

Appendix A

Download the Jupyter Notebook file called "DynamicPredictedMineralogyForDistribution.ipynb" here: <https://github.com/charraden/DynamicPredictedMineralogy.git>

Python package prerequisites (user to install): *pandas, numpy, pandarallel, matplotlib, seaborn, docplex, squarify, ipywidgets, and openpyxl*. The first block of the Jupyter notebook contains the pip install code for these packages (just run this cell). NOTE: DOCPLEX can be a bit tricky, but there is good documentation online to help troubleshoot.

This Jupyter notebook contains Python code to execute the predicted mineralogy using VNIR-SWIR and user-defined geological constraints to dynamically change the mineral library for the linear programming. The incoming data should have all of the bulk chemistry (+/- pXRF Si), and VNIR-SWIR mineralogy integrated for each sample into a single CSV. While this can be automated, variations in data exports and formats make universally applicable code challenging to provide. In this study, data integration was done manually, combining geochemistry and VNIR-SWIR data into a single CSV file used as the input for LP-based mineral predictions (this integrated datasheet can be downloaded in the supplemental materials).

To use this Jupyter Notebook:

- 1) Install the required packages by clicking inside the top code block, holding down the SHIFT key and hitting the ENTER key (this will execute the package installation). This only needs to be done once!

```
[ ]: 1 # INITIALIZE Packages (JUST DO THIS ONCE!)
2 pip install pandas
3 pip install numpy
4 pip install pandarallel
5 pip install matplotlib
6 pip install seaborn
7 pip install docplex
8 pip install squarify
9 pip install ipywidgets
10 pip install openpyxl
```

- 2) Then, check the mineral library (starting on line 46) to ensure that all minerals expected in the samples are included, all of the minerals reported in the VNIR-SWIR data are included, and that the chemistries for each mineral are fit-for-purpose. Minerals can be added (following the code conventions in this block), and chemistry of existing minerals can be modified here as well.
NOTE: if you do update the mineral library, go up to "Kernel" at the top and select "Restart Kernel". It will ask you if you are sure (you are). This ensures the mineral library changes you have made are recognized
- 3) Click inside the third block, holding down the SHIFT key and hitting the ENTER key (this will execute the code and a series of interactive widgets will appear at the bottom, below the block of code).

INFO: Pandarallel will run on 19 workers.
INFO: Pandarallel will use standard multiprocessing data transfer (pipe) to transfer data between the main process and workers.
WARNING: You are on Windows. If you detect any issue with pandarallel, be sure you checked out the Troubleshooting page:
<https://nalepae.github.io/pandarallel/troubleshooting/>

Dynamic Mineral Predictions

Mapping dropdowns populate after you upload a file.

Step 1: Upload your data (.csv or .xlsx)

Upload data (0)

Step 2: Set units & conversions (BE SURE TO CHECK CONVERSION OUTPUTS AFTER PROCESSING

Inputs are oxides (wt%)

Convert ppm→% ($\div 10^4$)

Step 3: Map columns to expected elemental names

Map your columns to expected fields

(appears after upload)

Optional: Map VNIR-SWIR columns

SWIR A (select after upload) ▾

SWIR B (select after upload) ▾

SWIR C (select after upload) ▾

Step 4: Choose static mineral library

Active mine... Albite
Analcime
Anhydrite
Ankerite
Apatite
Biotite
Calcite
Chalcopyrite
Clinochlore
Diaspore

Optional: Add geologic constraint rules

Rule column (select after upload) ▾
Remove mi... Albite
Analcime
Anhydrite
Ankerite
Apatite
Biotite
Calcite
Chalcopyrite

Operator equals ▾ Values (com...)

Add rule

Step 5: Plotting options

Residuals% ... 50 Exclude 100% residual rows

Step 6: Run LP & Export

Export CSV LP_output.csv

Run Mineral Predict...

- 4) Click the "Upload data" button, and select the integrated CSV which contains chemistry, VNIR-SWIR, and any other columns required to define geologically-significant rules. Click OK.
- 5) Set the units (if required). The code expects the values to be coming in as elemental wt.% - use these check boxes to convert oxides or ppm to elemental wt.%.
- 6) Map the columns from your CSV (available in the drop-down menus) to the expected elemental columns (these are populated from the mineral library, so if you add a new mineral with a new element, other elements will appear here).

