Using Python to Map of the Impacts of Geology and Fluvial Processes on the Development of Drainage Networks on Mars

Principles of Geocomputing 5541

> The synopsis of this project is to map out linear geological features (ligaments) on the surface of Mars to show evidence of past fluvial/drainage activity (with some emphasis on Valles Marineris and Olympus Mons). Using original Mars DEMs and three raster analytic layers, to find evidence; these include: watershed, hillshade, and slope, there is substantial geological evidence that these networks determined the fluvial processes. Having performed the digitization and analysis in ArcMap for a prior undergraduate project, the goal here is to simply automate the maps with Python and visualize them. The whole project itself utilized statistical analysis in simulating three-dimensional imagery, calculating indices, and assessing differences in terrain, atmospheric pressure, oxidation, and night/day cycles. In conjunction with Earth-Sciences Department at Bemidji State University.
>
> Alexander Danielson danie861

In [1]:
```
import rasterio #Open imagery to be displayed
import numpy as np #Allows for arrays and matrices
import matplotlib.pyplot as plt #Allows to plot the raster/tif files
import geopandas #Display the rows and columns in the shapefiles
import matplotlib as mpl
import geopandas as gpd #Plot out geological linear features
from rasterio.plot import plotting_extent #Sets the plotting extent of geological linea
from matplotlib.colors import BoundaryNorm, LinearSegmentedColormap
import numpy as np



mars = rasterio.open('Mars_MGS_MOLA_Shade_global_463m.tif') #Opens Mars imagery from NA
watershed = rasterio.open('Watersh_Flow12.tif') #Watersheds of Mars to delinate areas o
meta = watershed.meta
slopemars = rasterio.open('slopemars1.tif') #Slope for steepness of terrain
hillshade = rasterio.open('hillelemars1.tif') #Hillshade for downhill direction of terr
ligament = ('Ligament_Lines.shp') #Assigned ligaments
slopelinear = ('SlopeLinear.shp')
hillshadeline = ('HillshadeLinear.shp')

mars = mars.read() #Assigned variables to map
watershed = watershed.read()
slopemars = slopemars.read()
hillshade = hillshade.read()

print(mars.shape) #Prints out the dimensions of the Mars DEM
print(np.amin(mars[0]))
print(np.amax(mars[0]))
print(np.amax(mars[0]) + abs(np.amin(mars[0])))
```

```
print(watershed.shape) #Prints out the dimensions of the Mars Watershed
print(np.amin(watershed[0]))
print(np.amax(watershed[0]))
print(np.amax(watershed[0]) + abs(np.amin(watershed[0])))

print(slopemars.shape) #Prints out the dimensions of the Mars Tharsis Area via Slope
print(np.amin(slopemars[0]))
print(np.amax(slopemars[0]))
print(np.amax(slopemars[0]) + abs(np.amin(slopemars[0])))

print(hillshade.shape) #Prints out the dimensions of the Mars Tharsis Area via Hillshad
print(np.amin(hillshade[0]))
print(np.amax(hillshade[0]))
print(np.amax(hillshade[0]) + abs(np.amin(hillshade[0])))
```

```
(1, 22528, 46080)
0
254
254
(3, 10433, 21339)
0
255
255
(1, 12346, 14529)
0.0
79.75194
79.75194
(1, 12346, 14529)
0
255
255
```

