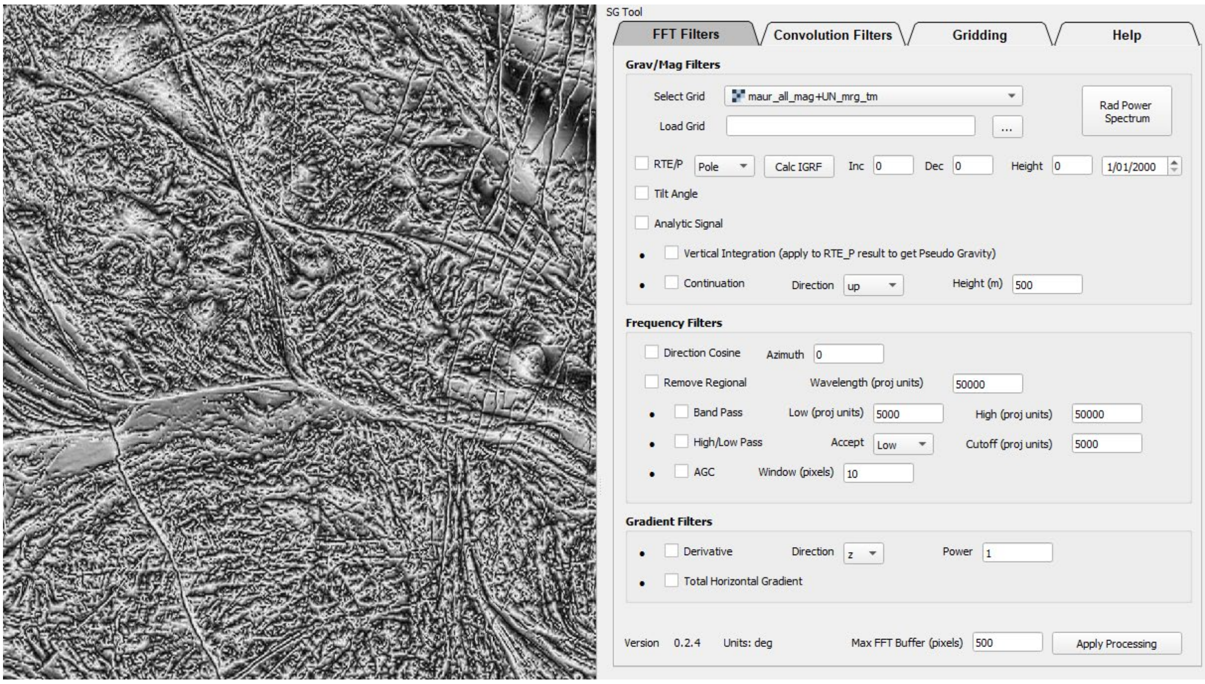
Structural Geophysics Tools v 0.2.7



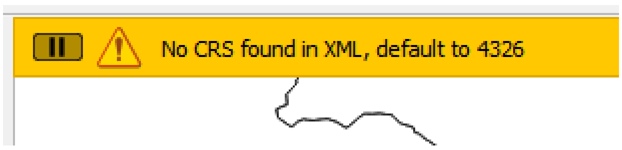
1. Load and Process Grid

a) Load a raster image from file

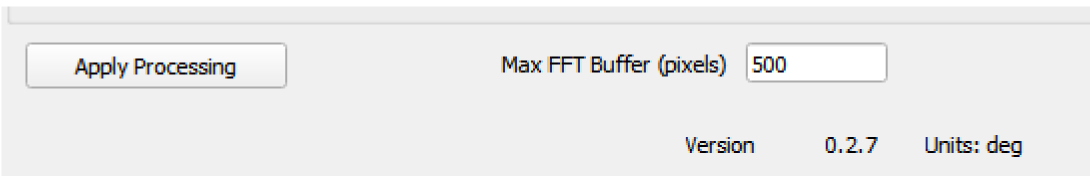
- If a GRD grid (Oasis Montaj) is selected, the plugin will attempt to load CRS from the associated xml file, unfortunately (and by design) xml files come in a huge number of variations, and the sgtools code searches for an EPSG definition, and if it fails it defaults to WGS84 (i.e. a degree-based projection).

- In any case the grid is saved as geotif in the same directory as the original grid.

