How to work with maps in the CAPRI GUI with a focus on the spatial HSMU layer

- Wolfgang Britz, July 2007 –

The author wants to thank Hans-Josef Greuel, Andrea Zintl and others for the manifold ideas which over the years had been incorporated into software at the Institute for Agricultural Policy to generate table and maps. Equally, contributions by Alexander Gocht to the Java Code are acknowledged. Not at least, feedback from the users helped to shape the product, especially to Fabien Ramos who also reviewed the user manual and tested the product.

What are the HSMUs and what information is available?

The HSMU are the so-called Homogenous Soil Mapping Units. Each HSMU contains one or several 1x1 km grid cells, not necessarily adjacent, and are defined so that these are more or less homogenous regarding climate, soil, slope, CORINE land cover class and NUTS II region. There are about 110.000 HSMUs for EU15. The spatial downscaling introduced in CAPRI-Dynaspat provides the following information per HSMU:

- Cropping shares and animal stocking densities
- Yields
- Economic indicators per crop and animal, and in relation to UAA
- Fertilizer application rates
- Environmental indicators

How to visualize the HSMU information

Given the 1x1 grid resolution, the most obvious way to look at the information is to produce maps with the CAPRI GUI. There is a co-ordinate set available which is called "HSMU.zip" which comprises the geometry for about 1.8 Mio Polygon which represent the HSMUs.

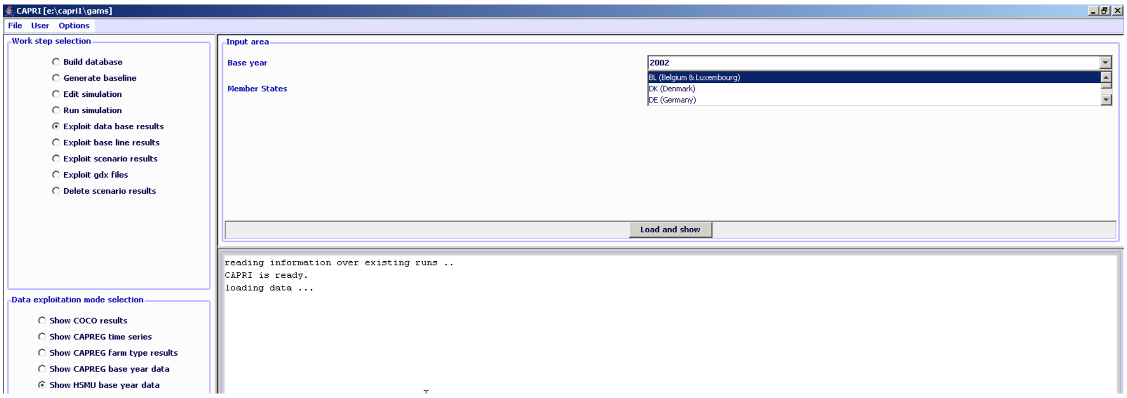
There are four options to view HSMU data:

1. Loading data for one or several Member States for the base year (dis-aggregated information from the NUTS II CAPRI data base).
2. Loading data for one or several Member States for the base year and the baseline, the latter representing dis-aggregated data from NUTS II results of the baseline calibration.
3. Loading data for one or more scenarios for a given year.
4. Loading data manually

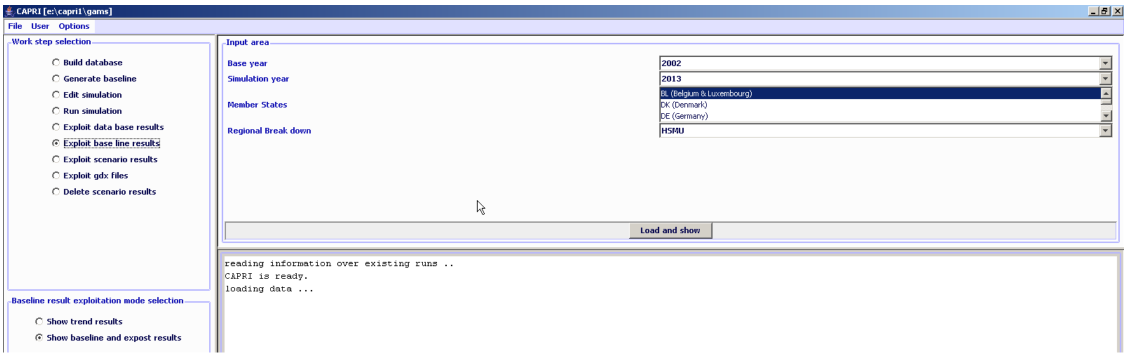
Loading data based on the GUI presets

As for the results at NUTS II level, there are three pre-defined exploitation possibilities included in the CAPRI GUI:

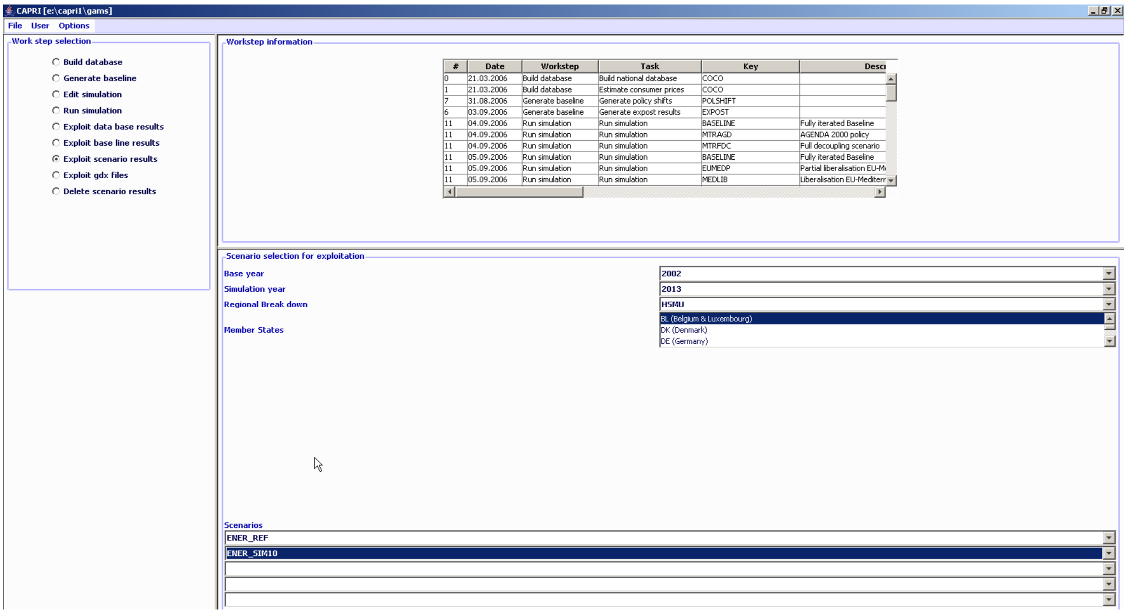
1. Viewing the results for the base year. Given the tremendous number of HSMUs, the user can select for which Member States to the load the information.



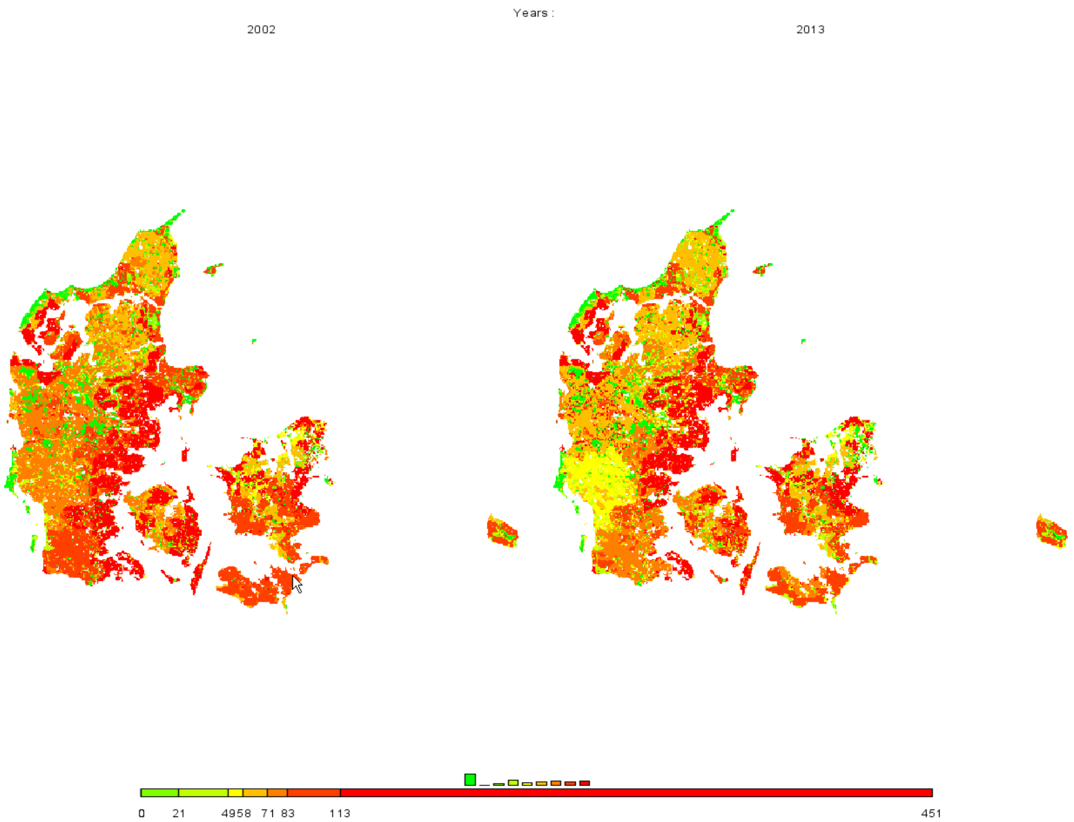
2. Comparing the results for the base with the baseline (projection results), comparison between two points in time. Again, the user can select the Member States.



3. Comparing results from different scenarios (comparisons for one point in time, but changes in drivers/assumptions relevant for the CAPRI economic model).



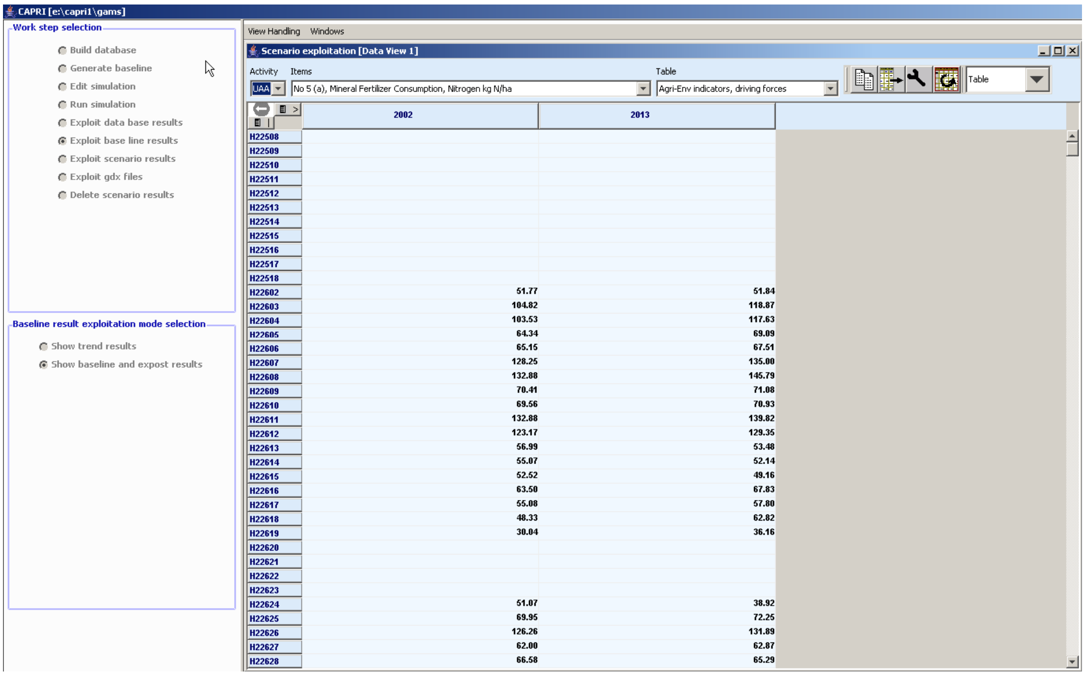
Once the loading is accomplished, the right hand side of the GUI is filled with a tabular view of the results, which can be turned easily into maps (other forms of exploitation as graphs are less suited given the large number of observations). Details how to work with the exploitation tools are found in a separate document. The screen-shot below shows results for Denmark as an example.



In order to ease exploitation of the results, pre-defined tables are set up, currently broken down into five categories:

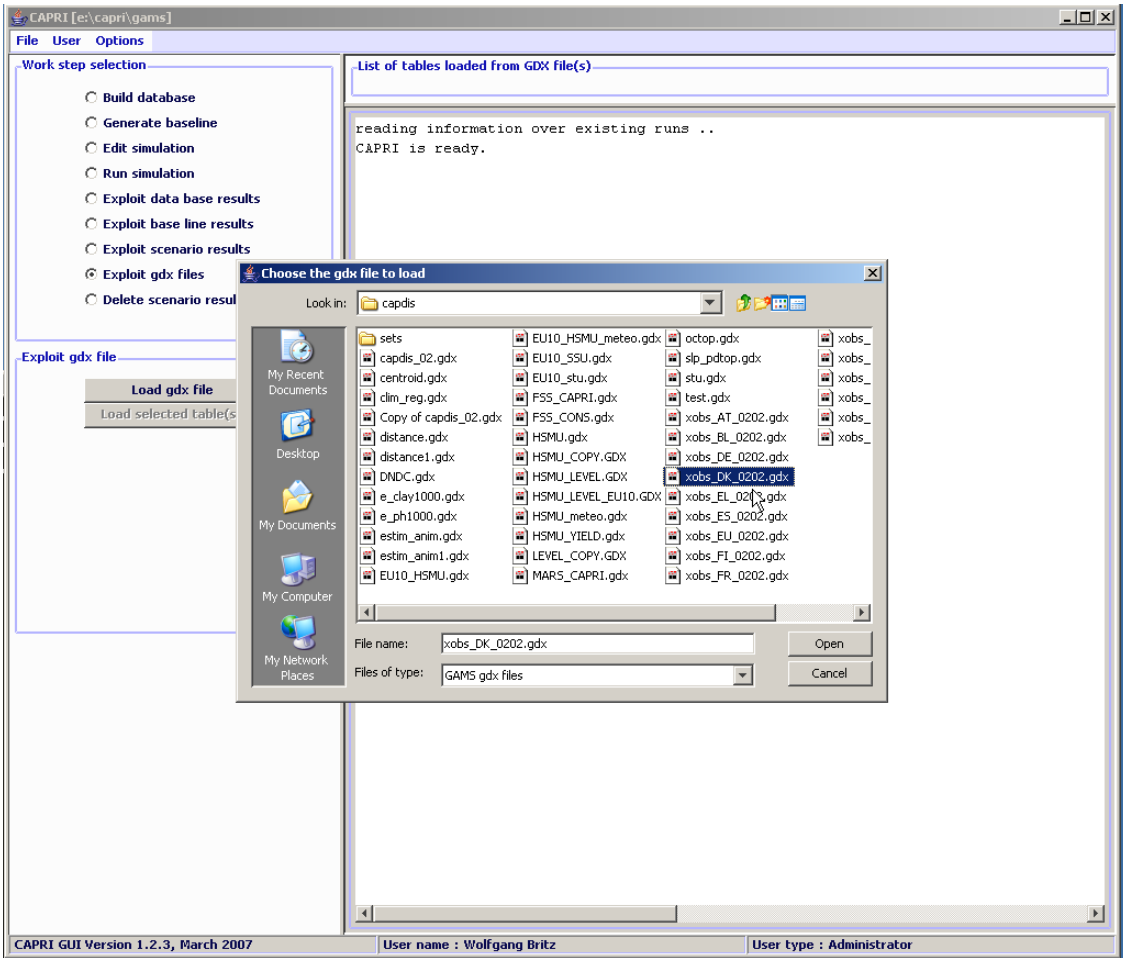
1. Agri-environmental indicators, driving forces (mineral fertilizer consumption, consumption of pesticides, irrigation shares, energy consumption, livestock densities, shares or arable / grass land or permanent crops)
2. Agri-environmental indicators, pressures and benefits (Gross nitrogen and phosphorous balance, green house gas emissions, High Nature Value Farm land indicator)
3. Economic indicators at HSMU level (market revenues, variable production costs, income)
4. Climate, soil, slope and altitude
5. Results from the DNDC meat model (gas losses for different nitrogen compartments, mineralization, leaching)

The tables on agri-environmental indicators (dirving forces, pressues and benefits) are set-up as close as possible according to the official EU Guidelines for Rural Development indicators.

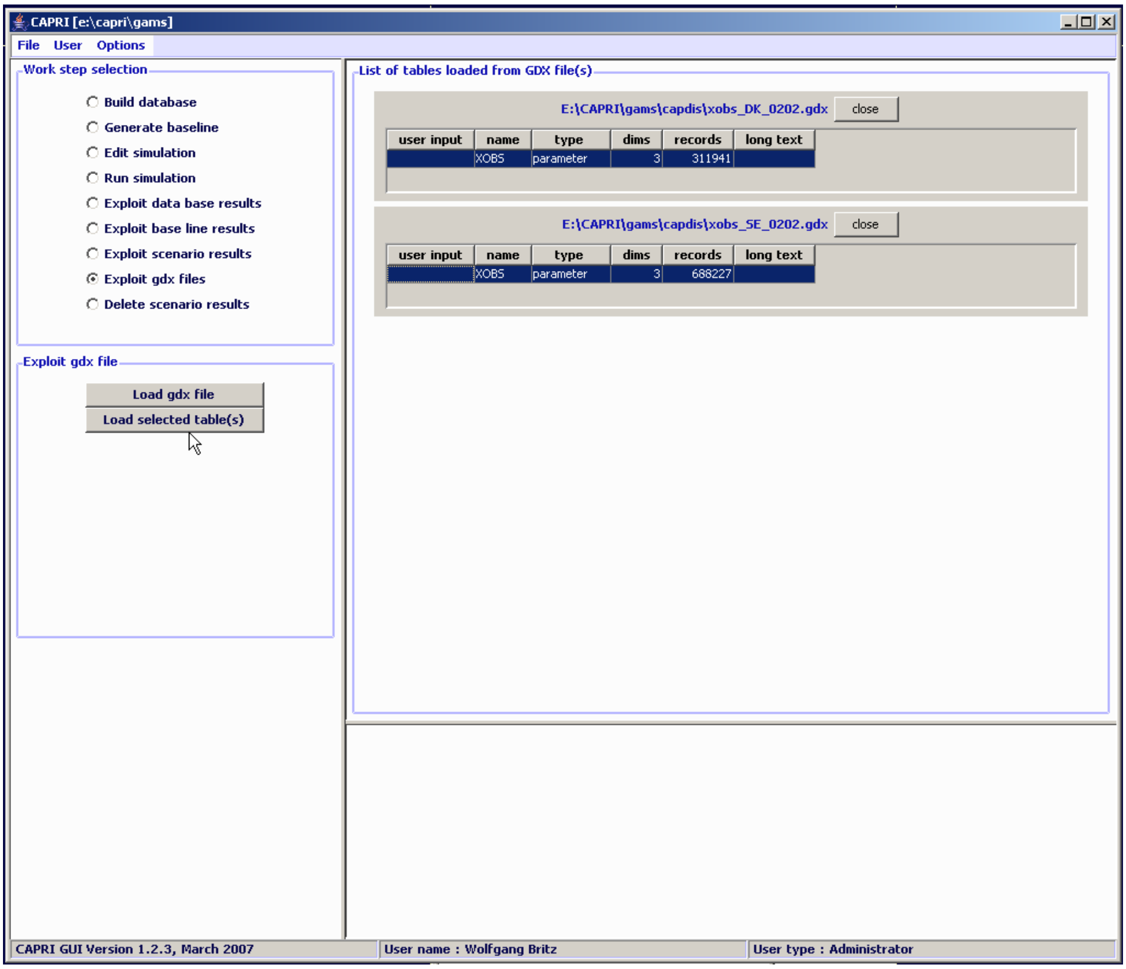


Loading data manually

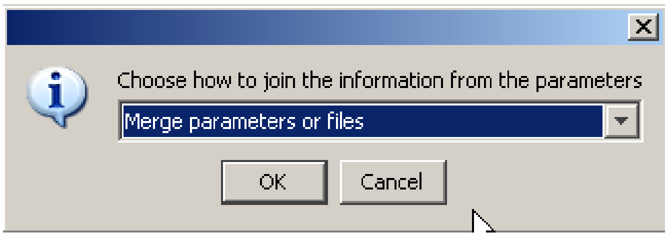
The option described here is introduced for completeness. The names of the file generated by the dis-aggregation programs start with “XOBS_” followed with the two character code of the Member state, then an underscore followed by the base year and the simulation year, and if applicable, the code for the simulation which is identical to the name of the GAMS files used from “pol_input” which was used to run the scenario.



If one wishes to see the information for several Member States simultaneously, one may click on “Load gdx file” again and add additional files. Afterwards, the lines with the parameters “XOBS” should be selected.



