NHSC/PACS Workshop

PACS spectrometer data storage

Jeff Jacobson & Dario Fadda

August 27, 2013

Exploring an observation

After retrieving an observation from the archive, it is possible to explore its metadata with the command obsSummary.

History | Log | Console

```
HIPE> obsSummary(obs)

Observation summary
   OBSID:       1342254274
   Instrument: PACS
   AOR label:  PSpecL - CII_G031.3+0.0
   Proposal:   OT2_pgolds01_6
   Target:     CII_G031.3+0.0
   Redshift:   0.0 (rad. vel. km/s)
   Concat.:    Undef.
   OD:         1267
   Start:      Thu Nov 01 08:15:37 PDT 2012
   Duration:   897.0 seconds (incl. spacecraft on-target slew time)

AOT and instrument configuration
   AOT:        PacsLineSpec
   Mode:       Pointed, unchopped grating scan
   Bands:      B2B + R1 (prime diffraction orders selected)
   Is bright:  NO (default spectral range mode)

Observation request summary (HSpot Line/Range Editor Table)
   Number of requested primary lines/ranges: 2
   List of prime- and parallel HSPOT line/range requests:

      Camera |  ID |  Band |  Wave |  Reps. |   Line Flux    |  Cont. Flux. |  Width  |  Out of band |  Channel |  Name
             |     |       |  [um] |        |  [E-18 W/m^2]  |     [Jy]     |  [km/s] |              |          |

      Red       1     R1      121.94  1        0              0             1         No             Prime      N II 3P2-3P1
      Blue      1     B2B     61.00   1        -              -             -         Yes            Parallel   -
      Red       2     R1      205.25  1        0              0             1         No             Prime      N II 3P1-3P0
      Blue      2     B2B     102.65  1        -              -             -         No             Parallel   -

   Number of wavelengths slices observable within nominal response range:  3

Observation block summary (Observation Context prime- and parallel channel wavelength slices)
   List of prime- and parallel observing blocks:

      Camera |  ID |  Band |  Wave min |  Wave max |  Reps. |  Capacitance  |  Channel |  Name
             |     |       |    [um]   |    [um]   |        |      [pF]     |          |

      Blue      2     B2B     102.10      103.19      1 x 2    [0.14]          Parallel
      Blue      3     B2B     60.07       61.97       1 x 2    [0.14]          Parallel
      Red       102   R1      204.20      206.30      1 x 2    [0.14]          Prime      N II 3P1-3P0, 205.24667079087783 micron
      Red       103   R1      120.13      123.91      1 x 2    [0.14]          Prime      N II 3P2-3P1, 121.93823265331103 micron

   Number of wavelength slices available in the Observation Context:  4

System configuration
   SPG pipeline version of archived products: v9.1.0
   Calibration tree version:  41
   SPG pipeline products creation date: Thu Nov 01 16:03:14 PDT 2012
   Mission configuration: MC_H102ASTR_P70ASTR_S66ASTR_RP

Archived products status
   Level 0 status: Processed
   Level 1 status: Processed
   Level 2 status: Processed
```

Main info

Obs mode

Line list

Pipeline & calib versions

Data levels processed

calTree

From the previous summary, as well as from the commands:

```
HIPE> print obs.meta["calVersion"]
{description="Version of Calibration Tree", string="PACS_CAL_41_0"}
HIPE> print obs.meta["creator"]
{description="Generator of this product", string="SPG v9.1.0"}
```

we infer the pipeline version used for reduction in the archive as well as the calibration products version. To know the current version of the calibration products, type:

```
HIPE> calTree=getCalTree()
HIPE> print calTree.version
58
```

To access the names of all the calibration products as well as their versions, we have to print calTree.spectrometer.

calTree

In a new version of the calibration, only one or few products change.

