MODULATING A POLYCHROMATIC IMAGE BY ONE PLOTTED AGAINST SATURATION – PROGRAM hsplot

Overview

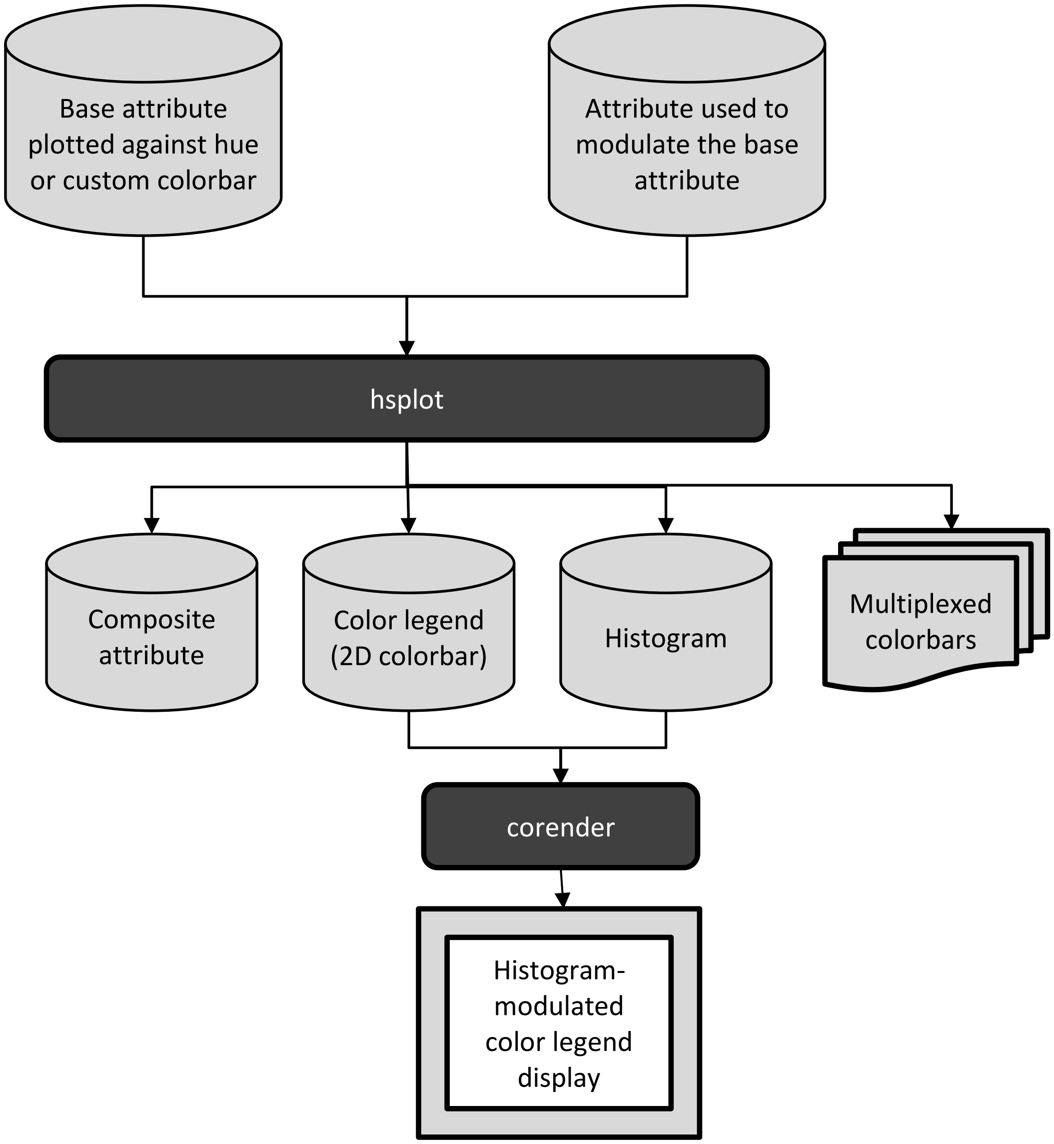
There are two main reasons to modulate one attribute by another. The first is when the two attributes form components of a 2D vector such as dip magnitude and dip azimuth, envelope and instantaneous phase, or spectral peak magnitude and spectral peak phase. The second is when we wish to modulate one attribute by another that measures the confidence in the first. For example, we may wish to map the acoustic impedance by a volume that measures the error misfit between the synthetics and the original data.

The AASPI software allows two ways to do this. Program **hlplot** modulates an attribute plotted against a polychromatic color bar by a second against lightness. Program **hsplot** modulates an attribute plotted against a polychromatic color bar by a second against saturation. Program **hlsplot** allows two levels of modulation. Program **corender** provides an interactive means to modulate one attribute by one or two others.

Computation Flow Chart

Program **hsplot** reads in two attribute volumes and outputs a composite volume, a color legend, a histogram, and a suite of multiplexed colorbars that can be used to load the composite volume into the more common interpretation workstations software products.

Display Tools: Program **hsplot**



Output file naming convention

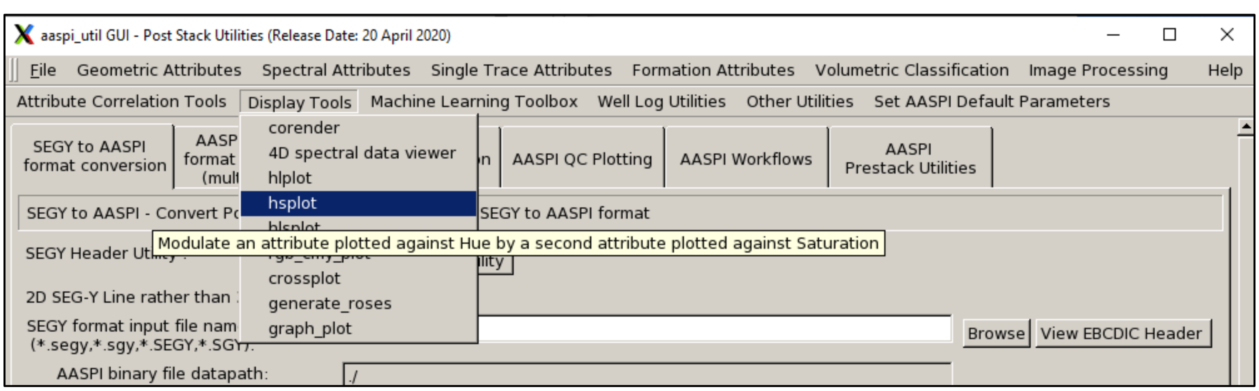
Program **nonparallelism** will always generate the following output files:

Output file description	File name syntax
Composite attribute	*saturation_axis_title*_vs_*hue_axis_title_unique_project_name*.H
Color legend (2D colorbar)	hsplot_color_legend_*saturation_axis_title*_vs_*hue_axis_title_unique_project_name*.H
2D histogram	hsplot_histogram_*saturation_axis_title*_vs_*hue_axis_title_unique_project_name*.H
Multiplexed 1D colorbars	hs_colorbar.alut, hs_colorbar.CLM, colorbar.geomodeling,…
Program log information	hsplot_*unique_project_name*.log
Program error/completion information	hsplot_*unique_project_name*.err

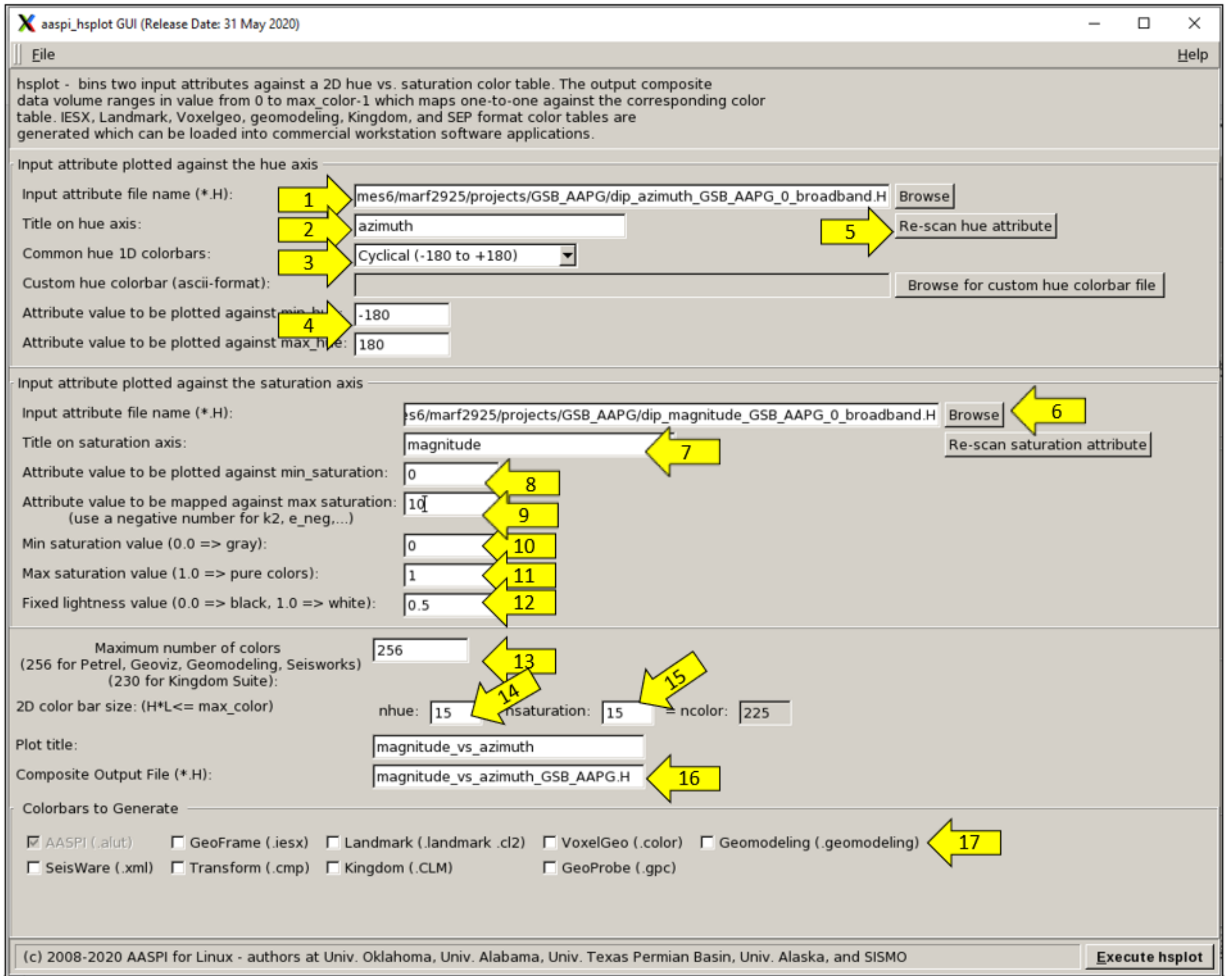
where the values in red are defined by the program GUI. The errors we anticipated will be written to the **.err* file and be displayed in a pop-up window upon program termination. These errors, much of the input information, a description of intermediate variables, and any software trace-back errors will be contained in the **.log* file.

