Use case: mapping sparse spatial data with TOPCAT

This use case describes a workflow related to large hyperspectral datasets.

In this example you will use data from the VIRTIS/Rosetta experiment and study the distribution of observation of the surface of comet 67P. The workflow consists in: retrieving the data files; preparing a data table restrained to spectra of interest; mapping and analyzing the footprints in TOPCAT.

1. Getting the data

The data are available on the PSA archive and on the PDS Small Bodies Node. The PSA has a graphic interface (https://archives.esac.esa.int/psa/#!Home%20View, Fig. 1) which helps select data files from their global properties. If the data are distributed in an **EPN**-TAP service, it can be searched and filtered using the VESPA portal (http://vespa.obspm.fr/, Fig. 2).

The PSA interface can help identify files containing data of interest, although to a lesser extent than the VESPA portal.

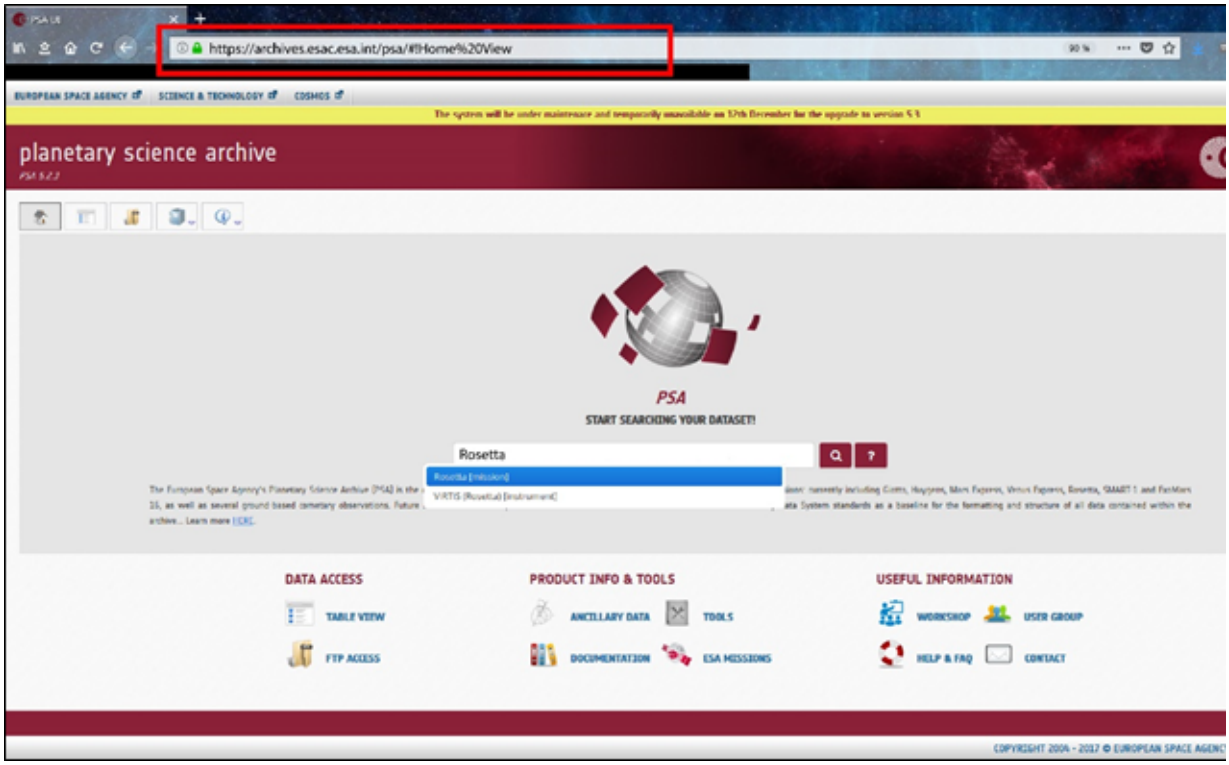


Fig. 1 PSA web interface.

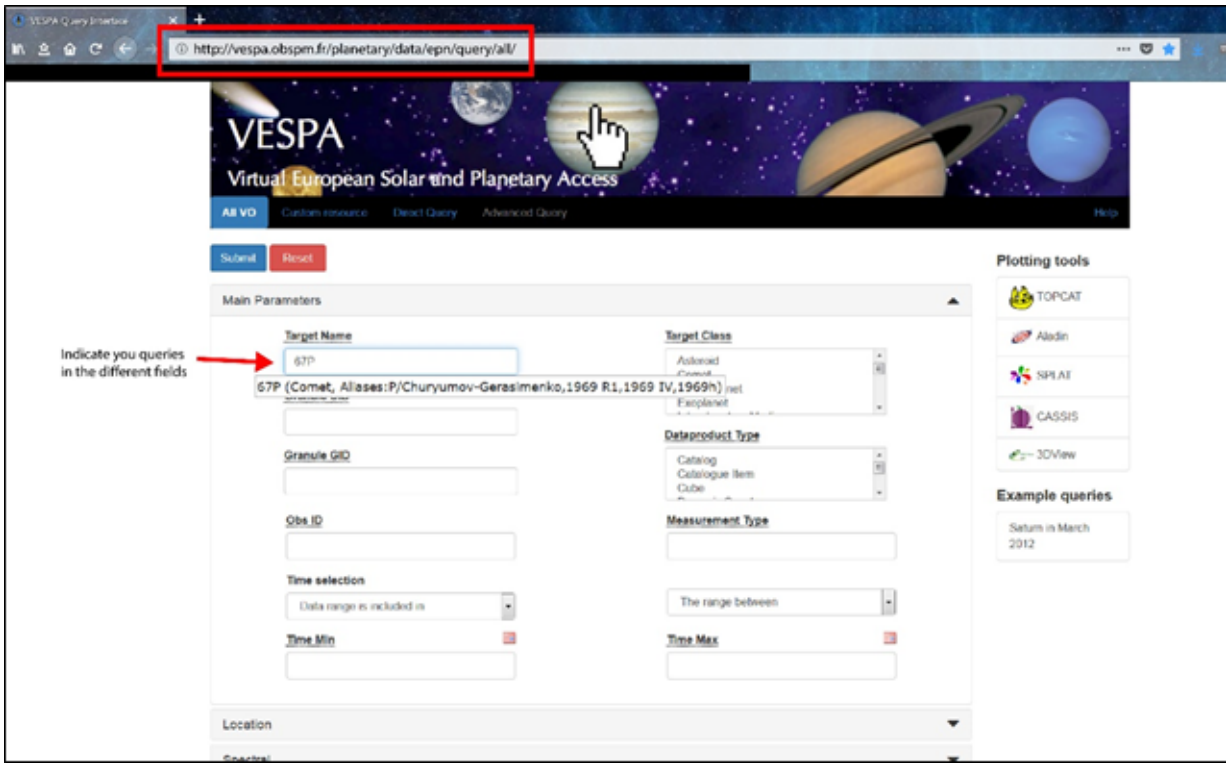


Fig. 2 VESPA web interface.

2. Preparing the data table

In this step, you want to prepare a table of individual spectra of interest and store it in a format suitable for TOPCAT. This file format can be VOtable or the FITS format which are bith current standards, as they contain self-described data. This step is important when studying a global dataset, because it settles the tedious process of going through a number of files once for all.

