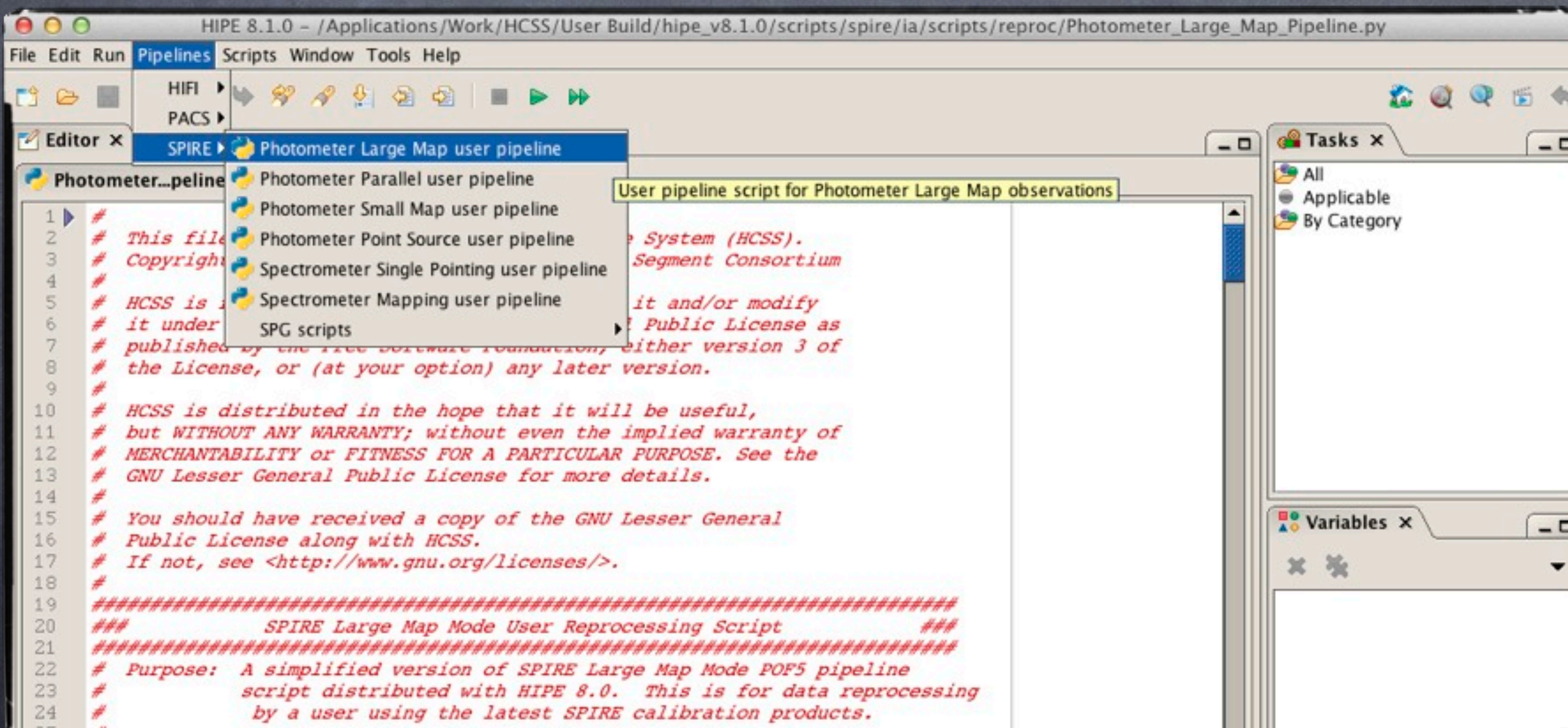
Do I Need To Reprocess?