Invoking the hsplot GUI

To invoke program **hsplot**, on the **aspi_util** GUI select *Display Tools* and then select **hsplot** on the drop-down menu:

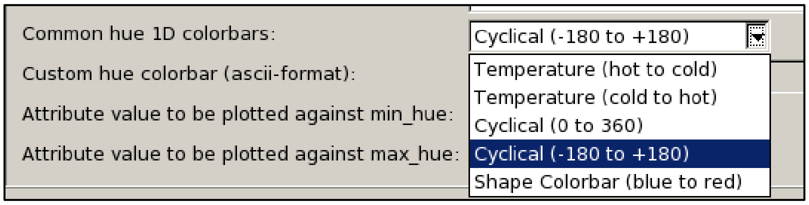


Program **hsplot** plots one attribute against hue (H) and a second attribute against saturation (S). You should have a GUI that looks like this:

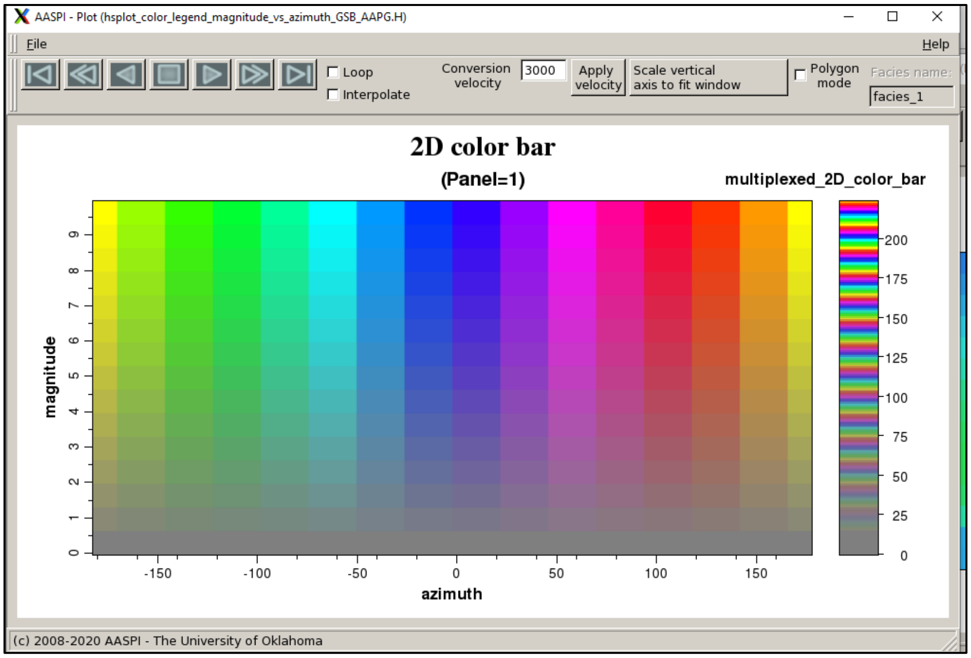


In this example I browse and select (1) the file *dip_azimuth_GSB_AAPG_0_broadband.H* as the file to be plotted against Hue. The current GUI is somewhat more clever than previous releases and will trigger off the substring "dip_azimuth" in the file name such that (2) the string *azimuth* appears for the *Title on Hue Axis*. (3) The string *cyclical (-180 +180)* color bar will appear as the appropriate color bar, and (4) the *Attribute value to be plotted against min_hue* (in this case, -180.0 degrees) and the *Attribute value to be plotted against max_hue* (+180.0 degrees) are read from the values of *min_amplitude* and *max_amplitude* from the input file entered after arrow 1. All of these default values can be altered, though for azimuth attributes these values are almost always the optimum values. If you mistype the range of minimum and maximum values, or if for some reason they are not in your *.H input file, you can always (5) *Rescan Hue Attr* to recompute the largest possible range in the data.

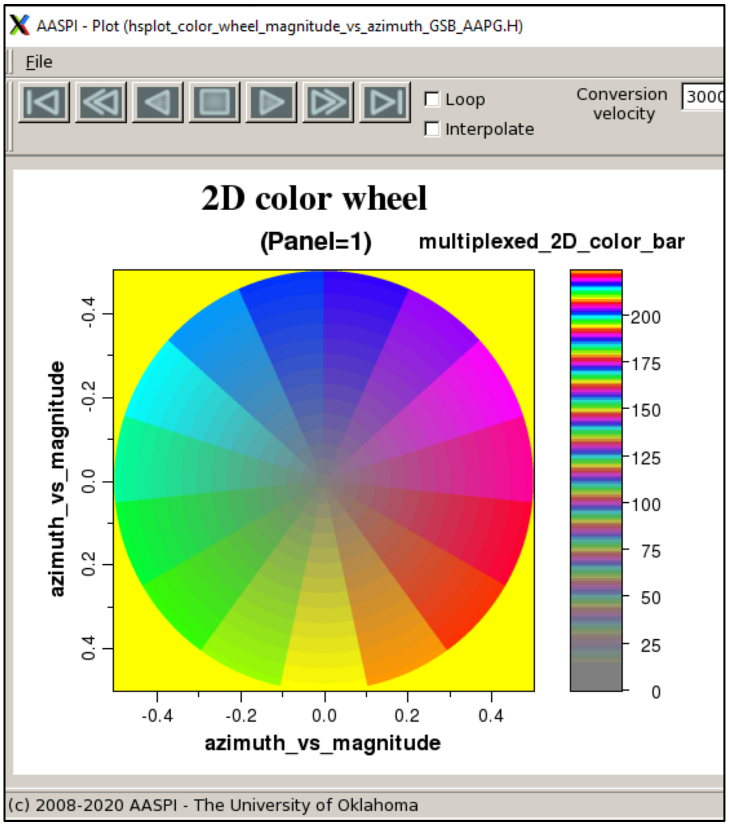
Other color bars useful in plotting AASPI attributes are found on the drop-down menu:



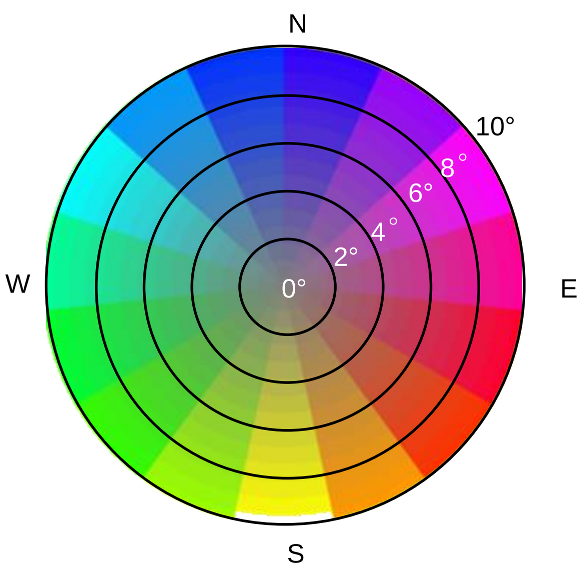
The saturation section is similar. The current GUI is again, somewhat more intelligent than our first releases, such that when (6) I click the *Browse* button, only the files that most commonly couple with those after arrow 1 will appear. Since I have previously entered a file name beginning with “dip_azimuth”, the GUI will present a list of those beginning with “dip_magnitude”. In this case I choose *dip_magnitude_GSB_AAPG_0.H* as the file to be plotted against Saturation. The (7) *Title on Saturation Axis* defaults to be “magnitude”, the (8) *Attribute value to be plotted against min_saturation* defaults at “0” and the (9) *Attribute value to be plotted against max_saturation* defaults at “90” where the latter two values were stored as “min_amplitude” and “max_amplitude” in the input *.H file. However, our previous dip magnitude images showed a maximum dip magnitude of about 10 degrees, so (9) in the *Attribute value to be plotted against max_saturation* box, I typed in 10. The *Minimum* and *Maximum saturation values* (10) and (11) respectively, allow you to define the saturation values to plot the attribute against, with a maximum saturation of 1.0, representing full saturation (or pure colors), and a minimum saturation of 0.0, representing gray. Leave the (12) Maximum number of colors to be 256 since most workstation software will not allow you to import files with more than 256 colors. If this is the case, (13) keep the number of Hue and Saturation bins at 17 and 15. If desired, (16) modify the name for your output file *magnitude_vs_azimuth_GSB_AAPG.H*. Finally, if you intend to convert the output to SEGY format and load them into commercial workstation software, place a checkmark in front of the software that you use. Finally, click *Execute*. The following images will appear when the job completes:



The first image is a 2D color bar. On the right of the 2D color bar is the 1D color bar that you will load into your interpretation workstation. Note that it has been multiplexed (or wrapped) horizontally with color zero at the bottom left and color 224 in the upper right. Gray corresponds to low dip magnitude, and the fully saturation colors correspond to high dip magnitude, with dirty colors in between. Note that an azimuth of 0^0 (North) appears as blue, while azimuths of both -180^0 and $+180^0$ (South) appear as yellow. The GUI recognizes this to be a cyclic 2D color bar or color wheel and also provides the following image:

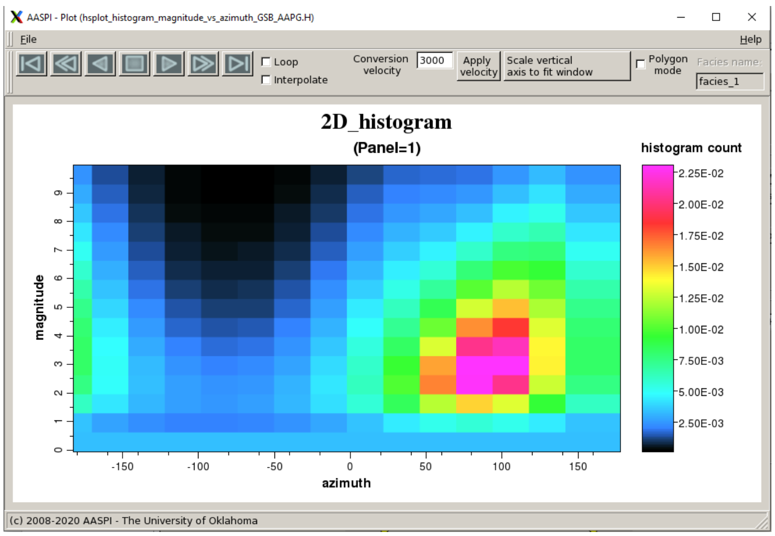


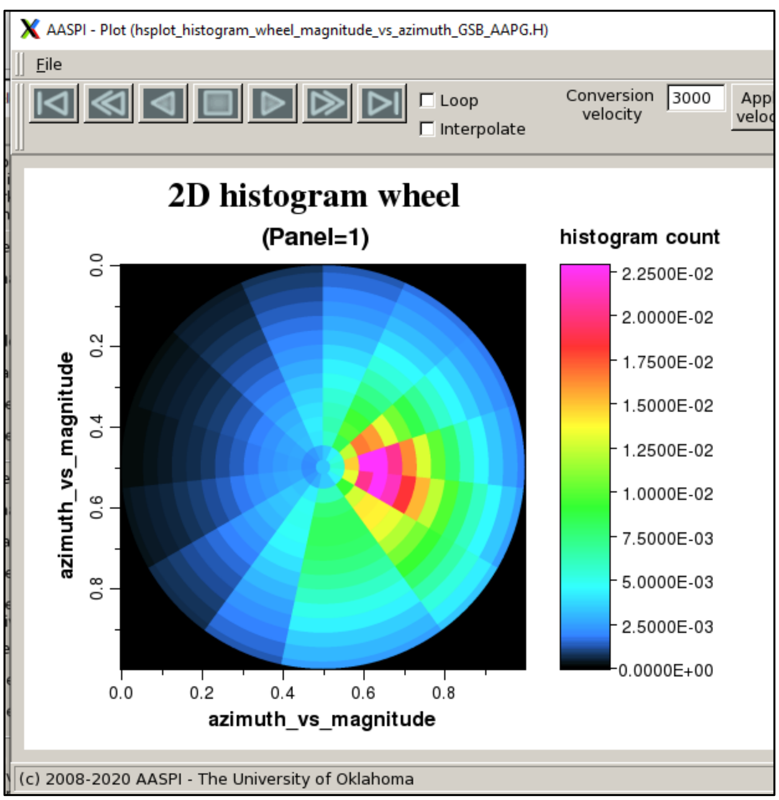
The color wheel is plotted using the **aaspi_plot** utility which is designed to display seismic amplitude and attribute data. I've annotated the image in PowerPoint to give a more explicit definition of the color wheel:



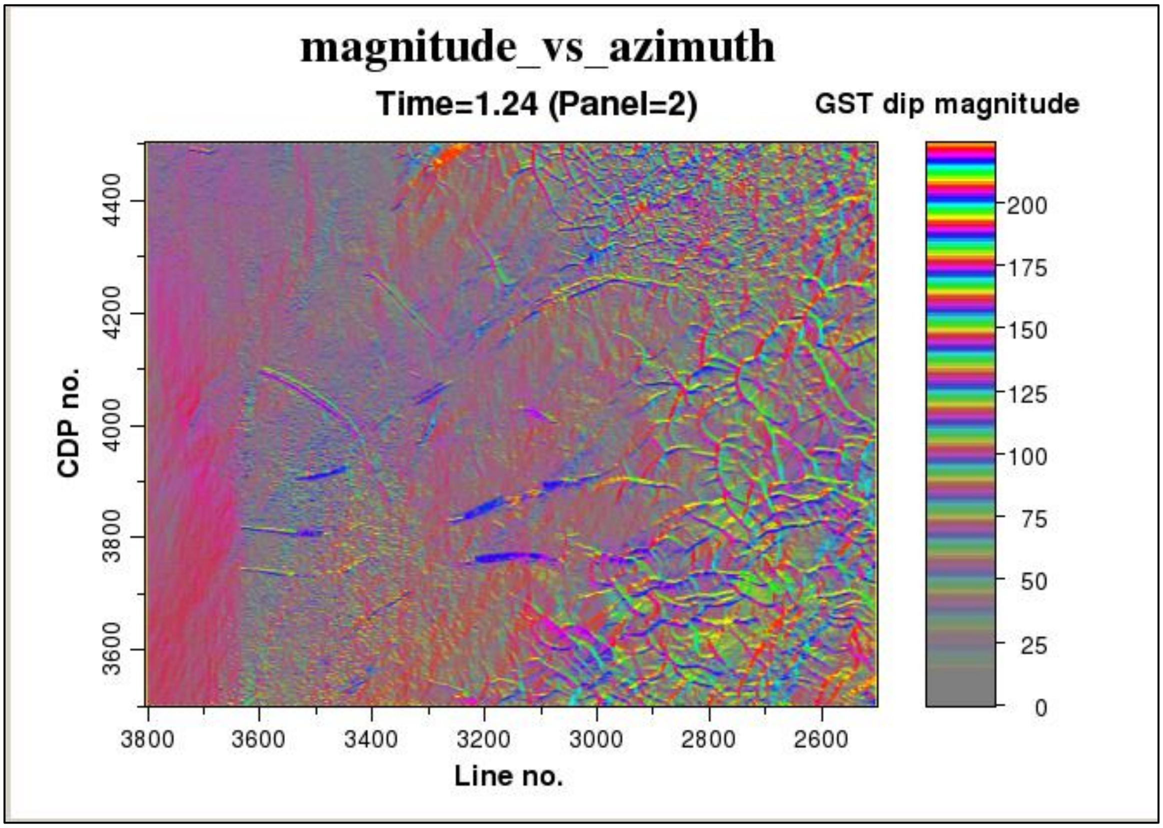
Flat (horizontal) events with a dip magnitude of 0^0 are plotted against white. Events with a dip magnitude of 50 and dip azimuth of 0^0 (North) appear as blue, 60^0 as magenta, 120^0 as red, 180^0 (South) as yellow, 240^0 as green, 300^0 as cyan, and 360^0 as blue. Less strongly dipping events appear as pastel colors.

We also have a histogram of how many voxels in the volume are plotted against each color. In this example we can see the data have a regional trend towards the east.



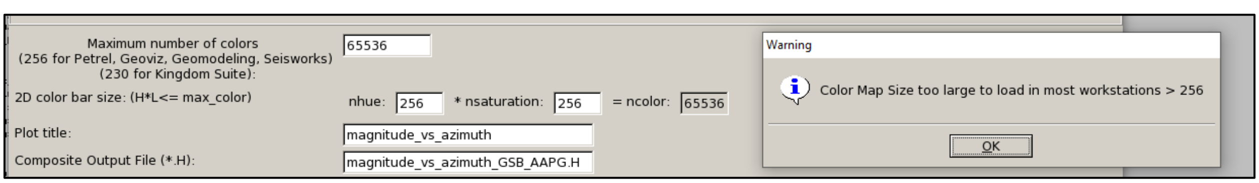


The final image will be an animation of time slices of the output data mapped to the 2D color bar. Strong dips are displayed as pure colors and weak dips as gray. In this example we see the strong dip adjacent to the fault blocks and the regional dip to east (in a magenta-red color).

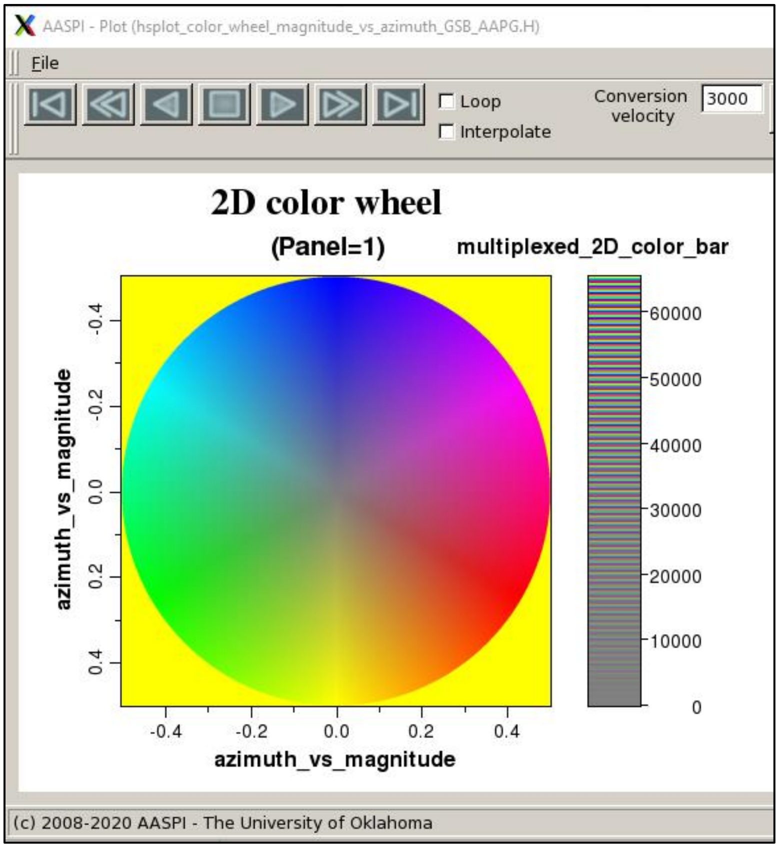


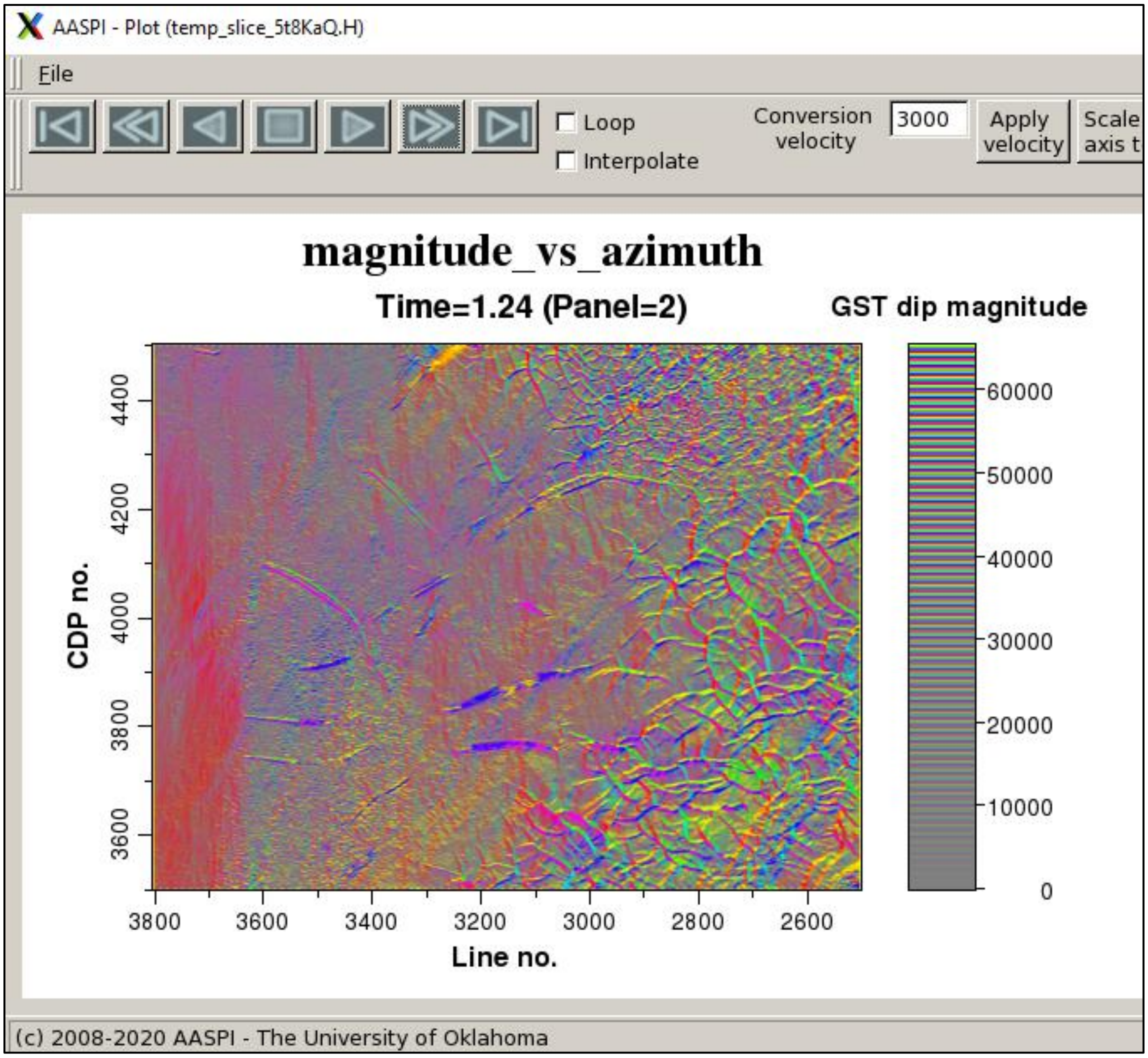
Generating images with more colors than supported by your commercial workstation software

