Appendix A

Download the Jupyter Notebook file called "MicroXRF_CompileFromTXT.ipynb" here: https://github.com/charraden/MicroXRF-Geochemistry-Compiler

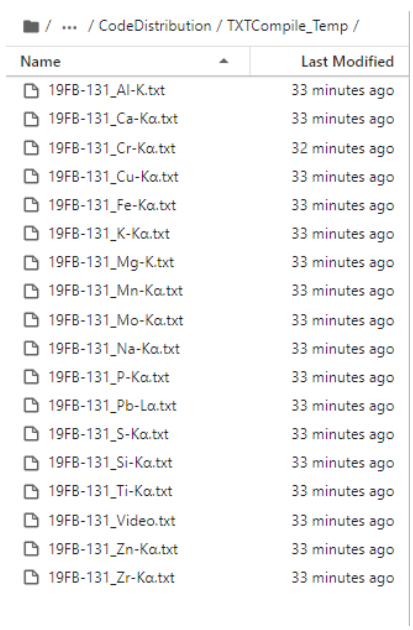
Python package prerequisites (user to install): pandas

This Jupyter notebook contains Python code to convert the multiple, elemental exported text files from the Bruker ESPRIT software into a single table assigning relative coordinates to preserve the spatial relationships of the pixels to one another. The data are collated into a wide-format geochemistry table where each row represents a single pixel and columns represent each element, indexed by the relative x and y coordinate of each pixel to preserve the spatial relationships and textural information. The Jupyter Notebook can handle and .txt outputs from ESPRIT (quantified or X-ray counts), assuming that each of the exported .txt file names contain the element the contained data represents (e.g. 19FB-131_Ca-Kα.txt contains Ca data for sample 19FB-131).

This step is required to produce the input data for the methods presented in this paper: linear programming, Random Forests, k-means clustering, and UMAP-HDBSCAN.

The methods in the paper were developed using µXRF images collected from a Bruker M4 TornadoPlus and data exports from Bruker ESPRIT. The instructions below are specific to this data/software, but could be modified for different data collection systems and software. To use this Jupyter Notebook for Bruker data:

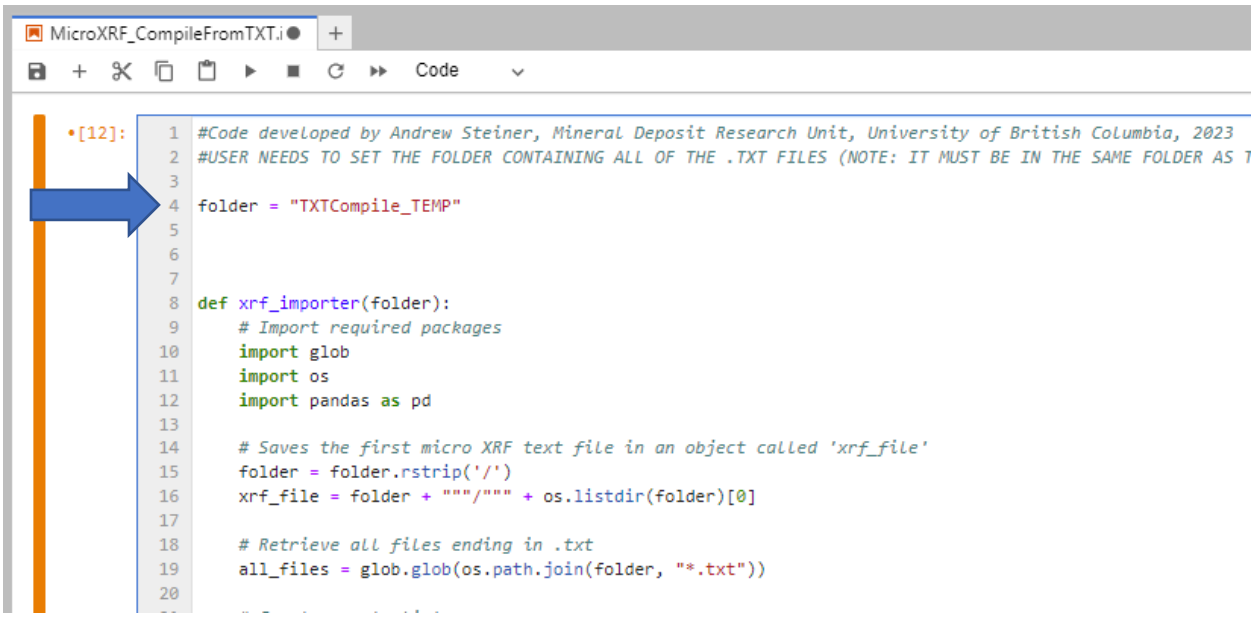
1) Create a new folder in the same folder where the file "MicroXRF_CompileFromTXT.ipynb" is saved. Name it something that makes sense for your project – we have called this folder "TXTCompile_Temp".
2) Copy all of the elemental .txt files for a single sample into this new folder. Note that this will include the Video file (single-band image of the sample) – the code can include this value if you would like it included in the final .csv output.