Of particular interest in the case of the spectrometer, are the:

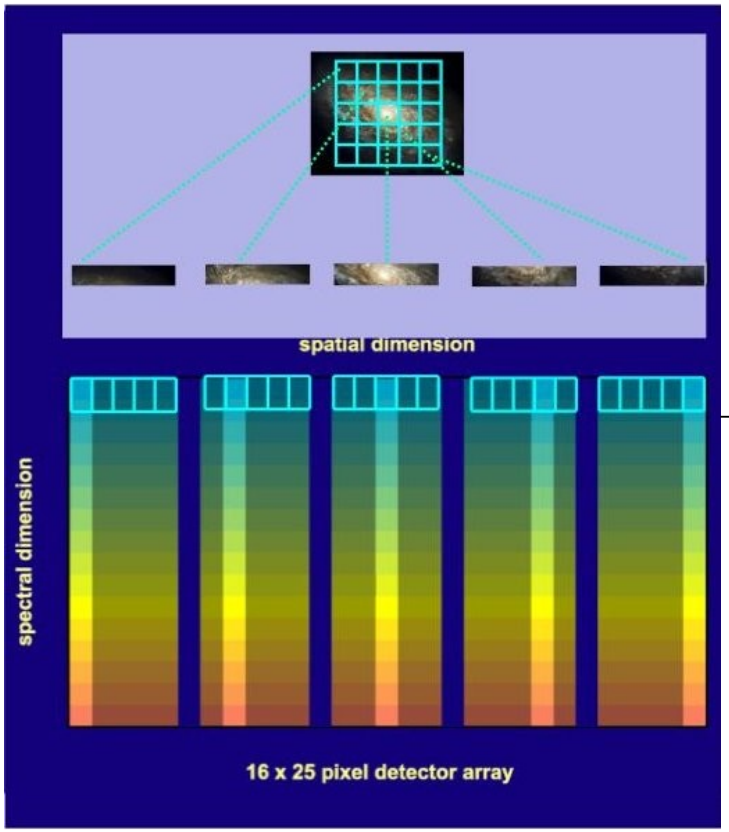
Calibration source Flux,

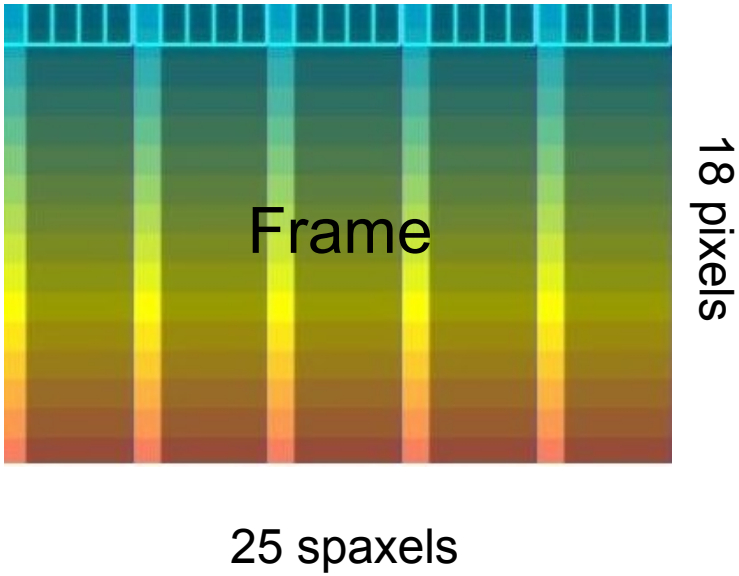
RSRF versions,

the telescope background.

```
HIPE> print calTree.spectrometer
PacsCalSpec Calibration Products:
  absoluteCapacitance              : FM, 4
  arrayInstrument                  : FM, 9
  badPixelMask                     : FM, 1
  beamSize                         : FM, 1
  beamsB2A                         : FM, 4
  beamsB2B                         : FM, 4
  beamsB3A                         : FM, 4
  beamsPerSpaxelB2A                : FM, 1
  beamsPerSpaxelB2B                : FM, 1
  beamsPerSpaxelB3A                : FM, 1
  beamsPerSpaxelR1                 : FM, 1
  beamsR1                          : FM, 4
  calSourceFlux                    : FM, 5
  calSourceFlux3x3                 : FM, 1
  capacitanceRatios                : FM, 7
  chopperThrowDescription          : FM, 2
  crosstalkMatrix                  : FM, 1
  darkCurrent                      : FM, 2
  detectorSortMatrix               : FM, 1
  discardRampHooks                 : FM, 1
  effectiveCapacitance             : FM, 1
  filterBandConversion             : FM, 1
  gprHall                          : FM, 3
  gprHallRedundant                 : FM, 3
  gratingJitterThreshold           : FM, 1
  keyWavelengths                   : FM, 1
  labelDescription                 : FM, 1
  littrowParameters                : FM, 1
  littrowPolynomes                 : FM, 1
  moduleArray                      : FM, 4
  noisyPixelMask                   : FM, 2
  nominalResponse                  : FM, 3
  nonLinearity                     : FM, 3
  observedResponse                 : FM, 1
  observedResponse3x3              : FM, 1
  offRatioB2A                      : FM, 4
  offRatioB2B                      : FM, 4
  offRatioB3A                      : FM, 3
  offRatioR1                       : FM, 4
  pointSourceLoss                  : FM, 3
  psf                              : FM, 1
  rampSatLimits                    : FM, 1
  readouts2Volts                   : FM, 1
  relCalSourceFlux                 : FM, 2
  rsrfB2A                          : FM, 4
  rsrfB2B                          : FM, 4
  rsrfB3A                          : FM, 4
  rsrfR1                           : FM, 3
  sensitivity                      : FM, 1
  signalSatLimits                  : FM, 1
  specProperties                   : FM, 1
  telescopeBackground              : FM, 5
  timedep                          : FM, 58
  wavePolynomes                    : FM, 3
  wavelengthGrid                   : FM, 1
```