Step 3: Map columns to expected elemental names

Map your columns to expected fields

Map your columns to expected fields

Al_pct	Al_pct
As_pct	As_pct
C_pct	C_pct
Ca_pct	Ca_pct
Cu_pct	Cu_pct
Fe_pct	Fe_pct
K_pct	K_pct
Mg_pct	Mg_pct
Mn_pct	Mn_pct
Mo_pct	Mo_pct
Na_pct	Na_pct
P_pct	P_pct
S_pct	S_pct
Si_pct	Si_pct
Ti_pct	Ti_pct

Al_pct	Al_pct
As_pct	(none)
C_pct	SampleID
Ca_pct	SampleType
Cu_pct	GeochemistrySource
Fe_pct	Lithology
K_pct	Al_pct
Mg_pct	Ca_pct
Mn_pct	Fe_pct
Mo_pct	K_pct
Na_pct	Mg_pct
P_pct	Mn_pct
S_pct	Na_pct
Si_pct	P_pct
Ti_pct	Si_pct
Optional: Map	
SWIR A	
Step 4: Choose	
Active mine...	
SWIR C	

7) Map the columns containing the VNIR-SWIR data as well.

Optional: Map VNIR-SWIR columns

SWIR A SWIR A SWIR B SWIR B SWIR C (none)

NOTE: if you don't link the SWIR columns, these minerals will not be considered dynamically (on a row-by-row basis).

- 8) Choose the static mineral library by highlighting them (you can select multiple minerals by holding down the CTRL key). These are the minerals that will be considered for **every sample/row** in the incoming data. DO NOT include the VNIR-SWIR active minerals you want to consider dynamically.

Step 4: Choose static mineral library

Active mine...

Albite
Analcime
Anhydrite
Ankerite
Apatite
Biotite
Calcite
Chalcopyrite
Clinochlore
Diaspore

- 9) Optional: add geologic constraints by selecting the rule column (perhaps lithology or magnetic susceptibility response), then use the operators to define the rule. When finished, click "Add rule" and the new rule will show below.

Optional: Add geologic constraint rules

Rule column	Lithology	Operator	equals	Values (com...)	basalt
Remove mi...	Pyrite Pyrophyllite Quartz Rutile Sanidine Siderite Tennantite Titanite	Add rule			

Added rule: if Lithology equals ['basalt'] -> remove ['Quartz']

- 10) Set the plotting options to show the chemistry of high residual samples. Default is to show chemistry for any row reporting >50% residual mineralogy.