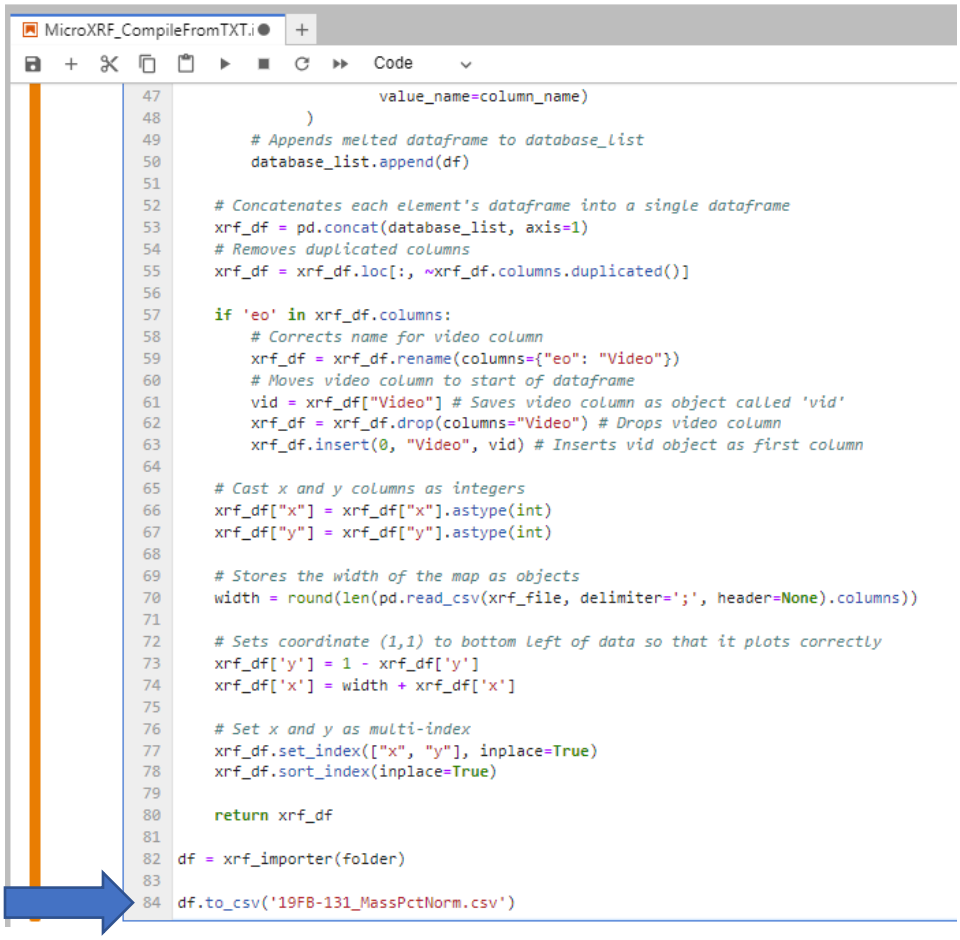
/ ··· / CodeDistribution / TXTCompile_Temp /

Name	Last Modified
19FB-131_Al-K.txt	33 minutes ago
19FB-131_Ca-Kα.txt	33 minutes ago
19FB-131_Cr-Kα.txt	32 minutes ago
19FB-131_Cu-Kα.txt	33 minutes ago
19FB-131_Fe-Kα.txt	33 minutes ago
19FB-131_K-Kα.txt	33 minutes ago
19FB-131_Mg-K.txt	33 minutes ago
19FB-131_Mn-Kα.txt	33 minutes ago
19FB-131_Mo-Kα.txt	33 minutes ago
19FB-131_Na-Kα.txt	33 minutes ago
19FB-131_P-Kα.txt	33 minutes ago
19FB-131_Pb-Lα.txt	33 minutes ago
19FB-131_S-Kα.txt	33 minutes ago
19FB-131_Si-Kα.txt	33 minutes ago
19FB-131_Ti-Kα.txt	33 minutes ago
19FB-131_Video.txt	33 minutes ago
19FB-131_Zn-Kα.txt	33 minutes ago
19FB-131_Zr-Kα.txt	33 minutes ago

