User manual for VgosDBpy

Hanna Ek
hannakristinek@gmail.com

Rickard Karlsson
rickkarl@student.chalmers.se

Summer 2019

Contents

1 How to start the program

The program is started by typing one of the following lines in the terminal from the location just outside the vgosDBpy folder.

- **python3 -m vgosDBpy** ****.wrp-path*** - The wrapper is parsed and its content displayed in the terminal
- **python3 -m vgosDBpy** ****.wrp-path*** **-g** - Opens the graphical interface with content from the wrapper file.
- **python3 -m vgosDBpy** ****.txt-path*** - Starts a script driven version of the program, no interface shown. Plots and tables produced according to the scripts instructions.

2 Running the program through the interface

Writing **python3 -m vgosDBpy** ****.wrp-path*** **-g** in the terminal will start the program using the interface.

2.1 The main features

The main features that are supported by the interface is,

- By a tree structure navigate though the content of a wrapper-file, such as files and folders.
- Generation plots and tables and save these
- Editing the data shown in tables and plots. Editing includes changing values of data points or remove data points.
- Saving changes that have been made. This will generate new netCDF files, a history file and a wrapper file.
- Print out information stored in netCDF files in an information box.

2.2 How to navigate though the files and folders

Navigating though the wrappers content is done by clicking at different files and folders displayed to the left. When a netCDF-file is selected its variables will be displayed in the lower left corner. There are two different tabs in which they are displayed. The first is labeled *Variables* which displays all variables with dimension either *NumObs, NumScan* or *NumStation.* (OBS: The names for these dimensions varies for different netCDF files). These variables can then be displayed in a table and if the data type (dtype) is not a string, *'S1'*, they can also be displayed in a plot.

2.3 The buttons

- **Buttons related to plotting**
  - **Plot** - Clears the plot canvas and then plots up the selected variables.
  - **Add to plot** -Expands the plot with one selected variable.
  - ***Checkbox:*** **Display time on x-axis** - Choose wheter to have the timestamps on the x-axis or not.
  - ***Checkbox*****: Show lines** - choose wheter to show the lines connecting the data point or not.
  - ***Checkbox:*** **Show marker** - choose wheter to show the data point or not.
  - ***Checkbox:*** **Show smooth curve** - choose wheter to show a approximated smooth curved of the data points or not.
- **Buttons related to the table**
  - **Table** - Clears the table and creates a new one with the selected netCDF variable as content.
  - **Add to table** - Adds new columns with data to an existing table.
- **Clear** - Clears both the table and the plot figure.
- **Buttons related to editing**
  - **Clear all marked data** - Removes the markers on data that have been marked either in the plot or in the table.
  - **Remove all marked data** - Removes the selected data point and replaces them with *NaN*.
  - **Restore original data** - Restores the file being edited to how it was before stating to edit it.
- **Save Changes**- Saves all the editing that has been done to new netCDF files. Creates a new wrapper and history file.
- **Print Table** - Creates a ASCII file of the table.

2.4 How does the plot function work

The plotting is done by using the python library *matplotlib.* This library has some functions which is used in the interface. These are located above the plot and are used to zoom in the plot, move around in the figure and saving the plot.

This is what will be displayed in the plot-figure for different inputs:

- **One variable selected + time on axis** - The selected data will be plotted against time.

- **One variable selected + not time on axis** - The selected data will be plotted against index.

- **Two variables selected + time on axis** - The two data arrays will both be plotted against time in the same plot with different y-axis and colors.

- **Two variables selected + not time on axis**- The two selected arrays will be plotted against each other, the one first selected will be on x-axis.

- **Three variables chosen + time on x-axis**- To many input arguments, no plot created.

- **Three variables chosen + not time on x-axis** - The three arrays will be plotted, the first selected will end up on the x-axis, the second on the left y-axis and the third on the right y-axis.

- **Some variables holds two data-arrays stored, this will be handled equally as if two variables where chosen** - An example on such a variable is the *AzTheo* variable in *AzEl.nc* file.

- **More then three variables can not be displayed in one plot**

To choice of having the time on x-axis or not is done by a clicking on the check-box under the plot canvas.

Every time you select to plot a variable it will also be displayed in the table, this is not true the other way around.