Pressing load selected tables will open the following dialogue, choose



As otherwise, the program will introduce a new dimension for the data loaded from the different files, and you will not be able to see the information for "DK" and "SE" together in one map. Afterwards, the selected records will be loaded from the files.

Depending on the amounts of records, that may take several seconds. Before turning to the mapping view, only one column should be selected. Press on the button left of "SWHE" in the table headers ("Open selection dialog for table column"), and select with the mouse one of the codes, then press o.k. The table should now comprise only one column. Afterwards use the drop-down list with the viewing options and choose map as shown below. Choose “HSMU.” to select the geometry for the HSMUs. The program will now load the geometry for the HSMU and draw the map which takes several seconds.

The software behind the mapping viewer and the CAPRI exploitation tools

Some words on the software behind the mapping viewer and its history may be interesting to the reader. The very first maps in CAPRI (in 1998) were produced with an MS-EXCEL mapping plug-in which was at that time a cost-free add-on. However, moving the data to EXCEL and then loading them in the viewer was not a real option for the daily debugging work on data base and model. Therefore, shortly before the first CAPRI project ended in 1999, a JAVA applet was programmed by W. Britz which was able to draw simple maps from CSV-Files automatically produced by the CAPMOD GAMS code. That code with slight modification remained active for quite a while, and some of the features are still to be found in the current mapping viewer. Then for a while, the exploitation tools were based on XML/XSLT+SVG and a mapping viewer in SVG was realized. However, the XML solution had the big dis-advantage of requiring a large amount of single ASCII input files, and was not really performant when complex pivoting was used. Therefore, the next evolution step was a pure Java GUI with direct access to GDX files which is the current state of the art. GDX files are an internal file format used by GAMS which allows a rather efficient I/O for large sparse tables. An API library allows to access GDX files from other applications.

When the HSMU spatial layer was added to CAPRI it became obvious that the existing JAVA code to produce maps needed some revision, especially regarding the way the geometry was stored. In that context the question of using an existing GIS independently from CAPRI or the use of existing GIS classes plugged-into the CAPRI GUI was raised again; and some tests with open-source products were undertaken. A stand-alone GIS as the sole option was certainly the less appealing solution. Firstly, it would have required producing rather large intermediate files, and would have left the user with the time consuming and often error prone task of exporting and importing the data. Secondly, the user would need to switch between two different programs and GUI standards. And thirdly, all the usual problems with installing and maintaining additional software on a work station would occur. However, as indicated later, the GUI naturally allows passing data over to external application and does hence not prevent the user from using a full-fledged GIS solution.

The main points taken into account during the search of a map viewing solution for CAPRI were: (1) possibility to import efficiently data from the CAPRI GUI, (2) user-friendliness, (3) performance and (4), in the case of plug-in libraries, expected realization and maintenance resource need, and naturally (5) license costs. It turned quickly that an ideal product was not available. Some of the products were not able to allow for the necessary link between newly imported tables with region codes and an existing geo-referenced geometry. Others had very complex user interfaces or produced run-time errors, took ages to draw the HSMU maps or very quite expensive. From the different option tested, solely gvSIG (http://www.gvsig.com/index.php?idioma=en) seemed to be the only allowing the user to import data from a CSV – which must however be semi-colon delimited – and join one of the columns to a shapefile. At least the version installed at that time as however running not very stable.

In the end, it was decided to build on the existing code base and let Wolfgang Britz write the additional code “on demand”. The main advantage of that approach is the fact that the mapping view is transparently integrated in the CAPRI GUI, it is sufficient to switch from “Table” to “Map” in a drop-down list to produce a colored map, and that user demands regarding additional functionality may be and had been realized.

Compared to ArcGIS, where the EU27 HSMU geometry plus codes and centroids requires about 340 Mbytes, the CAPRI version requires about 27Mbytes solely. Reading in the CAPRI GUI is somewhat slower compared to ArcGIS due to unzip on demand. The actual drawing operation takes about the same time as in ArcGIS (about 11 second for the full data set), classification in Java is typically faster.

The concept of the CAPRI exploitation tools is centred on the idea of a view. Content wise, that may be understood a certain angle to look at European agriculture as example its integration into international markets, its impact on the environmental, farm management, costs of the Common Agricultural Policy. Each view thus extracts a certain collection of numerical values, labels them so that they carry information to the user (long texts, units), chooses a matching presentation – as a table, as a map or a graphic, and arranges them in a suitable way on screen. The views can be linked to each others, allowing a WEB like navigation through the data cube. The user may have open several views in parallel, and he may change the views according to its needs, e.g. switch from a map to a tabular presentation, or change the pivot of the table, sort the rows etc.

Internally, each view is stored in a XML schema. A view could be also understood as a combination of a pre-defined selection query, along with reporting information. The XML schema allows to attach long texts, units and tooltips to the items of a table, and thus to show meta-data information to the user. The XML schema does hence replaces look up tables in a DBMS. It may equally store information regarding the pivoting, the view type (table, map, different graphic types), and for maps, classifcation, color ramp and number of classes. The views can be grouped into logical entities, and are shown as a popup menu to the user.

Tabular views may feature column and row groups. Empty columns and rows can be hidden, tables can be sorted by column, with multiple sort columns supported. Numerical filter can be applied to columns.

The underlying data model is very simple and straightforward. All data are kept in one large multi-dimensional data cube, and only float values are allowed. Currently, only read-only is supported. Each data dimension is linked to a vector a string keys. Those keys are the base for the filter definitions. Currently, data cubes up to six dimensions are used (regions – activities – items – trading partners – years – policy scenarios). The data storage model is equally optimised to the specific needs. As only float values are supported, all data can be stored as one primitive array of floats. To allow fast and efficient indexing, a linear index is constructed from the multi-dimensional data cube, and the non-zero data and their indices are stored in a hash table. That renders data retrieval very fast. All data are at initialisation time loaded in memory, for moderately long linear indices, about 10 Bytes are required to store a non-zero float and its index as an int. If the maximal linear index is very large, the index is store as a long and the storage needs goes up to about 16 Bytes. For moderately sized data cubes, 20 Million numbers can hence be hosted in about 200 Mbytes.

The data are read from a generic file format generated by GAMS (General Algebraic Modelling System, a commonly used software package in economic modelling) called GDX, the software package on which CAPRI is based. Access to GDX is handled via an API provided by GAMS.

Technically, the exploitation tool is completely client based. That reflects the specific user profile of the CAPRI modelling system where the exploitation tool is integrated with an economic model and tools building up its data base. The main aim of the tool is to support forward looking policy analysis, and user will create there own scenarios and in some cases even own variants of the export data, processes requiring considerable processing and storage resources. A client-server solution where the production process and data storage would need to be hosted on a web server is therefore not a preferred solution, also as users will often develop variants of the modelling system by code modification in GAMS, and contribute to its development. The structure of the data driver would however very easily support linkage to network or WEB based data bases.

The mapping viewer of CAPRI is based on very simple and straightforward concepts. First of all, it supports solely polygon geometries not comprising holes. The storage model is optimised to host rectangles, and is especially efficient if the polygons vertexes are all points in a raster. The topology is not read from a shapefile, but stored in a generic rather simple format. The vertices are stored in x,y coordinates already projected in a rectangular coordinate system, and the viewer does not deal with geographic coordinate system, but simply scale the rectangular coordinates in the viewport. The viewer in its current version solely supports one layer of quantities. Those restrictions naturally allow reducing memory needs, and, thanks to the rather simple data structures, also allow rather performing drawing operations. It should be noted that the JIT compiler of JAVA is indeed rather fast.

The biggest topology currently handled simultaneously covers an intersection of Corinne Land Cover, slope classes and Soil Morphological Units and comprises for EU27 around 2.7 Million polygons. As the majority of the polygons are rectangles, not more then 6-7 Million points needed to be stored. The topology handler and the drawing routines separate rectangles, for which only the two outer points are stored, from polygons, for which the vertices and centroids are stored.

The viewer is written in Java. There are two variants. One is a stand-alone version of the viewer realised as an applet. It reads from an internal portable binary data format, and java classes, data and geometry can be packed into one jar file, e.g. to ship it to a client. The second version is transparently integrated in the GUI of the CAPRI modelling system.

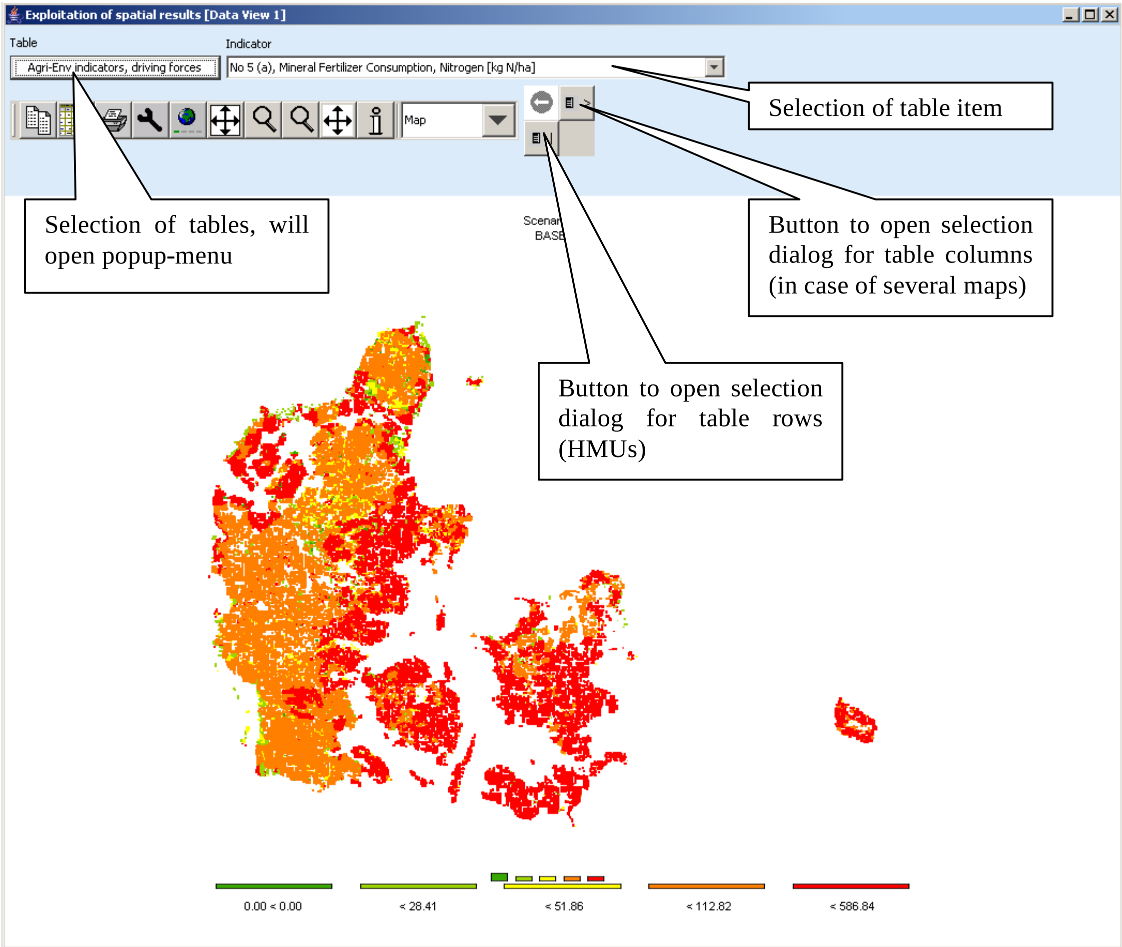
Swing is used for the GUI. In order to profit to the most of the simple implementation, the viewer has been written completely anew, and is not based on existing GIS libraries. Even certain standard JAVA classes as e.g. for hash tables, have been replaced by own implementations, to reduce implementation overhead. Some care was given to support flexibility in classification, given that only quantities are supported, so that the tool covers natural breaks, quantiles, equal spread, mean standard and nested means. Area weighting is supported as well.

In order to export data to other applications, the tools supports first of all tab delimited clipboard export, allowing import e.g. into EXCEL. Maps can be exported as JPEGs over the clipboard. Alternatively, the user may export to external file, in CSV format, DBF, to MS Access or to GAMS. DBF export will generate a second file comprising meta data.

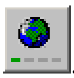
Working with the mapping viewer

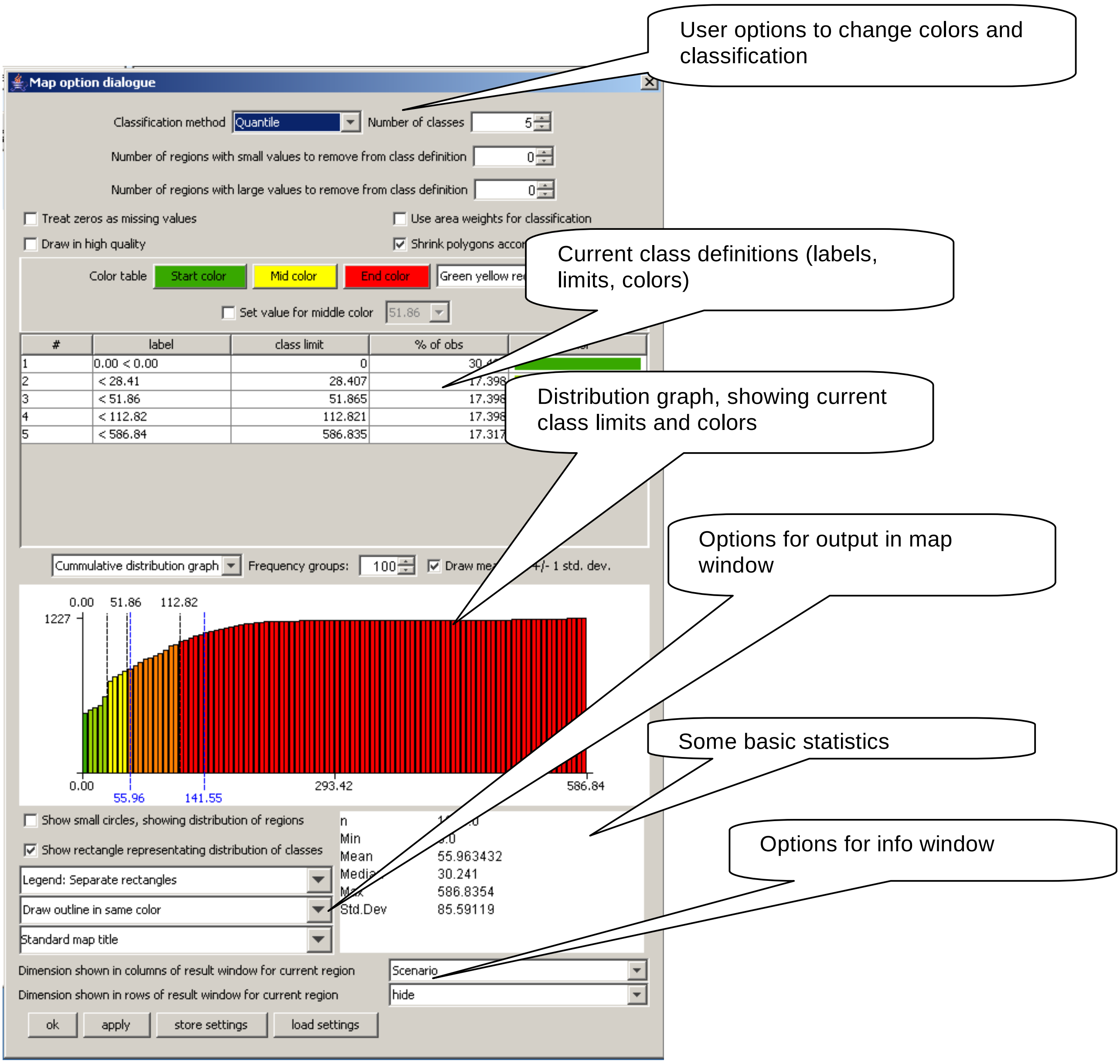
The most obvious way to work with the results at HSMU level is the use of maps. When starting the GUI, the mapping view uses some pre-sets which can be interactively changed as described below. The following screen-shot shows the result of loading the base year results from the spatial dis-aggregation for Denmark and then switch from the tabular to the mapping view.

As with other views, the content of the map can be changed by working with the drop-down boxes, or by (de)selecting columns and rows. The map specific possibility to change class limits and colors and further features are discussed in the following.



Changing the classification and the legend

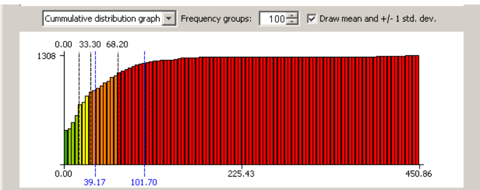
In order to change the layout of the map, click with the mouse in the area of the legend or press the map option button . The following dialogue will open.

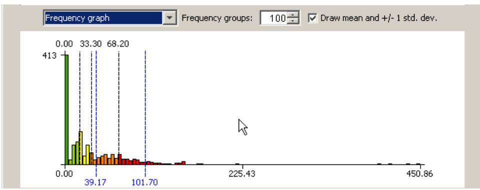


It offers different options to change the way the map is drawn on screen and information supporting the classification.

Controlling the classification

The classification can be controlled by the frequency diagram which can be either drawn as a cumulative distribution or as a frequency bar chart as shown below. The blue lines show the mean and +/- one standard deviation, the black lines the current class limits. The bars are drawn according to the current color model. The user can change the number of groups used to draw the diagrams, which does however not influence in the actual classification.



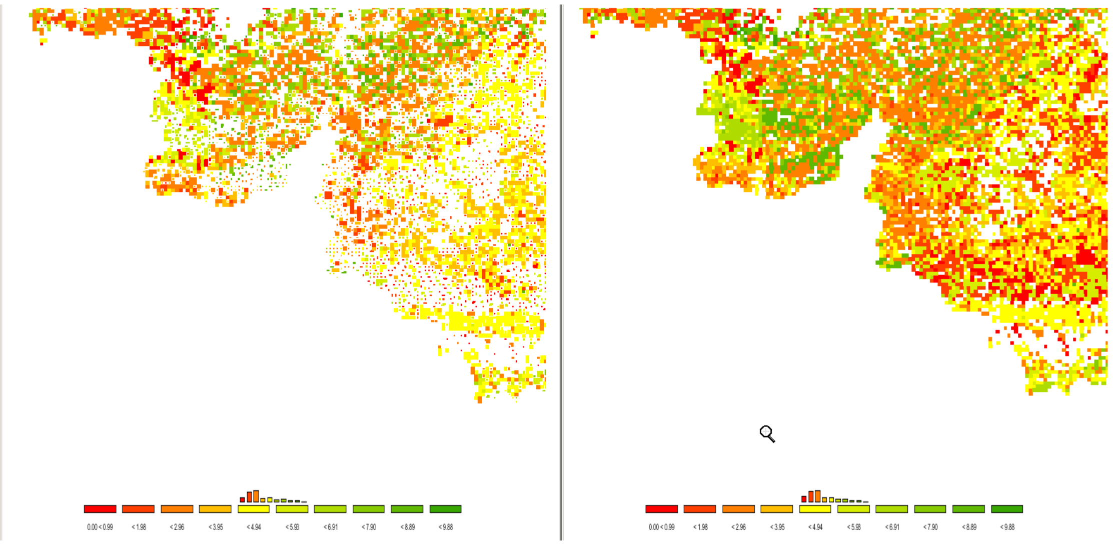


Shrinking polygons according to UAA share

The optical impression received from a map where colors are used to different between values is to large extent depending on the area covered by certain color. If the majority of the pixels is drawn in red, that will send a warning message to the user. In the case of the HSMUs and information relating to agriculture that message can be strongly biased as almost all HSMU comprise some other land cover then agriculture, and some of the HSMU comprise only very little agriculture, but e.g. forest, shrub lands, water bodies or artificial land cover. The HSMU geometry therefore comprises the information about the share of UAA assigned in the base year to each HSMU, and that information can be used to shrink the area of the polygons when drawn on screen accordingly. That is done by drawing all points of the polygons towards to the centroid of the polygon, by multiplying the distance between the point and the centroid with the square root of the share of the UAA. In the original HSMU geometry, such polygons had been broken down to simpler ones where the connection between a point and a centroid would cut through a segment of the polygon a in such case, shrinking could let the new polygon to hide other ones.

The graphs below show the very same map (same input data, classification and coloring) for the High Nature Value indicator for a part of Belgium. The right hand side map draws the HSMUs in there full size, the left hand side one using shrinking. The message perceived is probably mighty different. In the unshrinked right map, one may conclude that there is a lot of highly intensive agriculture (low HNV indicator drawn in red) in the lower diagonal triangle, and some important areas of high nature farmland in the protruding area. That optical impression is different with the polygons are drawn corrected for the share of agriculture. It turns out that in lower diagonal triangle, the density of agriculture is often low, and especially low in the intensively managed

HSMUs. Equally, it turns out, that the area covered by High Natural Farmland in the protruding part is relatively small.



Area weighted classification

The classification can be generally applied treating each “region” (a NUTS II or a HSMU) as an observation with equal weight or using the areas of the underlying polygons as weights. Those weights are multiplied with the share of UAA if shrinking is used as explained above.