Frames

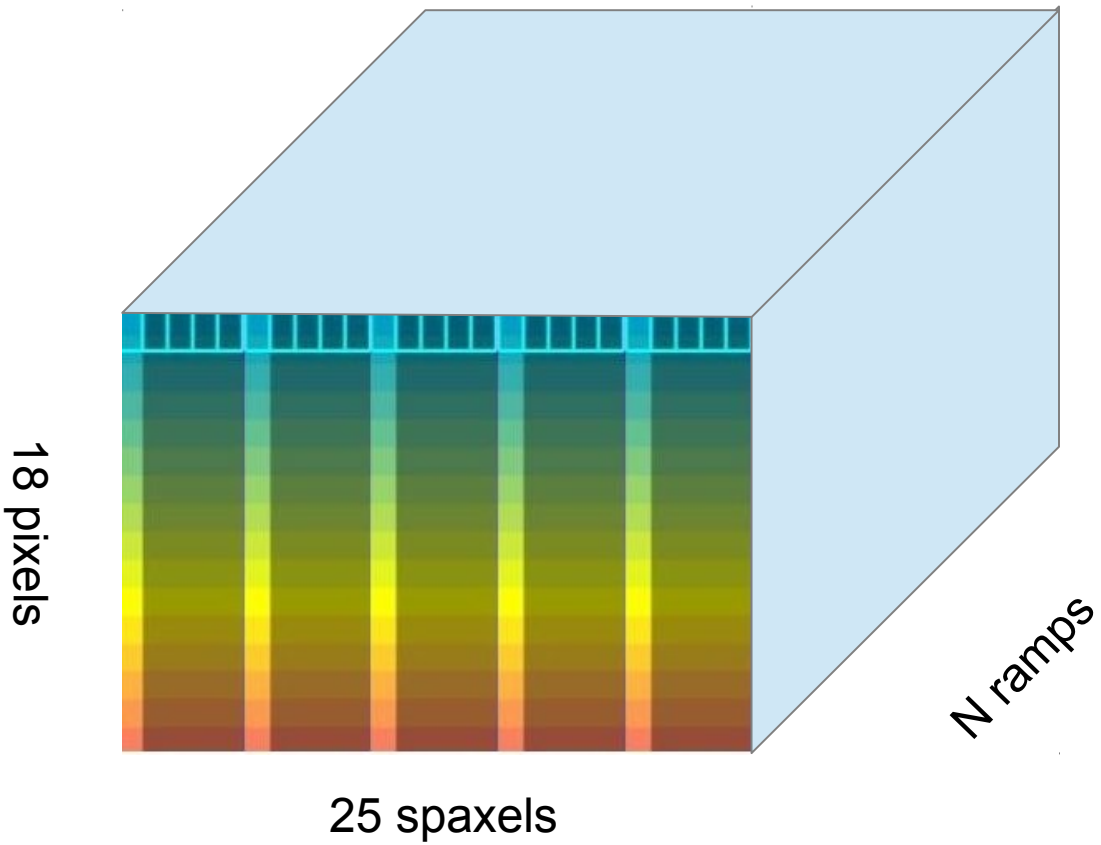




The natural structure to contain the data is the frame, a 2D double array 18x25.
In the spectral dimension, the two extreme pixels are engineering pixels and are not considered in the data reduction.