There are several possibilities to prepare the FITS/VOtable for TOPCAT:

1) Data selection and writing can be done in IDL (Fig. 3) or a similar language. This is the general case where data files are retrieved from the PSA or PDS archives (files need to be open and spectra need to be filtered). IDL (or its GDL open source clone) is anyway required to read PDS files and select individual spectra. Even if the data are loaded in IDL at this point, using TOPCAT is still useful because of its superior ability to integrate data into healpix cells in the 3rd step of this tutorial.

2) If the data are distributed as an EPN-TAP service describing individual spectra (a project for this particular dataset) the VESPA portal, or a TAP query issued from TOPCAT, can alternatively be used to retrieve the selected information and write it as a table; this would mainly save the next step.

The goal is to create a table where each line corresponds to a single pixel / observation / acquisition and each column to a physical (e.g. radiance at a given wavelength) or geometric (e.g. latitude/longitude) parameter or any other information (spectral parameters, reference of the hyperspectral cube, identification number of this acquisition, period/duration of the acquisition…). Only metadata are stored in the table, because the aim is to study the footprints. Pixel / observation / acquisition here refers to a pixel in a hyperspectral VIRTIS cube. In the present case, we want to retain only acquisitions that match several criteria: FoV intercepting the surface, given illumination conditions, minimum resolution, etc.

The Fig. 3 below illustrates the formatting with IDL.

```idl
pro tuto_Vo_VIRTIS, listofcube, listofcubegeometry

for i=0,n_elements(listofcube)-1 do begin                                    1.
  cube = virtispds(listofcube(i))
  geo = virtispds(listofcubegeometry(i))

; here you will filtered/make operations on data. For example:
  reflectance = mean(cube.qube(138:148,*,*),dim=1,/double) ; mean reflectance around 2.3µm
  incidence = geo.cube(17,*,*) ; extract the incidence
;  latitude = ....
;  longitude = ...
;  .
;  .
;  .
;  Thanks to the geometric cube I am able to extract information which allows to map the spectra

;  In order to write the information in a VOtable you must store these variable in an array at each iteration:
  all_reflectance = [reflectance,all_reflectance]
  all_incidence = [incidence,all_incidence]
endfor

; then put all the array corresponding to a parameter (a futur column in your VOtable) in a structure)   2.
my_structure = {Reflectance: all_reflectance,$
  Incidence: all_incidence}

;  And finally, write your VOtable thanks to write_vot.pro or to vobs_Struct2VOTable                      3.
vobs_Struct2VOTable,my_structure,'/here/is/my/pretty/VOtable.vot'
END
```

Fig. 3 Illustration of the data processing with IDL.

There are three steps:

1. Reading and filtering the data files

2. Packing them into a structure

3. Writing the structure as a VOtable or as a FITS table

1. The first step is data access and filtering, the most time-consuming process. PDS3 data files can be read under IDL with the readpds library from the PDS Small Bodies Node, or with the all-purpose virtispds library from LESIA (also running in the open source GDL environment). In this example, the input is a list containing the path of the hyperspectral cubes to process – the subset used here covers more than 6 months of VIRTIS/Rosetta observations in the visible range. All geometry parameters have been computed during the earlier calibration step using the SPICE library and are ready for use in the archive.

2. The information of interest is extracted from calibrated spectral cubes and geometry cubes: e.g. radiance, reflectance, incidence /emergence/phase angles, id numbers, altitude, elevation, acquisition time, cube from which it is extracted, latitude(s)/longitude(s), x/y/z coordinates… anything potentially useful to study the repartition of observations at the comet surface. Each parameter will constitute a column in the final table.

Cubes are processed in a loop; the information is extracted for the current cube, each parameter in a different array, then concatenated in a global array for all sessions. Spectra that cannot be exploited in a later phase may be filtered out during the extraction process, based on several criteria (observations angles, spacecraft altitude, signal threshold…). After the extraction process, the parameter arrays are associated in a structure, which is the required input to the table writing routines.

3. The final step consists in writing the FITS and/or the VOtable. While both formats will provide the same result, the FITS format is probably the most suitable since it is written in binary while VOTable in ASCII. Then, in the present case the table size is ~ 4GB in the VOTable format and only ~1GB in FITS format. As explain below, the loading time in TOPCAT is also faster for FITS table.

A FITS table can be write from IDL structure with:

- **mwrfits.pro** (https://idlastro.gsfc.nasa.gov/contents.html part of the Astron NASA Library - basic syntax is: "*MWRFITS, your_structure, filename ,/CREATE*")

At least two routines are available to write VOtables from IDL:

- **write_vot.pro** (https://github.com/epn-vespa/IDL_VOtable); requires the stilts java library to be accessible (current version runs under Linux, Mac and Windows)

- **vobs_Struct2VOTable.pro** (http://www.heliodocs.com/php/xdoc_list.php?dir=$SSW/vobs/gen/idl part of the SSW library)

Note that the FITS table must be written in binary (default behavior) to gain space and reading time.

Ideally, each parameter/column should be described with name, unit, and UCD for future reference and use in other tools. However, TOPCAT can ingest the data in this basic presentation.

3. Data visualization in TOPCAT

Our table gathers all relevant information from many files, which can now be handled easily. This file can be read by the TOPCAT tool (http://www.star.bris.ac.uk/~mbt/topcat/) which provides very powerful plotting functions (Fig. 4).

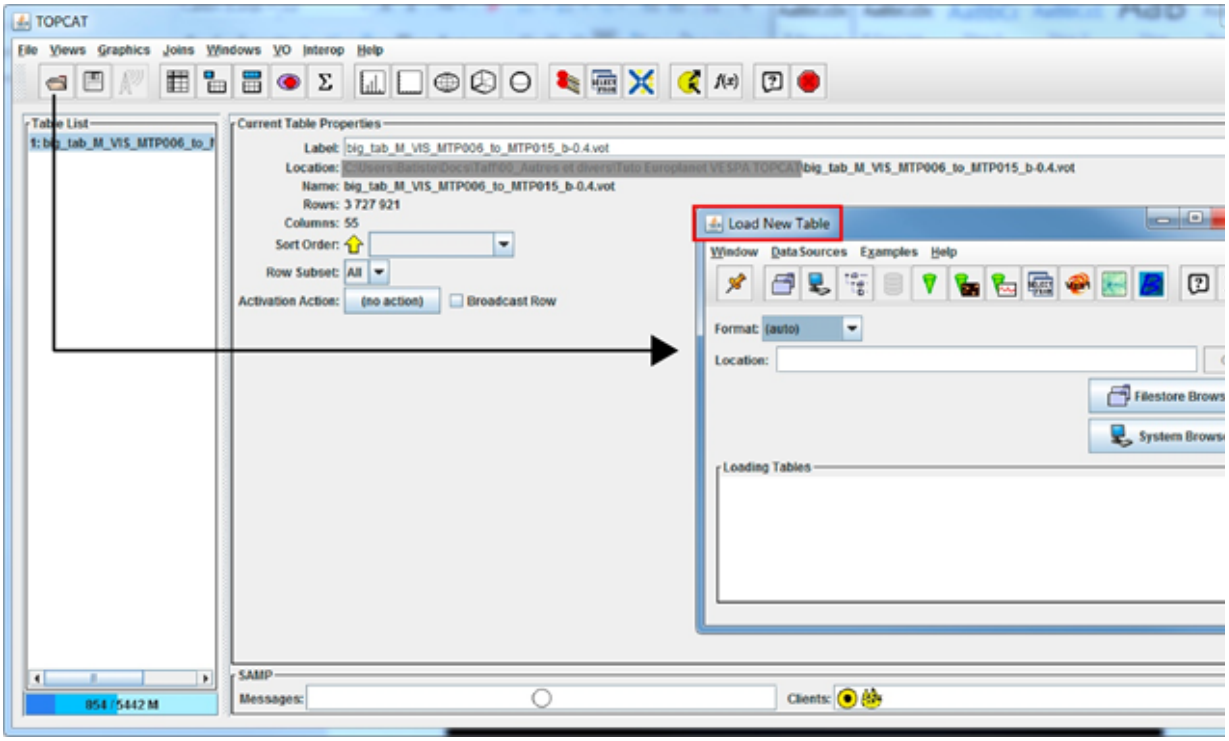


Fig. 4 Open a new table from the main interface by browsing your directory.