Excluding zeros from classification

In GAMS, zeros and missing values cannot be distinguished. For certain results, zero results are therefore coded as very small numbers to allow for that distinction. Zero observation can be excluded from classification and the polygons with zero observations will not be filled.

Classification method

A first important feature is called “classification method” and defines how internally the class limits are set. For all types of automatic classification methods a clean up procedure is used which removes classes with identical limits. It is generally recommended to use a number of classes which can be easily identified by the user, and to consult the frequency or cumulative distribution graphs present in the map option dialogue to check to what extent the class limits chosen represent well the data.

The following classification methods are currently supported:

Natural breaks

Natural breaks classification is a method to cluster the data into classes so that differences between the means of the classes become high while the standard deviation inside the classes becomes low (FISHER, W. D. (1958). "ON GROUPING FOR MAXIMAL HOMOGENEITY," JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION 53, 789-798. Code based on: HARTIGAN, J. A. (1975). CLUSTERING ALGORITHMS, JOHN WILEY & SONS, INC., NEW YORK. PAGES 130-142.). The algorithm does only find the approximate best solution, but gives often a rather appealing class limit definitions.

It works rather well if no extreme outliers are present in the distribution. In the latter case, classes solely comprising the outliers will be generated, and the vast majority of the values will be put in one or two classes.

The clustering algorithm is rather expensive to calculate, so that in cases of population exceeding 500 observations a somewhat simplified version is implemented in the CAPRI GUI. From the original observations, a “condensed” population is generated whose members represented means of consecutive observations of the original one. The members are set so that the number of observations from which the mean is calculated is not bigger then 1/500 of the original population size and that the spead of those observations is smaller then the minimum of 1/500 of the spread of the total population and 10% of the standard deviation. The actual calculations are then done taking the size of the resulting classes into account.

Quantile

The observations of the regions are split such that in each class fall approximately the same number of observations. Quantiles are cheap to calculate and are therefore the default setting, and often appealing as colors occupy similar areas in the maps as long as the polygons have approximately the same size.

If unique values are found at the end of a quantile, the algorithm will either exclude all observation with that unique value from the class or include all of them. The decision will be based on the fact if with or without inclusion the size of the class comes closer to the desired size. If the user has e.g. chosen five classes, the desired class size should cover 20% of the observations or area weights.

Equal interval

The differences between the current minimum and maximum value is divided into classes of equal spread. May lead to rather curious class limits when outliers are present. In those cases, it may be appropriate to exclude some regions from the classification. See below for details how to exclude regions from the classification.

Mean standard dev

The class limits are defined according to the mean and portions of the standard deviation of the data. Works best with normally distributed data, but may result in very small classes if the distribution is skewed, e.g. long tailed. The algorithm will always introduce at least four classes, then six, eight, ten and twelve. More then twelve classes are neglected.

The algorithm takes into account the spread of the data, and sets the class limits accordingly. If all observations fall into +/-25% of a standard deviation, class limits are introduced at 25% and 10% for four classes. If the number of classes is higher, new limits are introduced at 5%, 2.5%, 1% and 0.5%. In case of +/-50%, the smallest class is dropped and +/-50% added, and so forth up to +/- 3 standard deviations.

Nested mean

The nested mean classification will only work with 2, 4 or 8 classes. The classes will be defined such that one break is found at the mean of sample. The resulting two halves of population are then again divided by their mean to get four classes, and the resulting quarters divided by their means to define eight classes. Works well with rather skewed distributions.

Manual classification

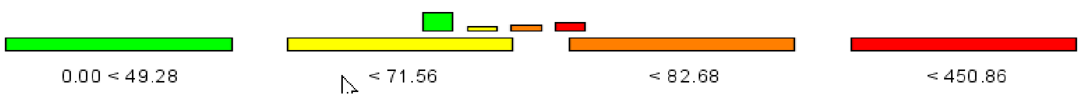
Finally, the user may set the class limits by hand. In order to so, double click with the mouse on the appropriate row in the table with the classification results in the column “class limit”. The value can now be changed with the keyboard. When done, click into another cell. The labels will be adjusted accordingly. Afterwards, when all the class limits are defined, the user may also overwrite the label (e.g. using words as “low” or “high”).

Please keep in mind that currently the values will be lost if you load other data or change the classification, number of classes etc..

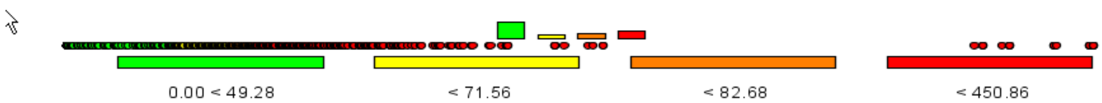
#	label	class limit	% of obs	color
1	0.00 < 0.00	0	30.488	
2	< 28.41	28.407	17.398	
3	< 51.86	60	17.398	
4	< 112.82	112.821	17.398	
5	< 586.84	586.835	17.317	

Integration distribution information in the map window

The GUI allows the user to enter in different ways distribution information in the map. The first possibility is to print a simple frequency diagram above the legend. That gives a rather intuitive feel on how well the class limits represent the data distribution. In our example below, it is obvious, that the majority of the values in the first class.



Less suitable for final out, but useful while playing around with classification methods and class definition are the distribution dots which can be added. They carry additionally information about the where the values of in different classes can be found.



Finally, switching to linear or logarithmic may be a way to help reading the map.

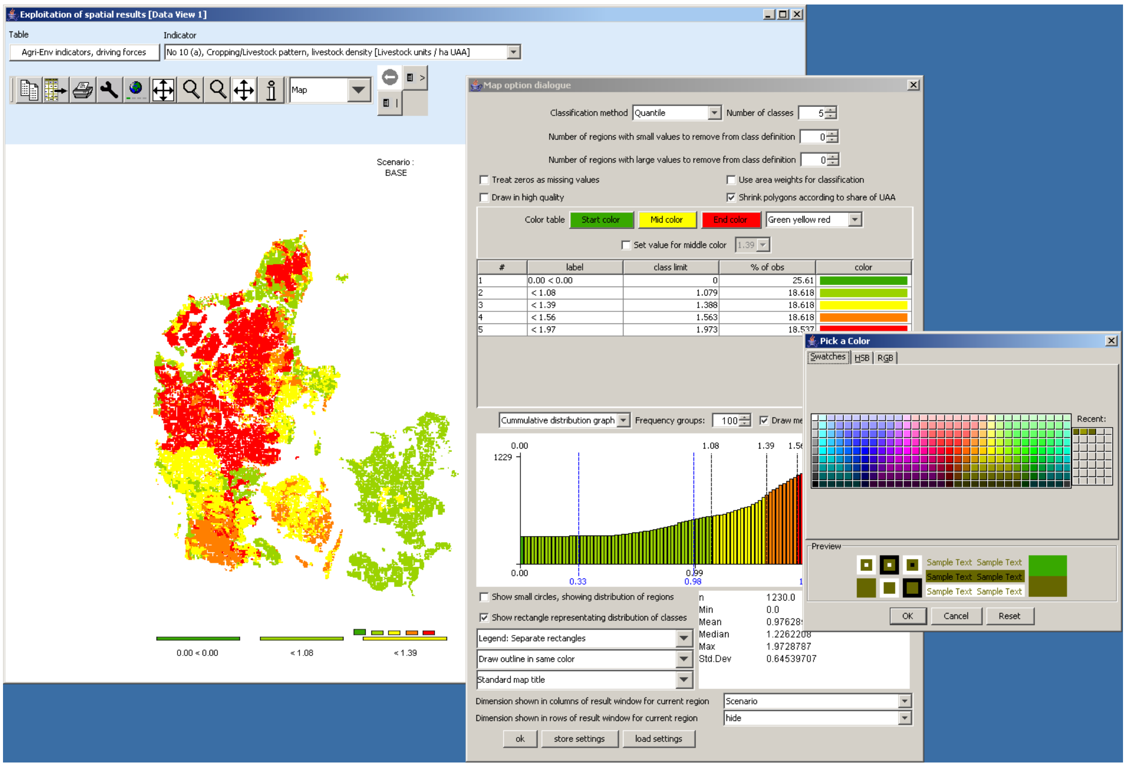
Color table

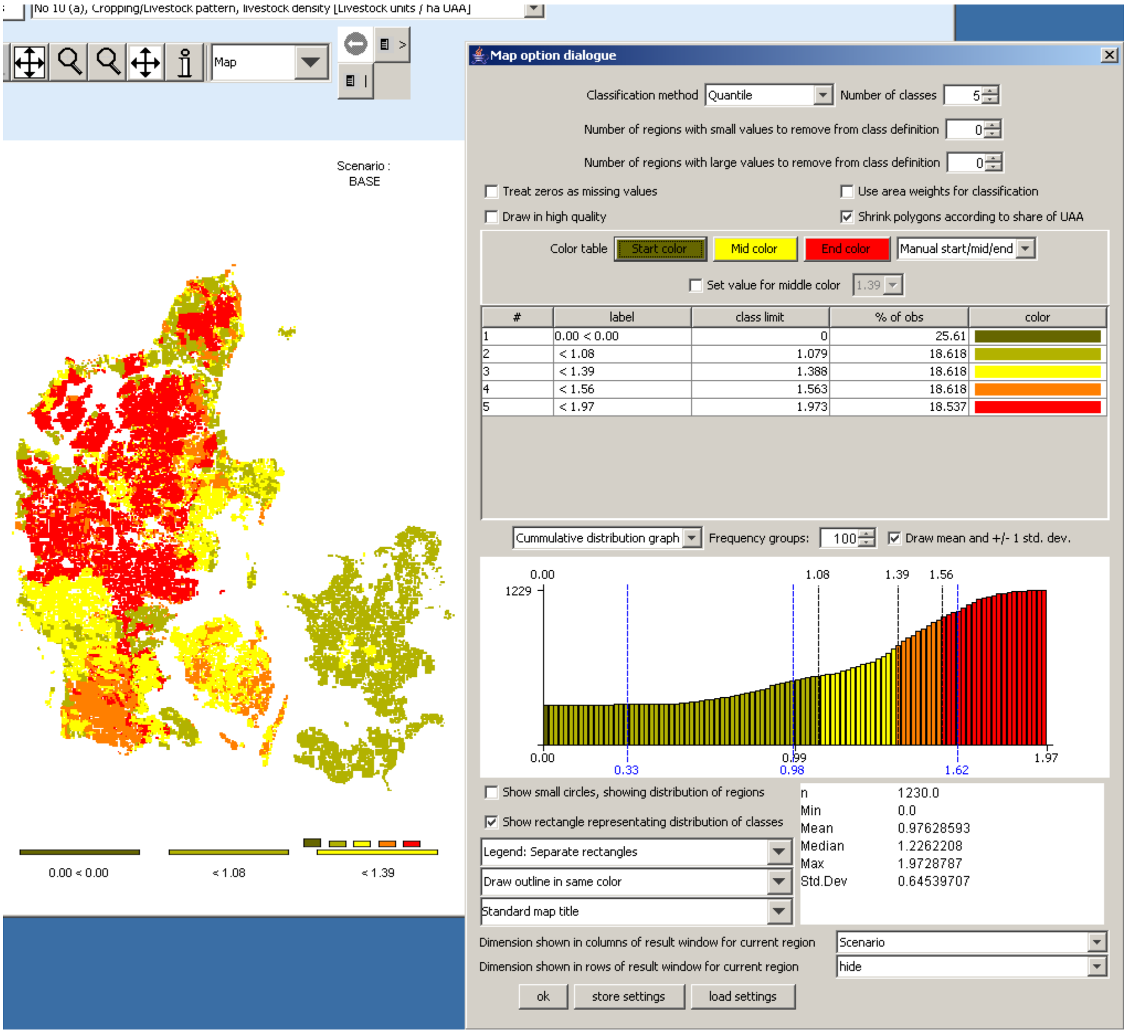
The color table defines the colors used for the classes. When choosing the color model, keep in mind that colors carry a meaning; red e.g. is generally interpreted as dangerous. Equally, it is important to think about the final medium with which the map will be published. Exporting colored maps to a black-white device will render it almost impossible to read the map. Is it best to try different color tables and different classification methods on your data. The following color models are currently available, named according to the data order from minimal to maximal value:

- *Green – Yellow – Red (standard)*. Normally, the middle class is drawn in yellow, smaller values in shades been yellow and green, and larger ones from green to reed. Should be applied e.g. to environmental indicators where the damage increase with the value of the indicator.
- *Red – Yellow – Green*, as above, only that high values are shown in green. Should be used e.g. for income indicators or environmental benefits.
- *Red – Gray - Green / Green – Gray – Red,* more available for historic reasons as they mimic the color tables of the original JAVA applet.
- *Blue – Gray - Green / Green – Gray - Blue*; introduced on demand of DG-AGRI. A good choice if the “good”/”bad” interpretation of the distribution is to be avoided.
- *Shades of grey*; sometimes needed for publications when color printing is available in the final hardcopy. Beware to use a limited number of classes.
- *Shades of blue*; useful where the notion or “bad” or “good” inheritably comprised in greenish and reddish colors is to be avoided.

Defining an own color model

Once a color model is chosen, the user can re-define the start, middle and end color using the three buttons below the color table selection row, as shown below, given a lot of freedom to generate color ramps.

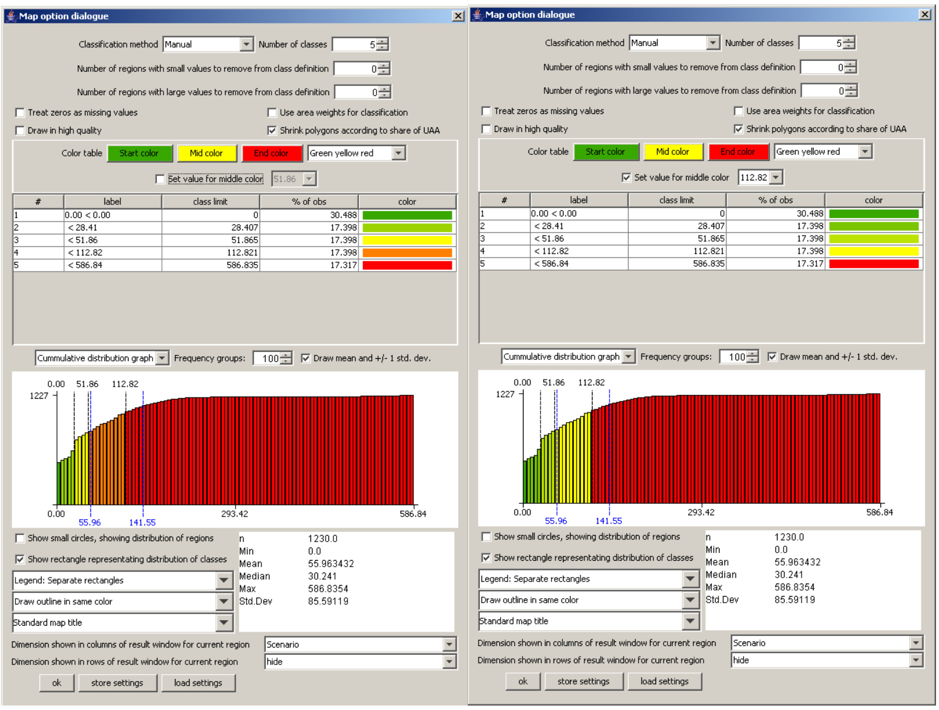




#	label	class limit	% of obs	color
1	0.00 < 0.00	0	25.61	
2	< 1.08	1.079	18.618	
3	< 1.39	1.388	18.618	
4	< 1.56	1.563	18.618	
5	< 1.97	1.973	18.537	

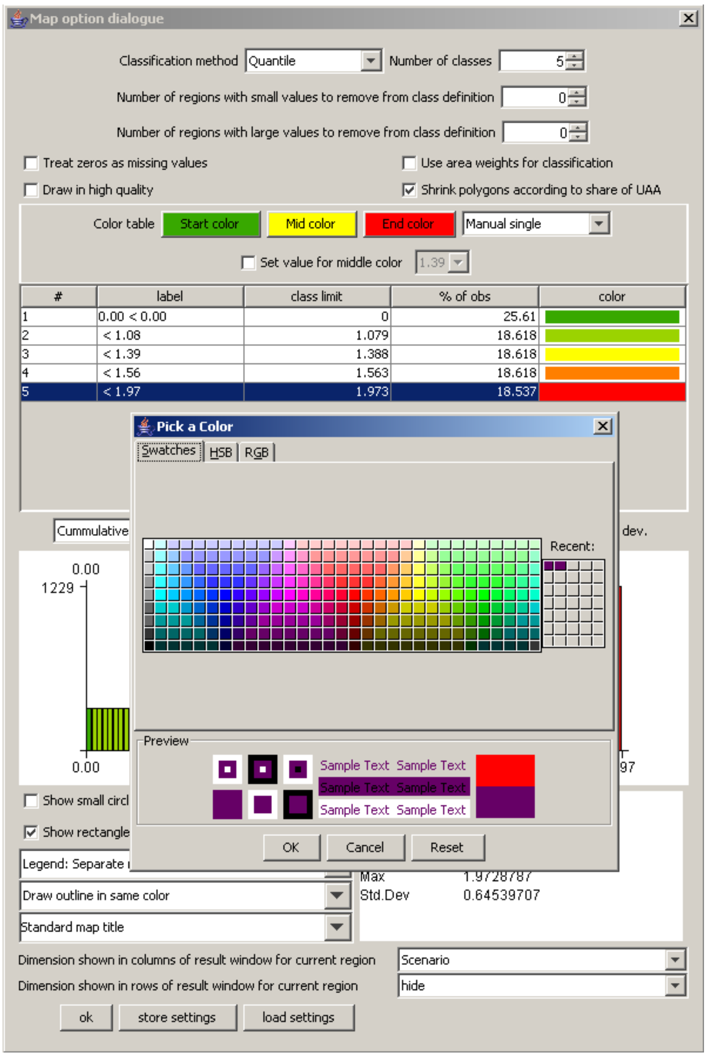
Changing the value for the medium color

Normally, the medium color (yellow or gray) is assigned to the middle class. Sometimes, the user may whish to change the class where the color switches. First, the “Set value for color change ..” must be ticked on. Next, in the now enabled drop-down box, choose the class limit for which the middle color should be used. The effect is shown below. Whereas before, values in the class below “392.70” – the middle class – was drawn in yellow. When the user now selects another class limit, the colors assigned to the classes change, here one of the shades of green is dropped and shade of red is added.



Manual set colors

Finally, the user can choose its own colors by double clicking in a color field in the legend table. That should only be done after the final definition of the class limits is set as otherwise, the manually set color will be lost.



Changing the way the legend is drawn

The map viewer puts the legend always below the map, and currently offers three options how legends are drawn:

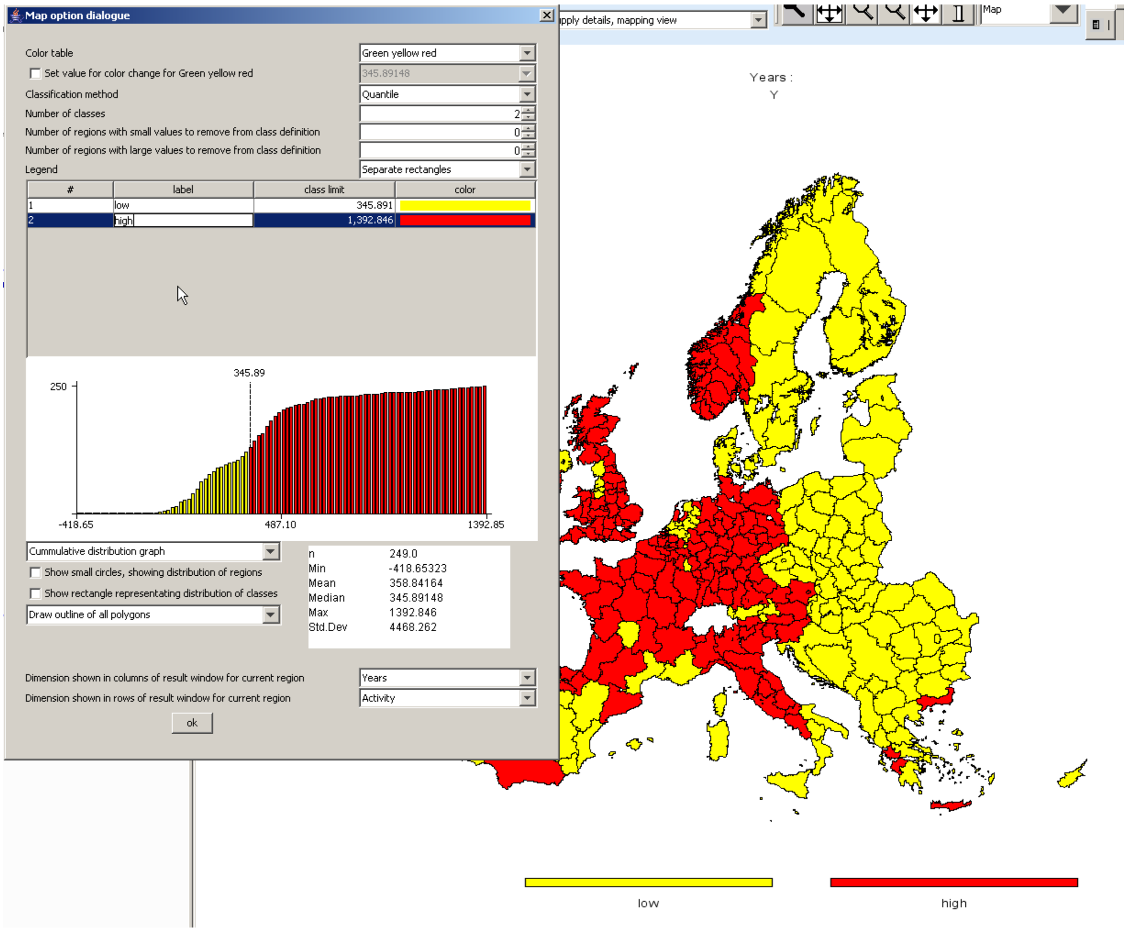
1. Separate, equally sized rectangles which show the upper class limit with the exemption of the lowest class, which shows the lower limit.