If you want to plot an image with greater color depth, you can. The only restriction is that you probably will not be able to import the resulting color table into your commercial workstation software. In this example, I set the limits to be those of my Dell display device which supports 32-bit color (256 values for R, G, B, and alpha):



A warning appears. I check OK and obtain the following (more continuous) colorbar and image with greater color depth:


AASPI - Plot (hsplot_color_wheel_magnitude_vs_azimuth_GSB_AAPG.H)
File
Loop
Interpolate
Conversion velocity
3000
2D color wheel
(Panel=1)
multiplexed_2D_color_bar
azimuth_vs_magnitude
-0.4
-0.2
0.0
0.2
0.4
azimuth_vs_magnitude
60000
50000
40000
30000
20000
10000
0
(c) 2008-2020 AASPI - The University of Oklahoma



Importing and modulating a custom colorbar with saturation

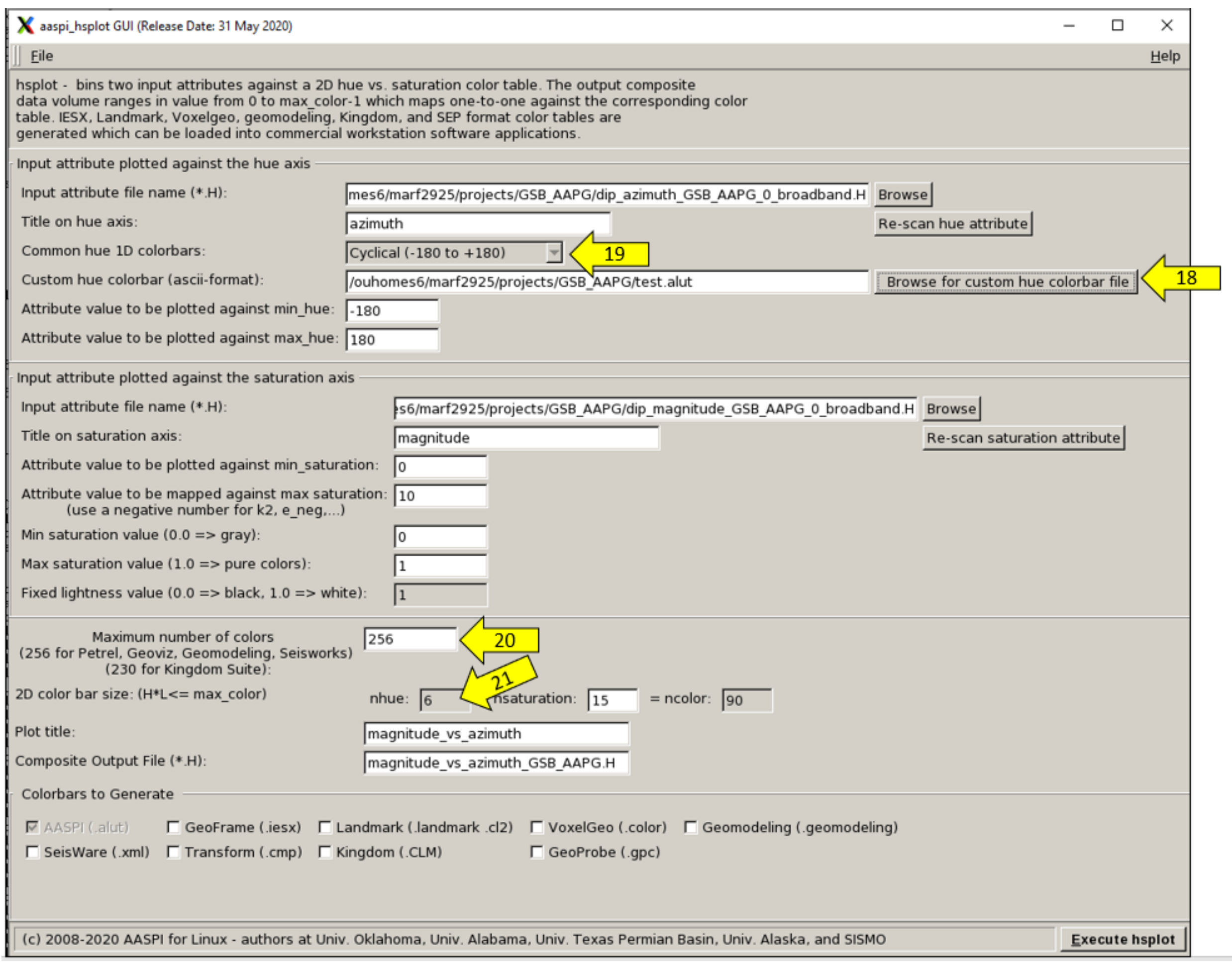
Although the dropdown menu described above for the more common continuous, fully saturated color bars constructed along the hue axis works well for the AASPI attributes, different companies and different software packages may have their own colorbars. For example, it is common in impedance inversion to use color bars that do not follow a smooth curve in RGB or HLS space, but rather "jump around" to enhance discrete changes. In this case, you need to export the corresponding RGB color bar. At present, our format looks like this (which is equivalent to the color *.CLM format used by Kingdom Suite):

255	0	0
255	255	0
0	255	0
0	255	255
0	0	255
255	0	255

Or alternatively with an unused alpha value as 4th column (the *.alut format used by Petrel):

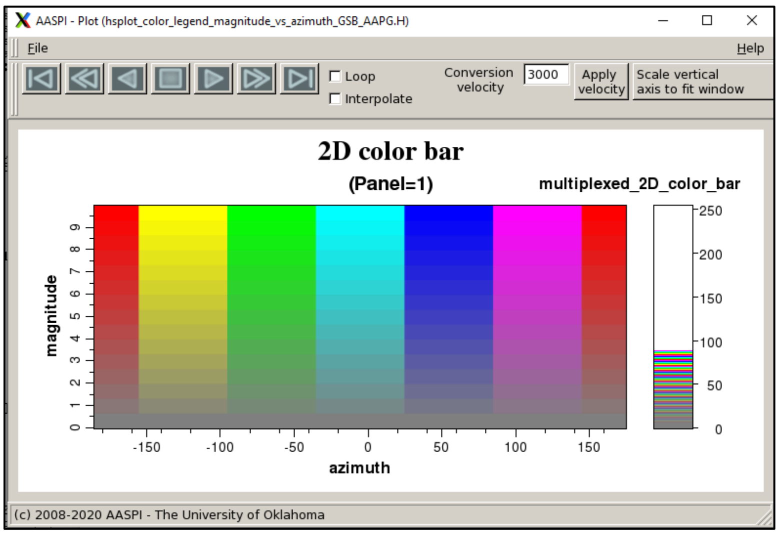
255,	0,	0,	255
255,	255,	0,	255
0,	255,	0,	255
0,	255,	255,	255
0,	0,	255,	255
255,	0,	255,	255

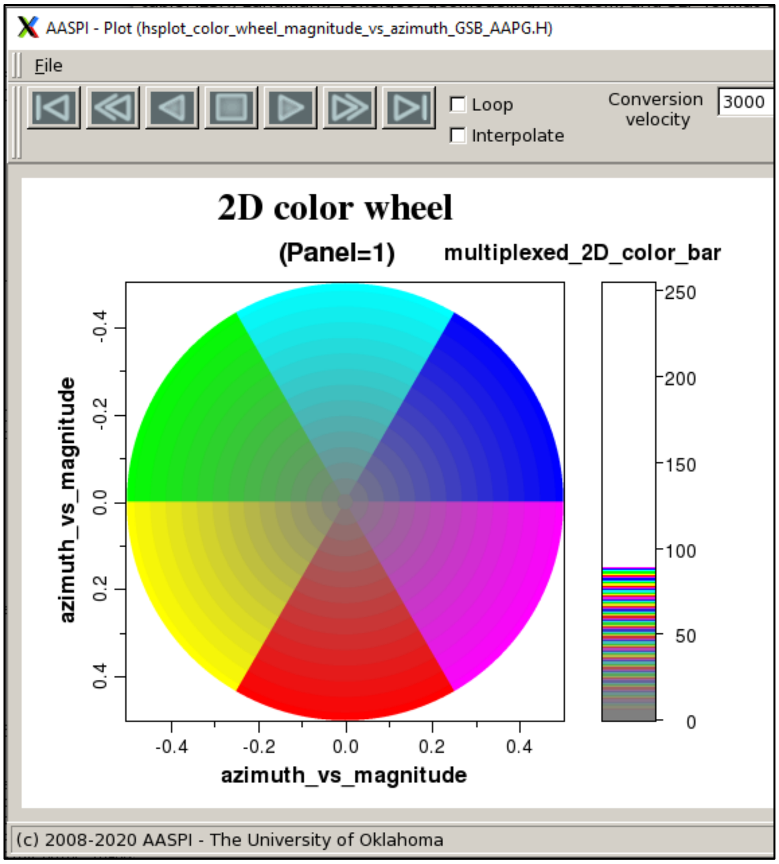
In both files, *test.CLM* and *test.alut,* there are 6 rows containing RGB triplets whose values range between 0 and 255. The values can be separated by spaces, commas, or tabs and do not need to be in strictly aligned columns. If you construct these files by hand, be aware that there can be *no blank lines* at the end of the file, which would add additional undefined RGB triplets! You may recognize these colors as being fully saturated values of red, yellow, green, cyan, blue, and magenta. Be sure the add the first line indicating the number of custom colors. Reexamining the GUI