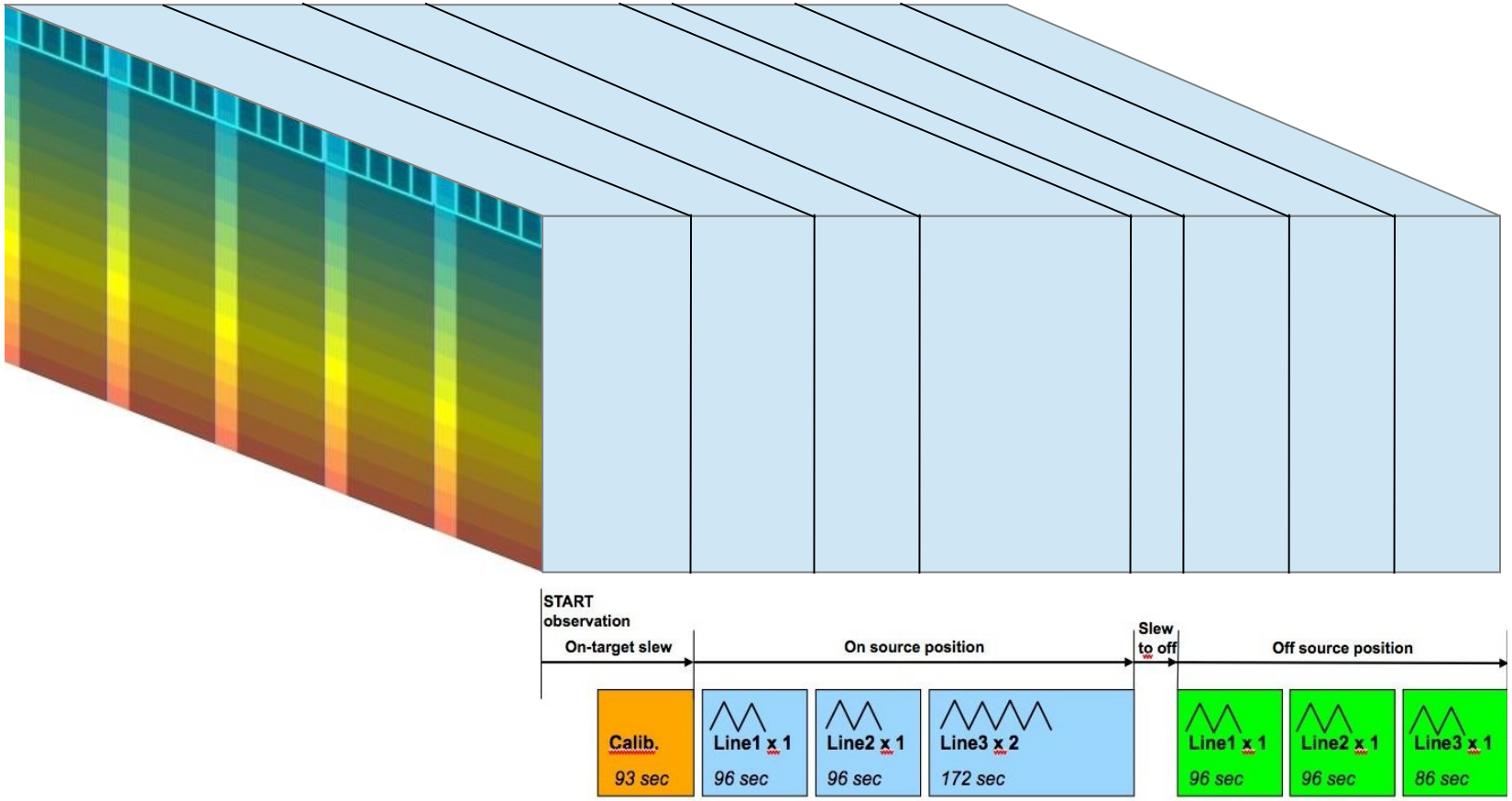
A set of frames

One observation is composed by a set of frames.
So, the entire observation is contained in a 3D array with size 18x25xN, with N the total number of ramps.



Slicing

The set of frames is naturally sliced into blocks of frames according to different pointings and wavelength ranges observed.



Slicing

The set of frames is contained in a structure called slicedFrames. Once the data are sliced, the structure slicedFrames will have several parts.
Parts containing only non-science data, such as slewing parts, are deleted and the module of the pipeline can run on separate slices.

Also the choice of the camera (red or blue) allows one to consider only one detector. Blue and red array are done in parallel.

The first slice is called the calibration block and it is used to compute the response of the detector for the observation. After this computation, also this slice is discarded and we are left only with science blocks.

slicedFrames

The command slicedSummary(slicedFrames) allows one to explore the slicedFrames.

```
HIPE>
HIPE> lev0 = obs.refs["level0"].product
slicedFrames = obs.refs["level0"].product.refs["HPSFITR"].product
HIPE> slicedSummary(slicedFrames)
noSlices: 1
noCalSlices: 0
noScienceSlices: 1
slice#  isScience  onSource  offSource  rasterId  lineId      band                    dimensions      wavelengths
0       true       both      both       0 0       [0]         ["R1","UNDEF"]          [18,25,8096]    wave not defined
HIPE> lev1 = obs.refs["level1"].product
slicedFrames = obs.refs["level1"].product.refs["HPSFITR"].product
HIPE> slicedSummary(slicedFrames)
noSlices: 4
noCalSlices: 0
noScienceSlices: 4
slice#  isScience  onSource  offSource  rasterId  lineId      band                    dimensions      wavelengths
0       true       yes       no         0 0       [102]       ["R1"]                  [18,25,1500]    204.198 - 206.302
1       true       no        yes        0 0       [102]       ["R1"]                  [18,25,1500]    204.197 - 206.301
2       true       yes       no         0 0       [103]       ["R1"]                  [18,25,1500]    120.127 - 123.910
3       true       no        yes        0 0       [103]       ["R1"]                  [18,25,1500]    120.127 - 123.910
```

Level 0

Level 1

We will discover that, at the beginning, there is only one slice. Then, at more advanced levels of reduction, there are more slices.

