

## **Spectral Cube Visualisation and Explorer Tool from the Herschel Interactive Processing Environment (HIPE)**

Jeroen de Jong,<sup>1</sup> Alain Gueguen,<sup>2</sup> Jaime Saiz,<sup>3</sup> Katrina Exter,<sup>3</sup>  
Wim De Meester,<sup>4</sup> Werner Salomons,<sup>5</sup> Carolyn McCoe,<sup>6</sup>  
Edward Polehampton,<sup>7</sup> Philip Appleton,<sup>8</sup> Patrick Morris,<sup>8</sup> Eckhard Sturm,<sup>1</sup>  
and the Herschel Data Processing Software Development Team

<sup>1</sup>*Max-Planck Institute for Extraterrestrial Physics (MPE), Garching bei München, Germany*

<sup>2</sup>*Observatoire de Paris Meudon, Paris, France*

<sup>3</sup>*Herschel Science Center, Madrid, Spain*

<sup>4</sup>*Instituut voor Sterrenkunde, KU Leuven, Leuven, Belgium*

<sup>5</sup>*SRON, Groningen, The Netherlands*

<sup>6</sup>*Department of Physics and Astronomy, University of Waterloo, Canada*

<sup>7</sup>*RAL, Oxfordshire, UK and University of Lethbridge, Canada*

<sup>8</sup>*IPAC, Caltech, Pasadena, USA*

**Abstract.** We present the interactive tool for visualizing and exploring spectral cubes of the Herschel Interactive Processing Environment (HIPE). With this GUI tool, one can easily plot the spectra within the cube, compare several cubes from the same region of the sky, and run a number of analysis tasks on them (such as computing velocity/dispersion maps or position-velocity diagrams). The tool is integrated within the Spectral Explorer from HIPE, which offers additional functionality such as an interface to the spectral fitter tool. All the actions with HIPE tools produce Python code, which can later be reused in scripts for automation. Various results can be saved as FITS files and loaded into other astronomical analysis tools.

### **1. Introduction**

The Herschel Interactive Processing Environment (HIPE, see Balm (2012) and Ott (2011)) was originally developed for the Herschel Space Observatory (Pilbratt et al. 2010), but many of the tools it contains are useful for general astronomical data analysis and visualisation. HIPE has been designed to provide a full environment with which astronomers not only can reduce their Herschel data, but also analyse their results interactively, which is an important part of producing science-quality data. Since all the Herschel instruments (PACS, SPIRE and HIFI) produced spectral cubes, we also developed an interactive tool for exploring cubes. This tool, embedded in HIPE, can be used with the many other tools of HIPE for FITS I/O, numerical analysis, plotting and scripting in Python to do a full analysis on spectral cubes of any instrument.

In this paper we will show in detail what this tool can do, how you can obtain it, and also how you can use it with data from other instruments.

## 2. User Interface and Features

The GUI consists of four panels (see Fig. 1). In the cube selection panel (lower-left) one can select individual spectra or elliptical, rectangular or slit regions. These selected spectra can either be shown directly, or the coordinates sent to one of the analysis tasks to integrate over a spatial region, to crop the cube or to produce a position-velocity map over a slit. When an analysis task receives such coordinates, then they get filled in the input boxes of that Task's GUI. The spectrum plot panel (upper-left) can also be used to select spectral regions, e.g., for fitting a continuum.

The tool has the following main features:

- Visually explore the spectra within a cube.
- Perform the following analysis tasks on the cubes:
  - **extractCentralSpectrum**: Extract a 1D spectrum from a cube by summing or averaging a spatial region.
  - **computeVelocityMap**: Compute velocity and dispersion maps by means of fitting or moment computation.
  - **cropCube**: Crop a cube spatially and spectrally.
  - **subtractBaselineFromCube**: Fit a polynomial baseline to a selected continuum and subtract this from all the spectra in the cube.
  - **integrateSpectralMap**: Integrate over a given spectral region to produce a integrated map.
  - **computePVMap**: Compute a position-velocity map.
- Each of the above tasks is integrated in the GUI to select spatial and spectral regions.
- Provide an interface to other tools such as the Spectrum Fitter in HIPE. With the Spectrum Fitter one can fit any combination of models to the spectra in a cube.
- Compare the spectra of several cubes simultaneously. One can connect several cubes by world coordinates and then plot the spectra from the same region simultaneously.
- Produce Python code to automate tasks later in scripts. With such scripts one can analyse the results from the tasks further, make publication-quality figures with the HIPE plot package, or export to results to FITS files to further use them in other applications.

### 2.1. Data Analysis Algorithms

The tasks contain several advanced data analysis algorithms which we describe here in more detail.

For the velocity and dispersion maps (task **computeVelocityMap**) one can either choose a Gaussian fit or compute the statistical moments. The latter is recommended when the line deviates significantly from a Gaussian profile. The moments computation