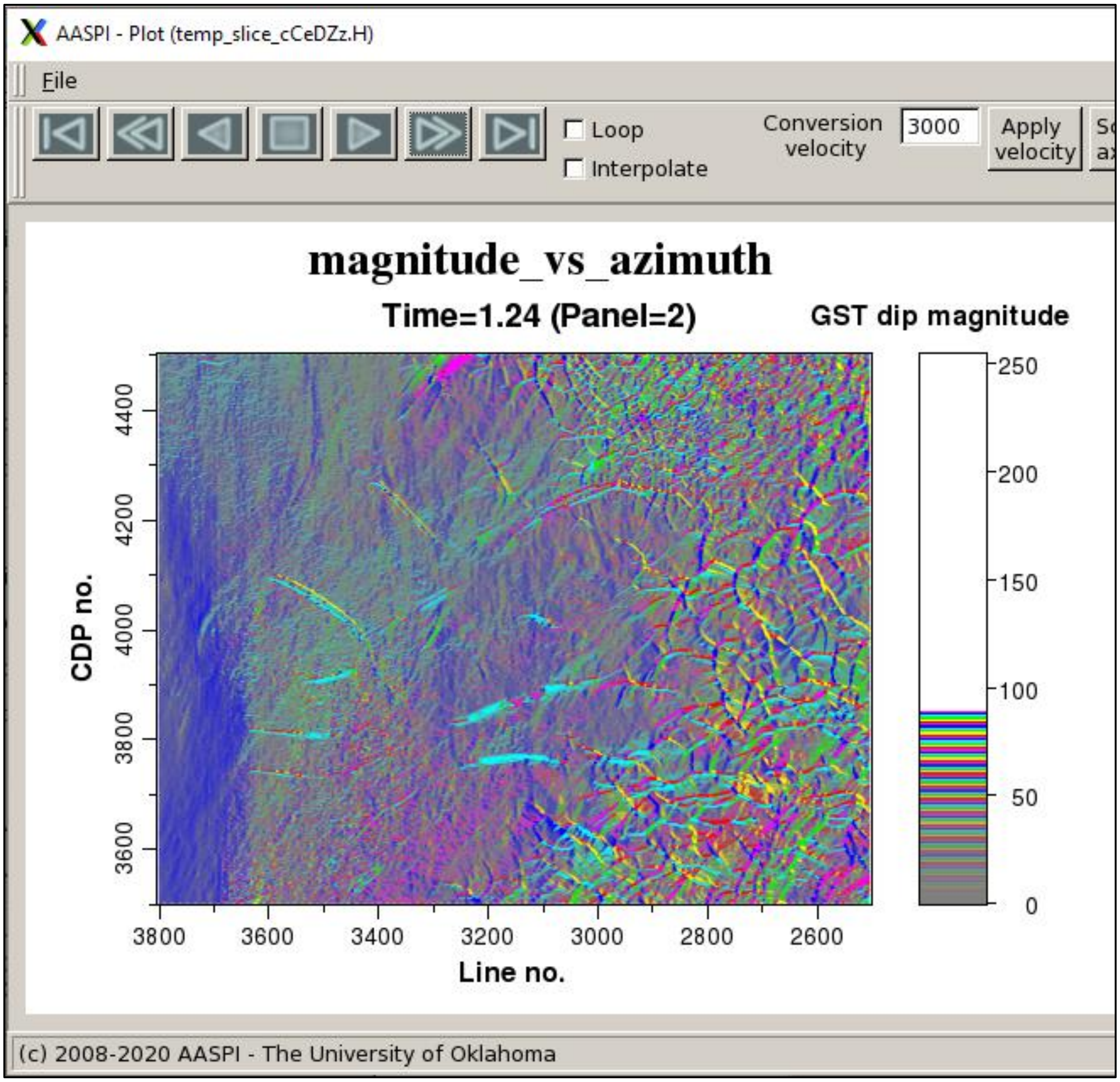
click (18) *Browse for custom hue colorbar file* and select from your suite of *.CLM or *.alut format files the one you wish to use (in this case, test.rgb). The *.alut and *CLM endings are optional, but helps minimize searching for the proper file from a potentially long list in your directory. After selection, three options are disabled – the drop down colorbar choices, the level of lightness (defined implicitly in your *.rgb file from the range of values) and the number of hues (defined explicitly in your *.rgb file). The value of lightness is changed to 1 and will multiply (preserve) the implicit values of lightness in the *.alut or *.CLM file.

The resulting colorbar and colorwheel look like this

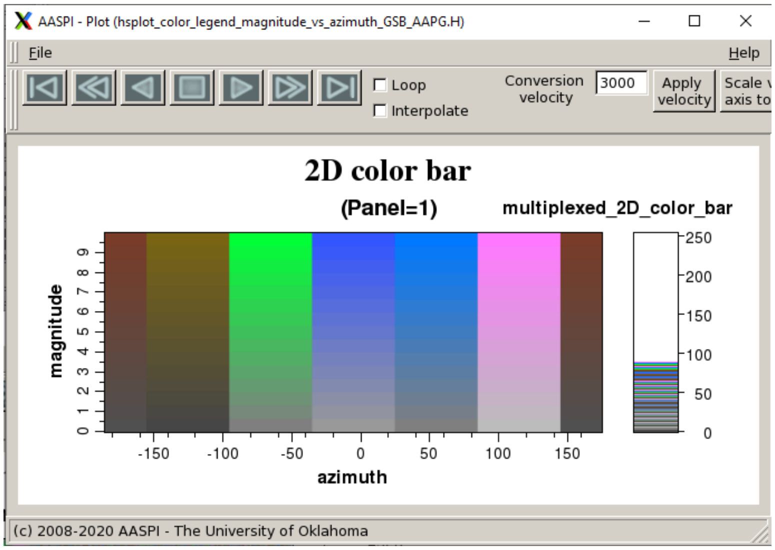


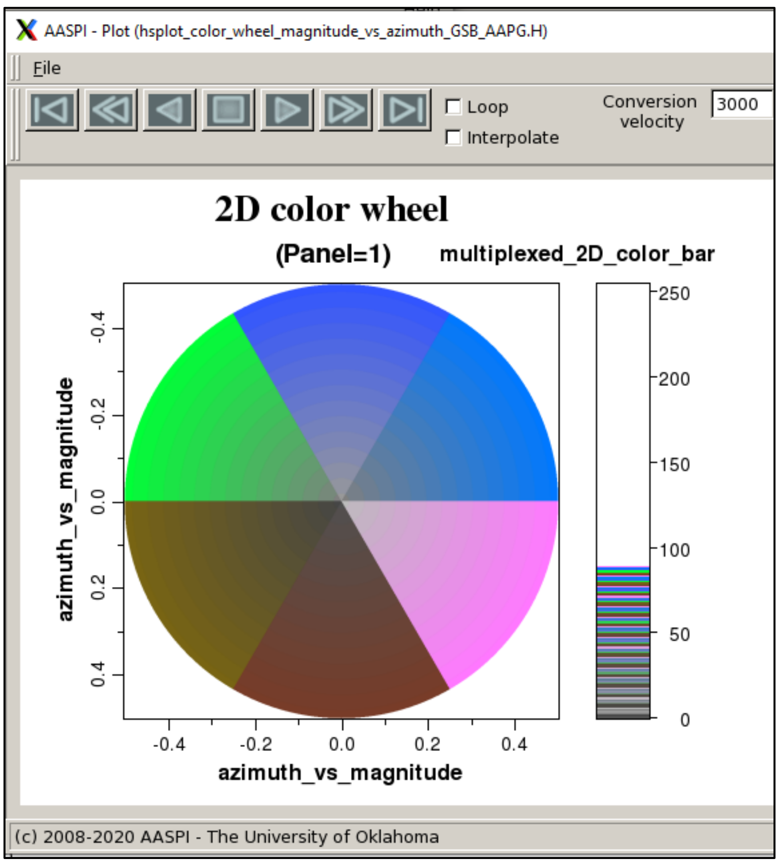


where the first color (red) is mapped to the minimum value of the attribute plotted against hue in the GUI. Unused colors are set to white. The resulting time slice in the composite data volume looks like this:



This first 6-color example showed fully saturated colors. Any RGB color triplets will work. Partially saturated colors will simply be multiplied by the saturation value of the second attribute. Here is another 6-color example that proves the point (but it is NOT one I would use to plot these data):