2. A continuous linear scaling bar. That gives an optical idea about the how distribution of the class limits. Overlapping of the number of avoided.

3. A continuous logarithmic scaling bar .

In all the cases, the tool dialogue can be used to set number of digits shown, e.g. reducing the number of digits to zero leads to a linear bar as shown below:

The reader is reminded that the label can be changed manually as shown below.



Copying the map to the clipboard or saving to disk

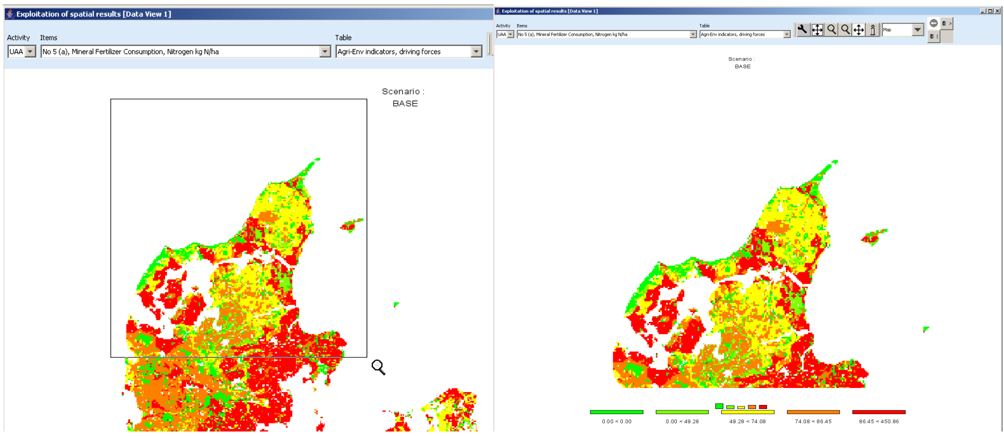
In order to export the map to other applications, the easiest way is to use the clipboard, in order to do so, press the “copy to clipboard” button. Afterwards, the current map can be imported into other applications as e.g. MS Word. Another possibility is to store the current map in jpeg format on disk, to do so, use the “export” button which will open a file dialog to choose the name of the file.

Changing the title of the map

When using output to clipboard or disk, the user may often prefer an own title or no title at all on top of the map in order to produce a caption for the map in another application. In order to refrain from drawing a title on top of the map, click into the legend part of the map, and in the dialog at the bottom, choose “none” in the row labeled “Title on top of map”. Alternatively, the user can simply write something in the box.

Zooming in and out and navigating in the map

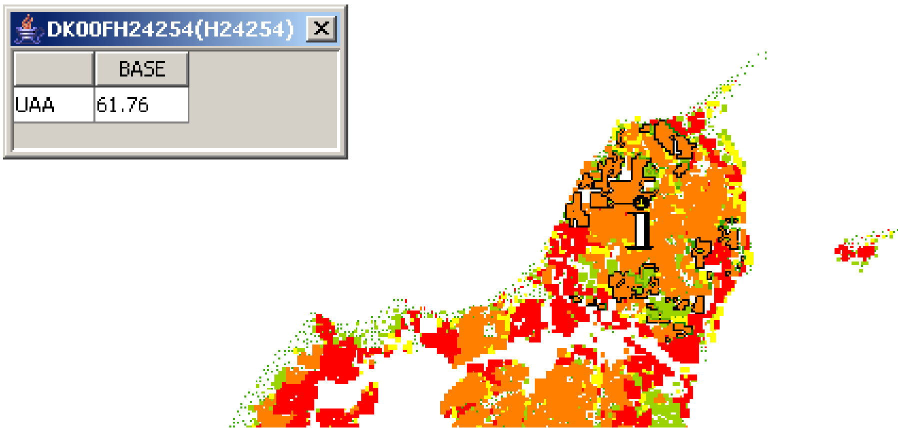
In order to zoom in part of the map, press the button. The mouse pointer will change to a magnifying class with a cross in it. You can then mark an area on the map by pressing the mouse button, dragging and releasing the mouse. After the mouse is released, solely the selected zone of the map will be drawn, without changing the class limits or any other setting. Clicking with the mouse while in zoom in mode will increase the map resolution step-wise by 25% and center the map at the current mouse position.



By clicking with the zoom out pointer on a point of the map, the point becomes the new center point of the map and the map resolution is reduced stepwise by 25%. Equally, you may drag the map while keeping the current resolution by choosing the drag pointer . And finally, in order to return the original full sized map, use the "full extent" button .

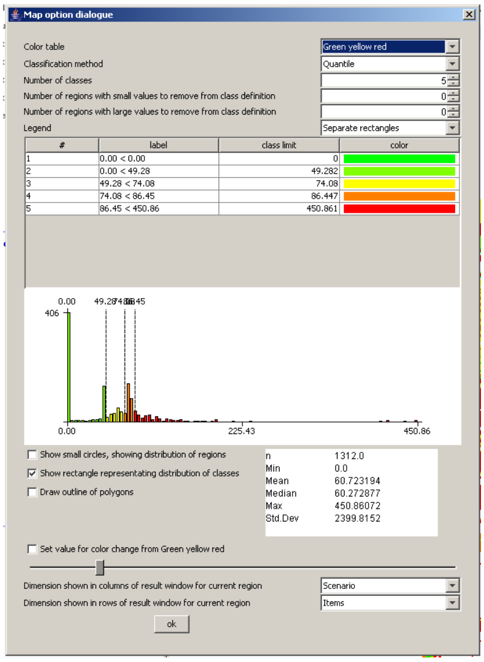
Getting data for specific polygons

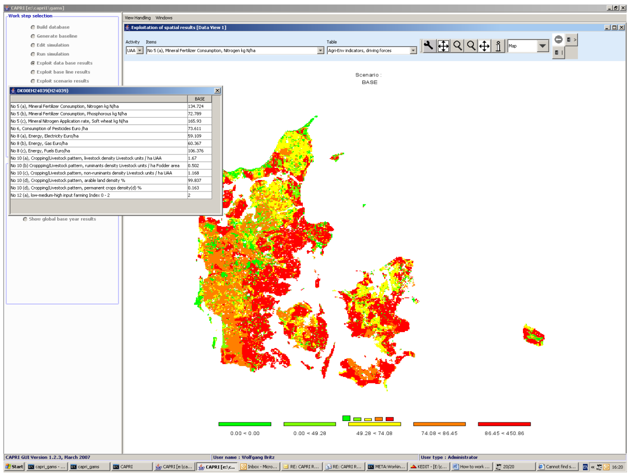
The info pointer will open an additional window as shown below which displays information on the current polygon – the circle above the "i" being the focus point. The new window shows in his title bar the code and, if available, the long text of the polygon currently pointed to with the info pointer. The content of the info window is continuously updated when the mouse is moved over the map, and all polygons belonging to the same region as the one where the mouse is over are highlighted.



If the user opts to use one of the comparison options to be shown (percentages, differences, normalization) by clicking on "customize" button, an additional column is automatically added to the info window showing the comparison value used. That is especially helpful when the map shows only differences.

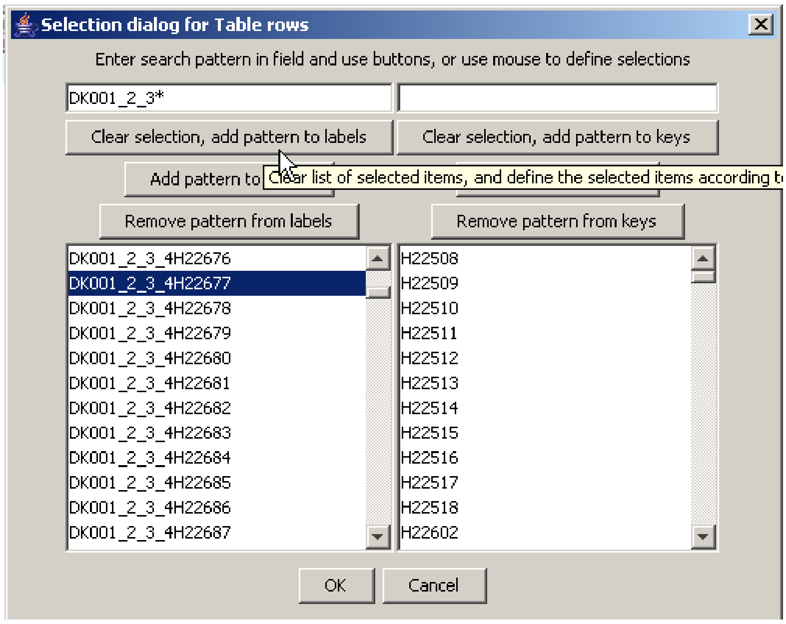
The content shown in the info window is not fixed, rather, the user can decide which data dimensions to use for the columns and rows by using the "map option dialogue" by clicking on the legend of the map. If the user e.g. switches to "items" instead of "activity", the "info" window will look like shown below. An alternative is to use a second tabular view in addition to the map.



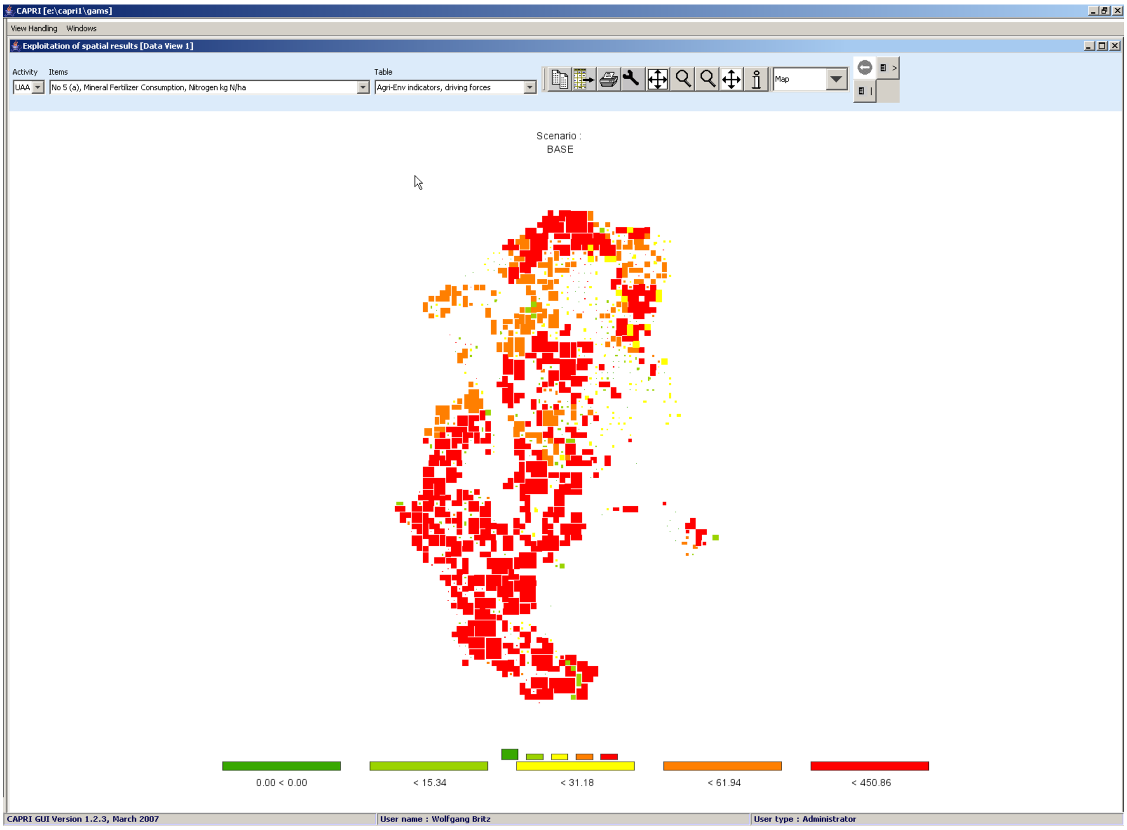


Highlighting specific regions in the map

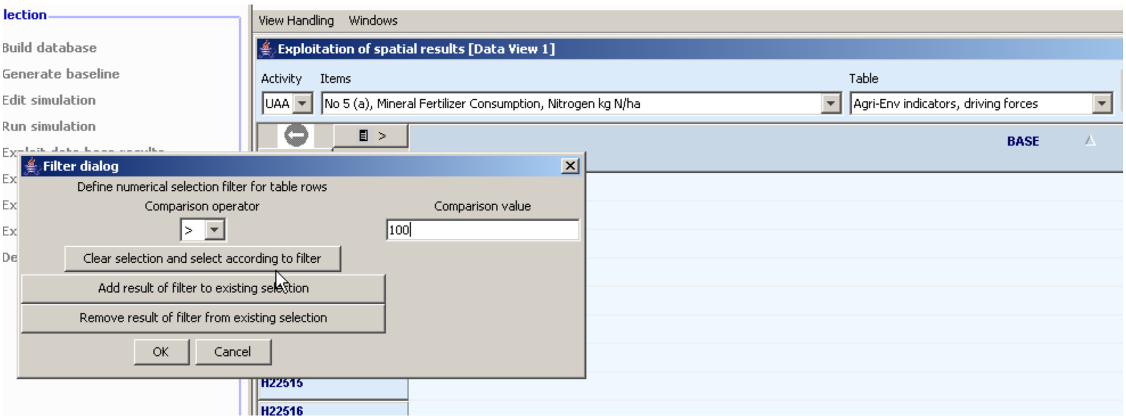
Sometimes it may be interesting to see the spatial distribution of specific data or data constellations. All views open the possibility to (de)select columns and rows, allowing e.g. to use the NUTS code in front of the numerical HSMU code to select only the HSMU belonging to specific administrative regions. That possibility is explained in short. First, press the row selection button (“Open selection dialog for table rows”) which will open the following dialogue.



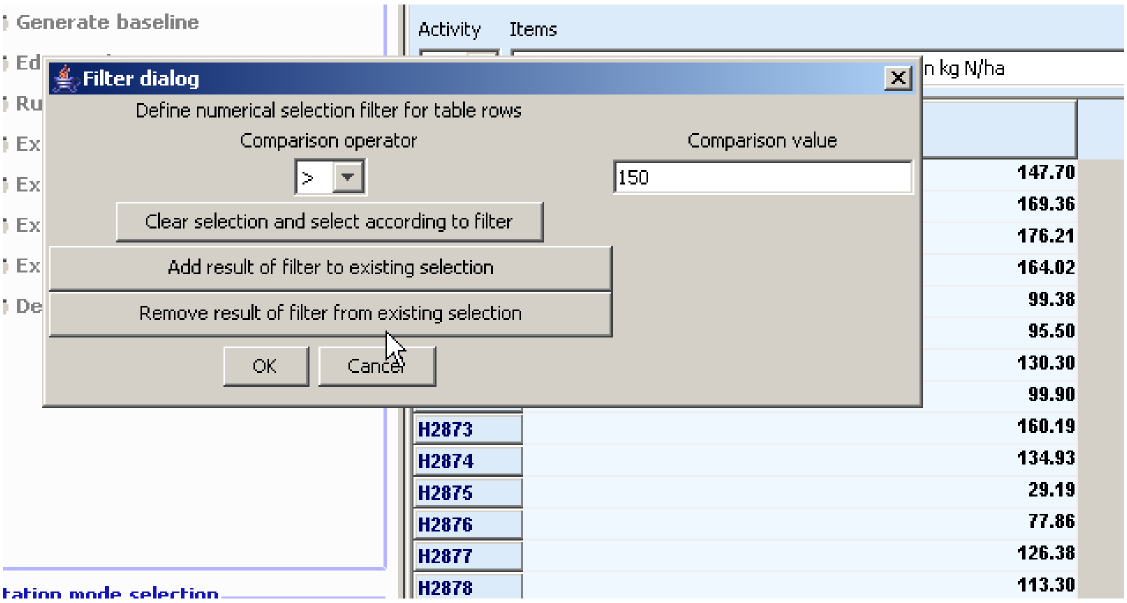
Now, we may e.g. selected only the HSMU belonging to the FSS region DK000_1_2_3 by typing “”DK001_2_3” in the left input box, and then choosing “Clear selection, add pattern to labels”. Afterwards, the map will look as shown below.



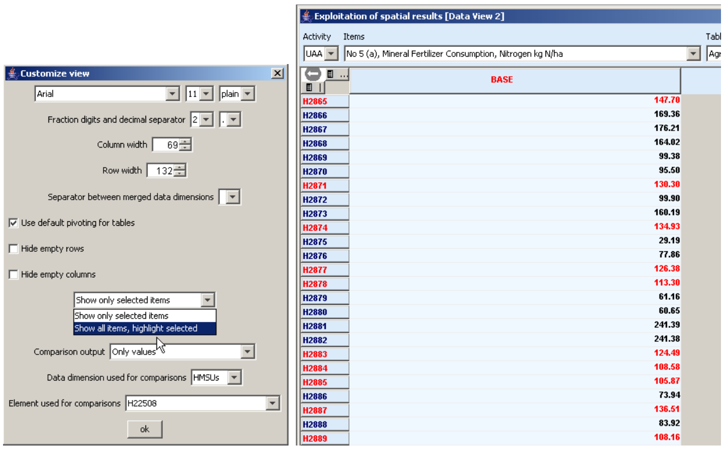
The tabular view opens up the possibility of using numeric filters, an option discussed in the following. Take as an example the task to select all regions where the Nitrogen Fertilizer Consumption is between 100 and 150 kg/ha. First, witch from map to tabular view. In the table click with the right mouse button in the column header of that column holding the values to which the filter should be applied, as shown below. We will need to apply the filter step-wise, first e.g. selecting all values greater then 100 and then removing those which are above 150.



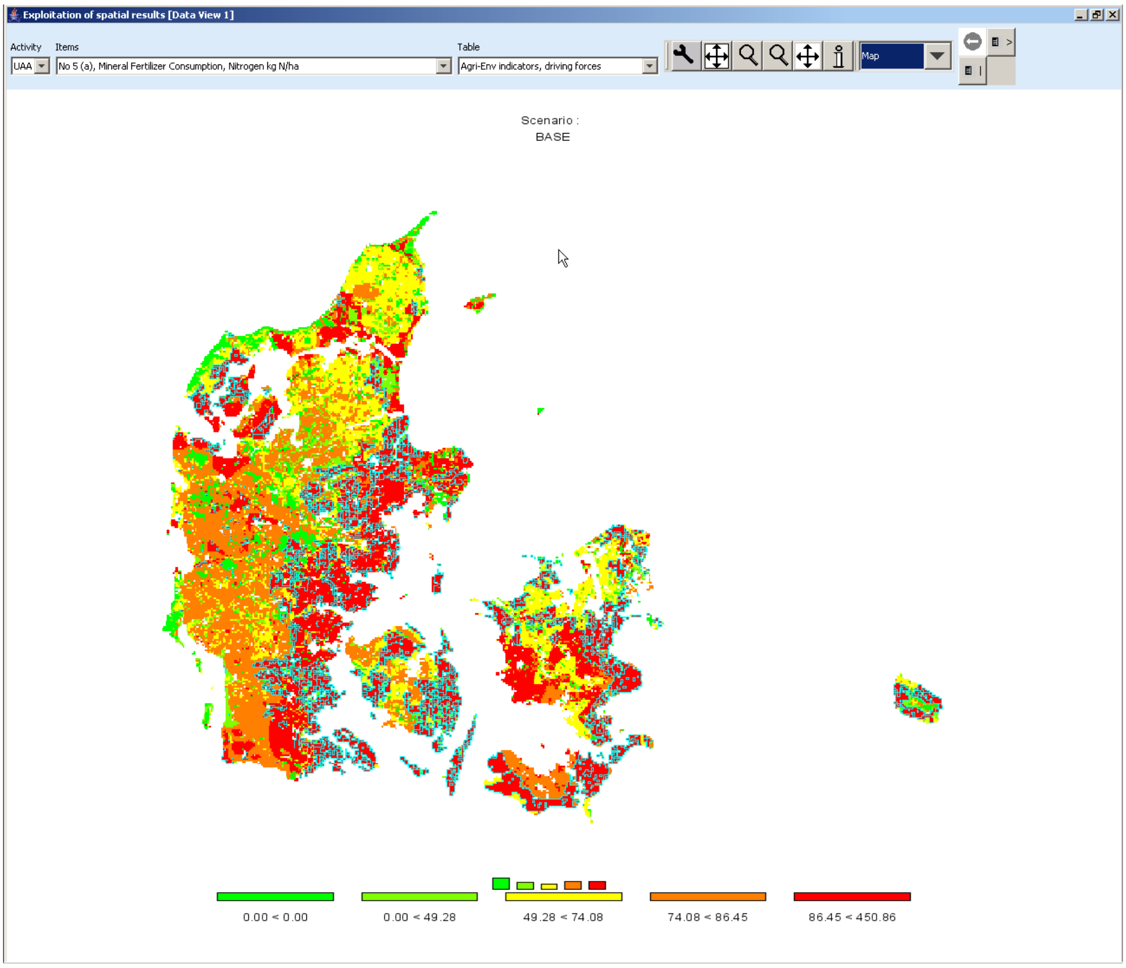
After pressing on “clear selection and select according to filter”, and then on “ok”, the table will only show such regions where the value in the column “BASE” is above 100, as shown below. Next, we most exclude the regions above 150 kg/ha. To do so, set the filter to “>” “150”m and then press “remove result of filter from existing selection”.



