Guidelines for TOF-SIMS-Based Electrode Microstructure Segmentation and Characterization

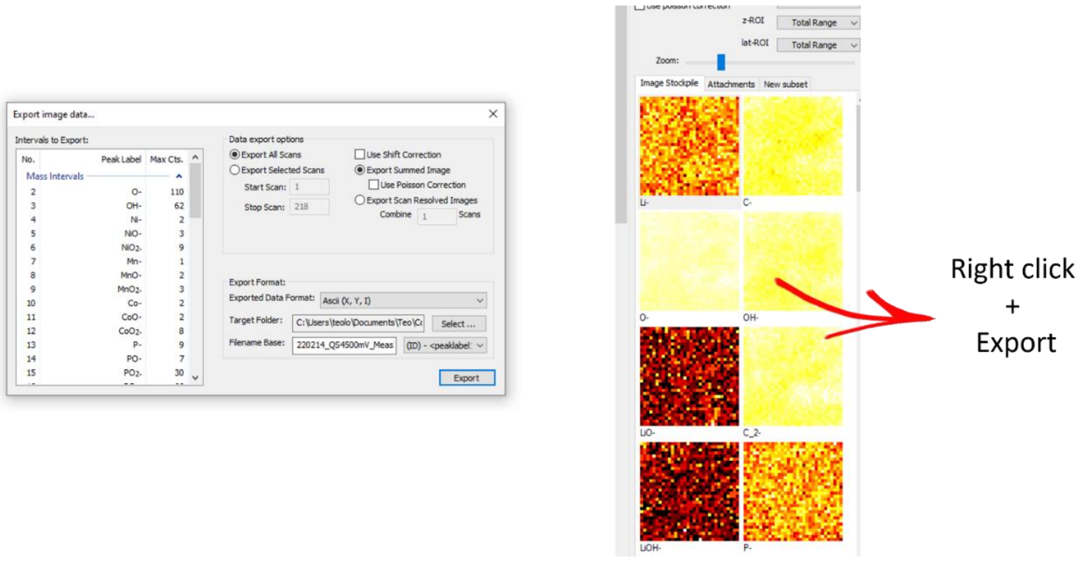
Scope

This code allows combining the spatial distribution coming from multiple secondary ions maps obtained through time-of-flight secondary ion mass spectrometry (ToF-SIMS) into a single image and segmenting it (*i.e.*, every pixel is assigned to one SI only). The SI should be selected as a fingerprint of a given (inter)phase of the battery electrode microstructure being analyzed. If this is the case, every pixel assigned to a given SI is also assigned to the (inter)phase of interest. In addition, the segmented image is analyzed to determine the overall volume fractions and interfaces, as well as the single particle/agglomerate size distribution, aspect ratio, and interfaces.

Data Exporting

The procedure needed to export the image data in a readable format ("Ascii") is depicted in Figure 1.

This procedure, and the associated reading section of the code, were developed for the data format exportable through SurfaceLab (the official software from IONTOF - https://www.iontof.com/). If your ToF-SIMS was purchased from a different provider, you will need to adapt the data exporting and reading procedures.



Export image data...

Intervals to Export:

No.	Peak Label	Max Cts.
Mass Intervals		
2	O-	110
3	OH-	62
4	Ni-	2
5	NiO-	3
6	NiO2-	9
7	Mn-	1
8	MnO-	2
9	MnO2-	3
10	Co-	2
11	CoO-	2
12	CoO2-	8
13	P-	9
14	PO-	7
15	PO2-	30

Data export options

[✓ Export All Scans]
Export Selected Scans
Start Scan: 1
Stop Scan: 218
Use Shift Correction
[✓ Export Summed Image]
Use Poisson Correction
Export Scan Resolved Images
Combine 1 Scans

Export Format:

Exported Data Format: Ascii (X, Y, I)
Target Folder: C:\Users\teolo\Documents\Teo\C
Select ...
Filename Base: 220214_QS4500mV_Meas
(ID) - <peaklabel
Export

Figure 1. Schematics of the procedure to be followed in SurfaceLab (IONTOF) to export all the images associated with a given measurement in a data readable format ("Ascii").

Code’s Inputs

The code’s inputs (Python - Jupyter Notebook) are reported and briefly discussed below (for each input, the comment “#Input” is reported at the same line in the code) in order of appearance:

POI: Here you should indicate the name of the SI that you want to retrieve to make your segmented image. Please note that the name here should be unique (if you have two SI having in common the indicated string/part of the name, the code will consider the last one in alphabetic order). The name of the identified SI files (in order of appearance) is printed to check that the correct ones have been retrieved.

Mains: Here report the name of the main phases (*i.e.*, not the interphases).

FoV: Here report the field of view of the image in micro meters. This parameter is needed to calculate the pixel resolution.

Threshold_cc: Here report the pixel intensity threshold to be used to eventually remove the current collector from the analysis. If this does not work in your case, there will be the possibility to crop the image afterwards.

Factor_l: To enhance the lightening of the retrieved image (for the moment this option is disabled and the parameter value kept to 0 to avoid using it – but you can uncomment the associated lines and try it, if you think it could be useful for your case).

Angle_r: Rotate the image of a given angle.

Manual_cut: Cut the image automatically (0) or manually (1). If 1 is selected, you can define below the upper (“limit_up”) and lower (“limit_down”) pixel limits for the cutting.