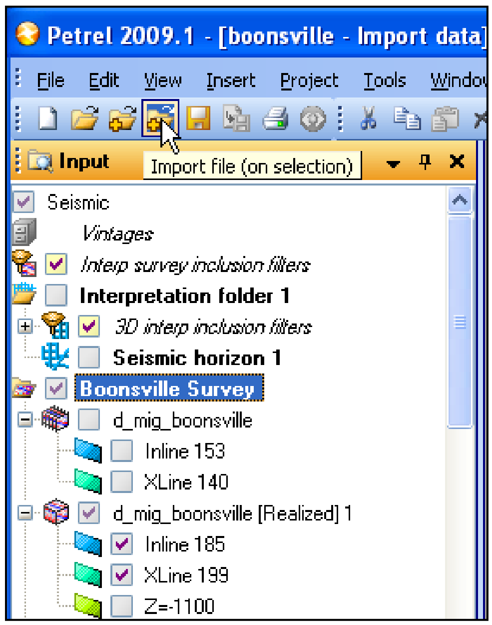




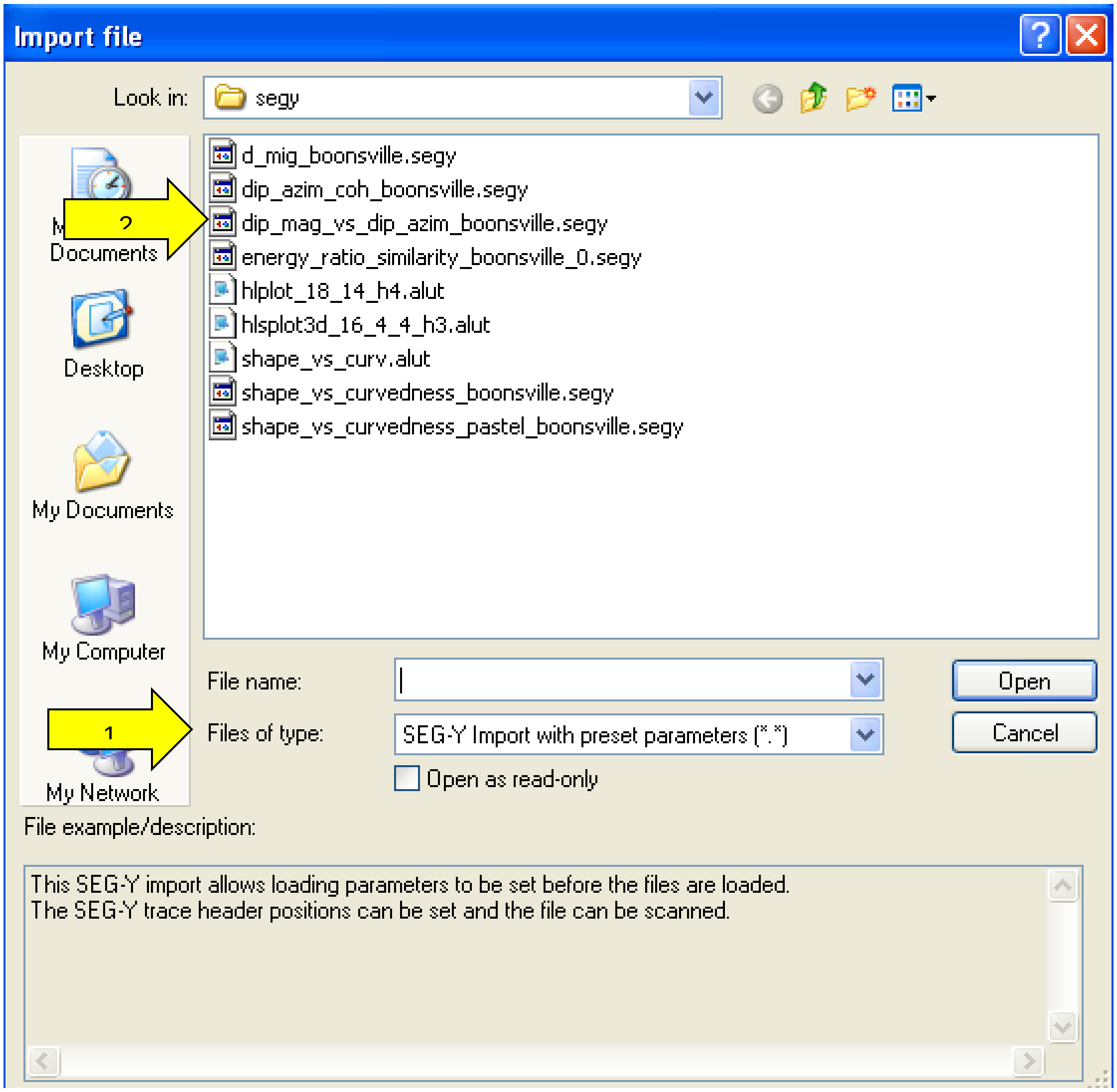
Loading and displaying multiattribute plots in Petrel

Schlumberger has been gracious enough to provide OU with licenses for education and research, hence the example on how to load a composite attribute volume into Petrel. Loading such volumes in Voxelgeo, Kingdom Suite, Landmark, Geoprobe, IESX, and Geomodeling are similar. Unfortunately, a few packages (e.g. Geographix) do not allow loading an ascii color bar from a file.

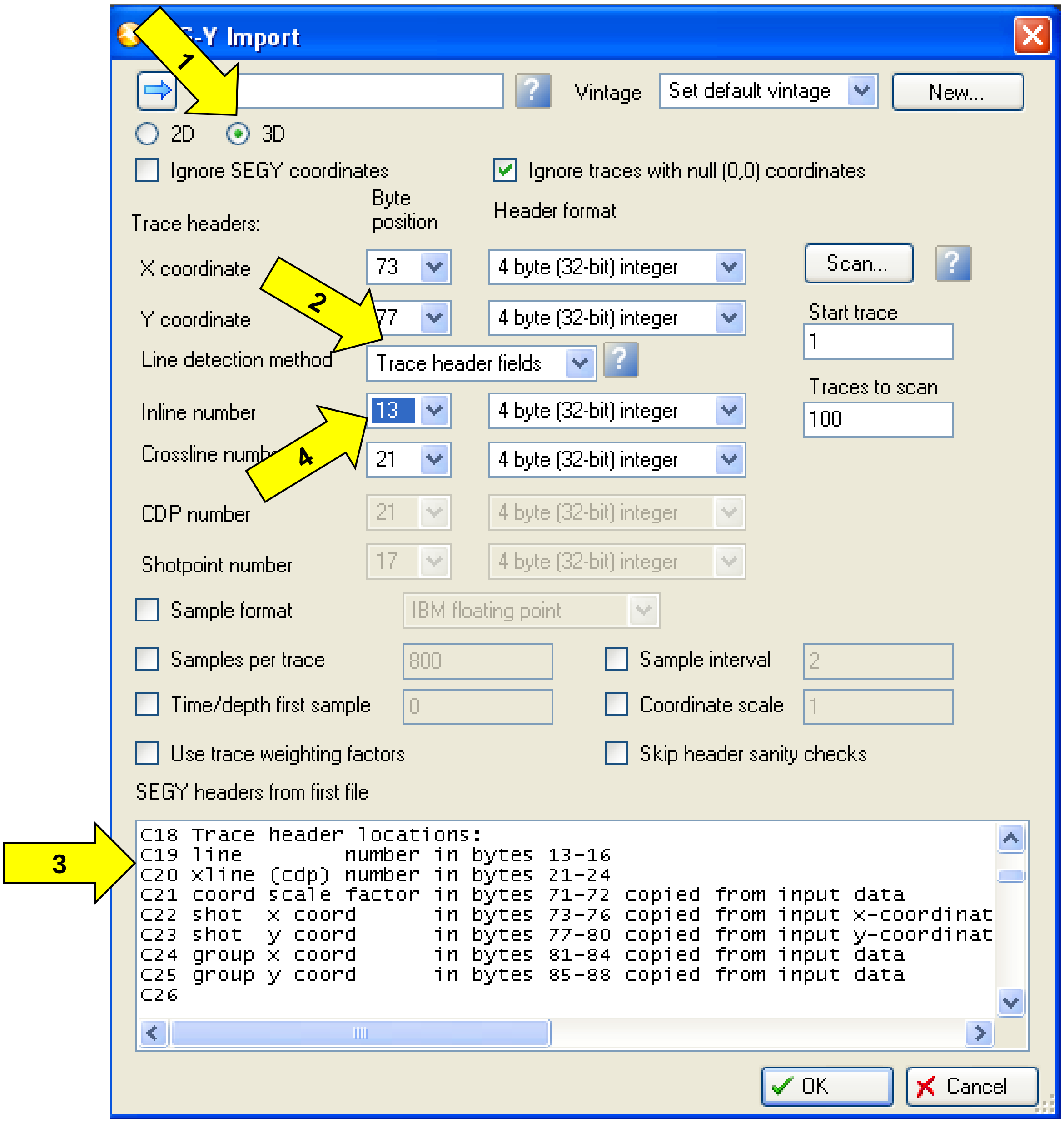
The first step is to convert the AASPI-format file dip_mag_vs_dip_azim.H to a SEGY-format file dip_mag_vs_dip_azim.segy using the *AASPI to SEGY format conversion (single file)* tab in **aaspi_util**. Then, depending on how your disk drives are mounted and shared, you will need to copy this data from your Linux system to your PC where you run Petrel. Once the data have been converted and copied, launch Petrel, open your project, and add the new SEGY file into the project by highlighting the survey name and clicking the *Import File (on selection)* icon:



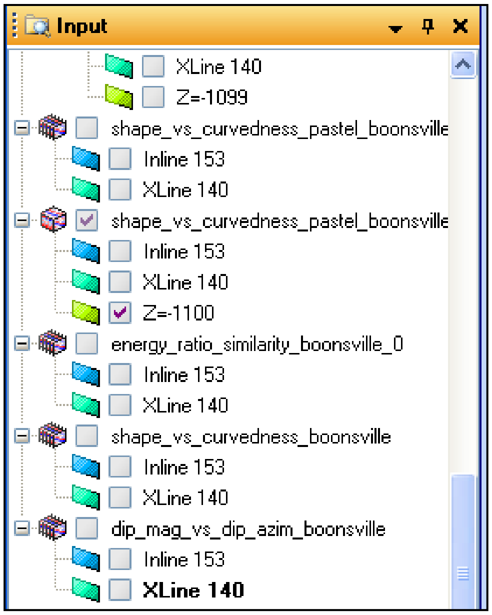
The following window should pop up:



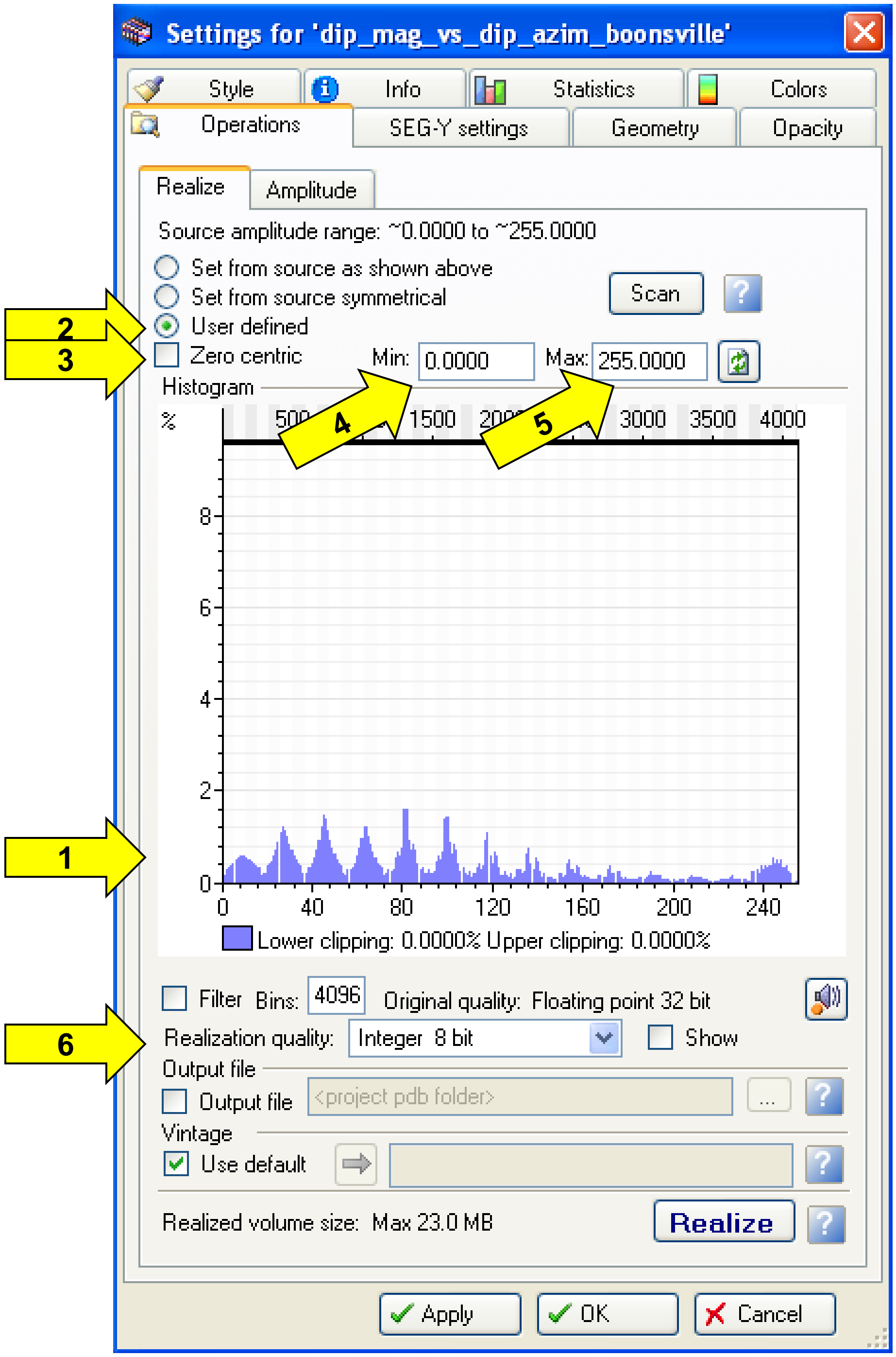
First, choose to load (1) *SEG-Y import with present parameters (*.*)*, since the header values for the AASPI volumes may be different from your default settings. (The Petrel loading utility is fairly clever, but let's just do this for clarity). Proceed to (2) select *dip_mag_vs_dip_azim_boonsville.segy* as the file to be loaded.
Click Open and the following panel is displayed (see next page):