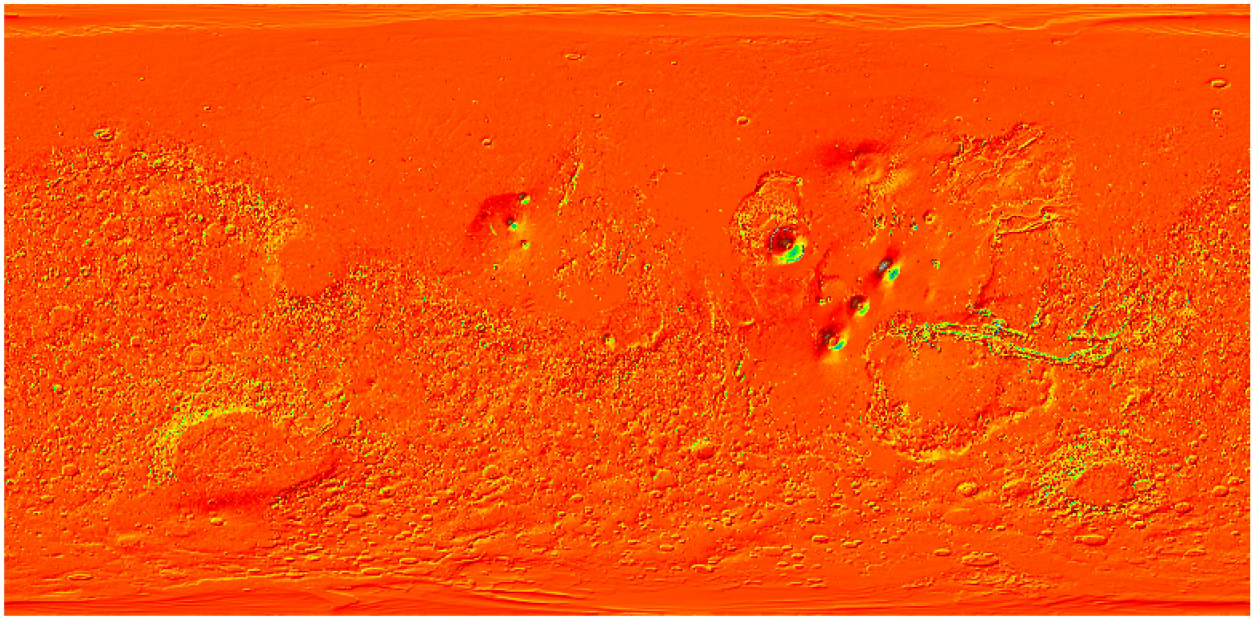
In [2]:

```
# Reads the shapefiles for the geological features to be displayed
gdfligament = gpd.read_file('Ligament_Lines.shp').set_crs("EPSG:9001")
gdfhillshade = gpd.read_file('HillshadeLinear.shp')
gdfslope = gpd.read_file('SlopeLinear.shp')
```

In [3]:

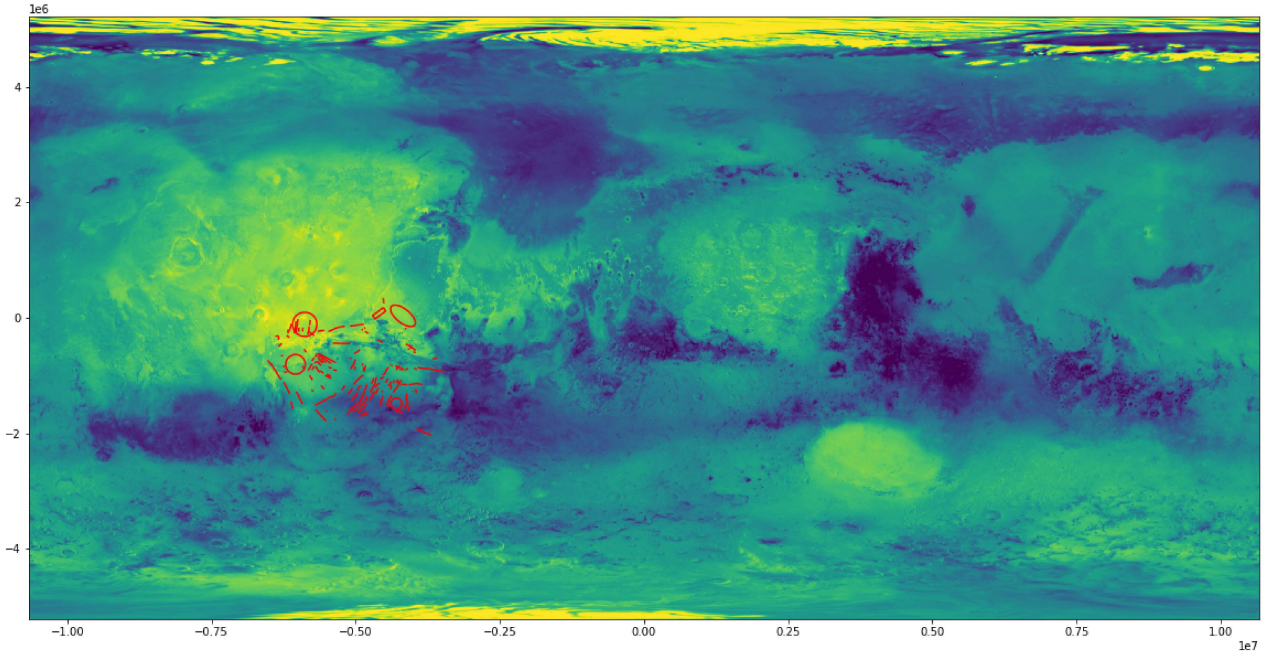
```
from matplotlib.colors import BoundaryNorm, LinearSegmentedColormap #Maps the raster la


custom_cmap = LinearSegmentedColormap.from_list('mars', ['#162252', #Creates a colored
                                                         '#104E8B',
                                                         '#00B2EE',
                                                         '#00FF00',
                                                         '#FFFF00',
                                                         '#FFA500',
                                                         '#FF0000',
                                                         '#8b0000',
                                                         '#964B00',
                                                         '#808080',
                                                         '#FFFFFF'], N=2221)

bounds = np.arange(-8210, 14000, 10) #Sets boundaries for raster
norm = BoundaryNorm(bounds, custom_cmap.N)

fig, ax = plt.subplots() #Makes subplots of area
fig.set_size_inches(14, 7)
```

```
ax.imshow(mars[0], cmap=custom_cmap) #Shows assigned color map of Mars
ax.axis('off')
newax = fig.add_axes([0.82, 0.13, 0.08, 0.08], anchor='NE')
newax.axis('off')
plt.show()
```



