1e_eds

November 28, 2022

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import hyperspy.api as hs

%matplotlib qt5
```

[2]:
```
s = hs.load("datasett/eds_linescan_data_1.hspy")
s2 = hs.load("datasett/eds_linescan_data_2.hspy")
s
```

[2]:
```
<EDSSEMSpectrum, title: SEM-EDS linescan, dimensions: (49|1500)>
```

[3]:
```
s.metadata
```

[3]:
```
    Acquisition_instrument
        SEM
            Detector
                EDS
                    azimuth_angle = 0.0
                    elevation_angle = 35.0
                    energy_resolution_MnKa = 130.0
            Stage
                tilt_alpha = 0.0
    General
        FileIO
            0
                hyperspy_version = 1.7.3
                io_plugin = hyperspy.io_plugins.hspy
                operation = save
                timestamp = 2022-11-25T18:42:24.619150+01:00
            1
                hyperspy_version = 1.7.2
                io_plugin = hyperspy.io_plugins.hspy
                operation = load
                timestamp = 2022-11-28T12:01:00.742175+01:00
        title = SEM-EDS linescan
    Signal
        signal_type = EDS_SEM
```

[4]:
```
s.plot()
# Observerte linjer sortert etter intensitet i slice 39
# 1.74 Si
# 0.52 O
# 8.45
# 8.01
# 1.50 (Al?)
# 4.50
# 9.69

# Fra autodetect
# ['Si_Ka', # Rimelig
#  'Rb_Lb1', # Usannsynlig
#  'Ta_Mb', # Usannsynlig
#  'W_Ma', # Usannsynlig, men passer veldig bra
#  'Ta_Ma', # Usannsynlig
#  'Hf_Mb', # Usannsynlig
#  'Rb_La', # Usannsynlig
#  'Sr_La', # Usannsynlig
#  'W_Mb', # Passer som sagt bra
#  'Si_Kb', # Rimelig
#  'Re_Ma'] # Usannsynlig
```

[5]:
```
s.set_elements(["Si", "O", "Al"])#, "W"])
s.plot(xray_lines=True)
```

[6]:
```
# Autodetect trial
P = s.find_peaks1D_ohaver(maxpeakn=1)[0]
hs.eds.get_xray_lines_near_energy(P['position'], only_lines=["a", "b"])
```

```
[########################################] | 100% Completed | 334.65 ms
```

[6]:
```
['Si_Ka',
 'Rb_Lb1',
 'Ta_Mb',
 'W_Ma',
 'Ta_Ma',
 'Hf_Mb',
 'Rb_La',
 'Sr_La',
 'W_Mb',
 'Si_Kb',
 'Re_Ma']
```

[46]:
```
s_sum = s.sum()
s_sum.plot(xray_lines=True, integration_windows="auto", only_lines=["Ka", "La"])
```

0.0.1 Tanker om datasett 1

Silisium kommer av prøven, Oksygen kommer av prøven. Aluminium kommer nok fra SEM-en selv. Wolfram finnes sikkert ikke, selv om det passer sykt bra med grafen

0.1 Prøve 2

[8]:
```
s2.plot()
# Observerte linjer
# 1.74 Si
# 0.52 O
# 0.25 C
# 9.45 Ga?
# 8.07 Cu?
# 11.12

# 2.12 ??? P????
```

[9]:
```
s2.set_elements(["Si", "O", "C"])
s2.plot(xray_lines=True)
```

[26]:
```
# Autodetect trial
P = s2.find_peaks1D_ohaver(maxpeakn=1)[0]
hs.eds.get_xray_lines_near_energy(P['position'], only_lines=["a", "b"])
# Bare tullball her
```

```
[########################################] | 100% Completed | 370.84 ms
```

```
  Input In [26]
    hs.eds.get_xray_lines_near_energy(P['position'], only_lines=["a", "b"], ␣
 ↪only_lines=["Ka", "La"])
                                                                             ^

SyntaxError: keyword argument repeated: only_lines
```

[45]:
```
s2_sum = s2.sum()
s2_sum.plot(xray_lines=True, integration_windows="auto",  only_lines=["Ka",␣
 ↪"La"])
```

0.1.1 Tanker om prøve 2

Hva i alle dager er den peaken på 2.12, og hvorfor forsvinner Silisium???

[23]:
```
s_sum.add_xray_lines_markers(xray_lines=s_sum.metadata.Sample.elements)
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
```

```
c:\Users\ivism\Progging\skole\5sem\tools\eksamen\1e_eds.ipynb Cell 14 in <cell␣
 ↪line: 1>()
----> <a href='vscode-notebook-cell:/c%3A/Users/ivism/Progging/skole/5sem/tools/
 ↪eksamen/1e_eds.ipynb#X23sZmlsZQ%3D%3D?line=0'>1</a> s_sum.
 ↪add_xray_lines_markers(xray_lines=s_sum.metadata.Sample.elements)

File c:
 ↪\Users\ivism\AppData\Local\Programs\Python\Python310\lib\site-packages\hyperspy\_signals\ed
 ↪py:1025, in EDSSpectrum.add_xray_lines_markers(self, xray_lines, render_figure)
   1015 """
   1016 Add marker on a spec.plot() with the name of the selected X-ray
   1017 lines
   (…)
   1022     A valid list of X-ray lines
   1023 """
   1024 if self._plot is None or not self._plot.is_active:
-> 1025     raise RuntimeError("The signal needs to be plotted.")
   1027 # in case of log scale, if some lines have intensity zero, then
   1028 # the line and label will not be displayed.
   1029 norm = self._plot.signal_plot.ax_lines[0].norm

RuntimeError: The signal needs to be plotted.
```

0.2 Pent

[56]:
```
fig, (ax1, ax2) = plt.subplots(2)
hs.plot.plot_spectra(s_sum, ax=ax1, fig=fig)
hs.plot.plot_spectra(s2_sum, ax=ax2, fig=fig)
ax1.annotate("a", xy=(0.005, 0.85), xycoords="axes fraction", fontsize="20",␣
 ↪color="red")
ax2.annotate("b", xy=(0.005, 0.85), xycoords="axes fraction", fontsize="20",␣
 ↪color="red")
fig.canvas.draw()
fig.savefig("1i/eds_begge", dpi=300)
```