First, select (1) *3D* since you will wish to load a 3D seismic volume. Then (2) select *Trace header fields* as the *Line detection method*. Next, scroll down under *SEGY headers from first file* and note that the (3) *Inline number* is in bytes 13-16 and the *Crossline number* in bytes 21-24. The value of 21 is the 1977 SEG-Y standard for CDP number and is usually set as the default crossline number for interpretation systems. However, most interpretation systems were started before the 2003 SEG-Y standard for 3D seismic data was adopted, such that finding the inline number takes some detective work. For all the AASPI volumes, the inline number will be in bytes 13-16. Enter (4) the value of 13 for the *Inline number* byte location. Click OK and the data begins to load. The file will then be under your *Input* tab:

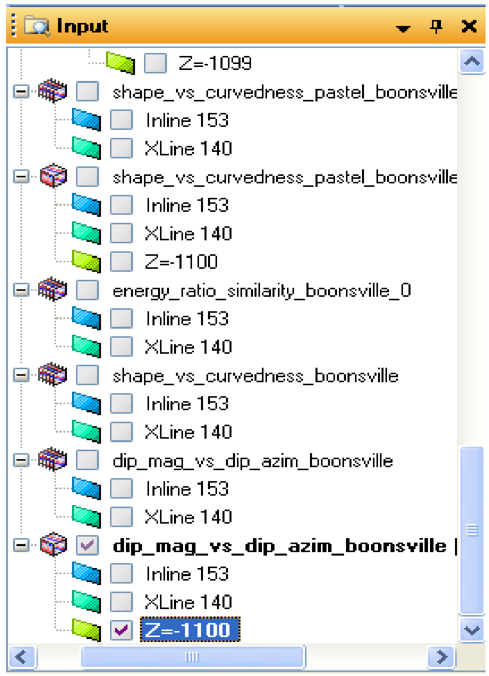


Double click the last file in the list *dip_mag_vs_dip_azim_boonsville* to open the *Settings* pop-up window:

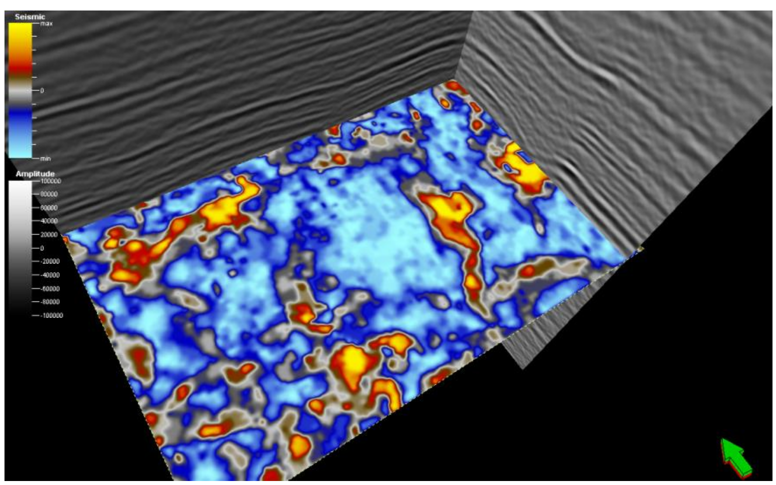


Under the *Operations* tab, you will note that (1) the histogram is neither Gaussian (like seismic amplitude) nor log-normal (like RMS amplitude and envelope). Rather, what you see is the histogram of the dual attribute data multiplexed into a single composite attribute histogram. Next, select (2) *User defined* and unselect (3*) Zero centric*. The composite attributes will range from 0 to 255. In order to map accurately against the 2D color bar, explicitly set (4) *Min* to be 0.0 and (5) *Max* to be 255.0. Since the data range between 0 and 255, choose the (6) *Realization quality* to be *Integer 8 bit*. Finally, click Realize and close the window to exit when you are done.

Your Input tab should have the Realized data below it:

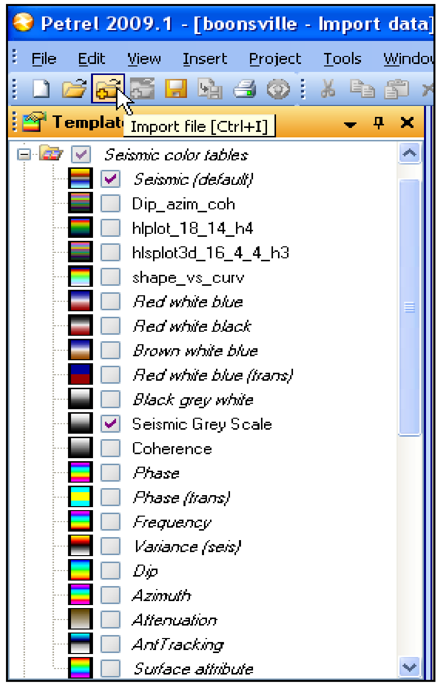


Choosing the plot, the time slice at t=-1100 ms (for those of you not familiar with Petrel, it has the time and depth axes as Positive Up) should give you the following display:

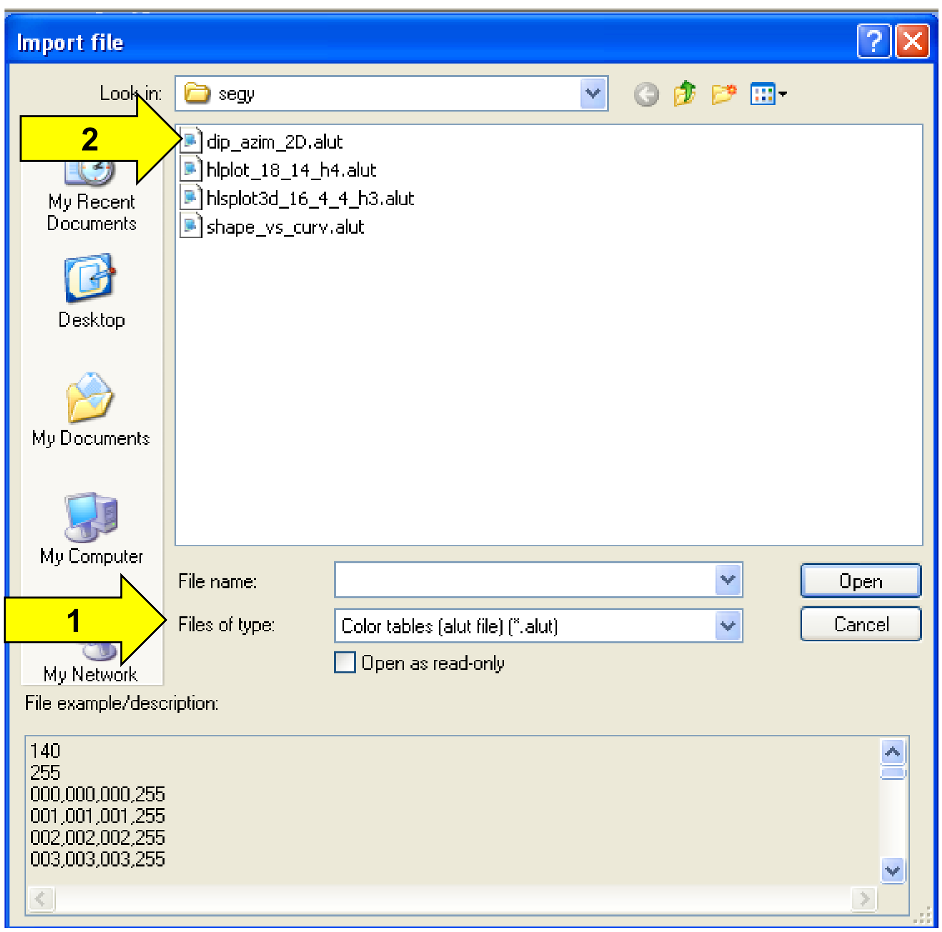


Petrel assigns the default Seismic color bar. Therefore, you will need to import the 2D color bar generated by the program **hlplot**. To do so, first copy the file *hsplot_18_14_360.alut* from your Linux workstation. Rename it to be something that makes sense, such as *dip_azim_2D.alut*. Then highlight your *Templates* tab and click the import icon:

Display Tools: Program **hsplot**

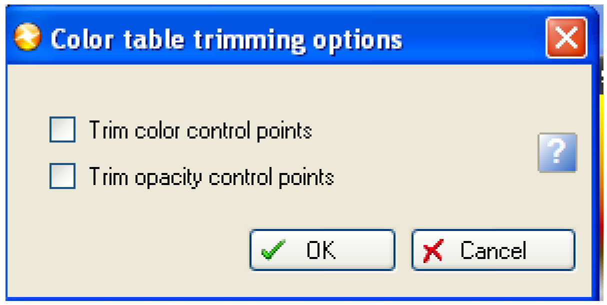


The following window opens:



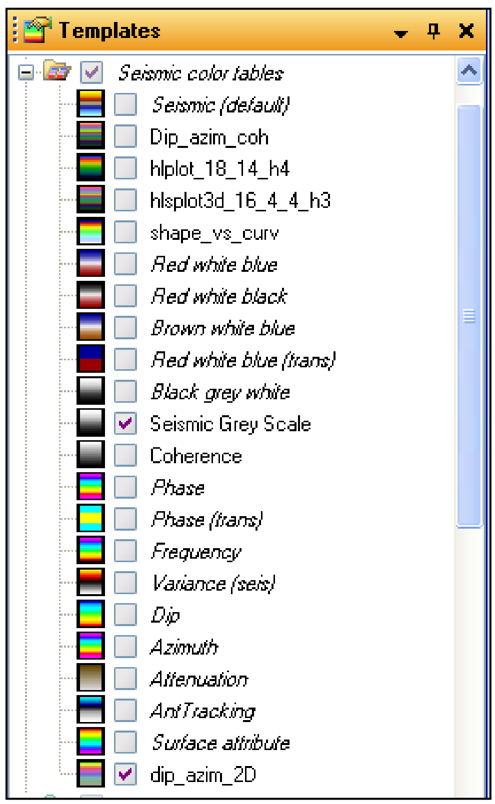
First, (1) select *Files of type* to be *Color tables (alut file) (*.alut)* . Then (2) select the file *dip_azim_2D.alut* and click *Open.*

The following pop-up window appears:

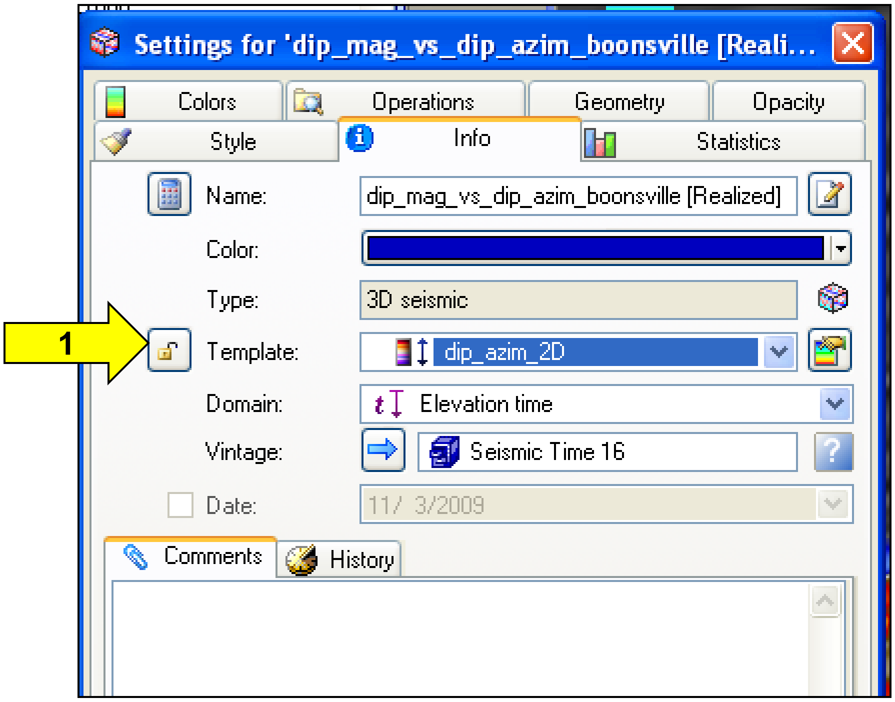


Petrel will want to Trim your color and opacity control points. <u>It is extremely important that you unselect this default option or your color mapping will not work!</u> Once unselected, click OK.

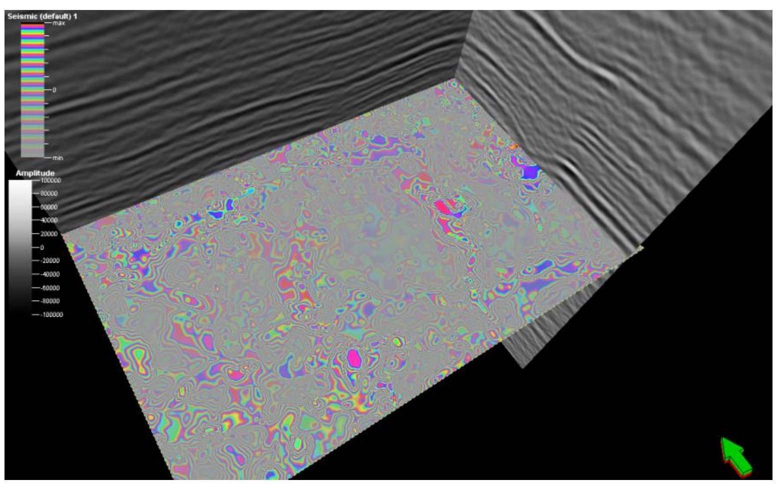
The new color bar will appear at the bottom of your *Seismic color tables* in the *Templates* tab:



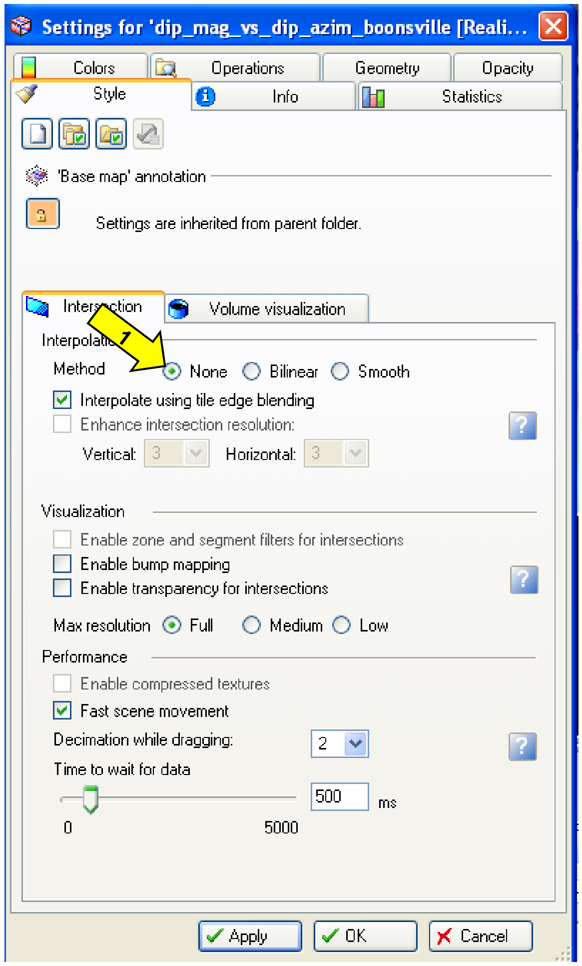
Return to the Input tab and double-click the realized data volume dip_mag_vs_dip_azim_boonsville (Realized) to open the Setting tab:



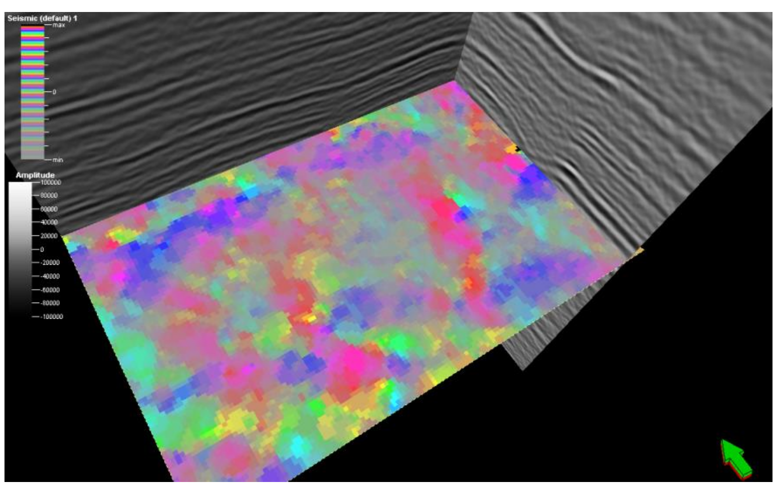
Under the *Info* tab, (1) select your new color bar, *dip_azim_2D*. Click Apply and your seismic attribute time slice may look like the following:



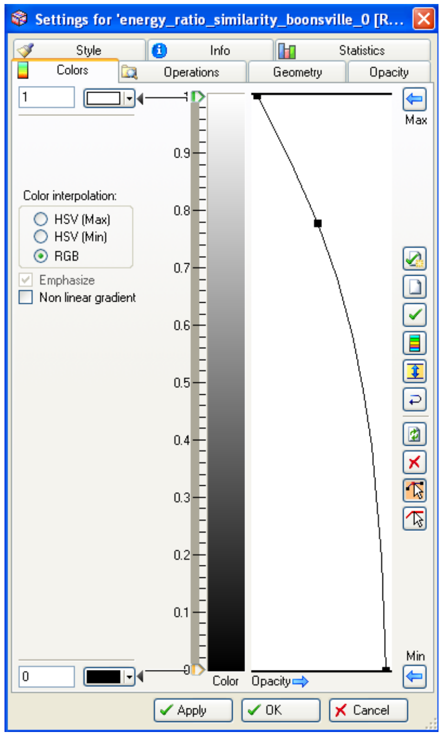
The problem here is that Petrel is interpolating colors. Go back to the *Settings* tab and under the *Style* tab, click (1) *None* as the *Interpolation Method*. You will encounter this problem when using any discontinuous color bar. In addition to multiattribute images, these may occur with discrete classifications (e.g. if you import classes from Stratimagic) or discrete color tables from facies inversion (e.g. green for shale, yellow for sandstone, blue for limestone).



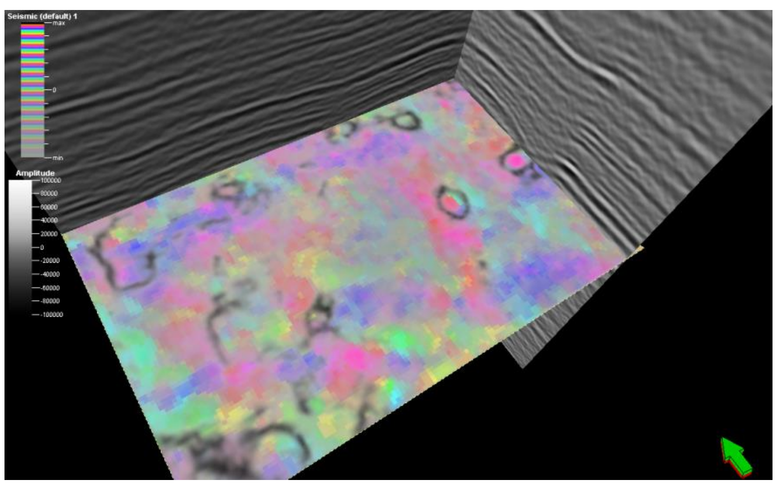
The new image looks like this:



Later, using program **similarity3d**, you will generate various coherence images. You will wish to set your transparency for energy-ratio similarity to be transparent for coherent regions, and black for incoherent regions:



Co-rendering the energy ratio similarity image with the composite dip magnitude vs. dip azimuth image results in the following:



Careful examination indicates that the color-encoded dip magnitude vs. dip azimuth images shows the reflectors dipping into the circular karst collapse features.

References

Marfurt, K. J., 2015, Techniques and best practices in multiattribute display: Interpretation, **3**, 1-24.