- The plugin flags if it can’t find a valid CRS with a warning but you have to manually set the CRS in QGIS:



- The plugin also provides the units at the bottom right of the plugin for the currently selected layer:



b) Whatever layer is shown in the layer selector will be the one processed by whatever combination of filters are selected by check boxes.

- All processed files will be saved as geotiffs or ERS format files depending on the original format, will be saved in the same directory as the original file, and will have a suffix added describing the processing step.

- If a RTP or RTE calculation is performed, it is possible to define the magnetic field manually or the IGRF mag field parameters can be assigned based on the centroid of grid, plus date and survey height

- If a file exists on disk, it will be overwritten, although QGIS plugins don't always like saving to disks other than C: on Windows.

- Length units are defined by grid properties except for Up/Down Continuation (so Lat/Long wavelengths should be defined in degrees!)

c) If multiple processing steps are required, first apply one process, select the result and then apply subsequent steps.

2. Grav/Mag Filters

Reduction to the Pole

$$H_{RTP}(k_x, k_y) = \frac{k\cos I\cos D + ik_y\cos I\sin D + k_x\sin I}{k}$$

Converts magnetic data measured at any inclination and declination to what it would be if measured at the magnetic pole. Where

- k_x and k_y : The wavenumber components in the x and y directions.
- k = The total wavenumber magnitude = sqrt{k_x^2 + k_y^2}
- I : Magnetic inclination (in radians).
- D : Magnetic declination (in radians).
- i : Imaginary unit.

Reduction to the Equator

$$H_{RTE}(k_x, k_y) = \frac{k\cos I\cos D + ik_y\cos I\sin D + k_x\sin I}{k\cos I\cos D - ik_y\cos I\sin D + k_x\sin I}$$

Converts magnetic data measured at any inclination and declination to what it would be if measured at the magnetic equator. Where

- k_x and k_y : The wavenumber components in the x and y directions.
- k = The total wavenumber magnitude = sqrt{k_x^2 + k_y^2}
- I : Magnetic inclination (in radians).
- D : Magnetic declination (in radians).
- i : Imaginary unit.

Analytic Signal

$$A(x,y) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial z}\right)^2}$$

Computes the total amplitude of the gradients, independent of field inclination or declination. Useful for locating edges of potential field sources (e.g., faults or contacts).

Tilt Angle

$$T = \tan^{-1}\left(\frac{\frac{\partial f}{\partial z}}{\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}}\right)$$

Enhances the contrast of geological features by highlighting gradients relative to the vertical component. Where
df/dz : Vertical derivative of the field. df/dx , df/dy : Horizontal derivatives of the field.

Continuation

$$H(k) = e^{-kh}$$

Where
h > 0 for upward continuation.
h < 0 for downward continuation.

Vertical Integration

$$H(k_x, k_y) = \frac{1}{k}$$

When applied to an RTE or RTP image provides the so called Pseudogravity result
Where
k = sqrt{k_x^2 + k_y^2} .

3. Frequency Filters

Band Pass

$$e^{-(k-k_c)^2 / (2\sigma^2)} - e^{-(k+k_c)^2 / (2\sigma^2)}$$

The band-pass filter retains frequencies within a specified range, suppressing both low and high frequencies outside this range. Where
k_c : The central frequency of the band. sigma : The width of the frequency band.

Directional Cosine Filter

$$H(k_x, k_y) = \left|\cos(\theta - \theta_c)\right|^p$$

The Directional Cosine Filter emphasizes or suppresses frequency components along a specific direction.
H(k_x, k_y): Filter response as a function of wavenumber components k_x and k_y.
theta = \arctan\left(\frac{k_y}{k_x}\right) : Angle of the frequency component.

$theta_c$: Center direction (in radians), representing the direction to emphasize.
p : Degree of the cosine function. Higher (p) sharpens the directional emphasis.

High Pass

$$H(k) = 1 - e^{-k^2 / (2k_c^2)}$$

The high-pass filter removes low-frequency components (long wavelengths) while retaining high-frequency components (short wavelengths). Where
k_c : The cutoff frequency where the filter begins attenuating lower frequencies.

Low Pass

$$H(k) = e^{-k^2 / (2k_c^2)}$$

The low-pass filter suppresses high-frequency components (short wavelengths) while preserving low-frequency components (long wavelengths). Where: k_c : The cutoff frequency where the filter begins attenuating higher frequencies.

Remove Regional

$$H(k) = e^{-k^2 / (2k_c^2)}$$

The low-pass filter suppresses high-frequency components (short wavelengths) while preserving low-frequency components (long wavelengths). Where
k_c : The cutoff frequency where the filter begins attenuating higher frequencies.