2.4.1 The *add to plot* button

This button expects **one** netCDF variable to be selected, it will then add that variable to the existing plot, by appending it. What will be displayed will follow the same pattern as the list above. If the figure is already full, nothing will happen.

The main purpose of this button is to be able to plot data from different netCDF-files in the same plot.

2.5 How does the table function work

The data that can be put into a table is all the data shown in the lower left corner in the tab *variable*. It is not possible to put variables with different dimensions in the same table.

To create a table mark the name of netCDF-variables, in the lower left tab, wanted in the table, and press table. Use the *add to table* button to expand the table with more data once it been created. The tables is saved to an ASCII-file if the button *print table* is pressed.

2.6 How to edit data

Some general facts about editing

- The data can be edited by removed certain data points or change value of them.
- It is **only** possible to edit the data that can be visualized in a plot. This is netCDF-variables with dimension *NumObs*, *NumScan* or *NumStation* and any dtype except *S1*.

When a plot is generated a table with the same netCDF-variables is by default also generated. When data is marked in the plot the same data is marked in the table. The editing done in the plot is also shown in the table. It works the other way around as well,if you change the value of a data point in the table the change is seen in the plot. There are three buttons associated with editing which are listed above.

3 Running the script-driven program

Writing **python3 -m vgosDBpy **.txt path*** in the terminal will start the script driven version of the program.

The script driven version can create plots and tables. Editing data in not supported by the script driven part only through the interface.

The script driven code parses the **.txt* file given as an input argument and from the instructions in the file generates and saved tables and plots automatically without any further information form the user.

Since the code is based on parsing the **.txt*-file this file needs to have a very specific format. An example of how it should look is given in the file *example_script.txt*, displayed below.

The plots and tabels generated follows the exact same rules as the plots and tables made in the interface and same data input will create the same input.

3.1 Format for script

```
!To use the script driven function the format of the input must be very exact.
!
! Here is an example of how it can look:
!__________________________________________________________________________________
!
!
! '!' means line of comment and will not be read
! Keywords: 'save_at', 'new_wrapper', 'begin_table', 'end_table', 'begin_plot', 'end_plot'
!
!
!
save_at ./../some_folder
!
new_wrapper ../../path_to_wrp
!
```

```
!
!Everything between begin_table and end_table is shown in ONE table,
! if you want several just write many scopes like this:
!
begin_table
!
path_to_netCDF_from_wrapper -- name_of_variable_one -- name_of_variable_2 -- ... -- name_of_variable_n
path_to_netCDF_from_wrapper -- name_of_variable_one ...
!
end_table
!
!
!Everything between begin_plot and end_plot is shown in ONE plot,
! if you want several just write many scopes like this:
!
begin_plot
!
path_to_netCDF_from_wrapper -- name_of_variable_one -- ... -- name_of_variabale_n
path_to_netCDF_from_wrapper -- name_of_variable_one ...
end_plot
!
!
!It is possible change wrapper just use the keyword 'new_wrapper' again
!
new_wrapper ../../path_to_wrp
!
begin_plot
path_to_netCDF_from_wrapper -- name_of_variable_one --- ... -- name_of_variable_n
path_to_netCDF_from_wrapper -- name_of_variable_one ...
end_plot
!
!
!
!__________________________________________________________________________________
!
! Here is an example of how it can look:
!__________________________________________________________________________________
!
!
save_at ../saved
!
new_wrapper ../../Files/10JAN04XK/10JAN04XK_V005_iGSFC_kall.wrp
!
begin_table
!
TSUKUB32/Met.nc -- TempC -- RelHum
TSUKUB32/Met.nc -- AtmPres
!
end_table
!
!
!
begin_plot
!
TSUKUB32/Met.nc -- TempC
TSUKUB32/Met.nc -- AtmPres
end_plot
```

```
!
new_wrapper ../../Files/10JAN04XU/10JAN04XU_V005_iGSFC_kall.wrp
!
begin_plot
KOKEE/Met.nc -- AtmPres
end_plot
!
!
!
!
! This will generate two plot figures saved on the format *.png and
! one table saved as an ASCII *.txt file
! The three files will be saved in the folder
! ’../saved’ meaning ’../saved/*.txt’ for example

```