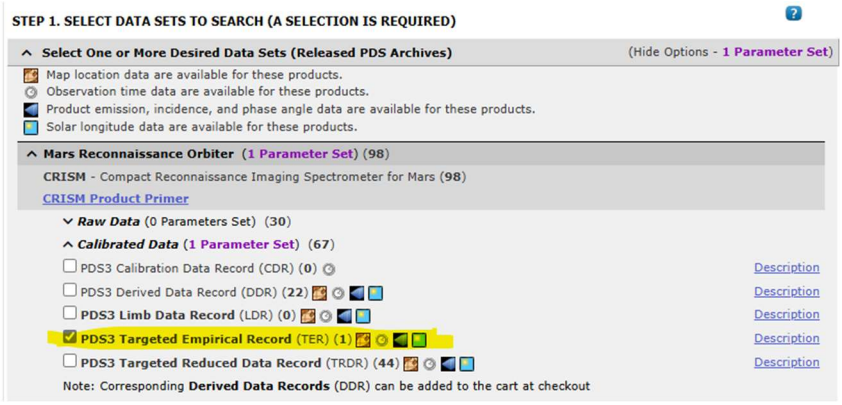
Step 1 – Import and Process CRISM MRO Spectral Library Data

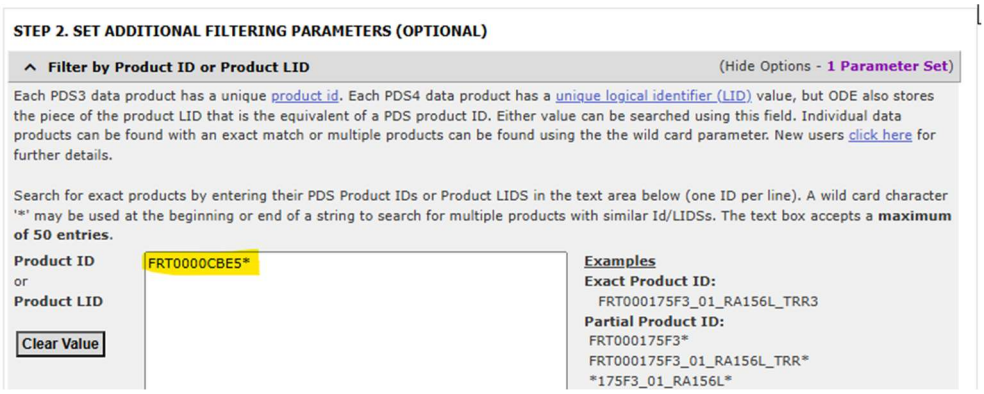
Navigate to the folder MRO_Spectra_Library_Conversion and run the script CRISM_Replication_Library.py to process the spectral library data.

Step 2 – Download CRISM TER3 Data for a Target Region

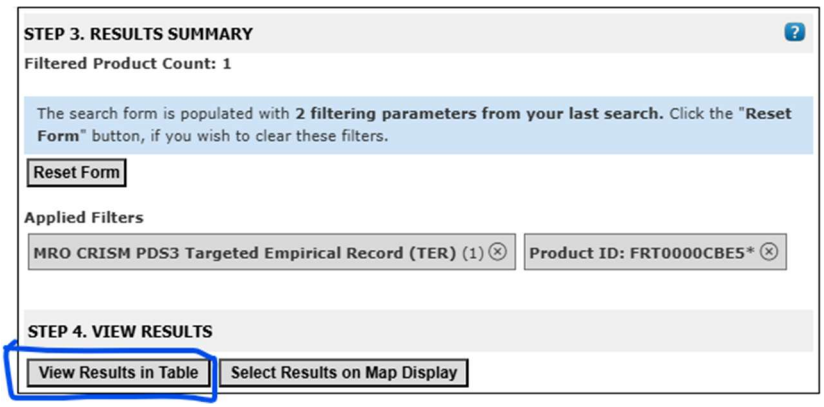
1. Go to https://ode.rsl.wustl.edu/mars/productsearch.
2. In Step 1, select TER3 as the product type.