Automatic Gain Control

$$AGC(x, y) = \frac{f(x, y)}{\mathrm{RMS}(f(x, y), w)}$$

Where
RMS(f, w) is the root mean square of the data over a window w.

Radially averaged power spectrum (but needs testing!)

$$P(k) = \frac{1}{N_k} \sum_{(k_x, k_y) \in k} |\,\mathrm{FFT}(f)|^2$$

Where
P(k) is the radially averaged power spectrum, and N_k is the number of samples in the radial bin.

4. Gradient Filters

Derivative

$$\frac{\partial f}{\partial u} = \frac{\partial f}{\partial x}\cos\theta + \frac{\partial f}{\partial y}\sin\theta$$

Where
theta is the angle defining the direction of the derivative (x,y or z).

Total Horizontal Gradient

$$THG(x,y) = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Convolution Filters

Mean Applies a mean filter using a kernel of size n x n .

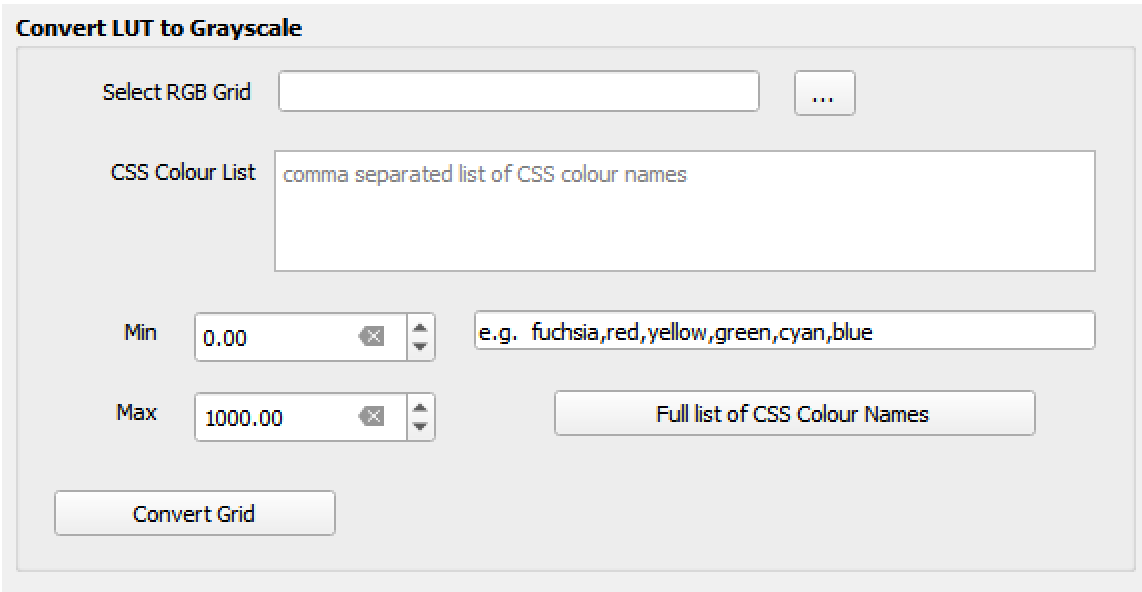
Median
Applies a median filter using a kernel of size n x n .

Gaussian
Applies a Gaussian filter with a specified standard deviation.

Directional
Apply directional filter (NE, N, NW, W, SW, S, SE, E)

Sun Shading
Computes relief shading for a digital elevation model (DEM) or other 2D grids.

5. RGB Importer



This tool takes a 3-band RGB image of some data and attempts to convert it to a monotonically increasing 1-band grid. The user provides the sequence of colours seen in the look up table using colour names from the CSS colour list as provided by matplotlib.

a) The **min max** values define the range of the data (if known)
b) The new grid (originalfilename_gray.tif) is saved in the same directory as the original grid
c) Assumes a linear look up table display (e.g. not histogram equalised, quantised…)
d) Best <u>without</u> shading applied to image, but not awful if it has been used
e) Reasonably close colour choice required.

f) Rename existing grey scale extract so you can try different colour lists, as you can't overwrite current layer