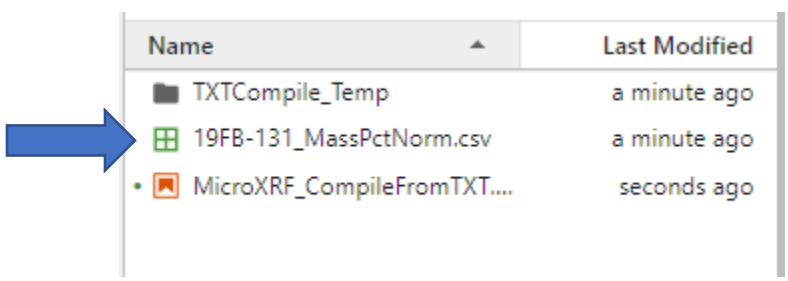
3) Open the "MicroXRF_CompileFromTXT.ipynb" Jupyter Notebook. Set the folder name (folder = code in line 4) to your new folder. Make sure the folder name is in quotes ("").
TIP: you can see the line numbers by clicking outside of the coding box once, holding down the SHIFT key on your keyboard, and hitting the "L" key on your keyboard.



4) Scroll to the bottom of the code (line 84) and give the output .csv file a name that makes sense for your project (suggest sample name + type of elemental data).



5) Click inside the code box once, then hold down the SHIFT key, and hit ENTER on your keyboard to run the code.
6) When the code is finished, the compiled .csv file will appear in the same folder as Jupyter Notebook.



The output .csv file now contains relative x and y coordinates of each pixel, the Video channel, and elemental values, with the element as the header name. This is the .csv file used in the supervised and unsupervised methods in the remaining appendices.

MicroXRF_CompileFromTXT.i × | 19FB-131_MassPctNorm.csv × | +

Delimiter: ,

	x	y	Video	Al	Ca	Cr	Cu	Fe
1	729	-415	65.971	0.0	4.3	0.0	0.058	0.36
2	729	-414	66.011	0.0	3.4	0.0	0.0	0.281
3	729	-413	64.68	0.0	3.699	0.0	0.119	0.201
4	729	-412	64.25	0.0	3.201	0.0	0.082	0.381
5	729	-411	63.899	0.0	4.3	0.0	0.076	0.339
6	729	-410	63.627	0.0	4.401	0.0	0.031	0.409
7	729	-409	63.554	0.0	3.9	0.0	0.067	0.299
8	729	-408	63.447	0.0	3.799	0.0	0.055	0.14
9	729	-407	65.855	0.0	4.199	0.0	0.052	0.281
10	729	-406	66.484	0.0	4.001	0.0	0.052	0.14
11	729	-405	65.791	0.0	4.3	0.0	0.015	0.22
12	729	-404	64.817	0.0	4.501	0.0	0.11	0.14
13	729	-403	64.063	0.0	3.9	0.0	0.052	0.32
14	729	-402	64.222	0.0	4.199	0.0	0.049	0.241
15	729	-401	64.396	0.0	4.901	0.0	0.0	0.421
16	729	-400	64.68	0.0	4.401	0.0	0.034	0.189
17	729	-399	64.204	0.0	4.199	0.0	0.0	0.339
18	729	-398	64.091	0.0	3.5	0.0	0.052	0.269
19	729	-397	64.228	0.0	4.001	0.0	0.0	0.339
20	729	-396	64.323	0.0	4.7	0.0	0.018	0.299
21	729	-395	64.393	0.0	4.599	0.0	0.0	0.259
22	729	-394	65.763	0.0	3.201	0.0	0.07	0.4
23	729	-393	65.193	0.0	4.199	0.0	0.0	0.311
24	729	-392	64.228	0.0	3.601	0.0	0.027	0.391
25	729	-391	64.1	0.0	3.699	0.0	0.0	0.339
26	729	-390	64.326	0.0	3.601	0.0	0.089	0.189
27	729	-389	65.461	0.0	4.501	0.0	0.119	0.159
28	729	-388	65.657	0.0	4.599	0.0	0.037	0.229
29	729	-387	65.791	0.0	4.7	0.0	0.131	0.339
30	729	-386	66.12	0.0	5.2	0.0	0.0	0.36
31	729	-385	66.383	0.0	3.601	0.0	0.049	0.18
32	729	-384	66.783	0.0	4.599	0.0	0.079	0.391
33	729	-383	66.783	0.0	4.501	0.0	0.018	0.33
34	729	-382	66.941	0.0	5.2	0.0	0.024	0.211
35	729	-381	67.564	0.0	4.001	0.0	0.055	0.119
36	729	-380	68.553	0.0	3.601	0.0	0.073	0.171