Our table contains 3 727 921 lines and 55 columns. TOPCAT is fast, even with huge tables. To read the VOTable on a busy Intel Core i7 @ 2.40GHz with 24Gb of RAM takes 2.15min (Java in 64 bits). However, the same table, written in FITS format, will be read instantaneously by TOPCAT. This table can be browsed from the interface (Fig. 5), which is useful to identify with precision an observation: when clicking on a dot in a plotting window, the corresponding line is highlighted in the table (and vice versa).

TOPCAT can perform further filtering of the data, so that you don't need to get back to the previous step. A standard way to achieve this is to define a new subset: click the **Display subset** icon (red and violet ellipses), then the **Define a new subset** icon (green cross) and enter an algebraic expression (you can also define a subset from the table itself using the top left icon in this window).

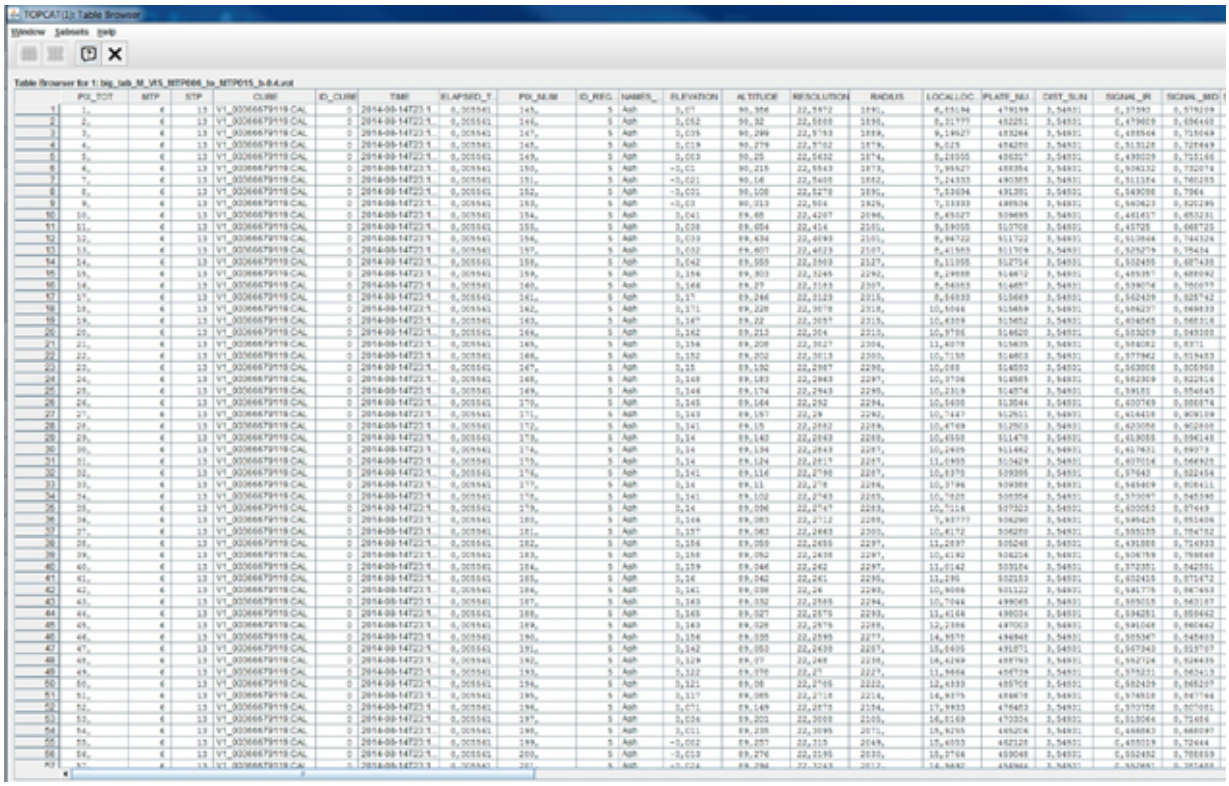


Fig. 5 Browse your table.

In the following, our goal is now to identity and extract observations in a specific location (on the Imhotep region of comet 67P) and observed under specific illumination conditions.

3.1 Quick-look maps

You first want to have a quick look of the spatial coverage of our data. To do this, use the “**plane plotting window**” mode, which allows us to plot any pair of columns against each other, with many options.

In the **Position** tab, select X= Lon_CENTER and Y= LAT_CENTER to display a cylindrical map (Fig. 6). Each line in the table (i.e. acquisition/spectrum) is plotted as a red dot. Since dots are plotted at the coordinates of pixel center, they do not represent the actual coverage of the pixel at the surface. By default, the **Auto mode** is on, and data are integrated inside the symbol used. Data density is represented with a given color: the more data points in the symbol footprint, the darker the area. If you enlarge the displayed area, the colors will change until the points are no longer superimposed. Changing the symbol size and shape in the **Form** tab affects the rendering. In Fig. 6 & 7, color variations represent the data coverage of the surface, which is partly controlled by the irregular relief.

Other customization options are available in the **Form** tab. For example, a more complete and user-configurable **Density** mode is available and can be adjusted with different parameters. Fig. 7 displays such a density map with a rainbow color table, in which denser regions are in red and less dense regions are in blue. The **Density** mode allows you to tweak the displayed range in several ways. The **Aux** mode makes it possible to map another parameter in the cylindrical frame, while the **Weighted** mode can display the statistics of the third parameter, computed from the points covered by the symbol.