Masks

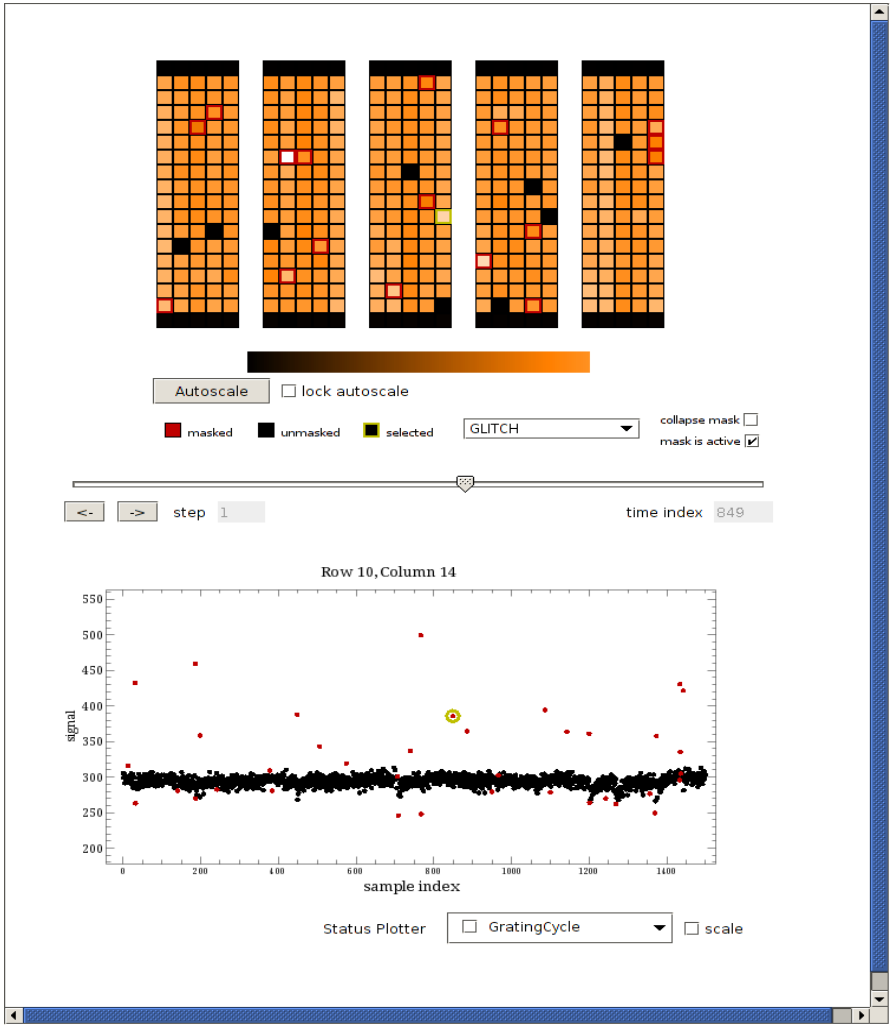
During the reduction, several masks are created to contain the flags of defective pixels and glitches. To see the list of masks created for each slice, type:

maskSummary(slicedFrames)

To see which pixel is masked:

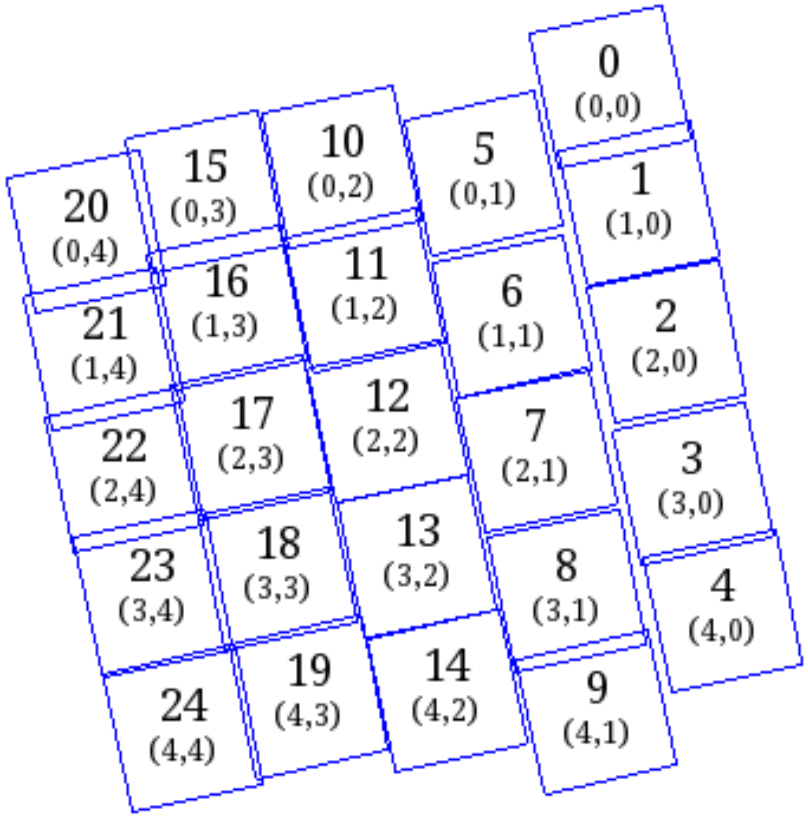
MaskViewer(slicedFrames.get(1))

where the number corresponds to the slice.



From frames to cubes

At the end of the pipeline, frames are reorganized in cubes with two spatial dimensions corresponding to the initial IFU array (5x5) and a wavelength dimension. SlicedSummary(cube) can be still used to read the metadata of cubes.



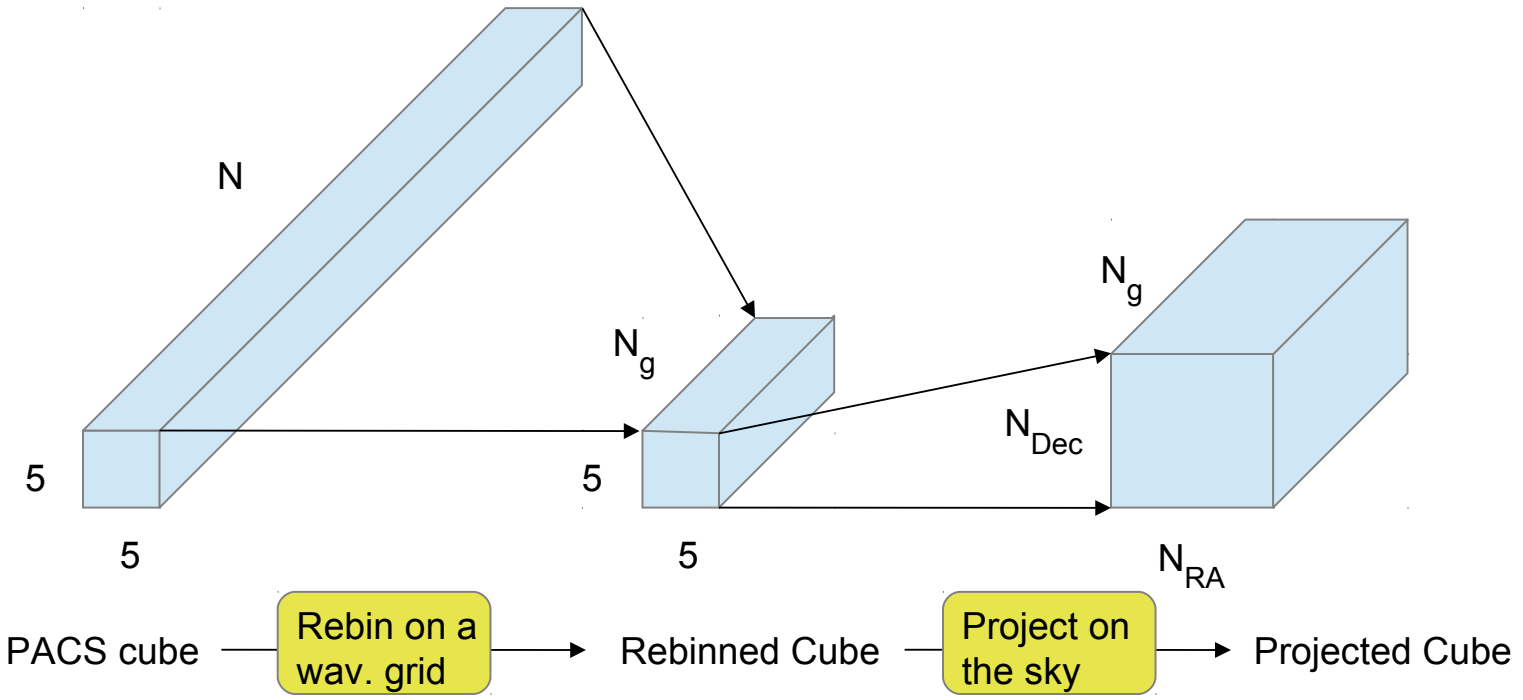
Map of the transformation from the 25 spaxels to the 5x5 spatial pixels.