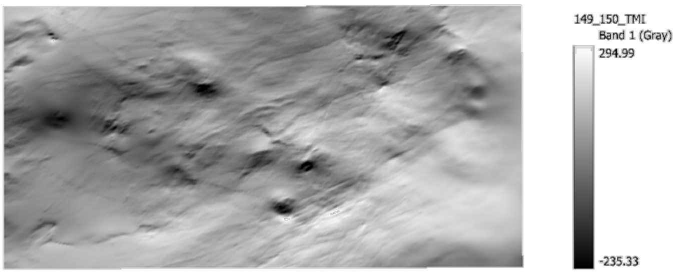
g) Could be modified to accept full csv LUT definition for more geeky users?

h) Resulting image usually needs a Gaussian filter to be applied first if high pass filters are to be used

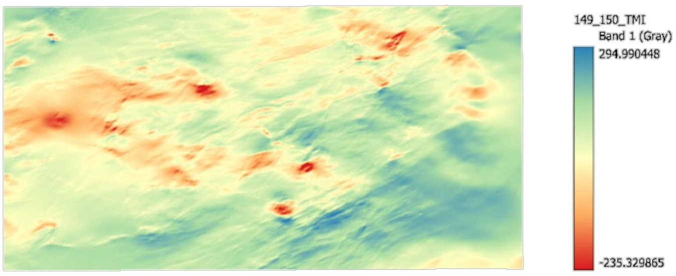
CSS Colour Names https://matplotlib.org/stable/gallery/color/named_colors.html#css-colors

Example usage

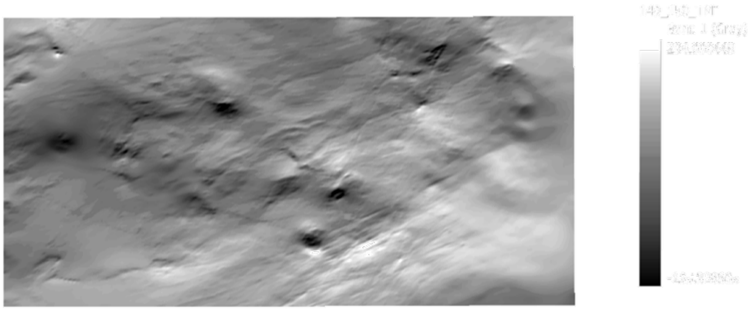
The original greyscale representation of a 1-band TMI grid :



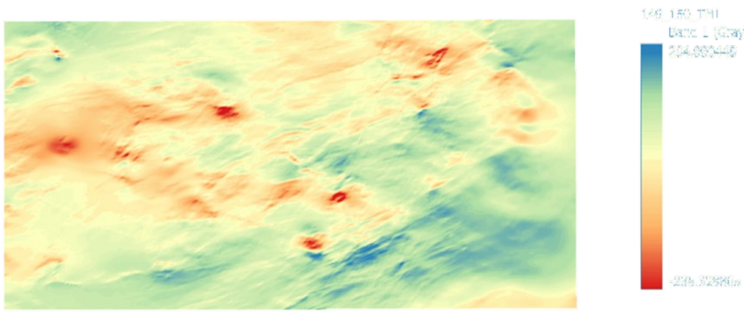
The colour representation of a 1-band TMI grid (QGIS Spectral LUT). <u>This image is saved out as a 3-band RGB representation</u>:



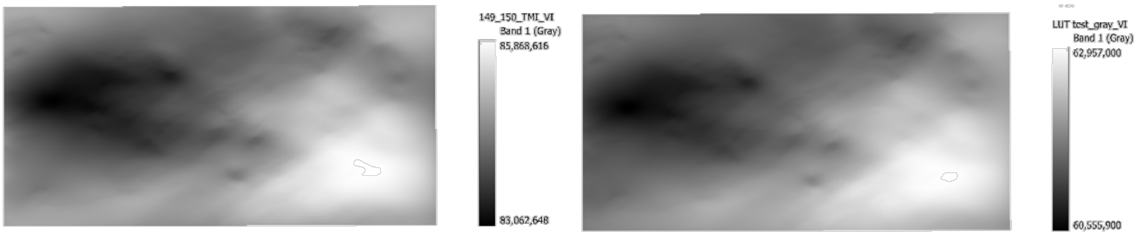
The greyscale representation extracted from the 3-band RGB using the colour LUT sequence [teal, lemonchiffon, red] :



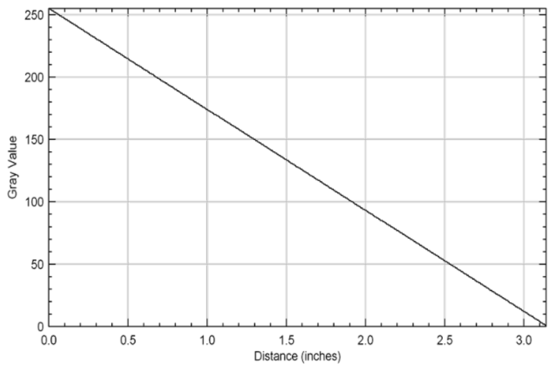
The colour representation of a the extracted TMI grid (QGIS Spectral LUT).