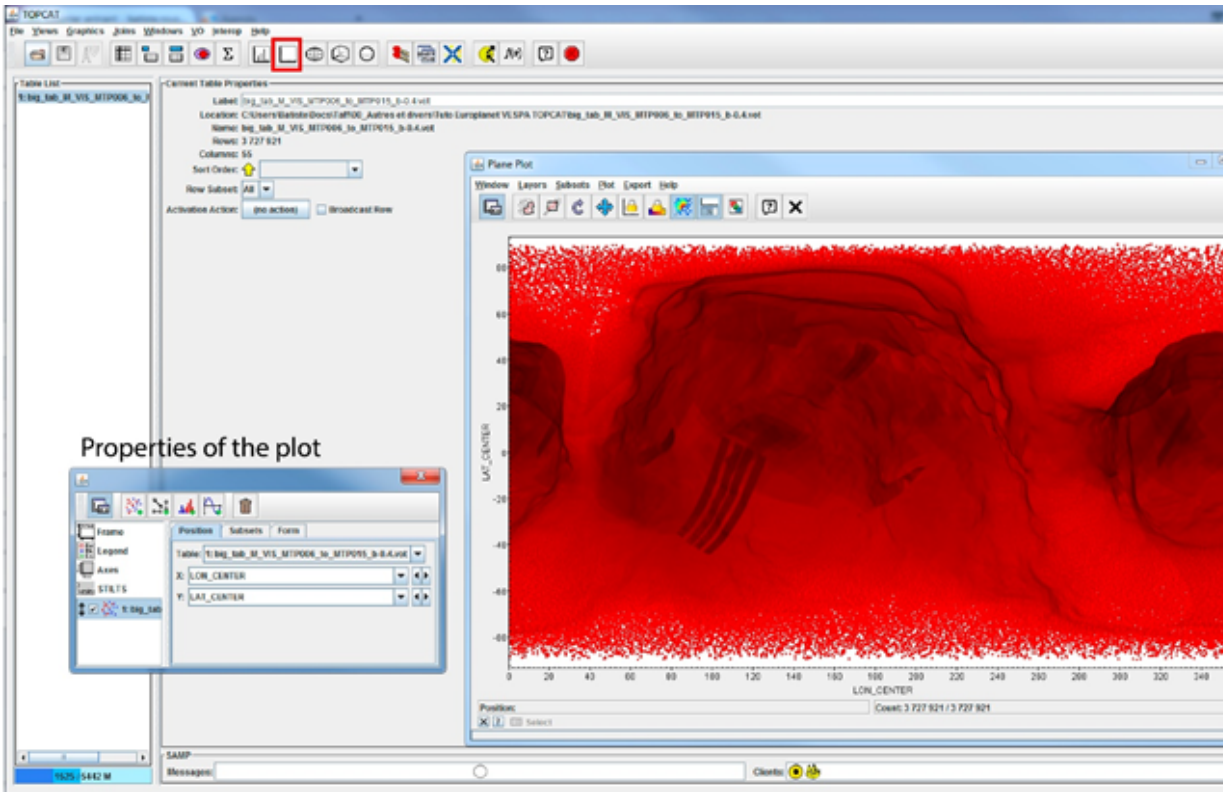


Fig. 6 Plotting longitude vs. latitude gives a...MAP ! By default, a single-color density representation is used.

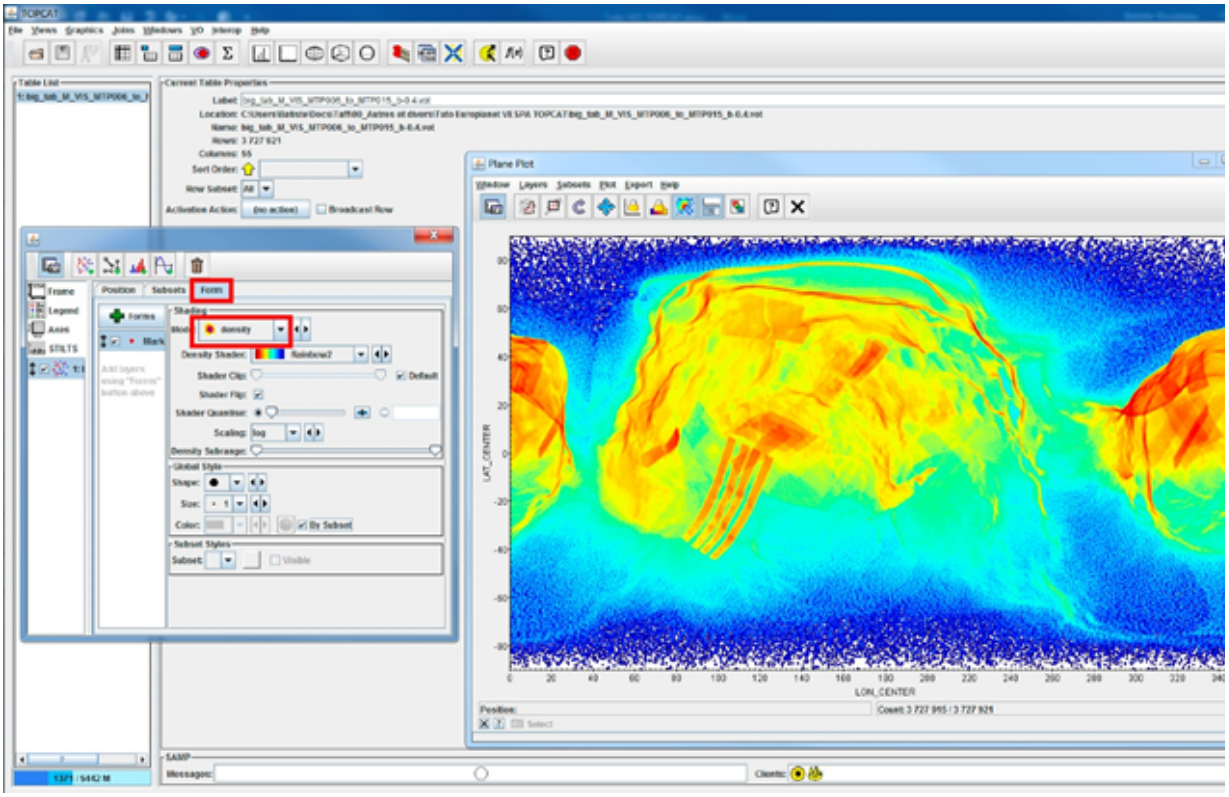
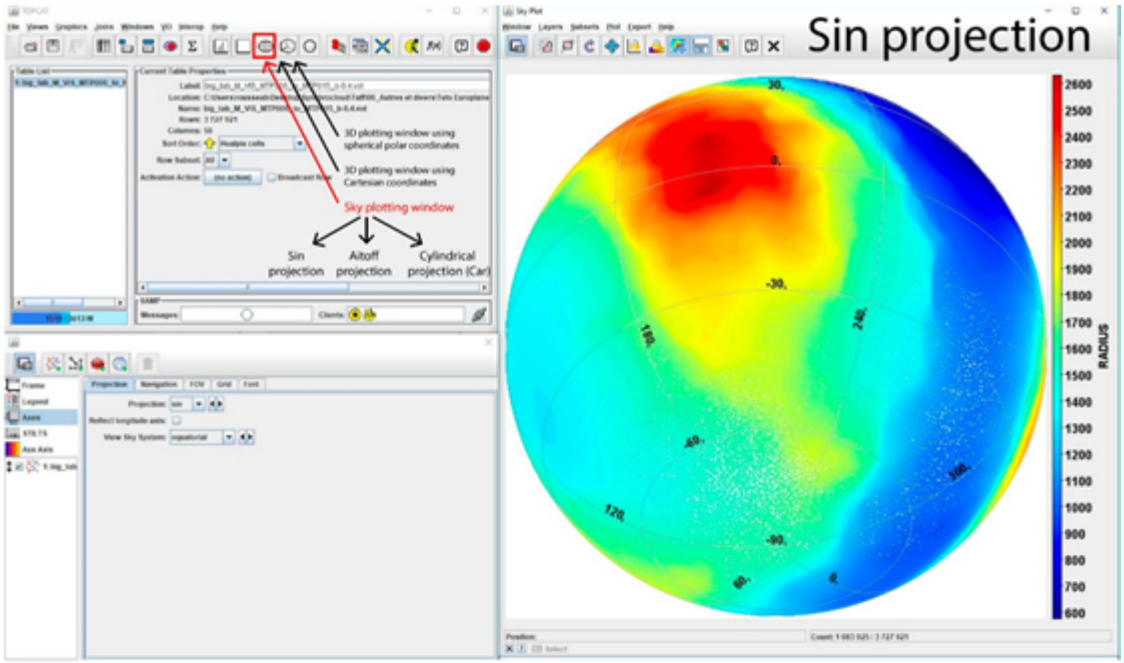
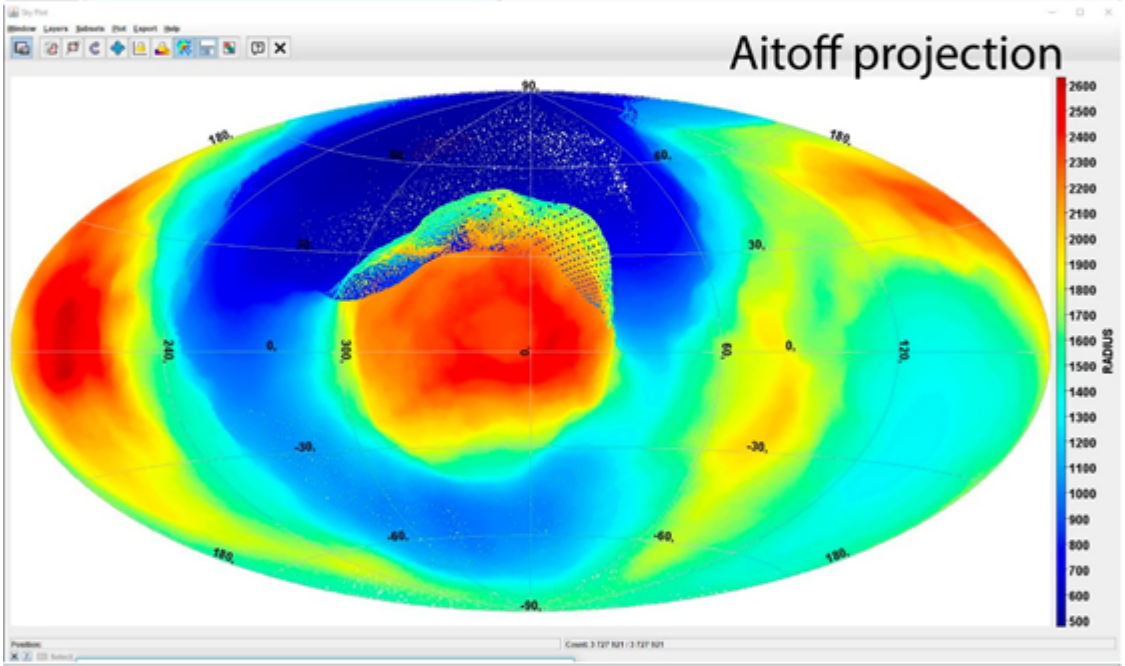