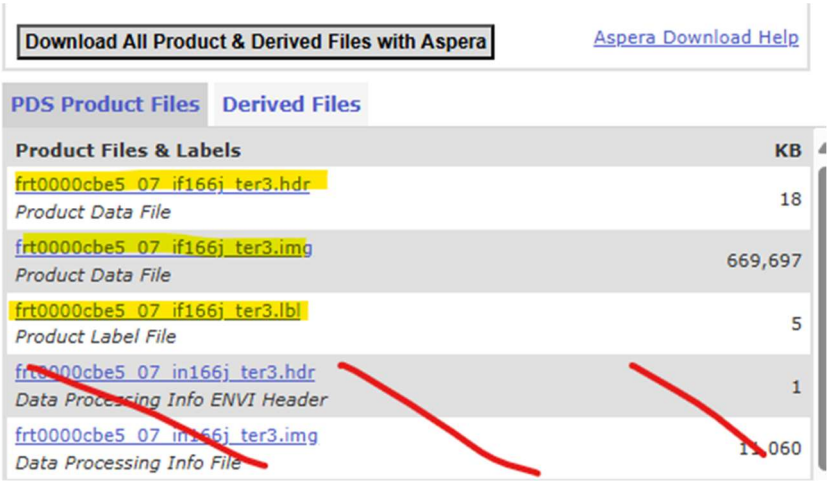
3. In Step 2, define your target region by entering the appropriate location ID.



4. Review the search results.



5. Download all files containing "if".



Step 3 – Convert TER3 Data into Summary Products

1. Place all downloaded TER3 files in a single folder.
2. Open the script CRISM_Spectral_Conversion.py located in the CRISM_Spectral_Conversion folder.
3. Specify the path to the TER3 .img files and set the desired output directory in the script.

```
#SPECIFY INPUTS AND OUTPUT LOCATIONS
#--------------------------------
TER3_Location = "D:/CRISM Data/FRT0000CBE5/TER3 Data/frt0000cbe5_07_if166j_ter3.img"
Output_Location = "D:/CRISM Data/FRT0000CBE5/Python_Converted"
h5_file_path = os.path.join(Output_Location, "stacked_frames.h5")
```

4. Run the script. *Note: This process may take several hours.*

Step 4 – Train the SOM

1. Open som_training.py in the SOM_Training folder.
2. Set the path to the .h5 file generated in Step 3.
3. Configure the SOM training parameters as needed.

4. Run the script to train the Self-Organizing Map.

```
# ----------------------
# Import Processed TER3 Data (H5 File)
# ----------------------
file_path = "D:/CRISM Data/FRT0000634B/TER_Data/Python_Converted/stacked_frames.h5"
with h5py.File(file_path, mode: "r") as f:
    combined_data = f['stacked_frames'][:]  # Load all data at once

# ----------------------
# Minimum Group Size
# ----------------------
#Kmeans is used to segregate data into groups until the size of the smallest group contains less than the number of
#instances specified here, instances are then sampled equally from each of these groups to ensure smaller groups, which
#are more likely to contain interesting features obtain more representation in the SOM training.

min_instance = 2000  # Number of instances for smallest group

# ----------------------
# SOM Training Parameters
# ----------------------
X = 10                # SOM Size (X)
Y = 10               # SOM Size (Y)
samples_iter = 5000    # Number of instances per iteration
l_rate = 0.5           # Initial learning rate
burn_itter = 50        # Burn-in iterations
num_itter = 1000       # Learning iterations
```

Step 5 – Visualise Mineral Locations

1. Open the script Layer_Visuals.py located in the SOM_Analysis folder.
2. Many data sources need to be specified, all of which should be familiar from steps 1-4.
3. There are many choices under the section “Identify Spatial Distribution of a Specific Mineral” to specify.
4. The output is a false or bespoke image view of the Martian surface showing the identified spatial distribution of the nominated mineral.

Step 6 – SOM Hierarchical Analysis

1. Open the script SOM_analysis located in the SOM_Analysis folder.
2. Specify the output locations from Step 4.
3. Run the code to obtain comprehensive topological analysis of the SOM model.

Step 7 – KMeans Comparison

1. The SOM model must be trained first to run this code as the inputs reshapped_data_loc and reshapped_locations_loc are generated during this process.
2. Open the script kmeans.py located in the KMeans folder, specify data locations and mineral information before running the script.