In [7]:

```
from matplotlib.colors import BoundaryNorm, LinearSegmentedColormap



custom_cmap = LinearSegmentedColormap.from_list('watershed', ['#162252', #Shows watersh
                                                              '#104E8B',
                                                              '#00B2EE',
                                                              '#00FF00',
                                                              '#FFFF00',
                                                              '#FFA500',
                                                              '#FF0000',
                                                              '#8b0000',
                                                              '#964B00',
                                                              '#808080',
                                                              '#FFFFFF'], N=2221)

bounds = np.arange(-8210, 14000, 10)
norm = BoundaryNorm(bounds, custom_cmap.N)

fig, ax = plt.subplots()
fig.set_size_inches(20, 14)

ax.imshow(watershed[0], extent = (minx, maxx, miny, maxy))# art = ax.pcolormesh(xx, yy,
# ax.axis('off')
#newax = fig.add_axes([0.82, 0.13, 0.08, 0.08], anchor='NE')
#newax.axis('off')
gdfligament.plot(ax=ax, color = 'r') #Creates plot of geological linear feature for ide
plt.show()
```



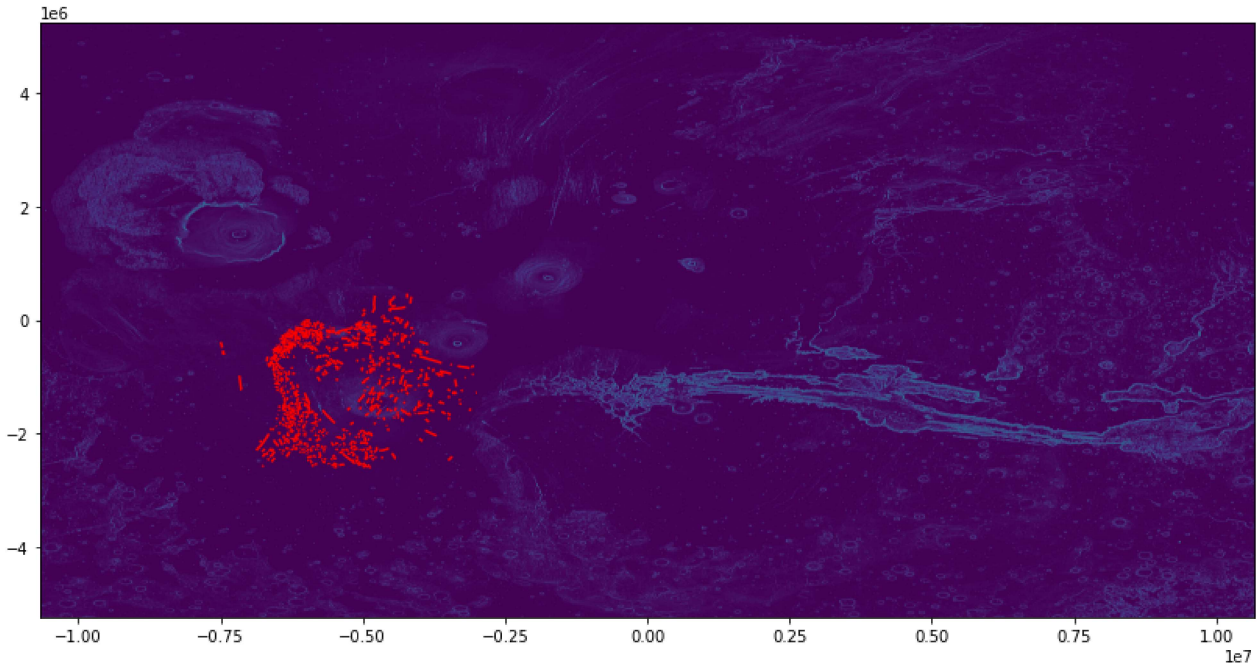
In [6]:

```
# xx,yy=np.mgrid[-10669444.87495712*0.01:-10669444.87495712+10433*.01:10433j,
               # 5216392.66818647*0.01:21339*0.01+5216392.66818647:21339j
               # ]

#CRS.from_wkt('PROJCS["Mars2000 Equidistant Cylindrical clon0",GEOGCS["GCS_Mars_2000_Sp
 #'transform': Affine(1000.0, 0.0, -10669444.87495712,
       # 0.0, -1000.0, 5216392.66818647)}
#(10433, 21339)

minx = -10669444.87495712
maxx = -10669444.87495712+21339*1000
maxy = 5216392.66818647
miny = 5216392.66818647-10433*1000
```

In [13]:

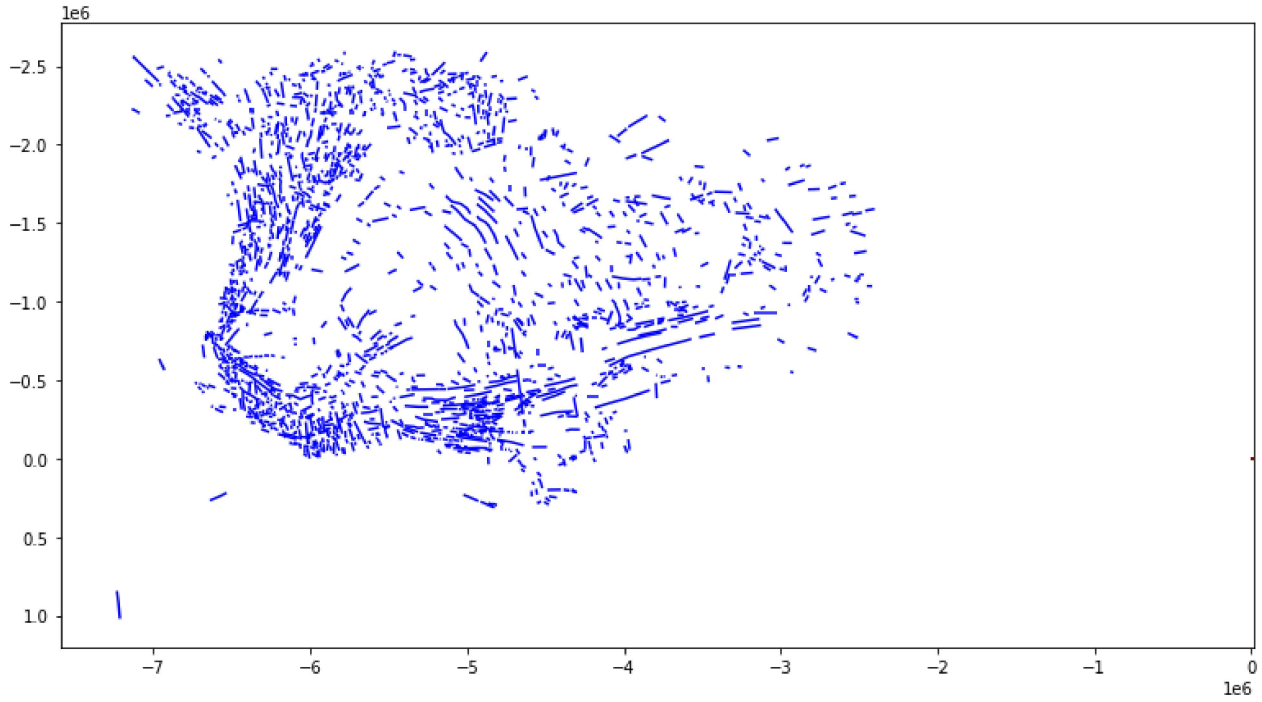
```
from matplotlib.colors import BoundaryNorm, LinearSegmentedColormap


custom_cmap = LinearSegmentedColormap.from_list('slopemars', ['#162252',
                                                              '#104E8B',
                                                              '#00B2EE',
                                                              '#00FF00',
                                                              '#FFFF00',
                                                              '#FFA500',
                                                              '#FF0000',
                                                              '#8b0000',
                                                              '#964B00',
                                                              '#808080',
                                                              '#FFFFFF'], N=2221)

bounds = np.arange(-8210, 14000, 10)
norm = BoundaryNorm(bounds, custom_cmap.N)

fig, ax = plt.subplots()
fig.set_size_inches(14, 7)

ax.imshow(slopemars[0], extent = (minx, maxx, miny, maxy))
```

```
#ax.axis('off')
#newax = fig.add_axes([0.82, 0.13, 0.08, 0.08], anchor='NE')
#newax.axis('off')
gdfslope.plot(ax=ax, color = 'r') #Displays slope geological linear features along slop
plt.show()
```



In [9]:

```
from matplotlib.colors import BoundaryNorm, LinearSegmentedColormap


custom_cmap = LinearSegmentedColormap.from_list('hillshade', ['#162252',
                                                              '#104E8B',
                                                              '#00B2EE',
                                                              '#00FF00',
                                                              '#FFFF00',
                                                              '#FFA500',
                                                              '#FF0000',
                                                              '#8b0000',
                                                              '#964B00',
                                                              '#808080',
                                                              '#FFFFFF'], N=2221)

bounds = np.arange(-8210, 14000, 10)
norm = BoundaryNorm(bounds, custom_cmap.N)

fig, ax = plt.subplots()
fig.set_size_inches(14, 7)

ax.imshow(hillshade[0], cmap=custom_cmap)
#ax.axis('off')
#newax = fig.add_axes([0.82, 0.13, 0.08, 0.08], anchor='NE')
#newax.axis('off')
gdfhillshade.plot(ax=ax, color = 'b') #Displays hillshade geological linear features al
plt.show()
```



In [15]:

```
colors_undersea = plt.cm.ocean(np.linspace(0.2, 0.8, 821)) #Assigns variables to be plo
undersea_map = LinearSegmentedColormap.from_list('mars', colors_undersea, N=821)

colors_land = plt.cm.terrain(np.linspace(0.25, 1, 1400))
land_map = LinearSegmentedColormap.from_list('watershed', colors_land, N=1400)

colors = np.vstack((colors_undersea, colors_land))
terrain_map = LinearSegmentedColormap.from_list('hillshade', colors, N=2221)

bounds = np.arange(-8210, 14000, 10)
norm = BoundaryNorm(bounds, terrain_map.N)
plt.show()
```