Fig. 7 Density map. This option is available is the form tab as well as other customization options.

Spatial information can be mapped with other projections using dedicated tools. Fig. 8 shows three map projections using the **sky plotting** option based on latitude and longitude. The projection type is available in the **Axes** settings (left menu of Sky plot window) + **Proj ection** tab, while the coordinate grid can be set in the **Grid** tab. In the example, the radius (from the 3D shape model) at each FoV center is plotted. Tools such as **3D plotting window using Cartesian coordinates** and **3D plotting window using spherical polar coordinates** (see top of the Fig. 8 for the access) allow you to use x/y/z coordinates (latitude, longitude and radius, respectively) to display a 3D representation.


3D plotting window using spherical polar coordinates
3D plotting window using Cartesian coordinates
Sky plotting window
Sin projection
Aitoff projection
Cylindrical projection (Car)
Sin projection
RADIUS


Aitoff projection
RADIUS

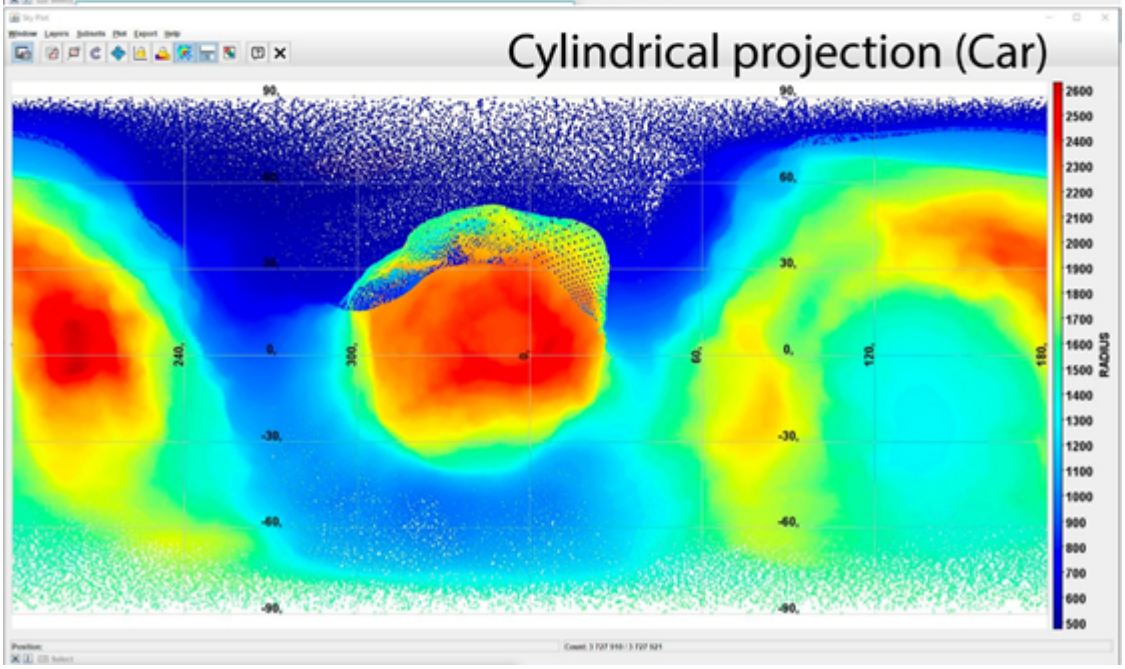

Cylindrical projection (Car)
RADIUS

Fig. 8 Different map projections of the "Sky plotting window". The radius is represented here as the auxiliary data.

3.2 Healpix maps

A powerful capacity of TOPCAT is to use the **Healpix** scheme to obtain a map with a homogeneous resolution / regular sampling. Healpix is a hierarchical tesselation of the sphere providing a grid of cells of the same size, available at various scales. Data points from our table are integrated in the Healpix cells, and different parameters can be displayed: sum of values, median or mean, min or max value, point density… TOPCAT is able to perform this entire task thanks to the "**Sky density**" form in the **Sky plotting window**.

As for the maps presented in Fig. 8, lon/lat coordinates are filled in with the columns "LON_CENTER" and "LAT_CENTER" (which correspond here to the name of our variables) to build the map. The Sky density is available from the **Sky plotting window** (step 1 in Fig. 9) in the **Form** tab. Uncheck or remove the Mark form, select **Sky density** in the "**Forms**" menu (step 2 in Fig. 9). A weight parameter can be chosen or left empty (step 3 in Fig. 9). In step 4, you can vary the size of the Healpix cells (the angular size is displayed at the bottom of this window, step 6). When a variable is chosen as weight (e.g. reflectance) the "**Combine**" option (step 5 in Fig. 9) allows you to display the sum, median, or other quantities computed into the Healpix cells. Finally, in step 6, it is possible to import /export the Healpix map into a table.