Now, drawing a map with just those regions is not so interesting. However, with the tool dialogue, we can highlight the selected value instead of hiding all others. The selected rows are now shown in red in the tabular view.



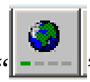
When we now draw the outlines of the selected polygons only (see map option dialogue), the map will draw the outline of the selected regions in cyan and thus highlight them. The row selection will be maintained when the pivot or the table is changed as long as one of the selected items can be found in the rows of the new table. The example map shown below is certainly not so interested, as changed class limits could have done basically the same job. However, we could switch e.g. to grass land shares to see if that fertilizer input is more often found on arable or grass land.

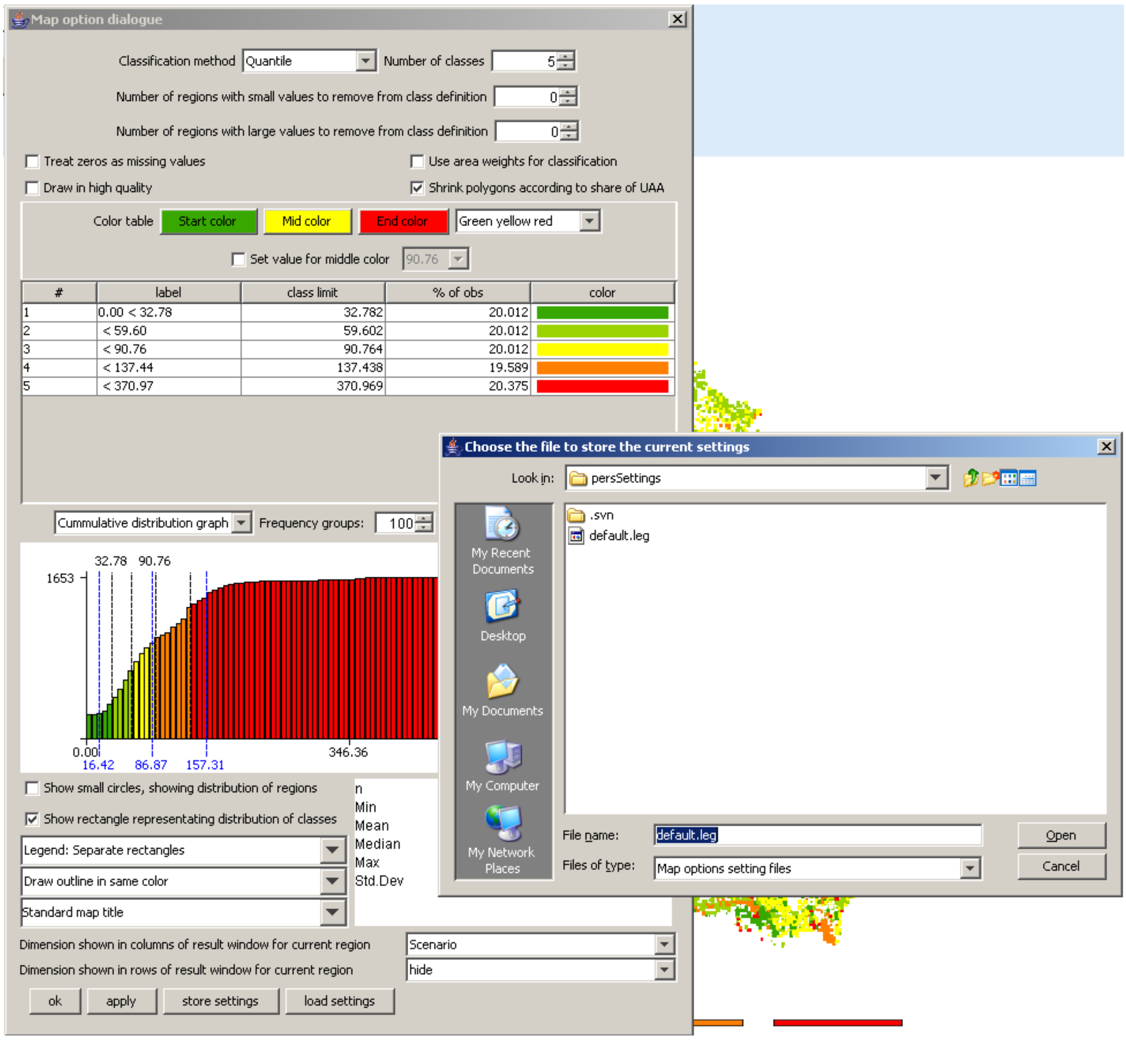


Updating the map

Generally, the map is updated automatically when the user changes an option impacting on its layout as long as long as the number of visible polygons is below 20.000. If that amount is exceeded, the classification dialogue is updated immediately, but not the underlying map. In order to apply the changes, the "apply" button must be pressed. The user is informed that the "ok" button will also update the map, so that an "apply" immediately before an "ok" is not necessary.

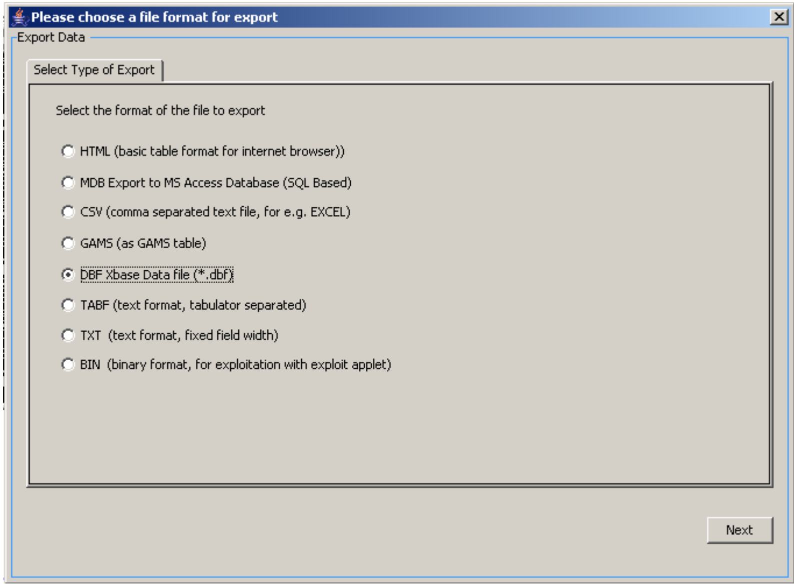
Storing and re-loading your settings

Open the map option dialogue by pressing the map option button "". Change the settings according to you needs, and then press the "store settings" button in the lower part of the dialogue. Choose a file name and location. You may later used "load settings" to retrieve them again, and apply them to another map.

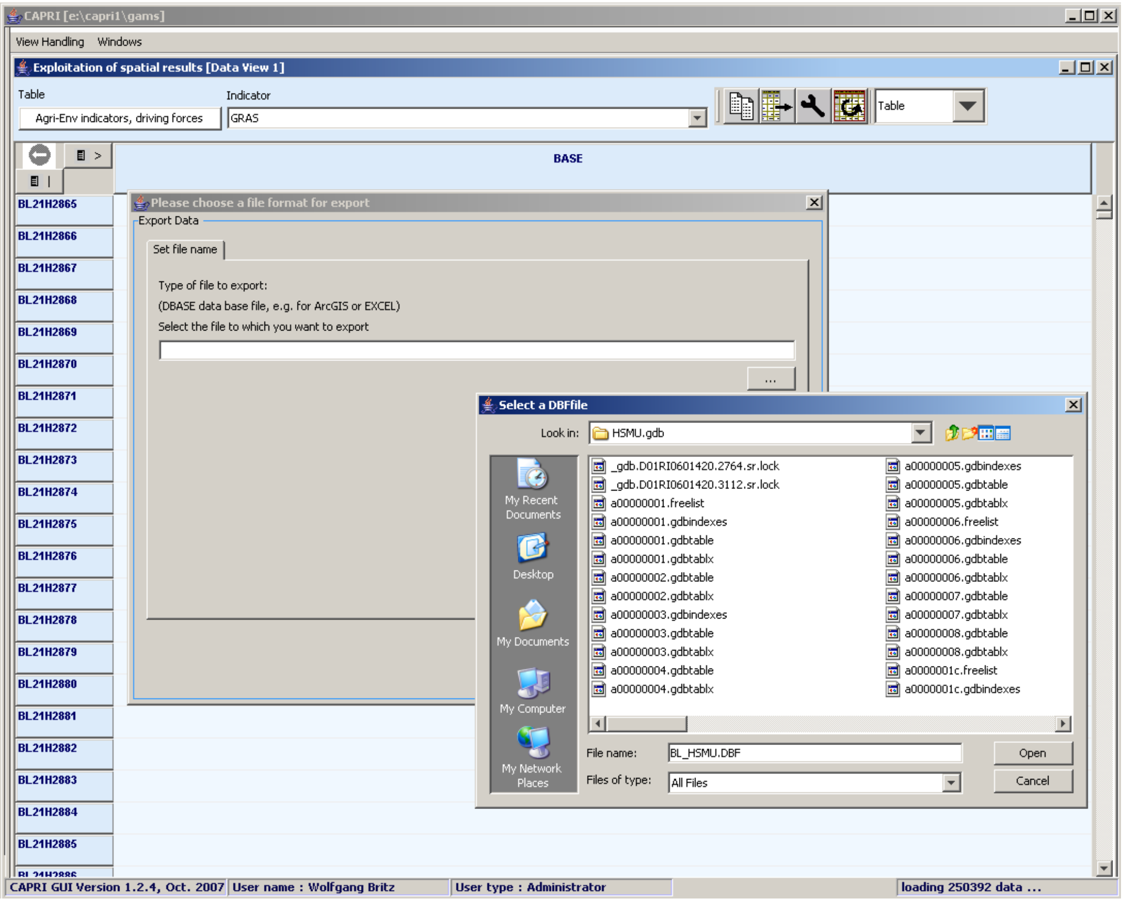


Exporting the data underlying the map

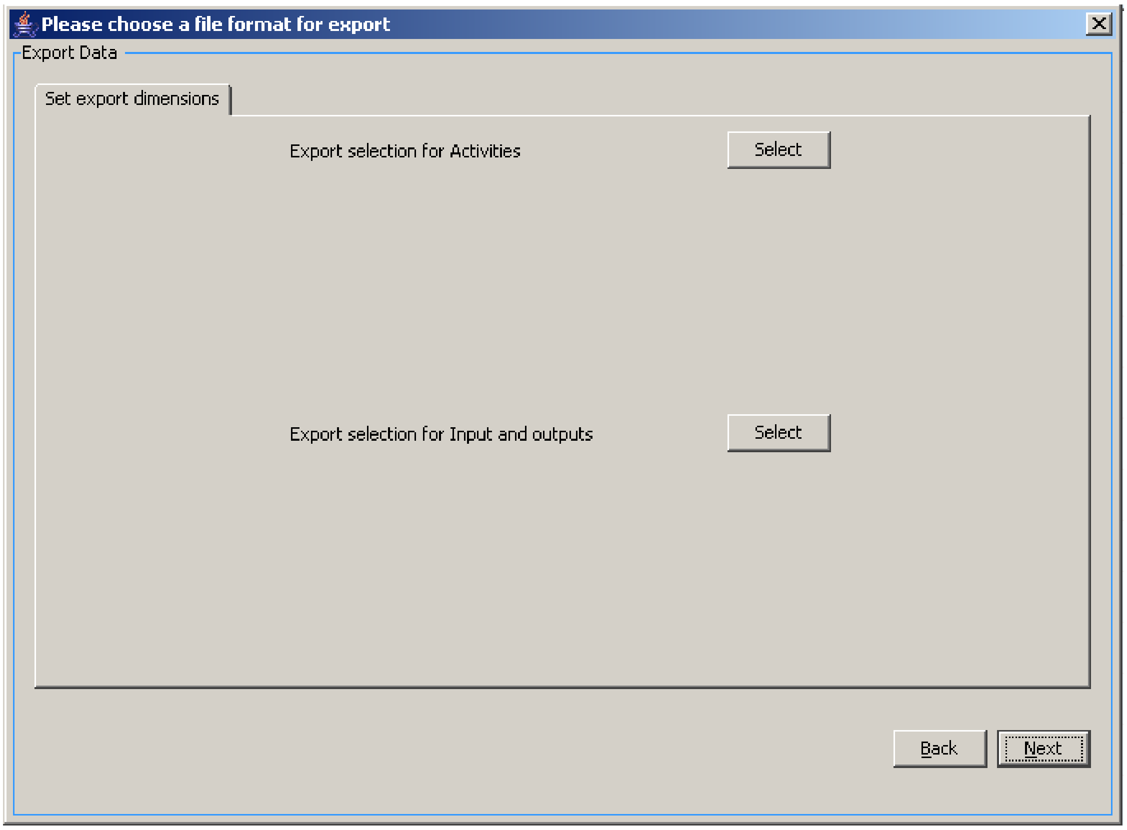
As mentioned above, the mapping viewer is part of the CAPRI exploitation tools which is in its core based on pivot tables. In order to export the data, e.g. to GIS system, the view must first be changed to tables. Afterwards, the  button which will open a file dialog as shown below. For GIS-export, e.g. to ArcGIS, DBF is the recommended format.



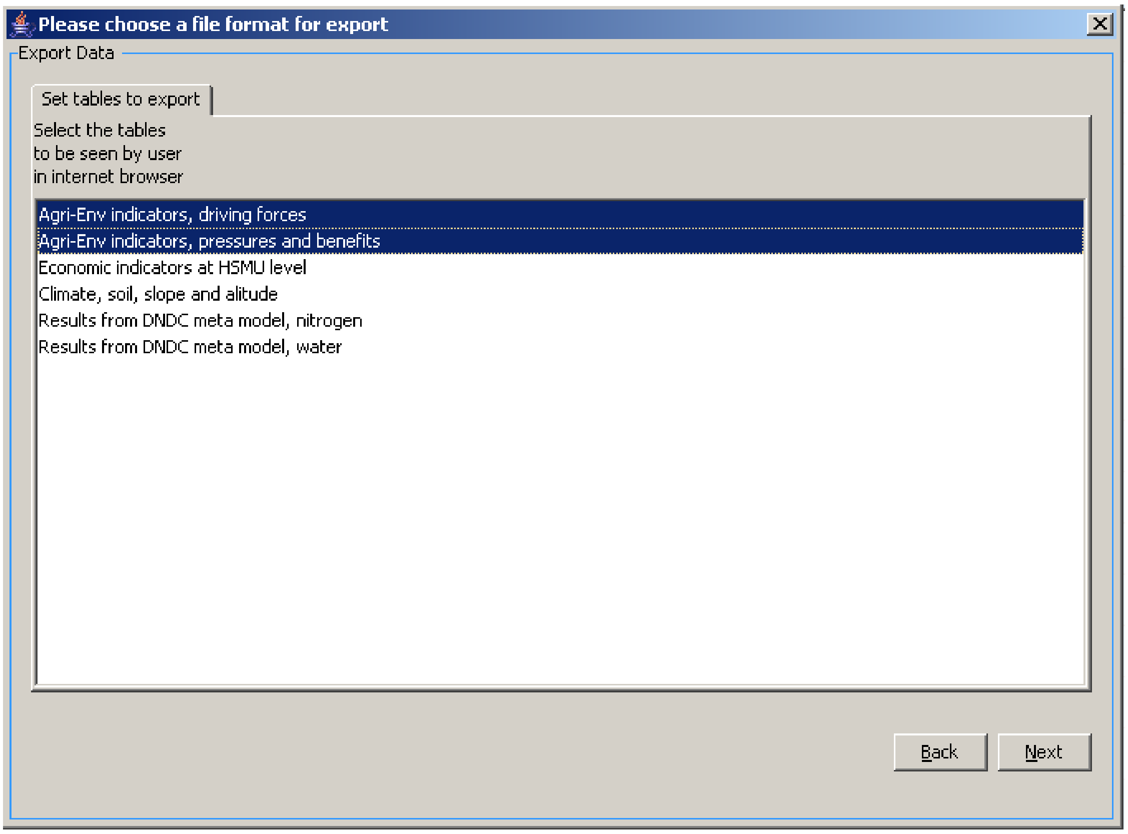
Once next is pressed, the next pane will open a file dialog to choose a file. In the case of export to a Microsoft Access Data Base, the file must exist.



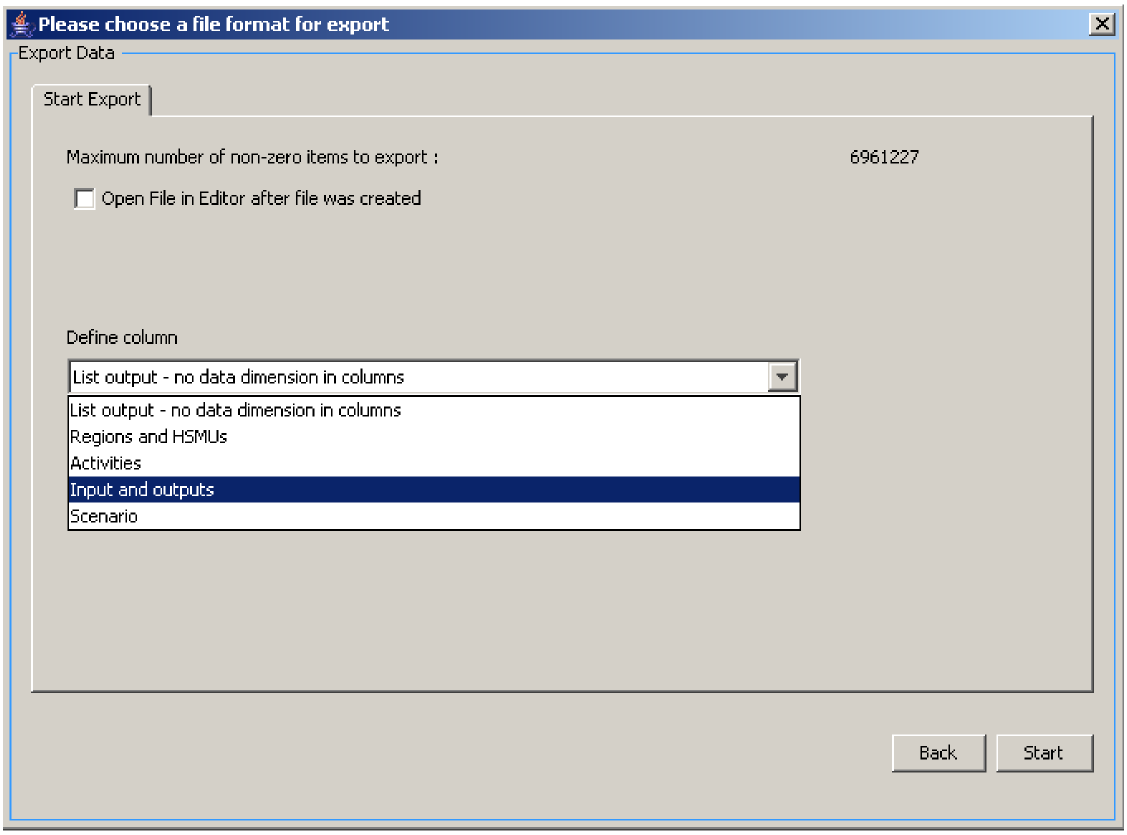
The next pane allows, if whished to open selection lists for the different data dimensions.



Next, the tables for export can be selected.



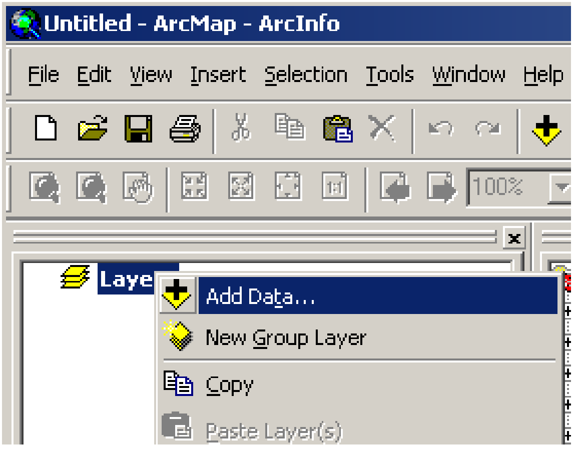
Beware: the pre-defined table structure will be lost, as will the long-texts and units attached to the tables. However, in the case of DBF-export, a second file with that information will be automatically created. If you solely want to export the table you have currently up front, use the “copy to clipboard” button. The clipboard export will retain the pivoting and further information.



