

Contents

I	Results	2
1.1	Introduction	2
1.2	Initial quantification	2
1.3	Analysis steps in HyperSpy	2
1.3.1	Loading the data and specifying the elements	3
1.3.2	Removing the background linearly	3
1.3.3	Quantification after linear background removal	3
1.4	Calibration	4
1.5	Peak and background modelling	4
1.6	Background models	4
1.7	Analysis failure	4

Chapter 1

Results

1.1 Introduction

The results are presented in this chapter. The sections follow the structure of the sub-problems in ??.

1.2 Initial quantification

The initial quantification was done on the data from the GaAs wafer in AZtec and in HyperSpy as out-of-the-box as possible. The results are presented in Table 1.1. The wafer is a 1:1 alloy of gallium and arsenic, so the atomic percent of Ga and As should be 50% and 50% respectively.

Table 1.1: Initial quantification of the GaAs wafer. The ratio in the wafer is 1:1, so the correct ratio is 50% and 50%, because the results are in atomic percent.

V_{acc}	AZtec		HyperSpy	
	Ga	As	Ga	As
5 kV	50 %	50 %	50 %	50 %
10 kV	50 %	50 %	50 %	50 %
15 kV	50 %	50 %	50 %	50 %
30 kV	50 %	50 %	50 %	50 %

1.3 Analysis steps in HyperSpy

(Question for Ton: Is this interesting to write about?)

(Question for Ton: This section is both presenting what and why we do the steps. That is leaning into the discussion. Should I separate it?)

The next sub-problem was to find out what is done with the data at the different steps in the analysis when using HyperSpy. In these steps it is assumed that the user have done qualitative analysis and want to do quantitative analysis on a set of elements. The analysis in AZtec is done as a black box, so it is not possible to see what is done with the data at the different steps. All variables inside crocodile need to be set by the user, e.g. `<element_list>` would be set to `['Ga', 'As']` for the GaAs wafer. An example notebook with quantification of the GaAs wafer is attached in APPENDIX. **(Brynjar: Make a notebook with GaAs quantification in HyperSpy, with the data somehow.)**

1.3.1 Loading the data and specifying the elements

```
s = hs.load(<filepath>, signal="EDS_TEM")  
  
s.set_elements(<element_list>)
```

The first step in the analysis is to load the data as a HyperSpy `signal` type, and specifying the signal as TEM. The `signal` type is a class in HyperSpy that contains the data and the metadata, and it has methods for analysis. The `signal` type must be specified as TEM, because the `signal` type for SEM is very limited and does not have a method for quantification. When using .emsa files from AZtec, as is done in this project, the metadata contains some relevant and some irrelevant information. The information relevant later in this project is: acceleration voltage, dispersion, zero offset, energy resolution Mn K α . After loading, it is possible to plot the data with `s.plot()`.

1.3.2 Removing the background linearly

```
bw = s1.estimate_background_windows(windows_width=<number>)  
  
iw = s1.estimate_integration_windows(windows_width=<number>)
```

The next step is to remove the background, which with the above code is done by a linear fit. The background can be removed through model fitting, which is covered in ???. The variable `windows_width` sets how wide the windows are for the background and integration, measured in FWHMs. A good value for `windows_width` is 2, but it should be tested by the user with a plot. The estimated windows can be plotted with:

```
s.plot(xray_lines=True, background_windows=bw, integration_windows=iw)
```

1.3.3 Quantification after linear background removal

```
s_i = s.get_lines_intensity(background_windows=bw, integration_windows=iw)  
  
k_factors = [<k-factor 1>, <k-factor 2>]  
  
quant = s.quantification(s_i, method='CL', factors=k_factors)
```

```
print(f'E1: {quant[0].data[0]:.2f} \%, E1: {quant[1].data[0]:.2f} \%' )
```

The quantification is done with the four lines of code above, where the last one prints the results. The first line gets the intensity of the peak corresponding to the lines of the specified element. HyperSpy selects automatically which lines to use for quantification. To see which lines are used, the `s_i` variable can be printed. The second line sets the k-factors. The k-factors in this project have been the one from AZtec, which are theoretically estimated. The third line does the quantification, where the method is specified.

1.4 Calibration

The next sub-problem was to calibrate the data with a self produced Python script.

1.5 Peak and background modelling

The next sub-problem was to find out how the peaks and the background are modelled in a way that is easy to understand.

1.6 Background models

The next sub-problem was to find out how different background models affect the quantitative analysis done in HyperSpy.

1.7 Analysis failure

The next sub-problem was to find out when the analysis fails, both in AZtec and HyperSpy.

Bibliography