Step 5: Plotting options

Residuals% ... 50 Exclude 100% residual rows

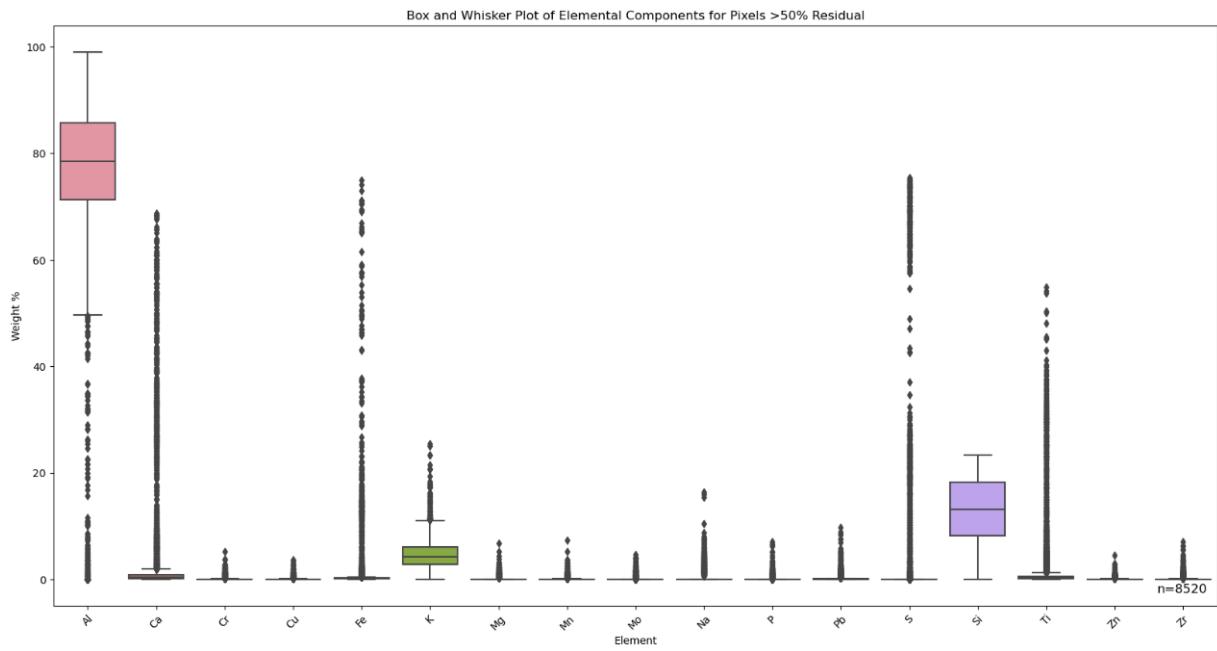
- 11) Create a filename for the output CSV (it will save to the same folder where the Jupyter Notebook is saved on your computer). Then click "Run LP & Export".

Step 6: Run LP & Export

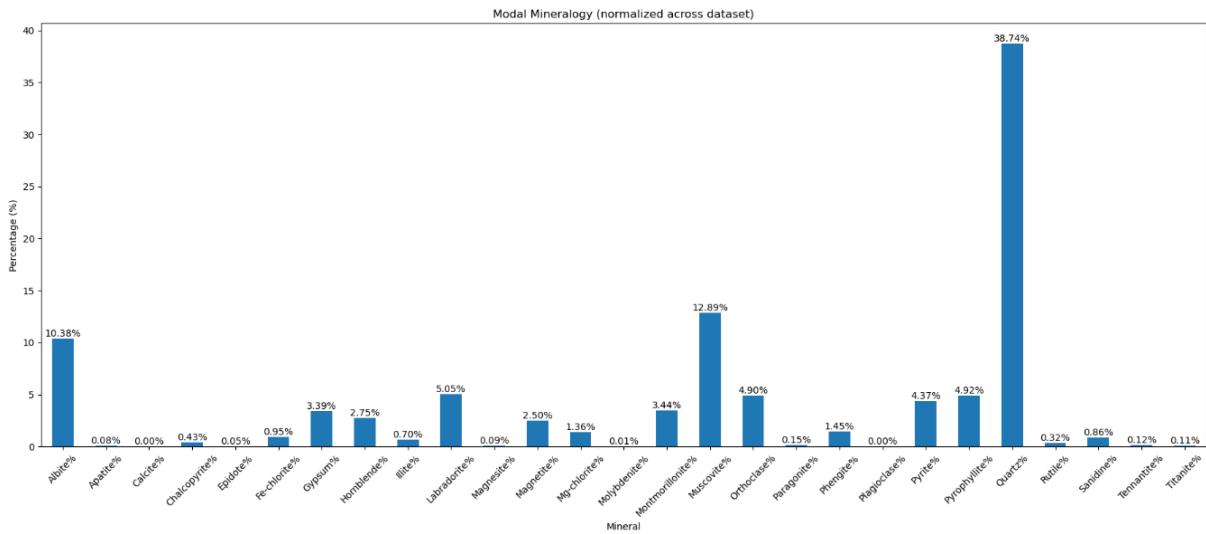
- 12) The code will execute, and print updates as it works through the predicted mineralogy. When it is finished, a message will appear that the new CSV was exported (with a file path).