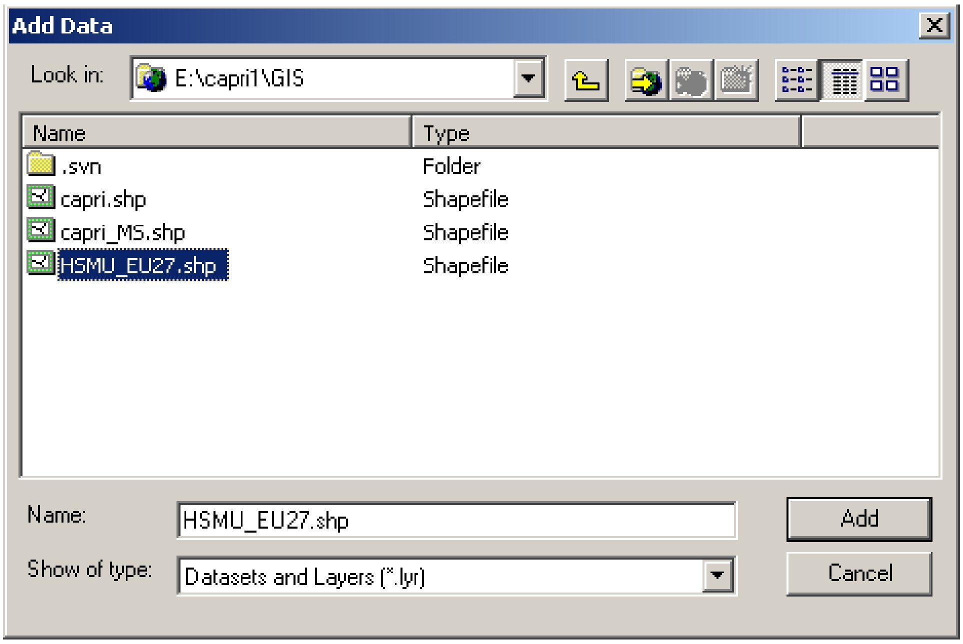
The last pane let's you decide for DBF-export if you want a list, or if you want of the data dimension spanned across the columns. For exporting the HSMU tables, it is recommended to put "Inputs and outputs" in the columns.

If everything has worked well, we should now find two files: one with the data, named as chosen in the file dialog, and a second one with "_meta" introduced before the file extension.

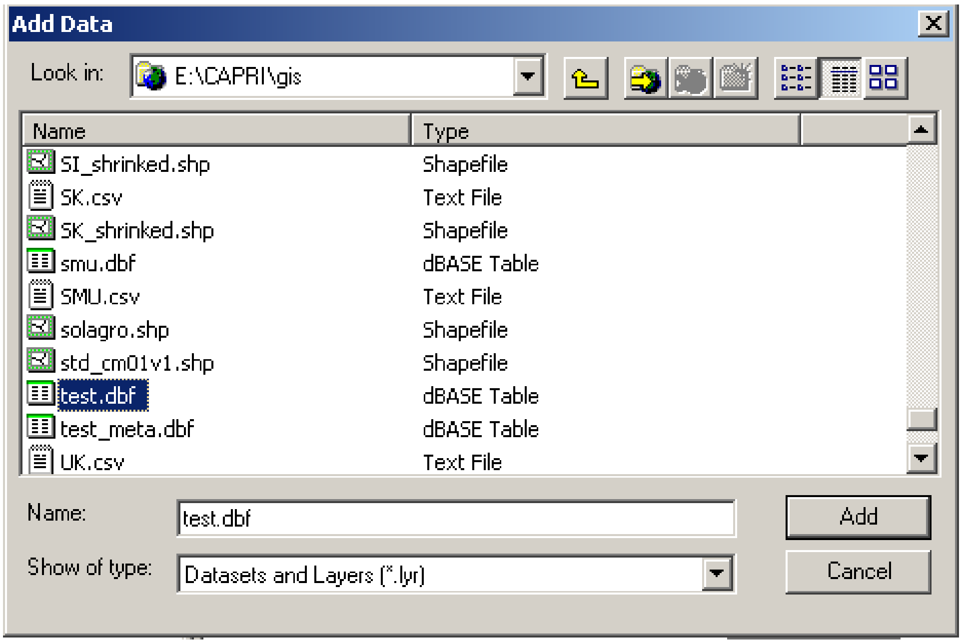
The following section will briefly explain how to the now work with the data in ArcGIS. Under Layers, choose add Data,



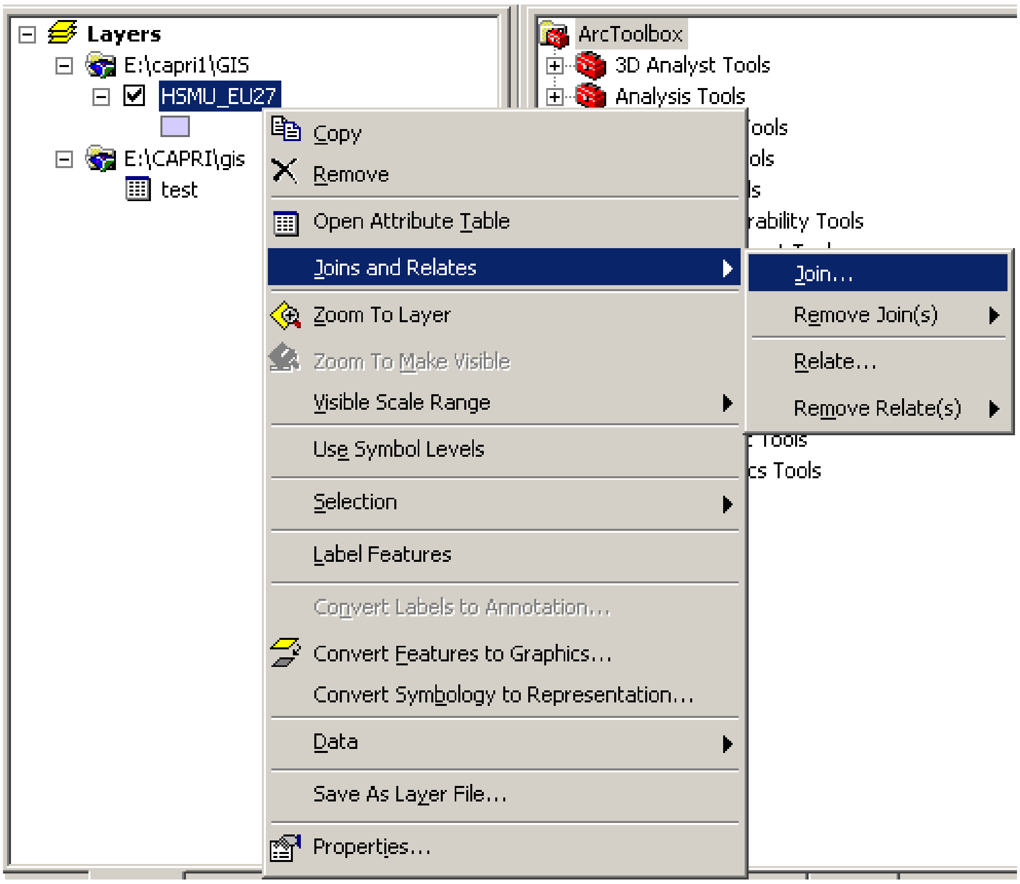
and in the case of the HSMUs, add the “HSMU_EU27.shp” shapefile.



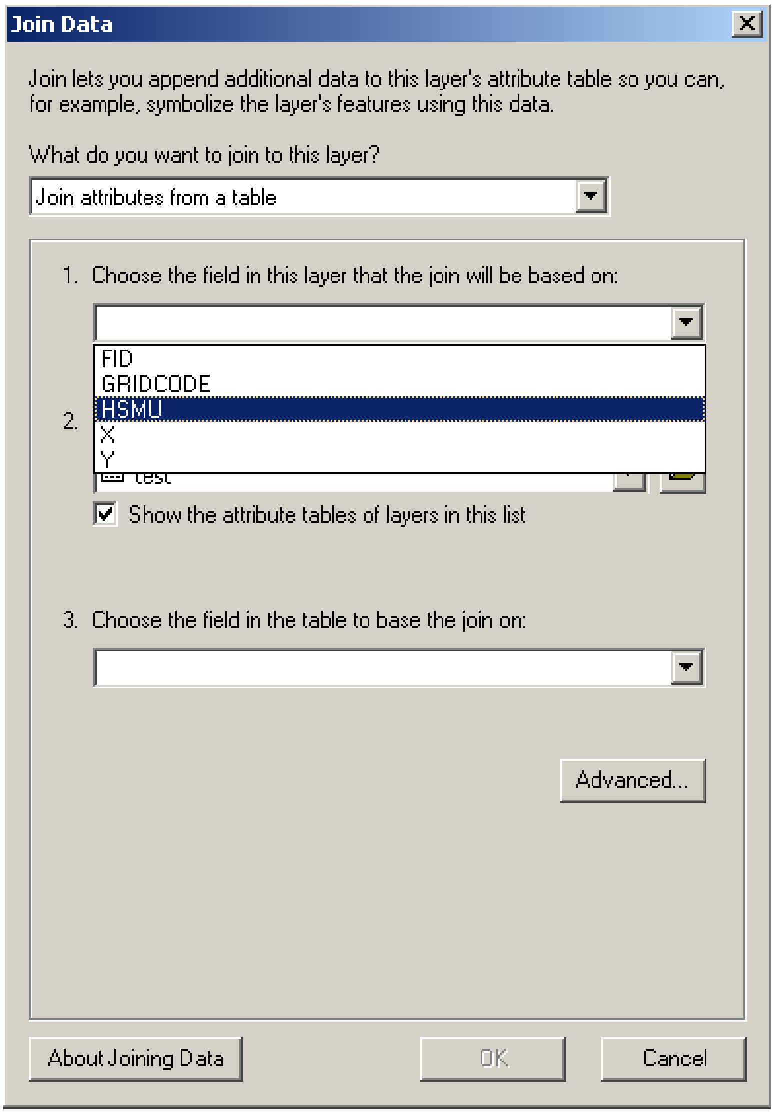
Then, choose add layers again, and add the dfb-file you generate in the step explained above. You may also add the file with the meta data.



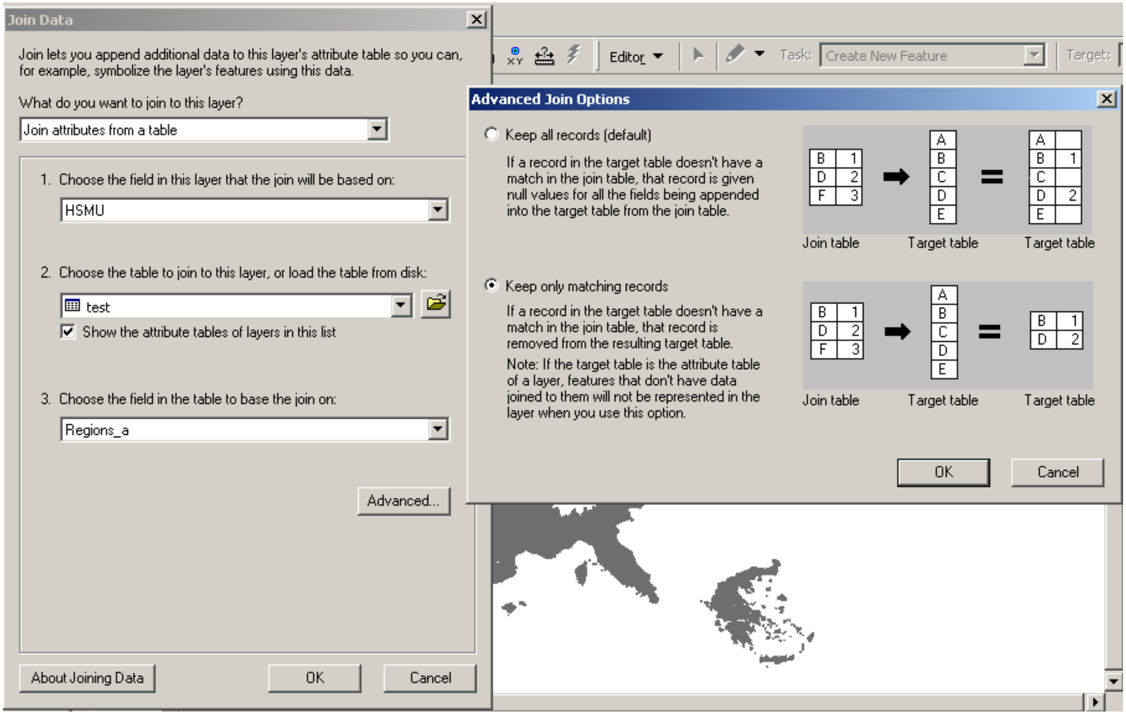
Next, we need to connect the HSMU geometry with the newly loaded data, a process called "joining" in ArcGis. In the context of "HSMU_EU27", choose "Join and Relates", then "Join …".



That will open the join dialogue as shown below.



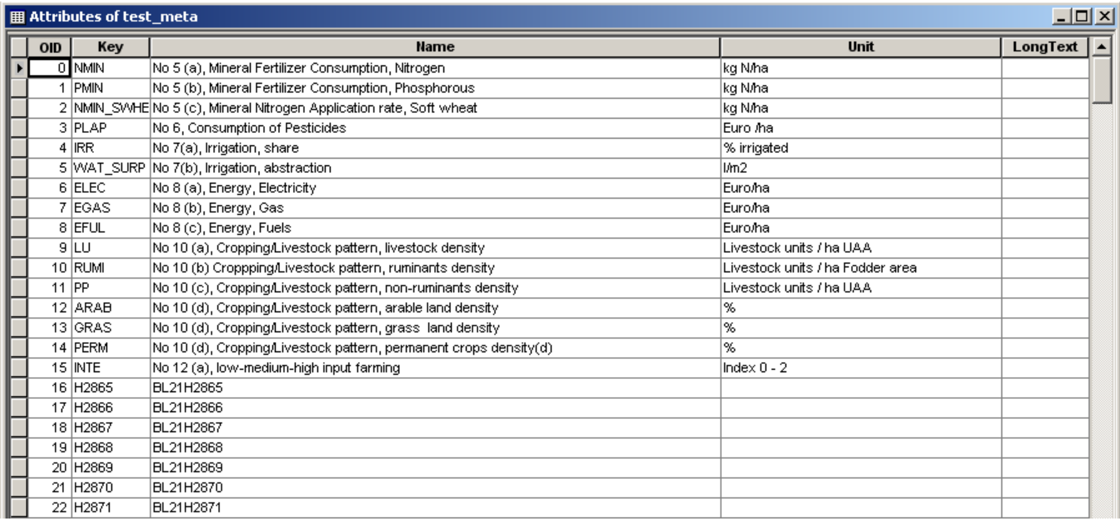
Make sure that “Join attributes from a table” is set in the first drop down box, and under 1., select HSMU, i.e. the filed in the HSMU_27 geometry where the codes for the HSMU polygons are stored. Use the name of the exported dbf-table under 2., and select the field “Regions_a” (the field name are restriced to 10 chars) under 3. Then press the button labeled “advanced”, and chose the radiobutton “keep only matching records”. If you are asked to build index, confirm.



If anything has worked well, you should now see the country or countries you had in the original map.

There is a trap, though. If you export several tables, or results for several scenarios, your table will normally have several fields used as a row header (e.g. year, scenario, activity). If that is the case, the join will not work properly as several rows for the same regions will be joined to the very same polygon. Unfortunately, ArcGIS will not warn you about that. You have *first* to execute a definition query in the table, selecting those rows you are later going to draw a map from.

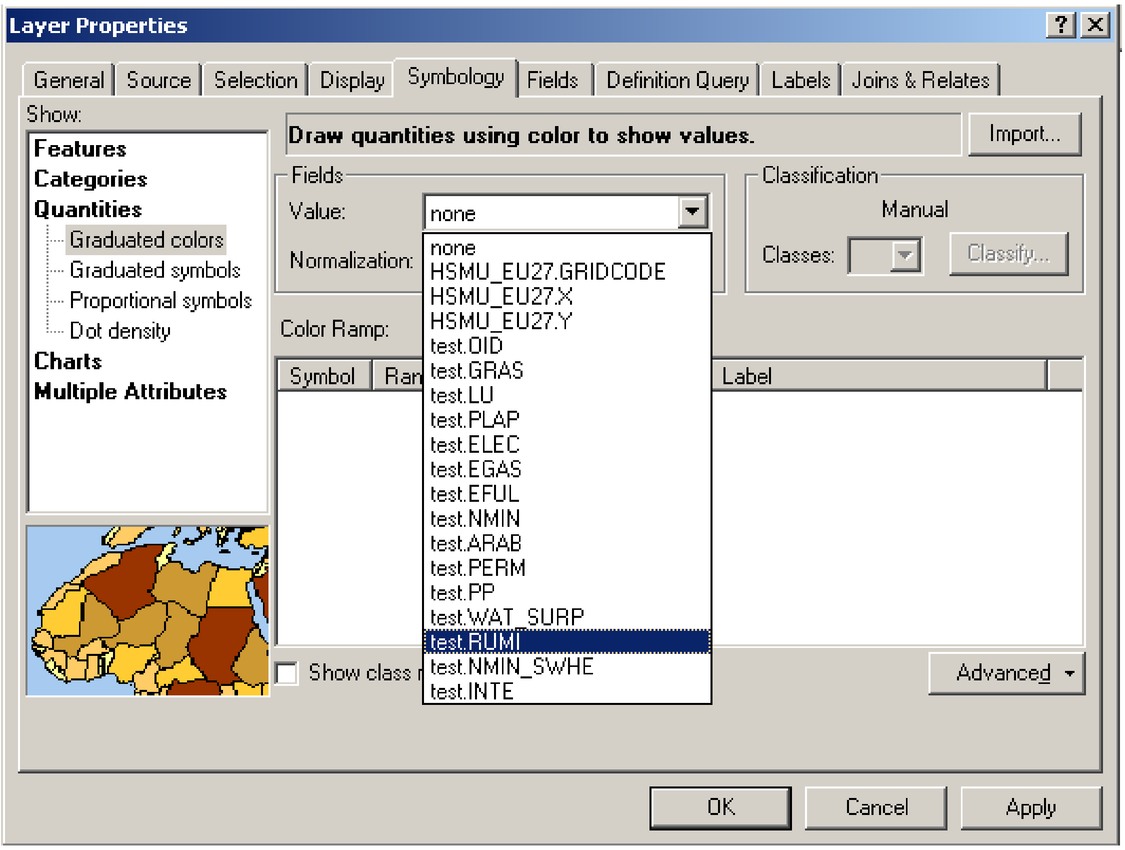
In order to draw now a thematic map, it may be helpful to add the file with the meta data to the map, and with the help of its context menu, open the meta-data table. It will give us the long description and units belonging to the data fields in the exported data table.