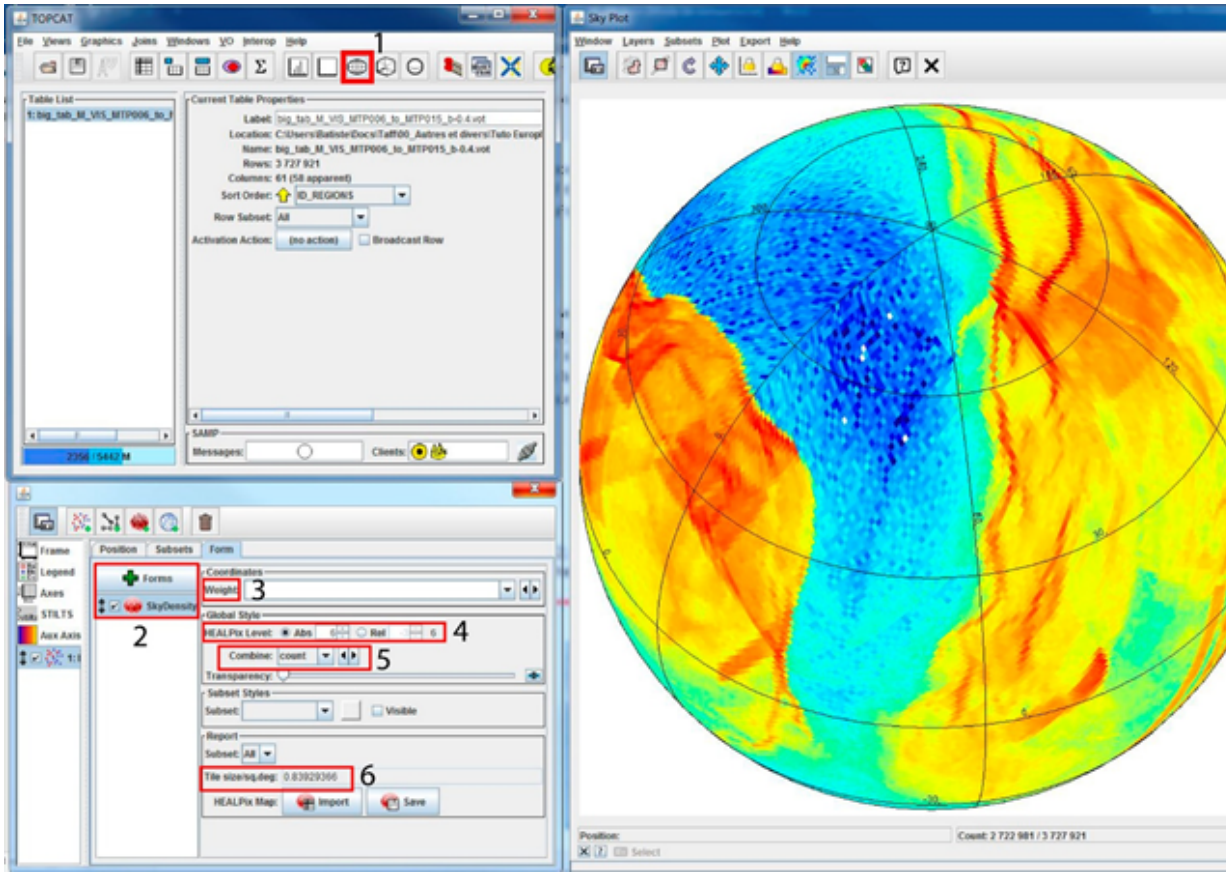


Fig. 9 An example of a density map using the Sky density function.

TOPCAT provides an alternative method to handle healpix maps (through the "**Add a new Healpix layer control to the stack**" tools). However, this method requires that the data are already averaged in the healpix cells, typically after aspecific database query.

3.3 Filtering data

You now want to study separately the observations of the Imhotep region, which is approximately centered on longitude/latitude 0°. There are several ways to do this:

1. It is possible to create a subset graphically by enlarging the plot and defining a new subset based on visible points, thanks to the option "**Define new row subset containing only visible points**". However, this is only relevant for rectangular regions, and not complex ones that are predefined from other parameters.

2. Arbitrary regions can be defined in freehand selection / lasso mode, thanks to the option "**Draw a region on the plot to define a new row subset**" (click, select, and click again). To make this operation easier, you can plot the identification number of each region (a parameter included in this table, see maps in Fig. 8) as auxiliary data. Regions will be colored independently. However, freehand selection is adapted to isolated groups of points, which is not the case here.

3. The easier way in this case is to use the region ID from the table to select only lines corresponding to Imhotep. Click Subset in the main TOPCAT window then click the green + icon (new subset), enter a subset name and expression **ID_REGIONS==13** (which stands for Imhotep)**.**

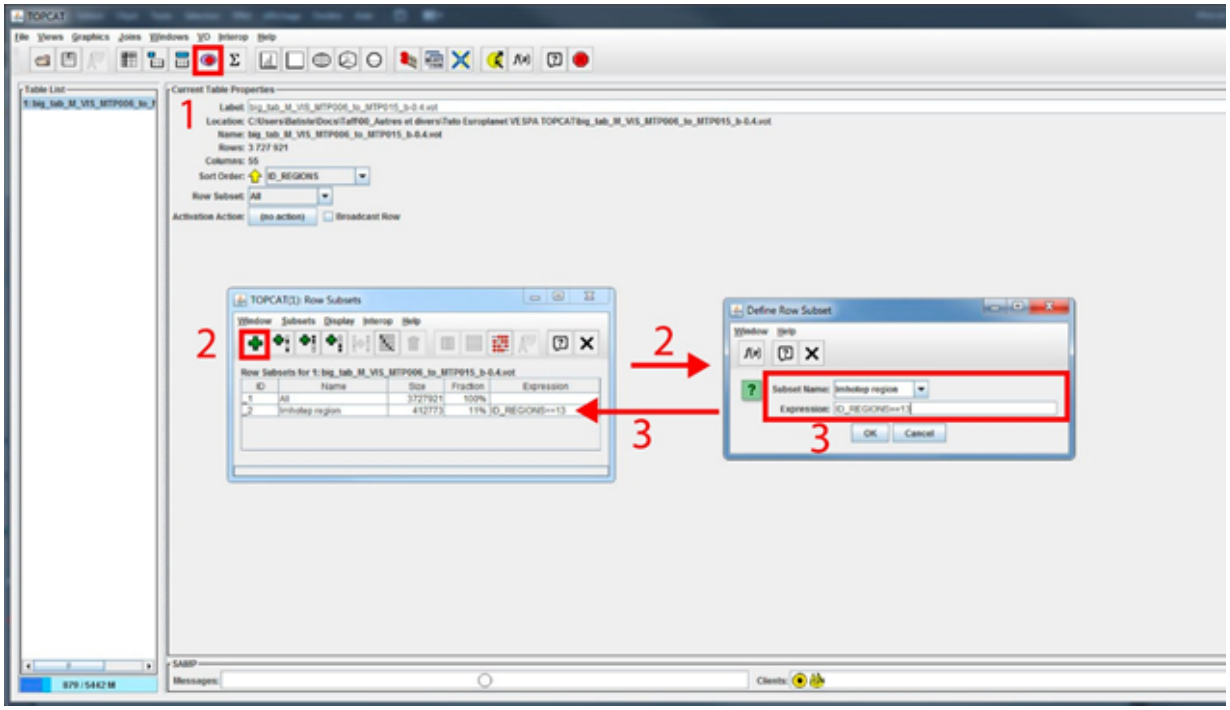


Fig. 10 How to define a subset, quickly and easily.

