Data Preparation – G&E Samples

1) What this script does

This Python script is a small processing pipeline for raw SERS .txt exports:

1. Trims each raw .txt file to start right after a known header line ("Pixel;Wavelength;Wavenumber"), and saves the trimmed version to a folder named Trimmed/.
2. Filters the trimmed file to keep only two columns:
    - Raman Shift
    - Dark Subtracted #1
      starting from row 551 onward, then saves the result as a .csv in Filtered/.
3. Plots all filtered spectra on one overlay figure and saves it as SERS_Spectra_Overlay.png in the same folder as the script.
4. Converts each filtered .csv into a .npy file (NumPy array) containing only the intensity signal (Dark Subtracted #1), shaped as (1, N).
5. Splits the .npy outputs into Training vs Testing folders based on the filename ending:
    - If the filename ends with R1_1 … R3_7 (i.e., index 1–7) → TestingData
    - If it ends with R1_8 … R3_36 (i.e., index 8–36) → TrainingData

Finally, it prints a short summary (how many files and an example shape per folder).

---

2) What the script expects from the user

Input files

- Put all raw input files (*.txt) in the same directory as this script.
- The script only processes files matching *.txt in the current working directory.
- Each .txt file must contain the line:
    - Pixel;Wavelength;Wavenumber
      If it is missing, that file is skipped.

Filename requirement for Train/Test split

For the .npy conversion and split, each file name (without extension) must end with:

- R1_<number> or R2_<number> or R3_<number>

Examples of valid endings:

- ...R1_1, ...R2_7, ...R3_12, ...R1_36

If the file does not end with that pattern, it will be skipped for .npy generation.

---

3) What folders/files it creates

In the script directory it creates:

- Trimmed/ (trimmed .txt outputs)
- Filtered/ (filtered .csv outputs)
- TrainingSet_tensors/ (created for compatibility; typically unused for saving here)
- SERS_Spectra_Overlay.png

In addition, it saves .npy outputs into these fixed absolute paths (important):

- Testing .npy files:
  /home/elhamm/links/projects/def-qianl/elhamm/AI-CysConcentrations-SERS/Real-SERS-Data/TestingData
- Training .npy files:
  /home/elhamm/links/projects/def-qianl/elhamm/AI-CysConcentrations-SERS/Real-SERS-Data/TrainingData

If those paths do not exist, the script will create them.

That path is specific to the original author's computer. On your computer, you must replace it with your own local path (where you want the output folders created).

Also, the saved .npy files will start with a cultivar name prefix. In the code, this prefix is currently:
• AAC_Liscard_

This is controlled by the line inside convert_all_csv_to_npy():
• npy_filename = "AAC_Liscard_" + stem + ".npy"

If you are processing a different cultivar, you must change "AAC_Liscard_" to the correct cultivar name, so the output filenames match your cultivar (for example "AAC_Chrome_", "CDC_Athabasca_", etc.).

---

4) How to run it on a local computer

Step A: Install Python requirements

You need Python 3 plus these packages:

- numpy
- pandas
- matplotlib

Install them with:

pip install numpy pandas matplotlib

Step B: Place files correctly

1. Put the script (e.g., DataPreparation.py) in a folder.
2. Copy all raw *.txt SERS files into that same folder.

Step C: Run the script

From that folder:

python DataPreparation.py

Or in JupyterHub, simply run the .ipynb code:

DataPreparation. ipynb

---

5) How to interpret the outputs

- Trimmed files: Trimmed/*.txt
  These are the raw files cut to start after the header line.
- Filtered files: Filtered/*.csv
  Each CSV has exactly two columns: Raman Shift and Dark Subtracted #1 (from row 551 onward).
- Overlay plot: SERS_Spectra_Overlay.png
  One figure with all filtered spectra overlaid.
- Model-ready NumPy arrays:
  - Testing arrays in .../TestingData/
  - Training arrays in .../TrainingData/
    Each .npy file contains the intensity vector expanded to shape (1, N).

The script prints how many .npy files were saved to each folder and notes any skipped files.