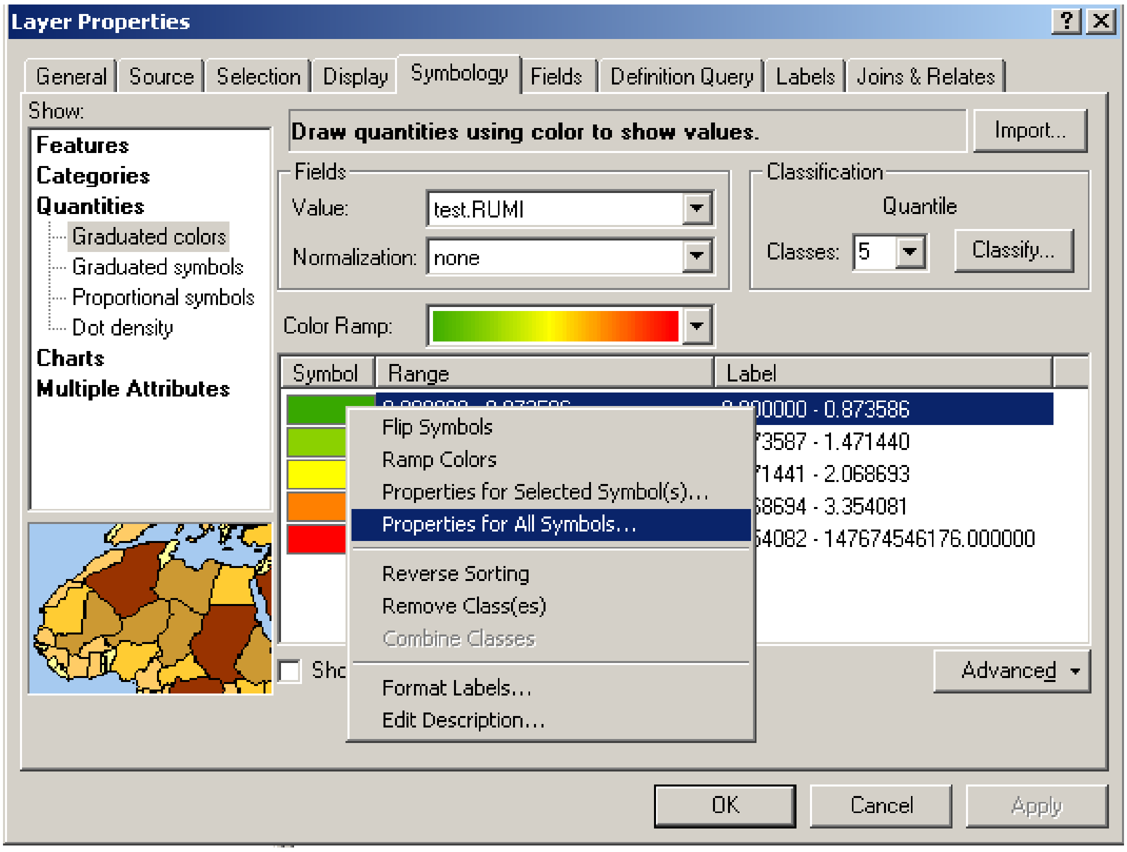
Attributes of test_meta

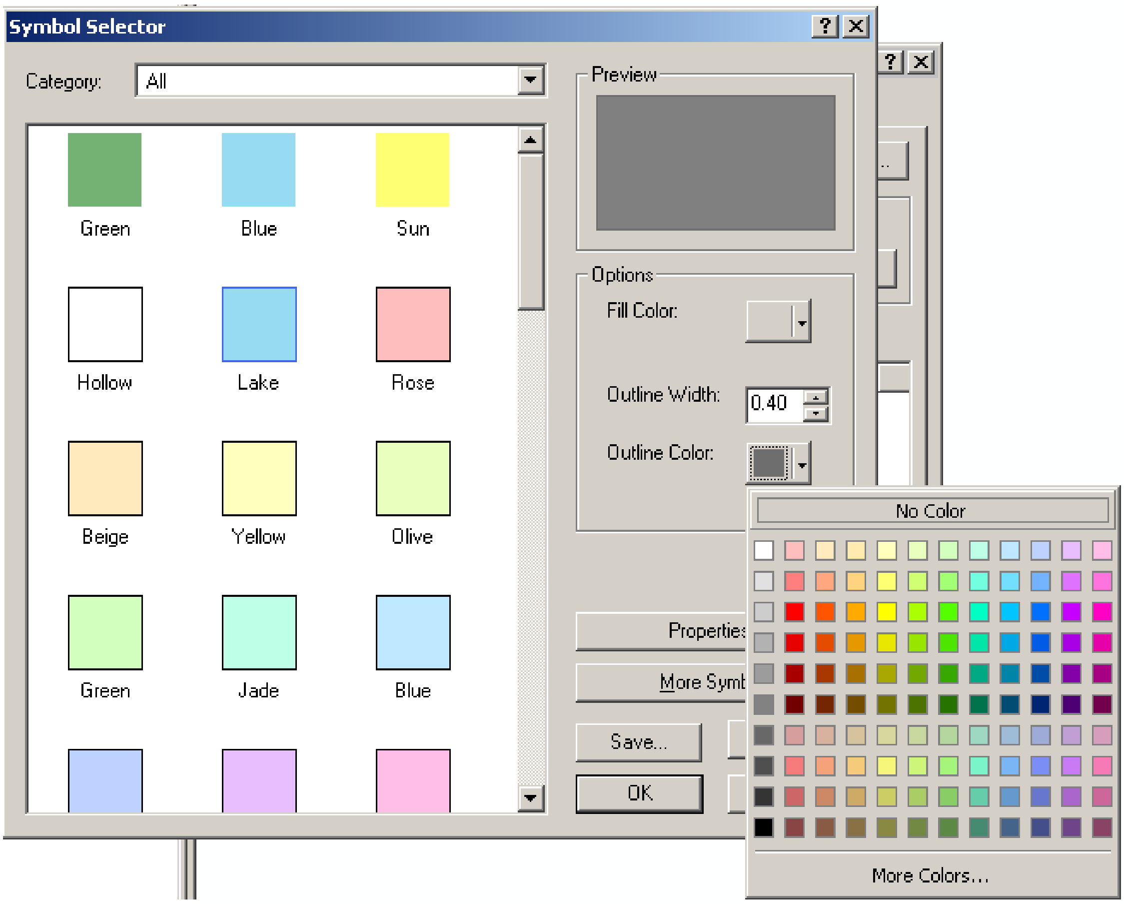
OID	Key	Name	Unit	LongText
0	NMIN	No 5 (a), Mineral Fertilizer Consumption, Nitrogen	kg N/ha	
1	PMIN	No 5 (b), Mineral Fertilizer Consumption, Phosphorous	kg N/ha	
2	NMIN_SWHE	No 5 (c), Mineral Nitrogen Application rate, Soft wheat	kg N/ha	
3	PLAP	No 6, Consumption of Pesticides	Euro /ha	
4	IRR	No 7(a), Irrigation, share	% irrigated	
5	WAT_SURP	No 7(b), Irrigation, abstraction	l/m2	
6	ELEC	No 8 (a), Energy, Electricity	Euro/ha	
7	EGAS	No 8 (b), Energy, Gas	Euro/ha	
8	EFUL	No 8 (c), Energy, Fuels	Euro/ha	
9	LU	No 10 (a), Cropping/Livestock pattern, livestock density	Livestock units / ha UAA	
10	RUMI	No 10 (b) Croppping/Livestock pattern, ruminants density	Livestock units / ha Fodder area	
11	PP	No 10 (c), Cropping/Livestock pattern, non-ruminants density	Livestock units / ha UAA	
12	ARAB	No 10 (d), Cropping/Livestock pattern, arable land density	%	
13	GRAS	No 10 (d), Cropping/Livestock pattern, grass land density	%	
14	PERM	No 10 (d), Cropping/Livestock pattern, permanent crops density(d)	%	
15	INTE	No 12 (a), low-medium-high input farming	Index 0 - 2	
16	H2865	BL21H2865		
17	H2866	BL21H2866		
18	H2867	BL21H2867		
19	H2868	BL21H2868		
20	H2869	BL21H2869		
21	H2870	BL21H2870		
22	H2871	BL21H2871		

Assuming we want to draw now a map with the ruminant stocking density, we find it in row 10 under the key “RUMI”. In order to produce now a map, we have to open the context menu of “HSMU_EU27”, and choose properties, symbology, and choose “Quantities”. Under values, choose “RUMI”, the name before is the name of the DBF-file.

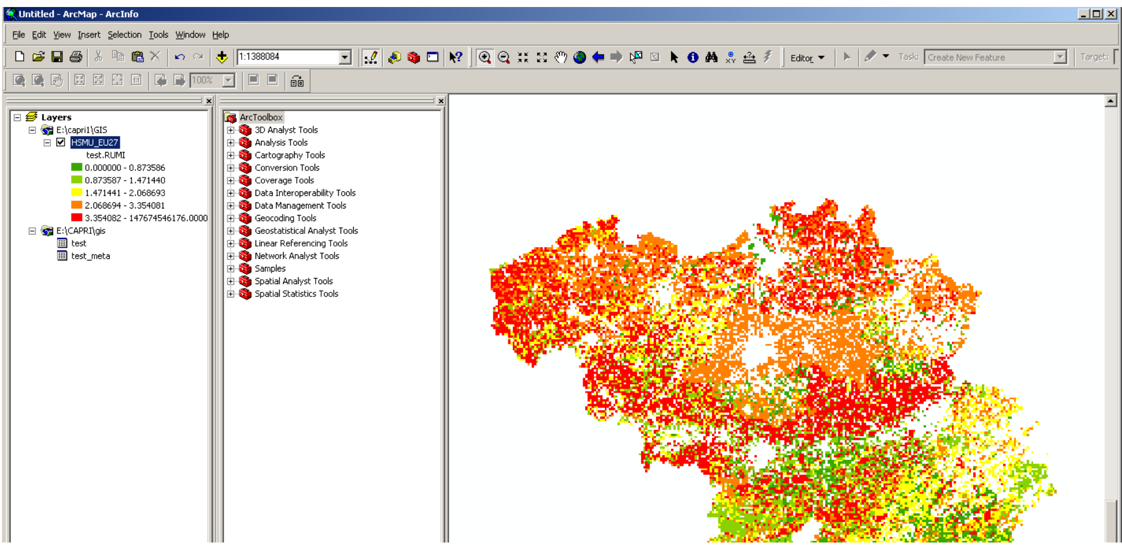


Afterwards, under classification, choose your preferred one. As there many small polygons, the outline of the polygons should not be drawn, click on one of the colors, choose “Properties for all symbols …” and under “Outline color” chose “No Color”.





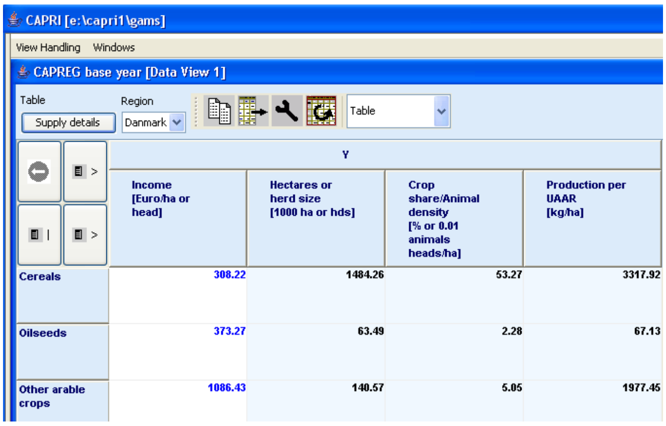
Afterwards, if anything went well, you should see your map.



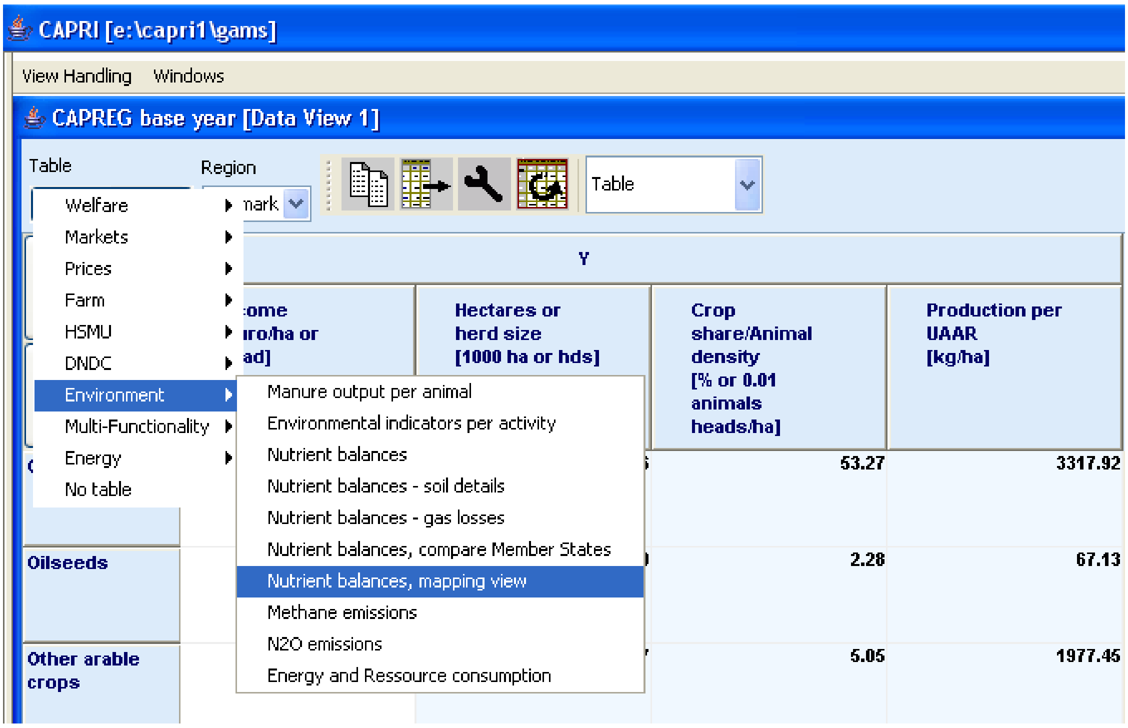
Examples

Drawing a map showing the nitrate surplus for EU27 at regional level in the base year

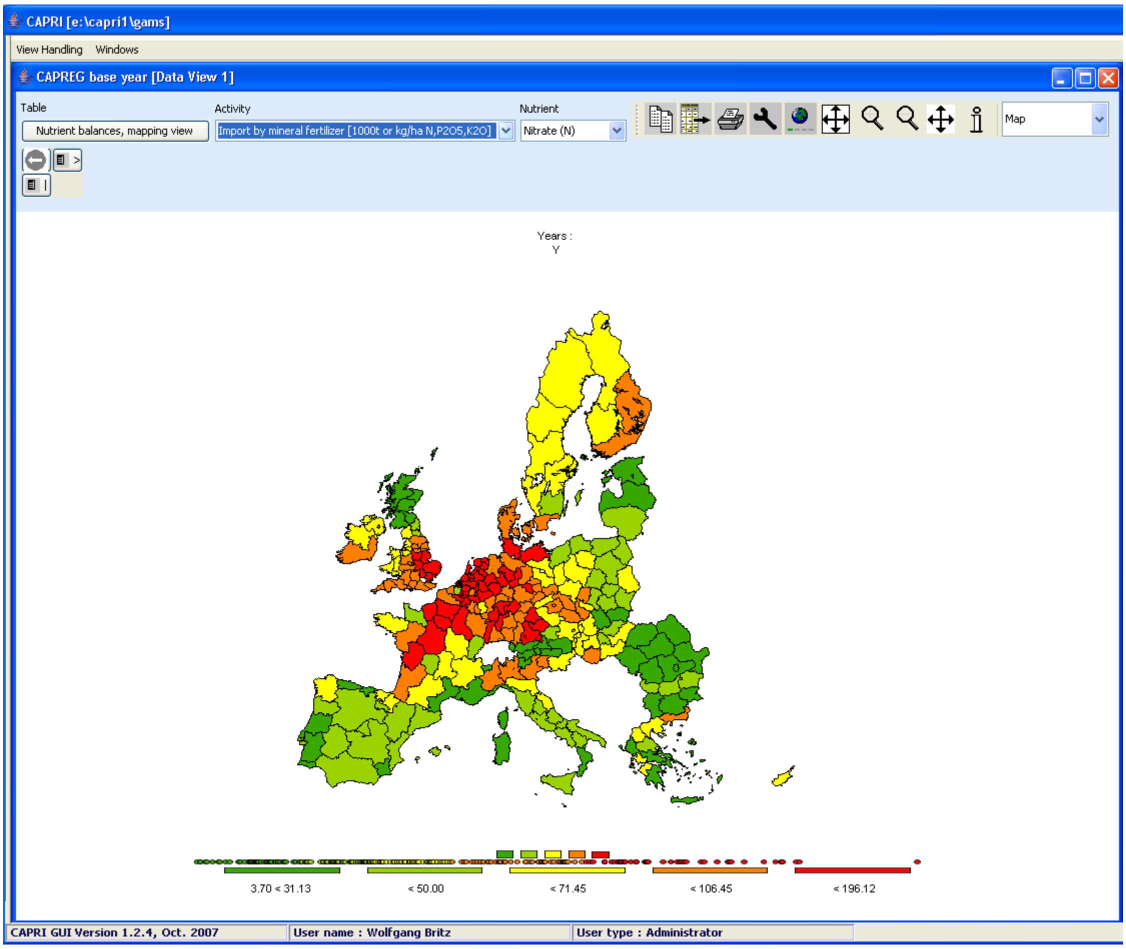
Firstly, we need to select "Exploit data base results" in the work step selection panel, and then choose the radio button "Show CAPREG base year data". Then, in the Member States drop-down list, right click the mouse, and select "EU27". The "Load and show button will then load the results.



Next, select a different selection of tables by pressing on the button below "Table", which currently shows the topic "Supply details". In the drop-down list, go-to "Environment" and select "Nutrient balances, mapping view".

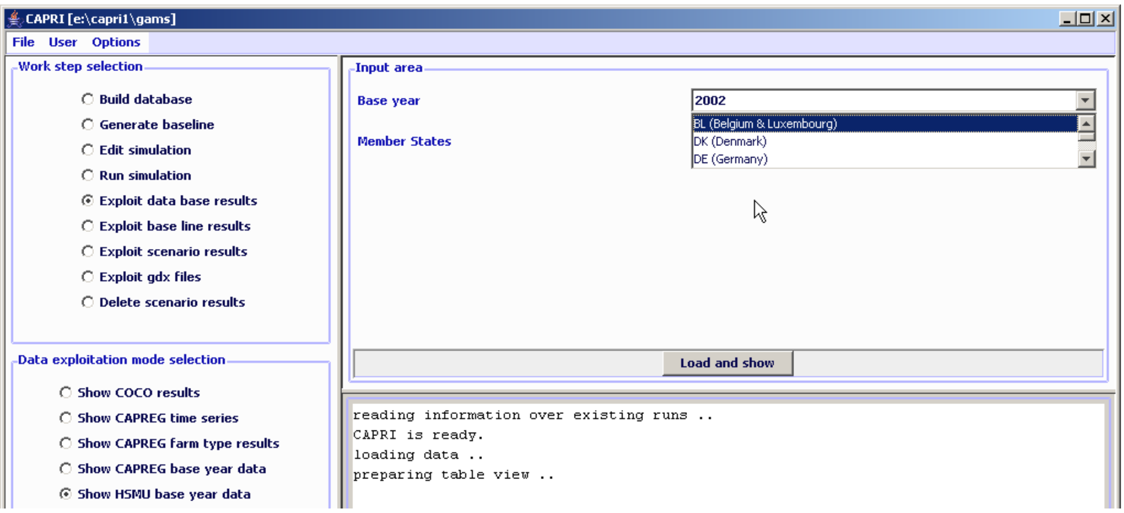


The following map should appear. You may select now different elements of the balance, by using the drop-down boxe under “activity” or change the nutrient, by using the drop-down box under “nutrient”.

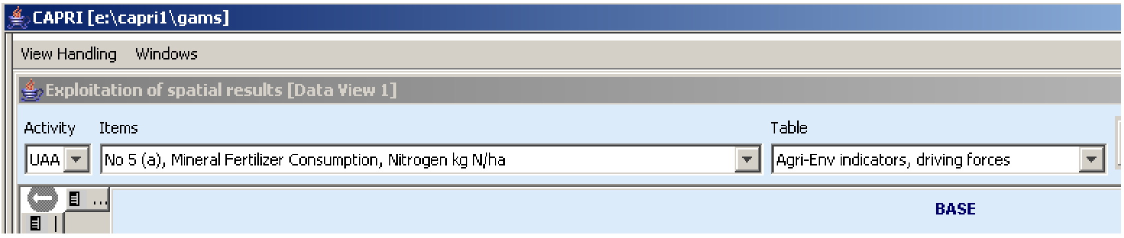


Drawing a map of the High Nature Value Farmland indicator for Belgium for the base year

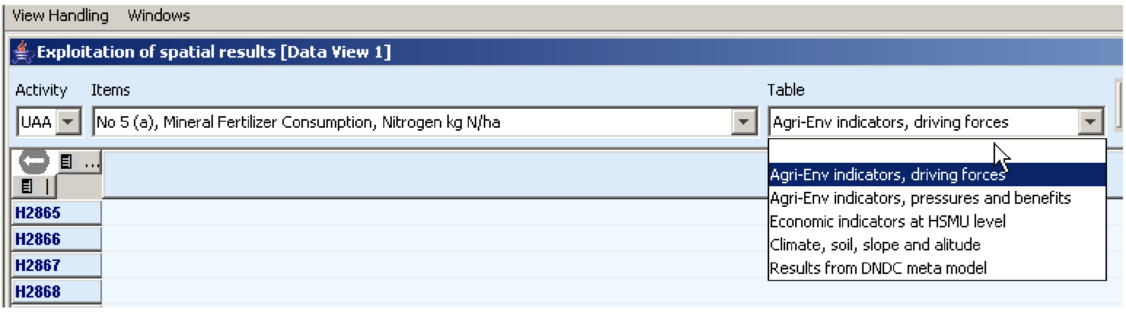
Firstly, we need to select "Exploit data base results" in the work step selection panel, and then choose the radio button "Show HSMU base year data". Then, in the Member States drop-down list, Belgium must be selected, as shown below.



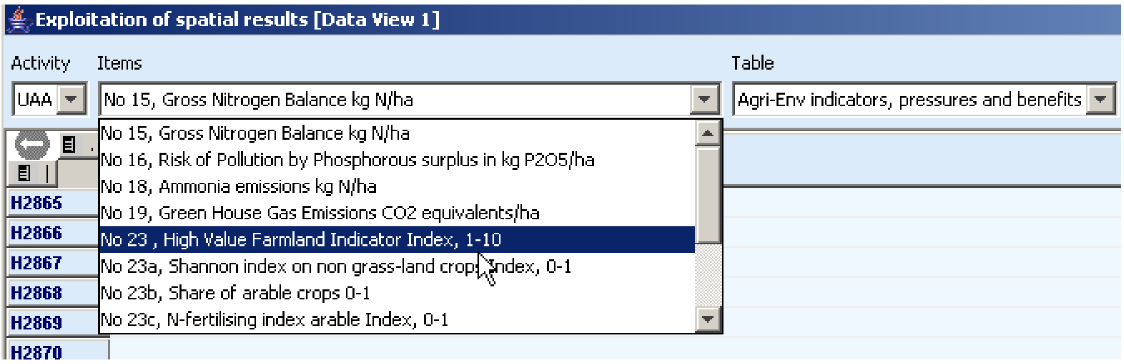
Pressing the load and show button will then bring up the first table links to the HSMU results as shown below.