- Often SPIRE maps look great straight out of the box, but problems can occur and artifacts can be seen
- Map artifacts may be visible in the final maps, often in the form of stripes or bands
- Reasons for striping, and their solutions:
  - Bias voltage drift during "burp": better baseline removal / destriper
  - Thermistor jump: run jumpDetector / mask bad thermistor
  - Missed bright (broad) glitches: alternative deglitchers / interactive deglitching
  - Bright extended source: better baseline removal / destriper



Users' Scripts - 1

- Data may need reprocessing using *User Reprocessing Scripts*, available under the *Pipeline* menu:
  - Simplified pipeline scripts accessible within HIPE
  - Well commented
  - Documented in SPIRE Data Reduction Guide
  - Contains all processing steps as the archive processing



Users' Scripts - 2

```
##########################################################################
###                       User Selectable Options                      ###
##########################################################################
# (A) Specific OBSID in the form of an integer or hexadecimal (0x) number:
# (B) the name of the data Pool in your Local Store:
# (C) Specify the output directory for writing the maps FITS files:
#
myObsid     =  enterOBSID
myDataPool = "Enter Pool name here"
outDir      = "/enter/path/here/"
# e.g.
#myObsid     =  0x50001833
#myDataPool = "OD117-ScanNGC5315-0x50001833"
#outDir      = "/Users/cpearson/jython/localstore/plots/"
#
# Additional Options
# (D) mapping: The mapping Algorithm to use naive or madmap
# (E) includeTurnaround: Include the scan line turnarounds in the pocessing and mapmaking
# (F) applyExtendedEmissionGain: Apply the relative gains for each bolometer for extended emission
# (G) baselineSubtraction: Subtract a baseline from each scan to avoid stripes
# (H) destriper: Determine and remove baselines to achive an optimum fit between all timelines
#      If the map maker is 'naive', at least one of the options G or H must be True.
# (I) coolerBurpDetection: Search and correct cooler burp recalculating the TempDrift calibration table
mapping = 'naive'
includeTurnaround = False
applyExtendedEmissionGains = False
useBaselineSubtraction = False
useDestriper = True
coolerBurpDetection = False
##########################################################################



#*************************************************************************
##  Load in an observation context from your data pool into HIPE:
obs=getObservation(myObsid,poolName=myDataPool)

print
print "Processing observation %s (0x%x) from data pool %s."%(myObsid, myObsid, myDataPool)

#*************************************************************************
# Calibration Context and Calibration Files
# Read the latest calibration tree relevant to HCSS v10 from the local disc:

cal = spireCal(pool="spire_cal_10_1")

# TO CORRECT AN ERROR ON THE ABOVE LINE, run the following command ONCE only
# to load and save the calibration tree from the Archive (may take some time):
# (for more details, see the "Calibration" chapter in the SPIRE Data Reduction Guide)

#cal = spireCal(calTree="spire_cal_10_1", saveTree=True)

# Attach the updated calibration tree to the observation context
obs.calibration.update(cal)
```