Cubes

Cubes come in two flavors:

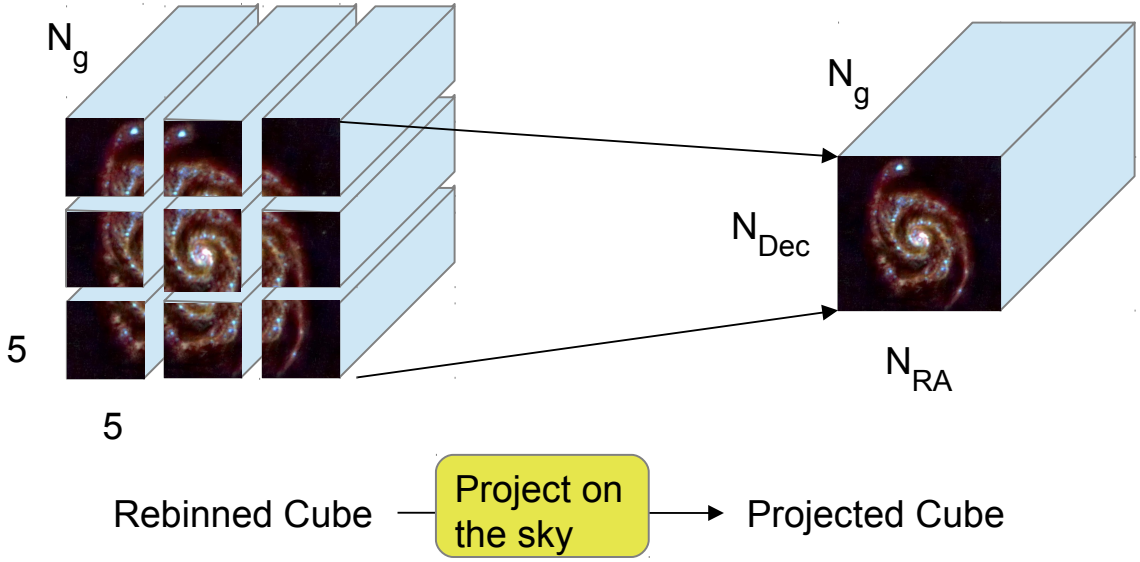
binned cubes, where the wavelength range has been binned into a grid,

projected cubes, where the spatial part has been projected on the sky and therefore is no more 5x5. The pixel size is 3 arcsec.



Cubes

In case of rasters, more than one rebinned cube is created for each line.
All the cubes are projected in a single projected cube.



Example of a 3x3 raster projected into a single Projected Cube.