Thresold_seg: The relative intensity (in 8-bit values, from 0 to 255) threshold for the image segmentation (to remove noise).

Factor_advantage_AM: Advantage factor over segmentation (if >1) or disadvantage (if <1) for the AM phase (the first one indicated in “POI”). This can be used if the accurate retrieval of a given (inter)phase has the priority in your analysis.

Thresold_binary: The threshold to be used for generating the binary images (one for each (inter)phase of interest). Binary segmentation is needed for the watershed-based procedure to identify the single particles/agglomerates. The current code was designed to account for three phases here (but you can modify it to account for more or less as a function of your specific needs).

Object recognition section: In this section, you find multiple parameters (each one commented in the code) for the watershed-based algorithm. More information on this procedure can be found here: https://youtu.be/M1mJsJ5M4iE. The code for single particles recognition is reported for the first phase only (the first one indicated in POI), but it can be applied to any others (changing the naming to other phases).

Results

An example of results can be found in the Example folder, and a general overview is reported in Figure 2.

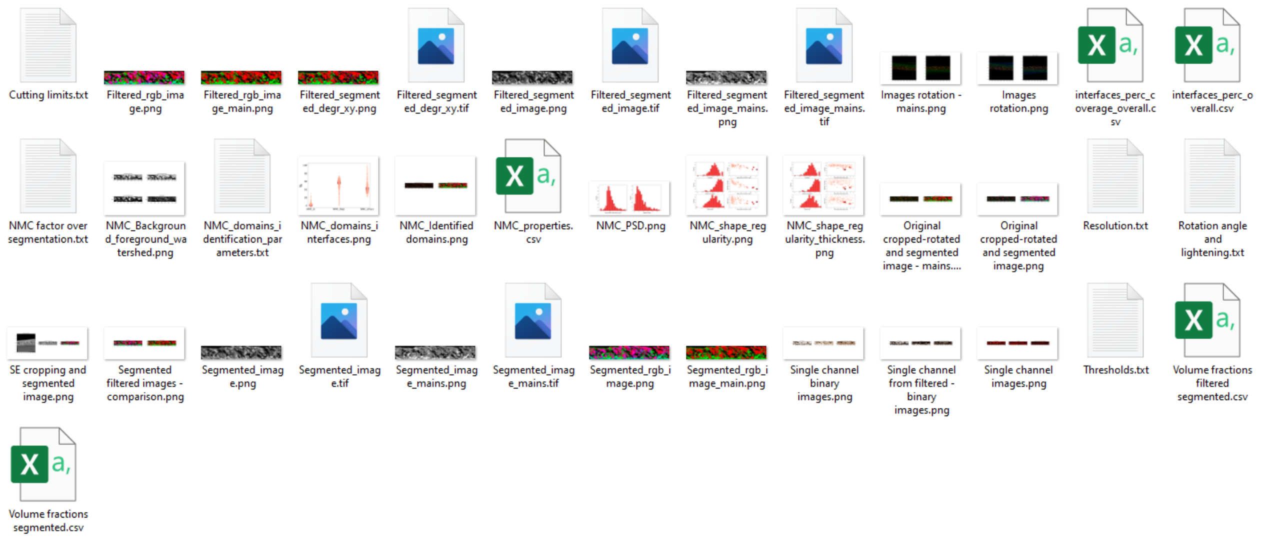

Cutting limits.txt
Filtered_rgb_image.png
Filtered_rgb_image_main.png
Filtered_segmented_degr_xy.png
Filtered_segmented_degr_xy.tif
Filtered_segmented_image.png
Filtered_segmented_image.tif
Filtered_segmented_image_mains.png
Filtered_segmented_image_mains.tif
Images rotation - mains.png
Images rotation.png
interfaces_perc_coverage_overall.csv
interfaces_perc_overall.csv
NMC factor over segmentation.txt
NMC_Background_foreground_watershed.png
NMC_domains_identification_parameters.txt
NMC_domains_interfaces.png
NMC_Identified domains.png
NMC_properties.csv
NMC_PSD.png
NMC_shape_regularity.png
NMC_shape_regularity_thickness.png
Original cropped-rotated and segmented image - mains....
Original cropped-rotated and segmented image.png
Resolution.txt
Rotation angle and lightening.txt
SE cropping and segmented image.png
Segmented filtered images - comparison.png
Segmented_image.png
Segmented_image.tif
Segmented_image_mains.png
Segmented_image_mains.tif
Segmented_rgb_image.png
Segmented_rgb_image_main.png
Single channel binary images.png
Single channel from filtered - binary images.png
Single channel images.png
Thresholds.txt
Volume fractions filtered segmented.csv
Volume fractions segmented.csv

Figure 2. Example of results (NMC case).

These results can be directly used in SliceGAN (for instance the RGB image “filtered_segmented_degr_xy.png”) using “colour” as image type (“image_type” parameter in the SliceGAN code). The RGB image can also be converted to a more classically segmented image (0,1, 2, *etc.* - instead of pure red [255,0,0], green [0, 255,0], or blue [0,0,255]) and use this as image type in SliceGAN. The SliceGAN code and more information on how to use it can be found here: https://github.com/stke9/SliceGAN

Contact for problems/doubts

If you have any problem using the code, feel free to contact me on my personal email: teo.lombardo3@gmail.com

You can also reach me on LinkedIn (https://www.linkedin.com/in/teo-lombardo-6626aa173) or Twitter (https://twitter.com/TeoLomb).