- 13) Output plots will also appear, showing:

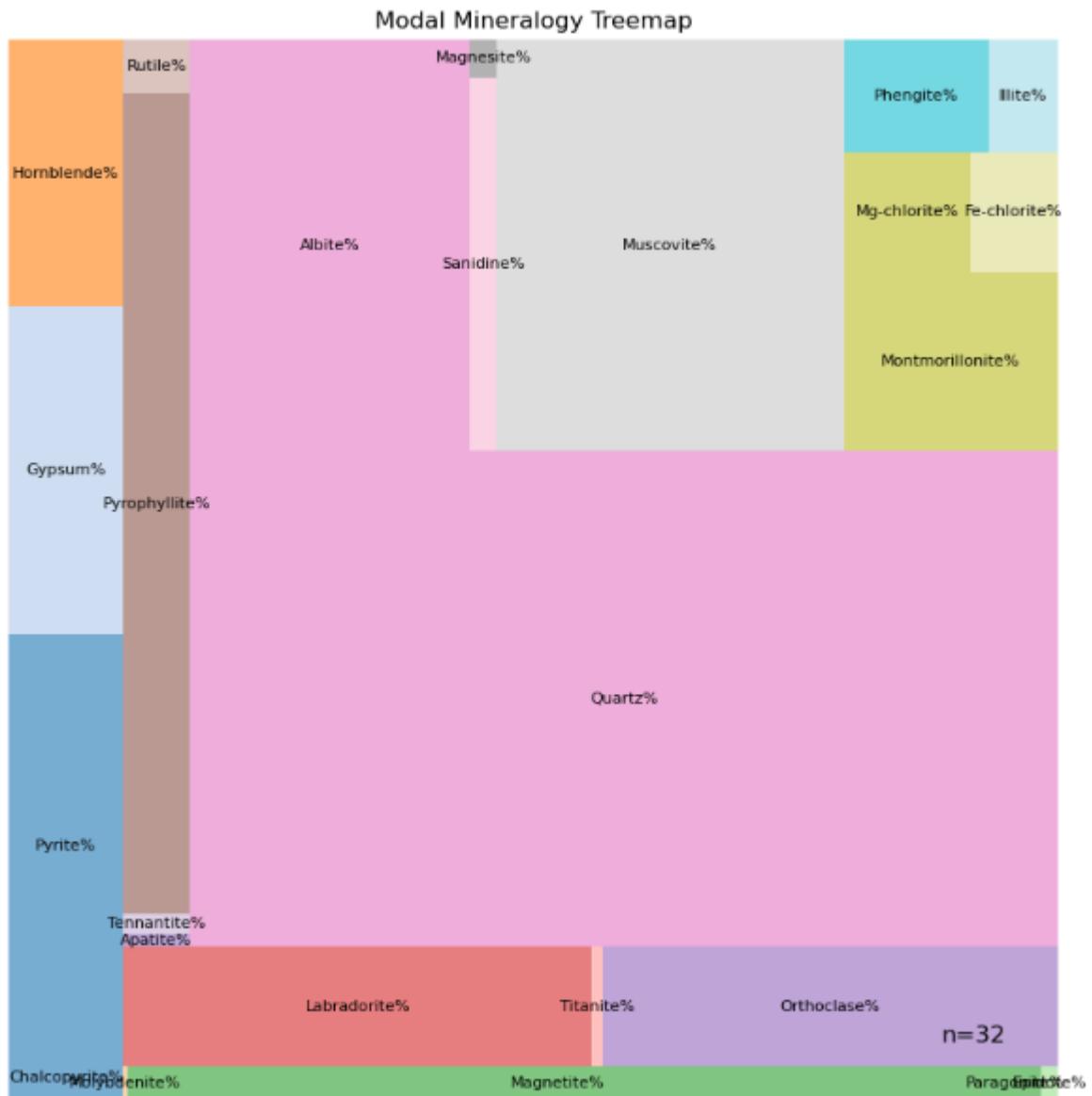
- a. A box and whisker plot showing the chemical distribution for any rows/samples with >50 (or whatever you set the threshold to) residual mineralogy. This is useful to determine if a mineral might be missing from the library.



b. Modal mineralogy across the entire incoming dataset as a bar plot



c. Modal mineralogy across the entire incoming dataset as a treemap



- 14) The output CSV now contains predicted modal mineral percentages reported in columns with the mineral names. The column "HighestProp" shows the highest abundance mineral reported for that row/sample.