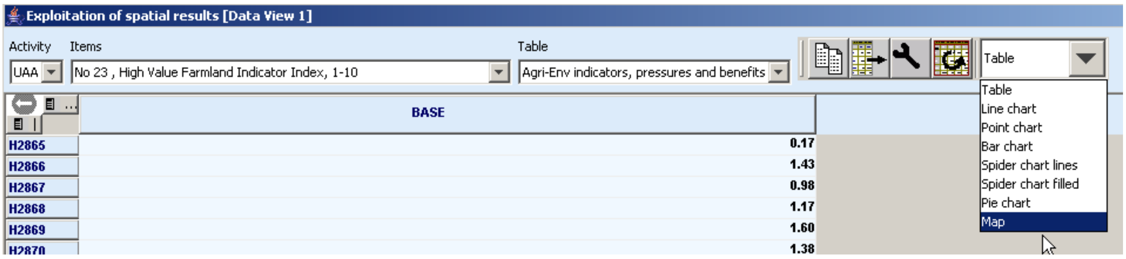
Now, according the guidelines for RD indicators, HNV is to be found under "Pressures and benefits", which is stored under another table. So opening the drop down list for the tables allows us to select the correct table.



Next, we need to find the correct item, to do so, the drop-down lists for the items must be selected, and the indicator 23 selected, as seen below.

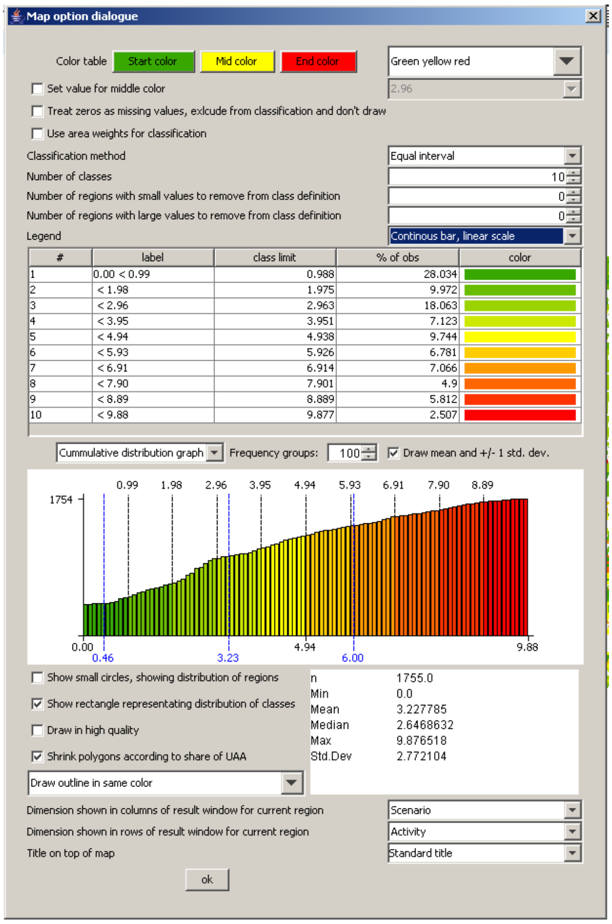


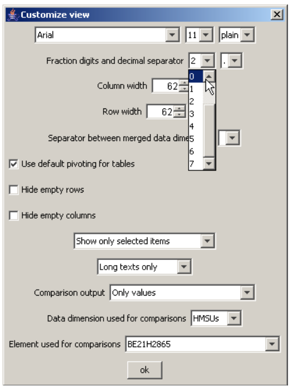
Those numbers should now be shown as a map, to do so, select “Map” from the drop-down list where “Table” is shown.



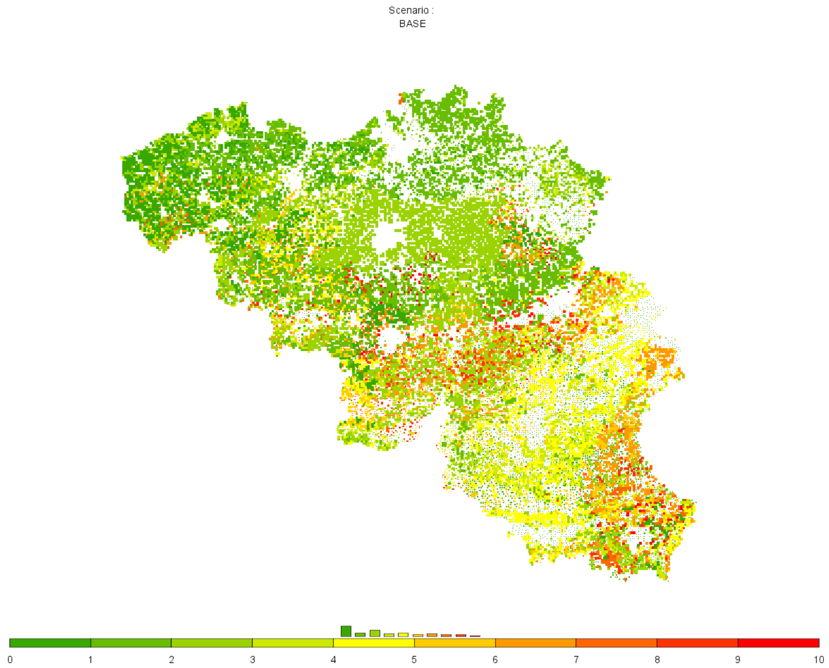
The “hour glass” cursor is shown and the geometry will be loaded which may take a few seconds. Afterwards the standard map comes up (green-yellow-red color model, quantile classification, polygons shrinked, no area weights, zeros included in classification). Now, for the HNV indicator ranging from 0-10 where 10 are the favorite numbers, and real zeros indicate missing values, the following settings could be appropriate:

- Equal interval classification with 10 classes
- Zeros treated as missing values
- And, using area weights may be appropriate so that the frequency graph below the maps shows the share of UAA in each of the ten classes
- A linear scale works nicer for our example, and if we are at it, as the data only range to 9.88, we should round the number to integer (use the “tools” button , and choose “0” in the “Fraction digits” scroll-down box.



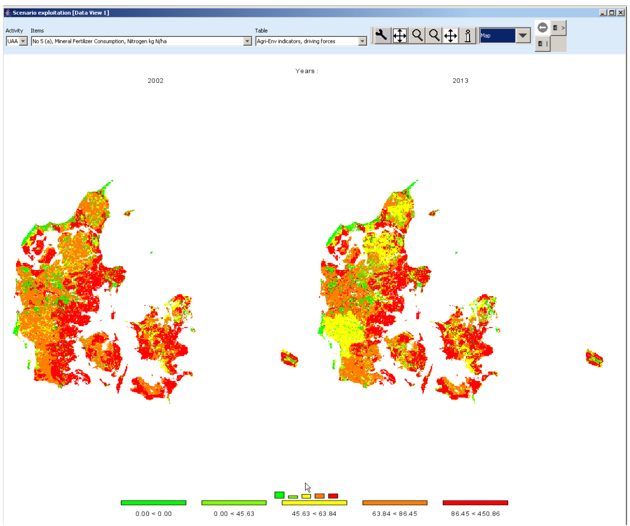


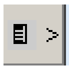
That should give the following map which then can be exported to other application via the clipboard  or can be send to the printer :

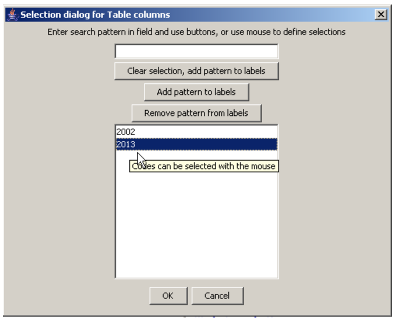


Drawing a map showing changes between the base year and the ex-ante results from the baseline

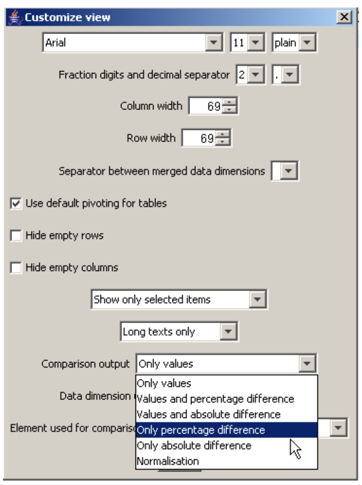
When scenarios or different points are compared with each other, it is often useful to draw maps which show relative or absolute changes. The following map is the typical starting point when the baseline is analyzed: two maps with identical class definitions, one for the base and one for projection year.

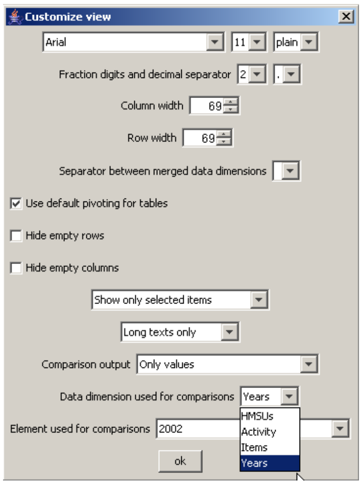


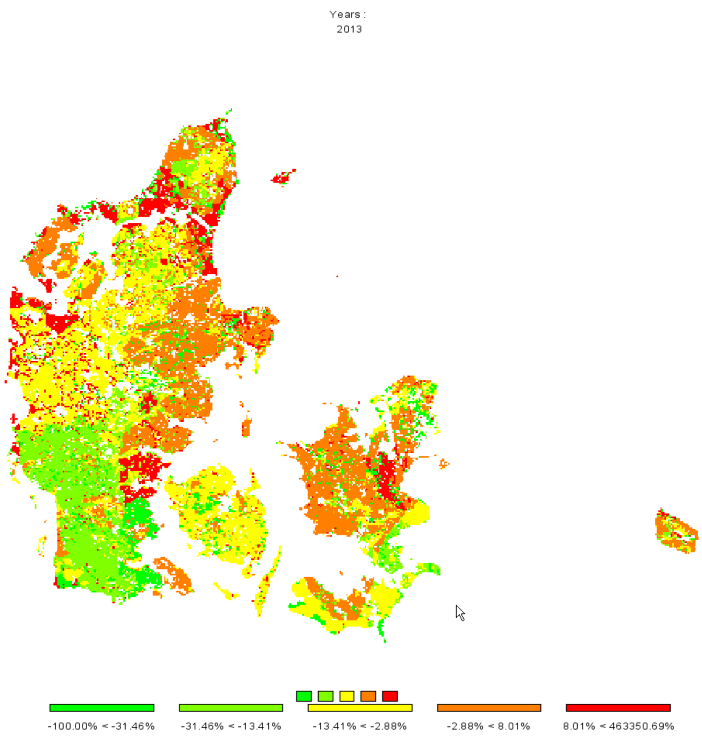
In order to draw a map with changes, we must first get rid of the base year by de-selected the first map. This is done by using the "column selection" button  which is found in upper right corner of the window. When the button is pressed, a dialog opens, and one can select with the mouse the projection year, only. Afterwards, the left map will not longer be present.



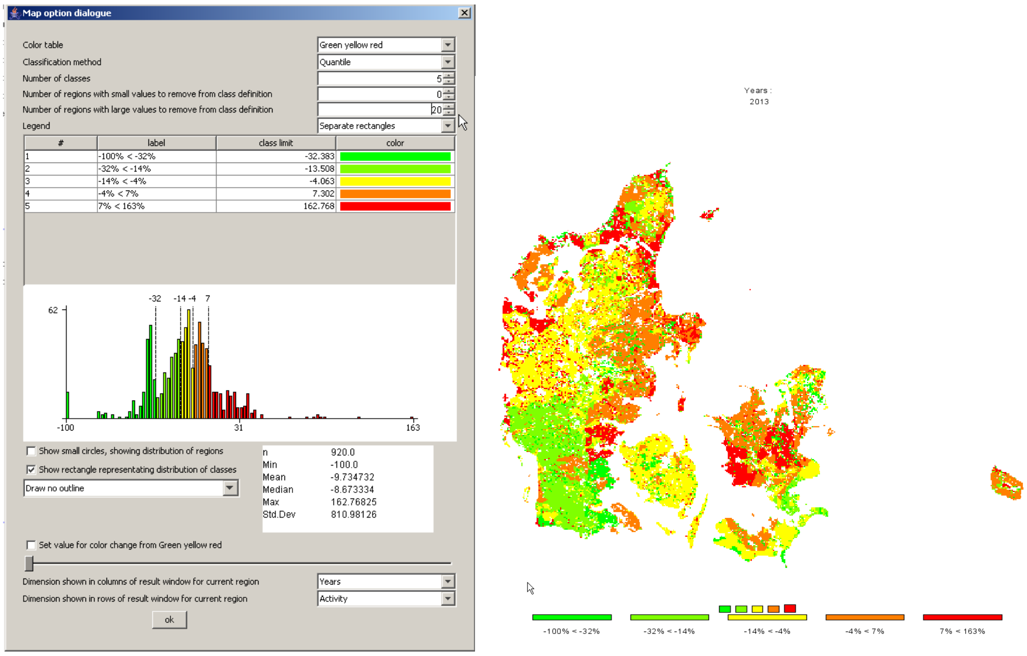
Next, we need to change what is shown in the map to relative changes to the base year. That can be accomplished by using the tool dialogue (press  button). In the tool dialogue, select "only percentage differences" in the drop-down box labeled "comparison output", and then put the "data dimension used for comparisons" to "Years". The "Element used for comparisons" should be "2002". After pressing o.k., the map will change as shown.





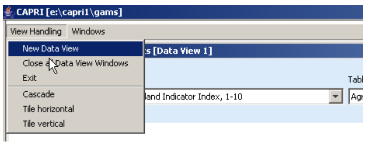


Now, there are two thinks we would most probably like to change: the number of digits shown in the legend, and getting rid of very large values shown in the legend. The number of digits can be changed again with the “tool” dialogue by changing the fraction digits. The large number can be excluded from the classification by increasing the “Number of regions with larger numbers to exclude from class definition”, in the example below the number had been set to 20.

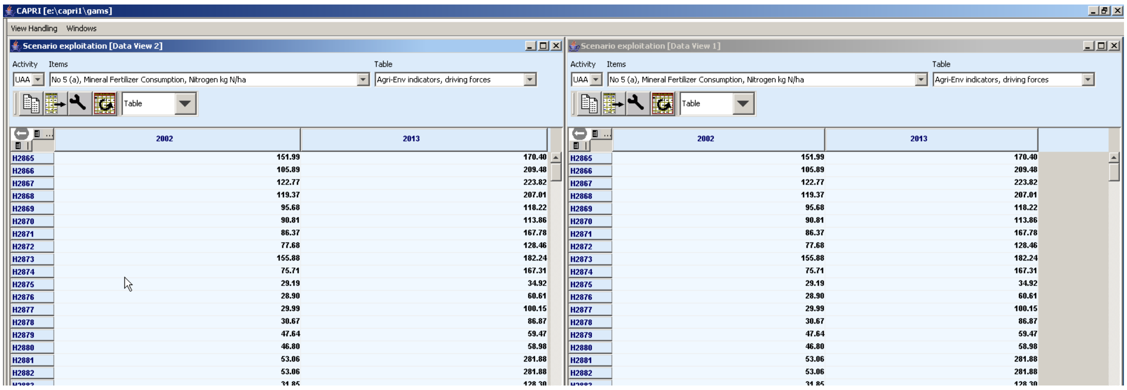


Drawing a map with the base year results next to one showing changes from the base year to the baseline results

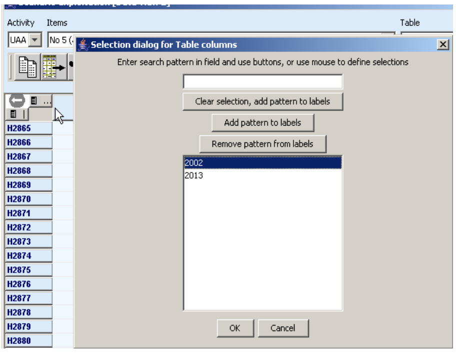
There are two ways to draw different maps. If more then one column is selected in the underlying table view, several maps with identical coloring and scaling will be drawn as shown in the sample above. That is not advisable – in our example we need to different scales, one appropriate for the absolute values and one appropriate for changes. In order to do so choose “View Handling”, “New Data View” and then “Tile vertical”.



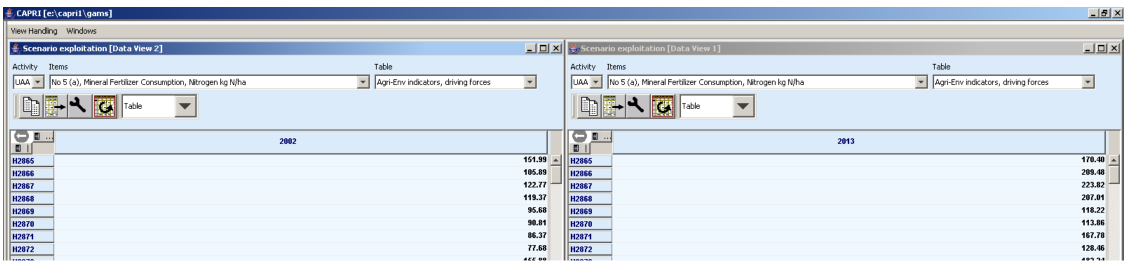
We will then see something as shown below.



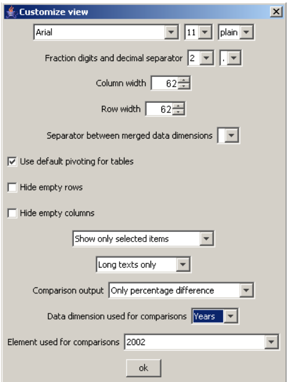
Now, in the left hand side, only the results for the base year should be shown. That can be accomplished by deselecting the column for "2013" – use the column selection button to do so.



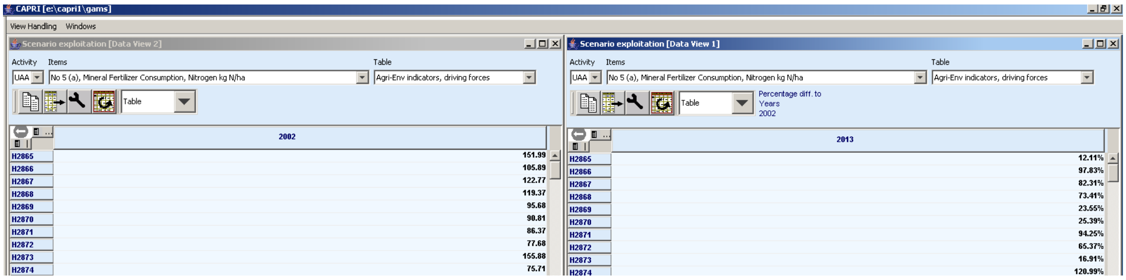
The very same trick should be applied for the other view, only deselecting the results for "2002". Now we see something as below:



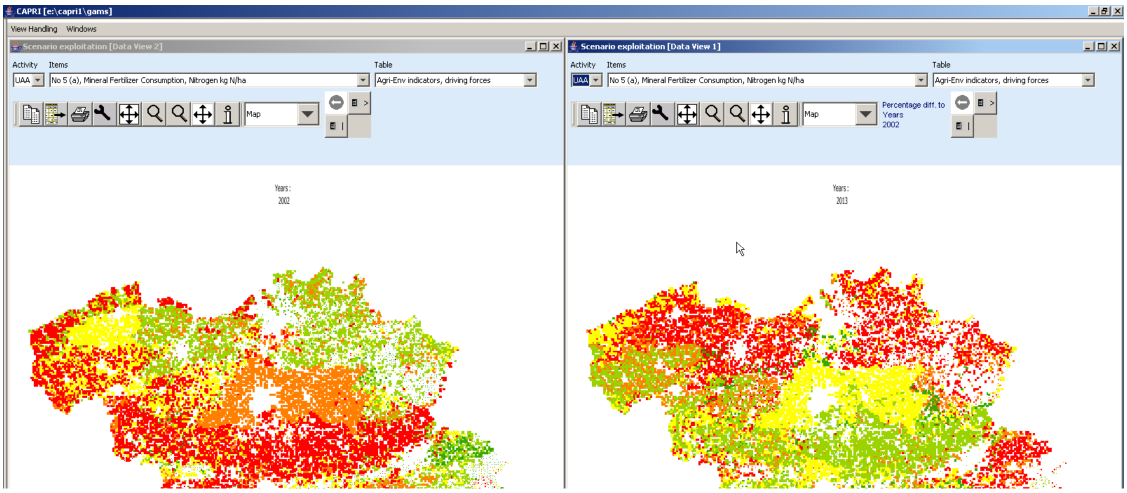
Now, for the map with the results for 2013, we should switch to percentage different to the "2002" by opening the "tool" dialog . There, under "Comparison output" choose "Only percentage difference", Select "Years" under "Data dimension used for comparisons" and ensure that "2002" is selected in "Element used for comparisons".



Now we should get a result as below.



Now, for both views, the output should be switched to maps, and there we are …



Index