In [16]:

```
def change_sea_level(new_sea_level): #Shows a colormap for Mars to delienate the differ
    new_sea_level = new_sea_level / 10
    undersea = int(800 + new_sea_level)
    land = int(1400 - new_sea_level)
    colors_undersea = plt.cm.ocean(np.linspace(0.2, 0.8, undersea)) #Based on ocean dep
    undersea_map = LinearSegmentedColormap.from_list('watershed', colors_undersea, N=un

    colors_land = plt.cm.terrain(np.linspace(0.25, 1, land)) #Based on land surface
    land_map = LinearSegmentedColormap.from_list('mars', colors_land, N=land)

    colors = np.vstack((colors_undersea, colors_land)) #Based on terrian/angles
    terrain_map = LinearSegmentedColormap.from_list('slopemars', colors, N=2221)

    bounds = np.arange(-8210, 14000, 10)
    norm = BoundaryNorm(bounds, terrain_map.N)
    return terrain_map, norm


terrain_map0, norm0 = change_sea_level(new_sea_level = 0)
terrain_map3000, norm3000 = change_sea_level(new_sea_level = 3000)
terrain_map_3000, norm_3000 = change_sea_level(new_sea_level = -3000)
plt.show()
```

In [ ]:
```
terrain_map1500, norm1500 = change_sea_level(new_sea_level = 1500) #Plotting Mars Terra

fig, ax = plt.subplots()
fig.set_size_inches(14, 7)

ax.imshow(mars[0], cmap=terrain_map1500, norm=norm1500)
ax.axis('off')
newax = fig.add_axes([0.82, 0.13, 0.08, 0.08], anchor='NE')
newax.axis('off')

plt.show()
```

In [14]:
```
gdfligament.crs #Coordinate Reference System inforamtion
```

Out[14]:
```
<Geocentric CRS: EPSG:9001>
Name: IGS97
Axis Info [cartesian]:
- X[geocentricX]: Geocentric X (metre)
- Y[geocentricY]: Geocentric Y (metre)
- Z[geocentricZ]: Geocentric Z (metre)
Area of Use:
- name: World.
- bounds: (-180.0, -90.0, 180.0, 90.0)
Datum: IGS97
- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich
```

In [11]:
```
meta #Geodata frame information
```

Out[11]:
```
{'driver': 'GTiff',
 'dtype': 'uint8',
 'nodata': 255.0,
 'width': 21339,
 'height': 10433,
 'count': 3,
 'crs': CRS.from_wkt('PROJCS["Mars2000 Equidistant Cylindrical clon0",GEOGCS["GCS_Mars_2
000_Sphere",DATUM["Mars_2000_(Sphere)",SPHEROID["Mars_2000_Sphere_IAU_IAG",3396190,0]],P
RIMEM["Reference_Meridian",0],UNIT["degree",0.0174532925199433,AUTHORITY["EPSG","912
2"]]],PROJECTION["Equirectangular"],PARAMETER["standard_parallel_1",0],PARAMETER["centra
l_meridian",0],PARAMETER["false_easting",0],PARAMETER["false_northing",0],UNIT["metre",
1,AUTHORITY["EPSG","9001"]],AXIS["Easting",EAST],AXIS["Northing",NORTH]]'),
 'transform': Affine(1000.0, 0.0, -10669444.87495712,
        0.0, -1000.0, 5216392.66818647)}
```