Once the subset is defined, you can highlight it in the table browser or display only this part of the data using the “Subset” item in the global TOPCAT menu, when the table window is selected (Fig. 11). It is also possible to create a new table containing this dataset by selecting the subset and exporting it in any format (Fig. 12). We can save this new table or the entire session to load it later.

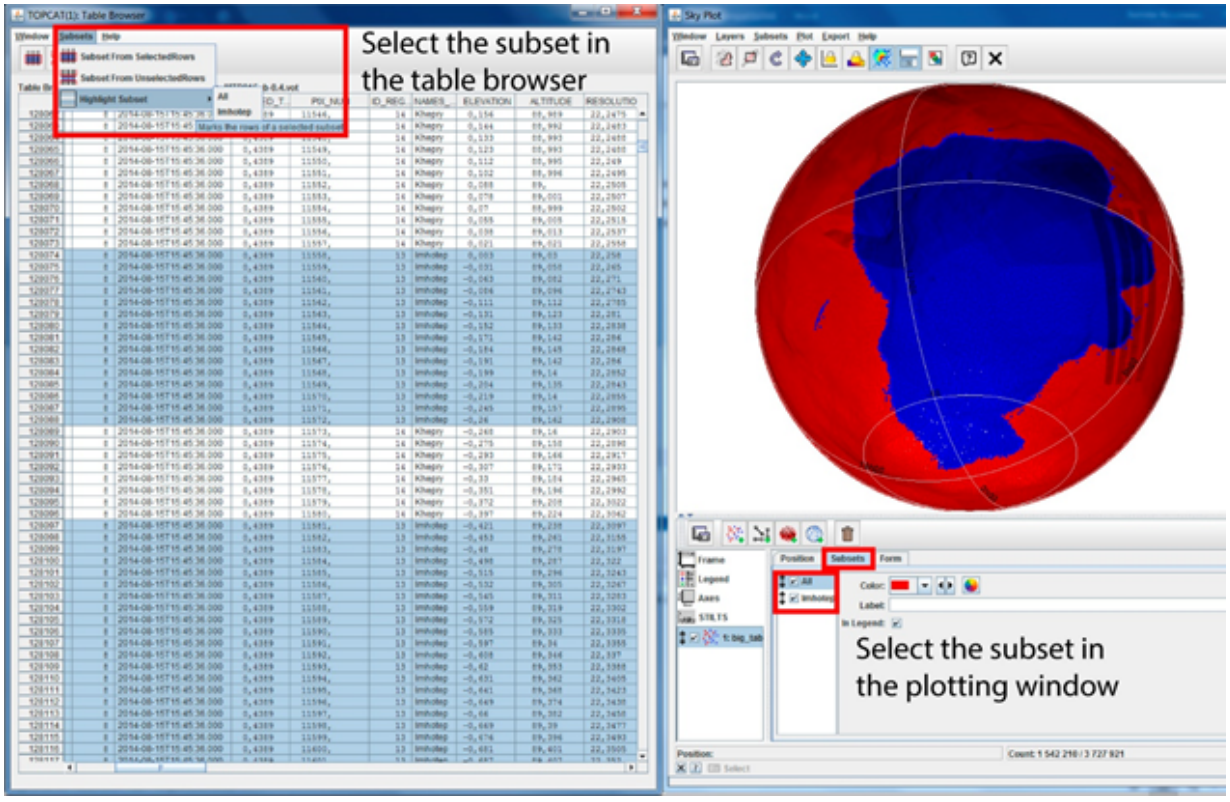


Fig. 11 After the creation of a new subset, it is possible to highlight it in the table browser (left) or plot it separately (right).

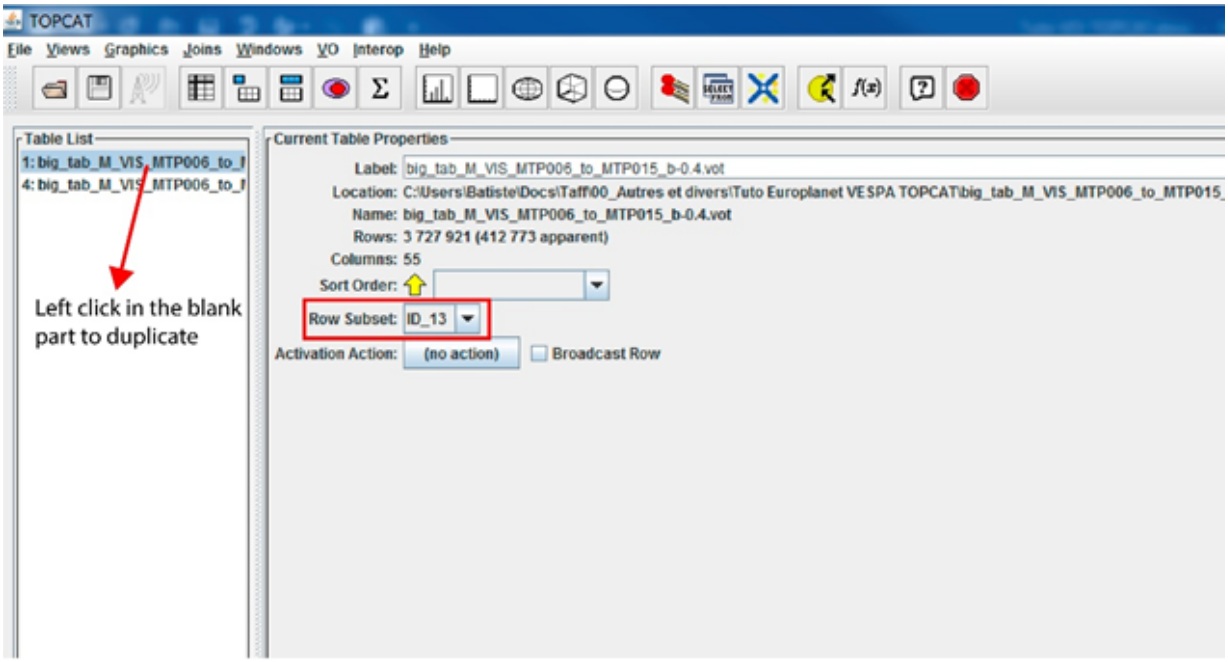


Fig. 12 Duplicate table or a subset by click and drag in the blank part.

You now want to get an idea of the illumination conditions in the selected subset. It is possible to plot **histograms** as shown in Fig. 13. Many options are available to explore the data: superposition of histogram, fitted curve… Here, the dataset ranges from 3.5° to 179° in incidence angle and from 0° to 129° in emergence angle. Another solution to obtain this information is to use the tool **Display statistics for each column** from the main window. From this window an appreciable number of quantities can be displayed (see Fig. 14).