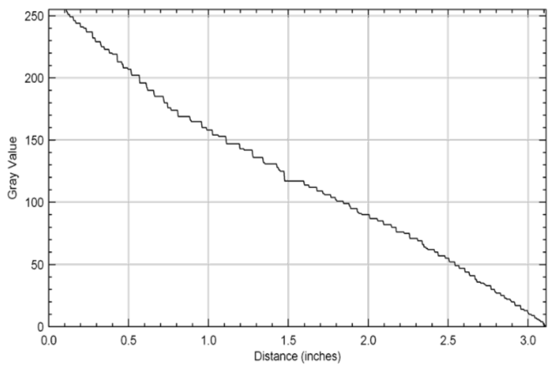
Comparison between Vertical Integration of original data (left) and extracted data (right)



LUT from Original



LUT extracted from RGB



Comparison between original data LUT and data LUT extracted from RGB image

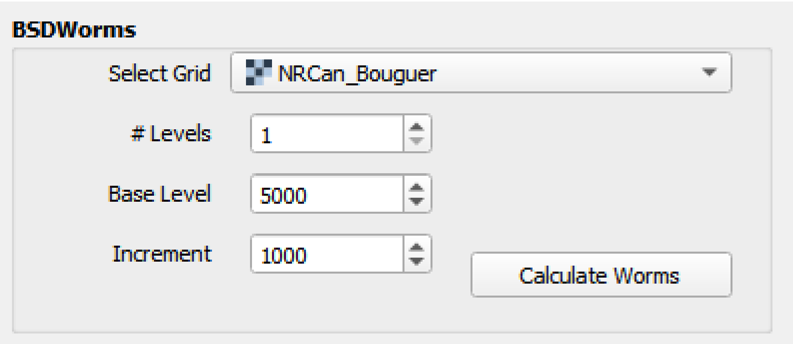
6. Import point or line data

This tool allows you to import a csv file (which of course you can do anyway in QGIS) or an XYZ format file (an ascii format that divides data up by flight line numbers). Optionally include tie lines for the latter option.

7. Gridding

This tool gets you started with gridding point data by allowing you to select an already-loaded point file and field to be gridded and to see the consequence of a given cell size in terms of number of rows and columns. Full dialog for each interpolation method allows other parameters to be set.

8. BSDWorms



This tool uses Frank Horowitz's bsdwormer tool (https://bitbucket.org/fghorow/bsdwormer/) to build wavelet transform "worms" for metre-based grids:

a) The grids must be of gravity or RTE/RTP + Vertical Integration grids of Magnetic data
b) This will not work for degree-based projections
c) **# Levels** are the number of levels of worms to calculate
d) **Base Level** is the height above 0 to calculate the first worm level (often the 0 level is very noisy so best ignored)
e) **Increment** provides the distance in metres between levels
f) Worms are saved out in the same directory as the original grid as a single csv file (originalfilename_worms.csv) that can be loaded into QGIS or a 3D renderer such as Geoscience Analyst
g) A padded version of the grid is also saved out and this can be removed after the worms are calculated but is provided for debugging purposes.

9. Code development

- Calcs ChatGPT and Mark Jessell

- Plugin construction - Mark Jessell using QGIS Plugin Builder Plugin https://g-sherman.github.io/Qgis-Plugin-Builder/

- IGRF calculation - pyIGRF https://github.com/ciaranbe/pyIGRF

- GRD Loader & Radially averaged power spectrum Fatiando a Terra crew & Mark Jessell https://www.fatiando.org/

- Example geophysics data in image above courtesy of Mauritania Govt. https://anarpam.mr/en/

- Worming of grids uses Frank Horowitz's bsdwormer https://bitbucket.org/fghorow/bsdwormer/

- Thanks to Lyal Harris for extensive beta testing!