In [7]:
```
!pip install earthpy #Attempt to use a different Python package by clipping the vector
```

```
Requirement already satisfied: earthpy in /opt/conda/lib/python3.9/site-packages (0.9.4)
Requirement already satisfied: matplotlib>=2.0.0 in /opt/conda/lib/python3.9/site-packag
es (from earthpy) (3.5.1)
Requirement already satisfied: rasterio in /opt/conda/lib/python3.9/site-packages (from
earthpy) (1.2.10)
Requirement already satisfied: geopandas in /opt/conda/lib/python3.9/site-packages (from
earthpy) (0.11.1)
Requirement already satisfied: scikit-image in /opt/conda/lib/python3.9/site-packages (f
rom earthpy) (0.19.2)
```

```
Requirement already satisfied: requests in /opt/conda/lib/python3.9/site-packages (from
earthpy) (2.26.0)
Requirement already satisfied: numpy>=1.14.0 in /opt/conda/lib/python3.9/site-packages
(from earthpy) (1.20.3)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.9/site-packag
es (from matplotlib>=2.0.0->earthpy) (4.28.5)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.9/site-packages
(from matplotlib>=2.0.0->earthpy) (8.4.0)
Requirement already satisfied: pyparsing>=2.2.1 in /opt/conda/lib/python3.9/site-package
s (from matplotlib>=2.0.0->earthpy) (3.0.6)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.9/site-packages (f
rom matplotlib>=2.0.0->earthpy) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.9/site-packag
es (from matplotlib>=2.0.0->earthpy) (1.3.2)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.9/site-pac
kages (from matplotlib>=2.0.0->earthpy) (2.8.2)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.9/site-packages
(from matplotlib>=2.0.0->earthpy) (21.3)
Requirement already satisfied: pyproj>=2.6.1.post1 in /opt/conda/lib/python3.9/site-pack
ages (from geopandas->earthpy) (3.3.0)
Requirement already satisfied: fiona>=1.8 in /opt/conda/lib/python3.9/site-packages (fro
m geopandas->earthpy) (1.8.20)
Requirement already satisfied: pandas>=1.0.0 in /opt/conda/lib/python3.9/site-packages
(from geopandas->earthpy) (1.4.2)
Requirement already satisfied: shapely<2,>=1.7 in /opt/conda/lib/python3.9/site-packages
(from geopandas->earthpy) (1.8.0)
Requirement already satisfied: click>=4.0 in /opt/conda/lib/python3.9/site-packages (fro
m rasterio->earthpy) (8.0.3)
Requirement already satisfied: cligj>=0.5 in /opt/conda/lib/python3.9/site-packages (fro
m rasterio->earthpy) (0.7.2)
Requirement already satisfied: snuggs>=1.4.1 in /opt/conda/lib/python3.9/site-packages
(from rasterio->earthpy) (1.4.7)
Requirement already satisfied: attrs in /opt/conda/lib/python3.9/site-packages (from ras
terio->earthpy) (21.2.0)
Requirement already satisfied: affine in /opt/conda/lib/python3.9/site-packages (from ra
sterio->earthpy) (2.3.1)
Requirement already satisfied: click-plugins in /opt/conda/lib/python3.9/site-packages
(from rasterio->earthpy) (1.1.1)
Requirement already satisfied: certifi in /opt/conda/lib/python3.9/site-packages (from r
asterio->earthpy) (2022.6.15)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.9/site-packages (fro
m rasterio->earthpy) (59.8.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.9/site-packages (f
rom requests->earthpy) (3.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /opt/conda/lib/python3.9/site-pa
ckages (from requests->earthpy) (1.26.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/lib/python3.9/sit
e-packages (from requests->earthpy) (2.0.9)
Requirement already satisfied: PyWavelets>=1.1.1 in /opt/conda/lib/python3.9/site-packag
es (from scikit-image->earthpy) (1.2.0)
Requirement already satisfied: networkx>=2.2 in /opt/conda/lib/python3.9/site-packages
(from scikit-image->earthpy) (2.8.6)
Requirement already satisfied: tifffile>=2019.7.26 in /opt/conda/lib/python3.9/site-pack
ages (from scikit-image->earthpy) (2021.11.2)
Requirement already satisfied: imageio>=2.4.1 in /opt/conda/lib/python3.9/site-packages
(from scikit-image->earthpy) (2.13.4)
Requirement already satisfied: scipy>=1.4.1 in /opt/conda/lib/python3.9/site-packages (f
rom scikit-image->earthpy) (1.8.1)
Requirement already satisfied: munch in /opt/conda/lib/python3.9/site-packages (from fio
na>=1.8->geopandas->earthpy) (2.5.0)
```

```
Requirement already satisfied: six>=1.7 in /opt/conda/lib/python3.9/site-packages (from
fiona>=1.8->geopandas->earthpy) (1.16.0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.9/site-packages (f
rom pandas>=1.0.0->geopandas->earthpy) (2021.3)
```