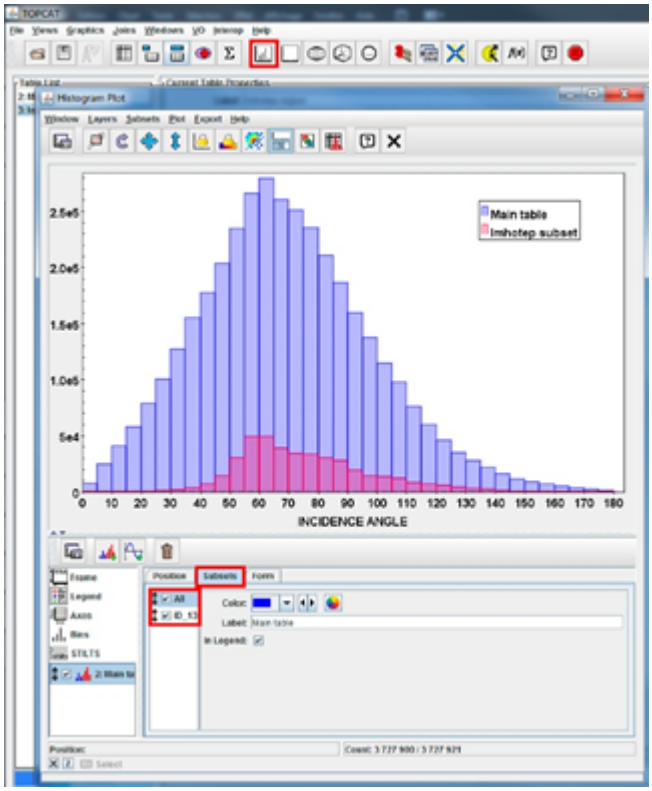


Fig. 13 Many options are available with the histogram and other plotting modes.

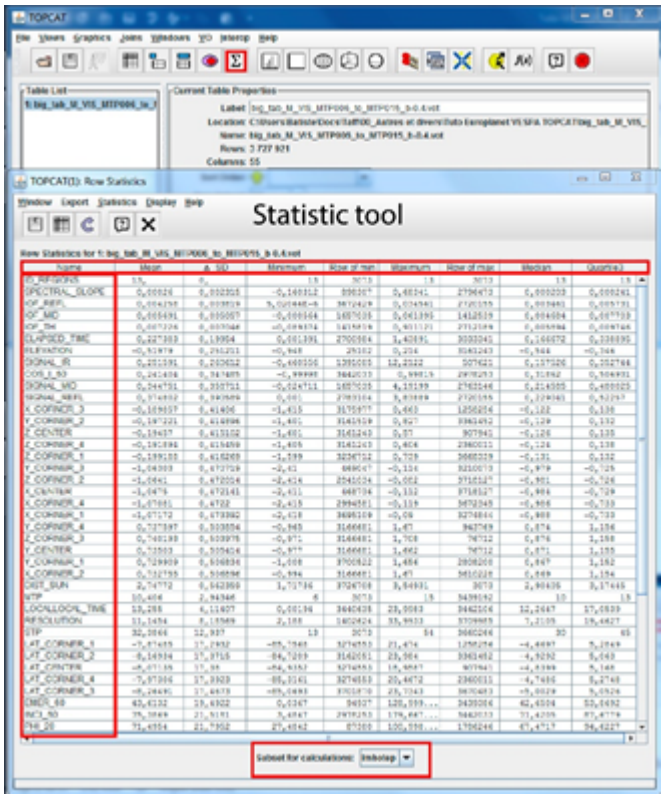


Fig. 14 Statistic tool, it's all good.

You now want to cut the dataset from 28° to 32° both in incidence and emergence angles. There are two possibilities to do this (Fig. 15).

1. The “graphical” way consists to plot the incidence vs. the emergence angle in a plane plot, to restrict the ranges of the X and Y axis to a minimum of 28° and a maximum of 32° and to press the button n°3 in red on the Fig. 15: **Define a new row subset containing only currently visible points**.
2. The “computational” way use the same tool used for Fig. 10. Click on **Display row subsets** (button #1 in blue) and then on **Define new subset using algebraic expression** (button #2 in blue). A new window is opened where you can enter a Boolean or any other compatible expression sorting the data. To know exactly the accepted commands and the available functions, we can report to the TOPCAT help (online or local).

Notice on Fig. 15 that the two methods give the same results (same number of selected line).

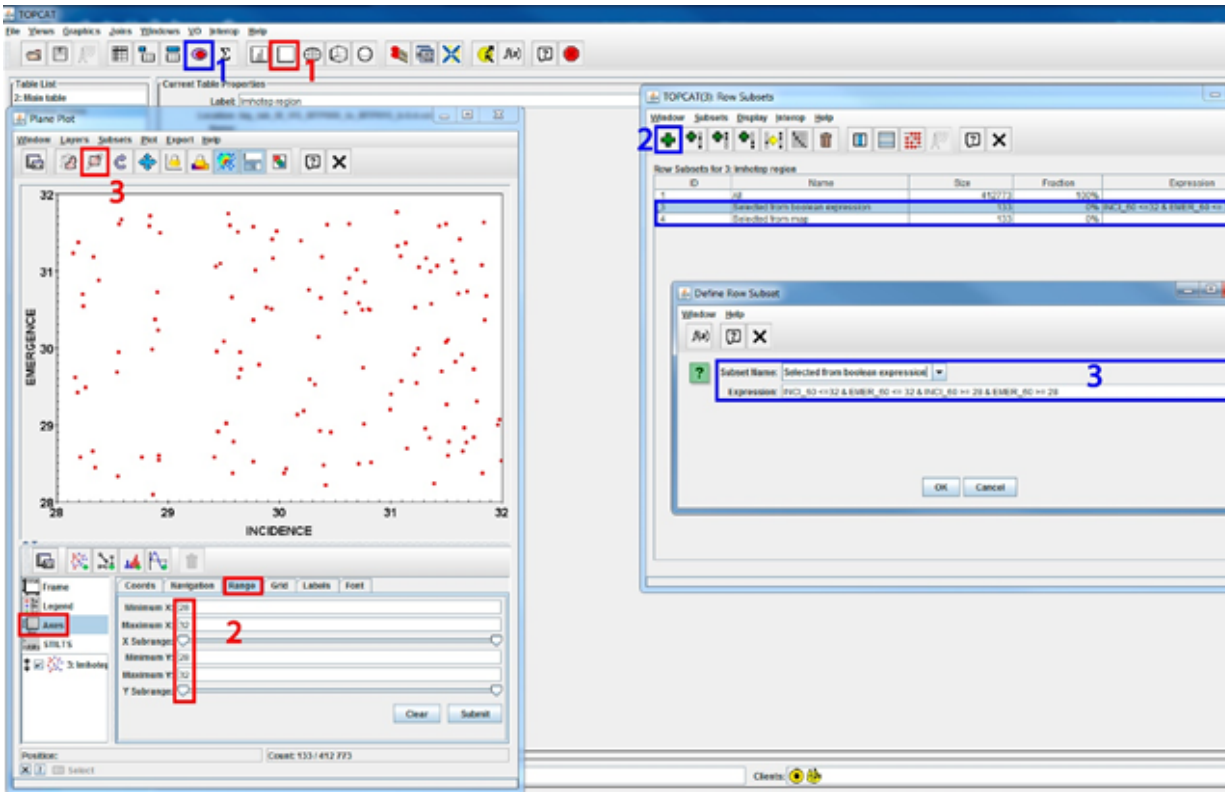


Fig. 15 Two ways of defining a similar subset.

When a specific dataset is defined, it is possible to save it or to broadcast the data to other VO tools. This method uses the SAMP protocol to communicate between the different applications. For example, data can be sent to the MATISSE tool (https://tools.asdc.asi.it/MatisseNoPermission.jsp) which can project data on the 3D shape model of 67P; to Aladin, e. g., to access other mapping options; to CASSIS for spectral plots, etc.