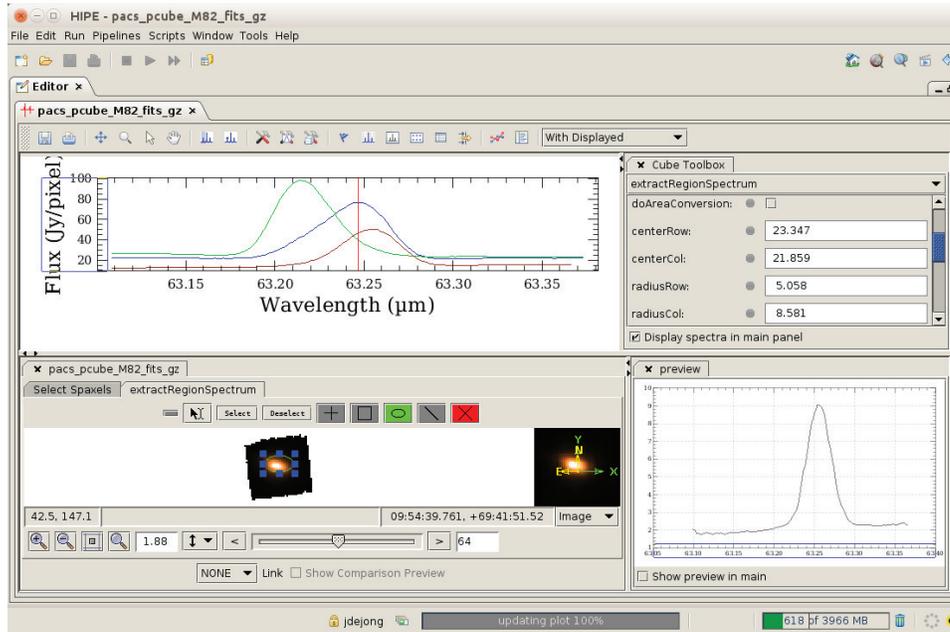


Figure 1. User interface of the spectral cube visualisation and explorer tool. The upper left panel shows the plots of the selected spectra or results of tasks. The upper right panel contains the GUI panels of the tasks. The lower left panel contains the cube selection toolbox and the lower right panel a preview of the spectrum under the mouse.

uses weights to determine how significantly a data-point belongs to the spectral line. This is especially important for the dispersion computation, since the second moment depends on velocity squared. This means that noise in the wings contribute significantly to the uncertainty in the dispersion. These “line strength” weights are computed as follows:

1. Divide the spectrum by its boxcar-smoothed version.
2. Compute the standard deviation  $\sigma$  over the residual.
3. Compute the  $N\sigma$  difference between the line and the continuum. The spectrum should have been baseline subtracted/ divided, so that the continuum is either 0.0 (emission) or 1.0 (absorption).
4. The weight is then  $w_i = \text{ERF}(N\sigma_i)$ . (ERF is the error function.)
5. Finally, a derivative filter is applied to remove single data-point outliers outside the line profile. These filtered weights are computed as  $w_{\text{filt}_i} = w_i / ((\alpha \bar{w}_i)^2 + 1)$  with  $\bar{w}_i = |w_{i+1} - w_i| + |w_i - w_{i-1}|$  where  $\alpha$  is a threshold factor, which determines the steepness of the filter.  $\alpha$  is set empirically to 1.0.

Other important algorithms relate mostly to computing the spatial weights of the overlap between the selected regions and the spaxels. These weights are used in the **extractRegionSpectrum** and **computePVMap** tasks to accurately compute the summed/ averaged spectra.

### 3. Using as Standalone Tool

One can use this tool after downloading and installing the core HIPE from <http://www.cosmos.esa.int/web/herschel/hipe-download>. This will require about 1 GB of disk space, but that is nowadays not a problem anymore. Other cube viewers like the CASA 3D viewer (Kümmel et al. 2012) require a similar amount of resources. HIPE can read FITS files from any instrument, but since there is not yet a good standard for spectral cubes, one probably needs to do a few conversions afterwards. Typically, it may be that the WCS keywords are not in the right Header Data Unit, or that some units are not read automatically. We tried e.g., a cube from the SINFONI instrument (Eisenhauer et al. 2003), which only required a bit of code to set the units:

```
from herschel.share.unit import FluxDensity, Length
map_masked_SC_j = fitsReader(file = 'map_masked_SC_j.fits')
scube = SpectralSimpleCube(map_masked_SC_j)
scube.waveUnit = Length.MICROMETERS # what should be the unit ???
scube.fluxUnit = FluxDensity.JANSKYS # what should be the unit ???
scube.fluxDescription = "Flux"
```

After running these commands you can open the spectral cube tool by just clicking on the `scube` variable in HIPE.

It is possible to extract this tool with the Spectrum Explorer from HIPE as a standalone tool. HIPE has been entirely written in Java and Jython (the Java implementation of Python) and has been split into many different modules (jar files). In order to make a standalone tool one has to collect all the modules needed by this tool and remove any references to parts of HIPE which are not relevant to this tool from the code. One will find out what is needed by compiling the code and running the test harnesses.

**Acknowledgments.** The Herschel Science Centre (ESA), the Instrument Control Centres (HIFI, PACS and SPIRE) and the NASA Herschel Science Center jointly manage and contribute to the Herschel Data Processing System

### References

- Balm, P. 2012, in ADASS XXI, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461 of ASP Conference Series, 733
- Eisenhauer, F., Abuter, R., Bickert, K., et al. 2003, in Instrument Design and Performance for Optical/Infrared Ground-based Telescopes, edited by M. Iye, & A. F. M. Moorwood, vol. 4841 of SPIE Conference Series, 1548. [astro-ph/0306191](http://arxiv.org/abs/astro-ph/0306191)
- Kümmel, M., et al. 2012, in ADASS XXI, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461 of ASPC Conference Series, 833
- Ott, S. 2011, in ADASS XX, edited by I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots, vol. 442 of ASP Conference Series, 347
- Pilbratt, G. L., Riedinger, J. R., Passvogel, T., et al. 2010, *A&A*, 518, L1+