In [8]:

```
!pip install rioxarray #Attempt to use a different Python package clip the vector to th
```

```
Requirement already satisfied: rioxarray in /opt/conda/lib/python3.9/site-packages (0.1
2.0)
Requirement already satisfied: rasterio>=1.1.1 in /opt/conda/lib/python3.9/site-packages
(from rioxarray) (1.2.10)
Requirement already satisfied: xarray>=0.17 in /opt/conda/lib/python3.9/site-packages (f
rom rioxarray) (2022.6.0)
Requirement already satisfied: pyproj>=2.2 in /opt/conda/lib/python3.9/site-packages (fr
om rioxarray) (3.3.0)
Requirement already satisfied: packaging in /opt/conda/lib/python3.9/site-packages (from
rioxarray) (21.3)
Requirement already satisfied: certifi in /opt/conda/lib/python3.9/site-packages (from p
yproj>=2.2->rioxarray) (2022.6.15)
Requirement already satisfied: numpy in /opt/conda/lib/python3.9/site-packages (from ras
terio>=1.1.1->rioxarray) (1.20.3)
Requirement already satisfied: click>=4.0 in /opt/conda/lib/python3.9/site-packages (fro
m rasterio>=1.1.1->rioxarray) (8.0.3)
Requirement already satisfied: affine in /opt/conda/lib/python3.9/site-packages (from ra
sterio>=1.1.1->rioxarray) (2.3.1)
Requirement already satisfied: click-plugins in /opt/conda/lib/python3.9/site-packages
(from rasterio>=1.1.1->rioxarray) (1.1.1)
Requirement already satisfied: attrs in /opt/conda/lib/python3.9/site-packages (from ras
terio>=1.1.1->rioxarray) (21.2.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.9/site-packages (fro
m rasterio>=1.1.1->rioxarray) (59.8.0)
Requirement already satisfied: cligj>=0.5 in /opt/conda/lib/python3.9/site-packages (fro
m rasterio>=1.1.1->rioxarray) (0.7.2)
Requirement already satisfied: snuggs>=1.4.1 in /opt/conda/lib/python3.9/site-packages
(from rasterio>=1.1.1->rioxarray) (1.4.7)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.9/site-packages (fr
om xarray>=0.17->rioxarray) (1.4.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/lib/python3.9/site
-packages (from packaging->rioxarray) (3.0.6)
Requirement already satisfied: python-dateutil>=2.8.1 in /opt/conda/lib/python3.9/site-p
ackages (from pandas>=1.2->xarray>=0.17->rioxarray) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.9/site-packages (f
rom pandas>=1.2->xarray>=0.17->rioxarray) (2021.3)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.9/site-packages (from
python-dateutil>=2.8.1->pandas>=1.2->xarray>=0.17->rioxarray) (1.16.0)
```

In [ ]:

```
# xx,yy=np.mgrid[-10669444.87495712*0.01:-10669444.87495712+10433*.01:10433j,
               # 5216392.66818647*0.01:21339*0.01+5216392.66818647:21339j
               # ]

#CRS.from_wkt('PROJCS["Mars2000 Equidistant Cylindrical clon0",GEOGCS["GCS_Mars_2000_Sp
 #'transform': Affine(1000.0, 0.0, -10669444.87495712,
       # 0.0, -1000.0, 5216392.66818647)}
#(10433, 21339)

minx = -10669444.87495712
maxx = -10669444.87495712+21339*1000
maxy = 5216392.66818647
miny = 5216392.66818647-10433*1000
```

```
#Extraneous Data Set-up Information
```

> The comments (#) denote where keycode/detail was implemented. A lot of the same interactions of code have been repeated, this it for simple visualization proposes as this isn't meant to be a complex operation. Only to show the juxtaposition of geological interactions on the surface and to show how the ligament features compare in conforming to shaping the fluvial valleys on the planet. As stated in the synopsis, most work was rendered in ArcGIS and the visualization and colormaps give inklings into how the terrain on Mars is geologically congruent to that of Earth and that they share a similar geological history.