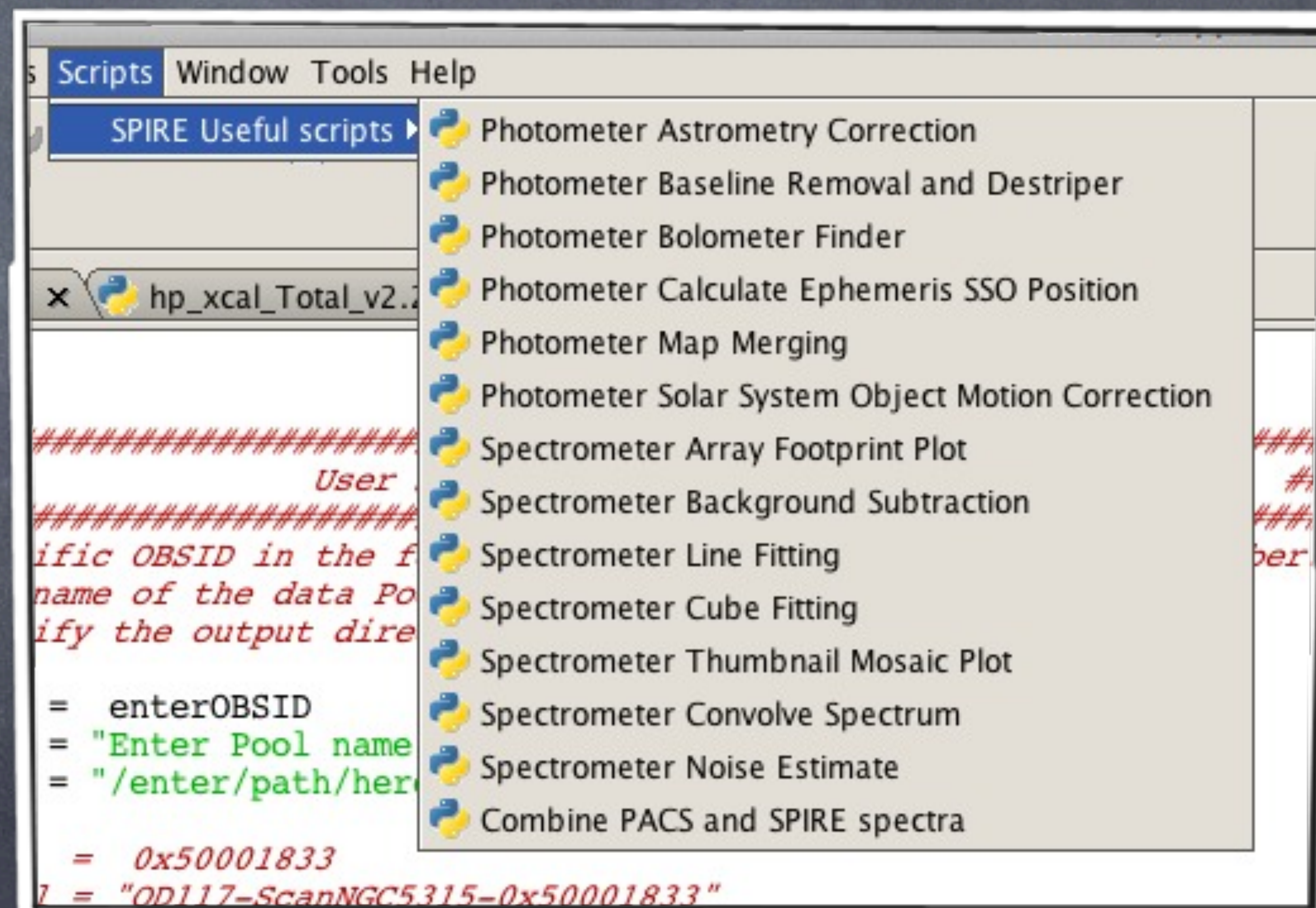
User Inputs:
1. OBSID
2. Pool name
3. Output Directory for FITS files

Options:
1. Choose map-maker
2. Include turn-around data?
3. Apply extended gains
4. Use baseline subtraction and/or destriper
5. Cooler burp detection & correction

Specify Calibration Tree

Useful Scripts

- As of HIPE 8, new “SPIRE Useful Scripts” menu available under the Scripts menu
- For the photometer:
    - Astrometry Correction script: correct astrometry of an observation given a reference map/observation
    - Baseline Removal and Destriper script: presents different ways on how to remove the AC component from SPIRE data
    - Bolometer Finder script: plots detectors data passing through a specified pixel
    - Ephemeris SSO Position script: it will provide the start and end positions for a moving target
    - Map Merging script: merge 2 observations together
    - SSO Motion Correction script: corrects S/C motion to focus the map on the SSO. Background will be blurred



SPIA

SPIRE Photometer Interactive Analysis:

- GUI to run SPIRE pipeline from level 0 to 2
- Provides other useful tools (e.g. concatenate level1 BBs)
- Available as HIPE plug-in. Latest version is 1.10.1
- More info at https://nhscsci.ipac.caltech.edu/sc/index.php/Spire/SPIA