Herschel online data processing workshop for newcomers

# PACS spectrometer data analysis

*Katrina Exter (Herschel Science Centre)*
*25/06/2014*

# PACS spectrometer data analysis

## Point sources and single pointings

At the end of Level 2/2.5

- The rebinned cubes — HPS3DR[B|R] — i.e. with the native IFU footprint, are calibrated for extended emission
- For point sources centred in the IFU (in spaxel 2,2), a task extractCentralSpectrum is provided to extract and point-source flux calibrate the spectrum
- It works by comparing the distribution of flux in the centre of the IFU to that for a correctly-centred point source, and at the same time applying a correction for slight pointing offsets and applies the point source flux correction (to account for the flux "missing" because the beam is larger than a single spaxel)
- It works well on sources of continuum flux level 5—10 Jy; for fainter sources the accuracy is much lower as the point source is hard to "see"
- This task was explained in more detail at the workshop in April 2014

http://herschel.esac.esa.int/twiki/pub/Public/OnlineSpectroscopyWorkshop2014/PacsSpectroFluxPointsources.pdf

# PACS spectrometer data analysis

## Semi-extended sources

At the end of Level 2/2.5

- The rebinned cubes — HPS3DR[B|R] — i.e. with the native IFU footprint, are calibrated for extended emission
- For slightly extended sources centred in the IFU (in spaxel 2,2), a task pacsExtendedToPointCorrection will turn point-source calibrated and extracted spectra (using extractCentralSpectrum) into extended source calibrated spectra
- It works by computing the correction to the point-source calibration based on the geometry (size, position angle, surface brightness distribution) of a slightly extended source compared to a point source
- You need to know this geometry to usefully use the task
- Slightly extended source could be from a few arcsec to a few spaxels in diameter
- This task was explained in more detail at the workshop in April 2014

http://herschel.esac.esa.int/twiki/pub/Public/OnlineSpectroscopyWorkshop2014/SpectrometerWorkshop_PACS_Session3_EPuga.pdf

# PACS spectrometer data analysis

## Extended sources: mapping

At the end of Level 2/2.5

- The projected cubes — HPS3DP[B|R] — are a mosaic of individual raster pointings and are the science-grade product mapping observations (i.e. with the beam fully-sampled)
- Two tasks can create these cubes: specProject (done in the bulk processing) and drizzle. Drizzle is recommended for line and short range scans, but you need to run this task yourself (i.e. redo the end of the pipeline)
- See the PACS Data Reduction Guide to learn more about the differences between these tasks
- These cubes have a regular spatial grid but a non-equidistant spectral grid: thus the WCS is incomplete
- To make them easier to read into other software, we will provide these cubes with an equidistant spectral grid in Track 13
- It is also possible to make line maps by fitting the rebinned cubes, and then doing the spectral—spatial projection on the fitting results (rather than the entire cube)

# PACS spectrometer data analysis

## Extended sources: not mapping

At the end of Level 2/2.5

- The rebinned cubes — HPS3DR[B|R] — are the science-grade product for extended sources observed in a single pointing or tiling mode (i.e. the beam is undersampled)
- Science measurements should be made on these cubes (bearing in mind the limitations of having undersampled the beam)
- The rebinned cubes have an irregular spatial grid and a non-equidistant spectral grid: thus the WCS is non-existent, although the RA and Dec of each spaxel is recorded in an extension in the cubes
- For tiling observations, with larger fields-of-view covered, it can be difficult to manage the potentially large set of cubes: we are working on a task to create an equivalent to the projected cubes, using interpolation rather than projection to create a regularly gridded cube
- The interpolated cubes will have a regular spectral and spatial grid, and be easier to read into other software

# Short demo script

```
## Script to accompany PACS Spectroscopy demo: Herschel Workshop for Newcomers, June 25 2014
## Katrina Exter
##
## Playing with data
##

"""
The Pacs Spectral Footprint Viewer: plot the footprint of a rebinned cube on an image to see
where you are pointed to. In this case: is the source well centred in the cube?
"""
# Spectroscopy of AFGL618
obs=getObservation(1342225838,useHsa=True)
# get one one of the "rebinned" cubes
aCube = obs.refs["level2"].product.refs["HPS3DRB"].product.refs[0].product
# a photometry observation of this source
obs1=getObservation(1342193133,useHsa=True)
# grab one of the blue maps (any one will do)
aMap = obs1.refs["level3"].product.refs["HPPPMOSB"].product.refs["l3mosaicB"].product
# open the viewer on the image and load the cube into it
fp = pacsSpectralFootprint(aMap)
# Then the "fp" variable has some methods to add more footprints, or to
# access the Display inside for e.g. custom annotations:
fp.addFootprint(aCube)
fp.display.zoomFactor = 4
fp.display.setCutLevels(100)
...
```

# Short demo script

```
...
# save the products to disk
#simpleFitsWriter(aCube,"/Users/katrina/aCube.fits")
#simpleFitsWriter(aMap,"/Users/katrina/aMap.fits")
"""
Look at a mapping observation and explore the "projected" cube with the Spectrum Explorer
"""
# NGC 40 chopNod, Nyquist mapping in the red (3x3 raster with offsets of 22"x24”)
obs=getObservation(1342246640,useHsa=True)
obsSummary(obs) # text summary of the observation
# The cubes are sliced on wavelength, which one do I want?
# Open the Observation Viewer on obs, to go the Level 2 "HPS3DPR" directory and click on the +
# See 4 cubes: hover over them with the mouse and the pop-up gives information on each cube, including the wavelengths
# Let's take the cube #2 (3rd slice as the counting starts at 0) and #3
bCube=obs.refs["level2"].product.refs["HPS3DPR"].product.refs[2].product
cCube=obs.refs["level2"].product.refs["HPS3DPR"].product.refs[3].product
# save the products to disk
#simpleFitsWriter(bCube,"/Users/katrina/bCube.fits")
#simpleFitsWriter(cCube,"/Users/katrina/cCube.fits")

# Now use the Spectrum Explorer, Cube Toolbox, and SpectrumFitterGUI
# See the demonstration recording as this was not done on the command line
```