**Supplemental Data**

**Supplemental Figure 1.** Hyperspy Code for FTIR Top View (referred to as “Aerial View”)

*# # HyperSpy Code for FTIR Aerial View*

*# ## Loading tools and libraries*

get\_ipython().run\_line\_magic('matplotlib', 'qt')

**import** **hyperspy.api** **as** **hs**

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

*# ## Reading Thermo OMNIC<sup>®</sup> FTIR data files*

**def** load\_omnic(filename, signal\_title, save=False):

**import** **os**

**import** **hyperspy.api** **as** **hs**

**from** **PyMca5.PyMcaIO** **import** OmnicMap

omnic\_file = OmnicMap.OmnicMap(filename)

reversed\_data = omnic\_file.data[::-1,:,::-1] *# Need to reverse y and energy dims*

sig = hs.signals.Signal1D(reversed\_data)

sig.axes\_manager[0].name = 'x'

sig.axes\_manager[0].offset = 0

sig.axes\_manager[0].scale = omnic\_file.info['OmnicInfo']['Mapping stage X step size']

sig.axes\_manager[0].units = '${\mu}m$'

sig.axes\_manager[1].name = 'y'

sig.axes\_manager[1].offset = 0

sig.axes\_manager[1].scale = omnic\_file.info['OmnicInfo']['Mapping stage Y step size']

sig.axes\_manager[1].units = '${\mu}m$'

sig.axes\_manager[2].name = 'Wavenumber'

sig.axes\_manager[2].offset = omnic\_file.info['OmnicInfo']['Last X value']

sig.axes\_manager[2].scale = -1 \* omnic\_file.info['OmnicInfo']['Data spacing']

sig.axes\_manager[2].units = '1/cm'

sig.metadata.General.title = signal\_title

**if** save:

sig.save(os.path.splitext(filename)[0] + '.hspy')

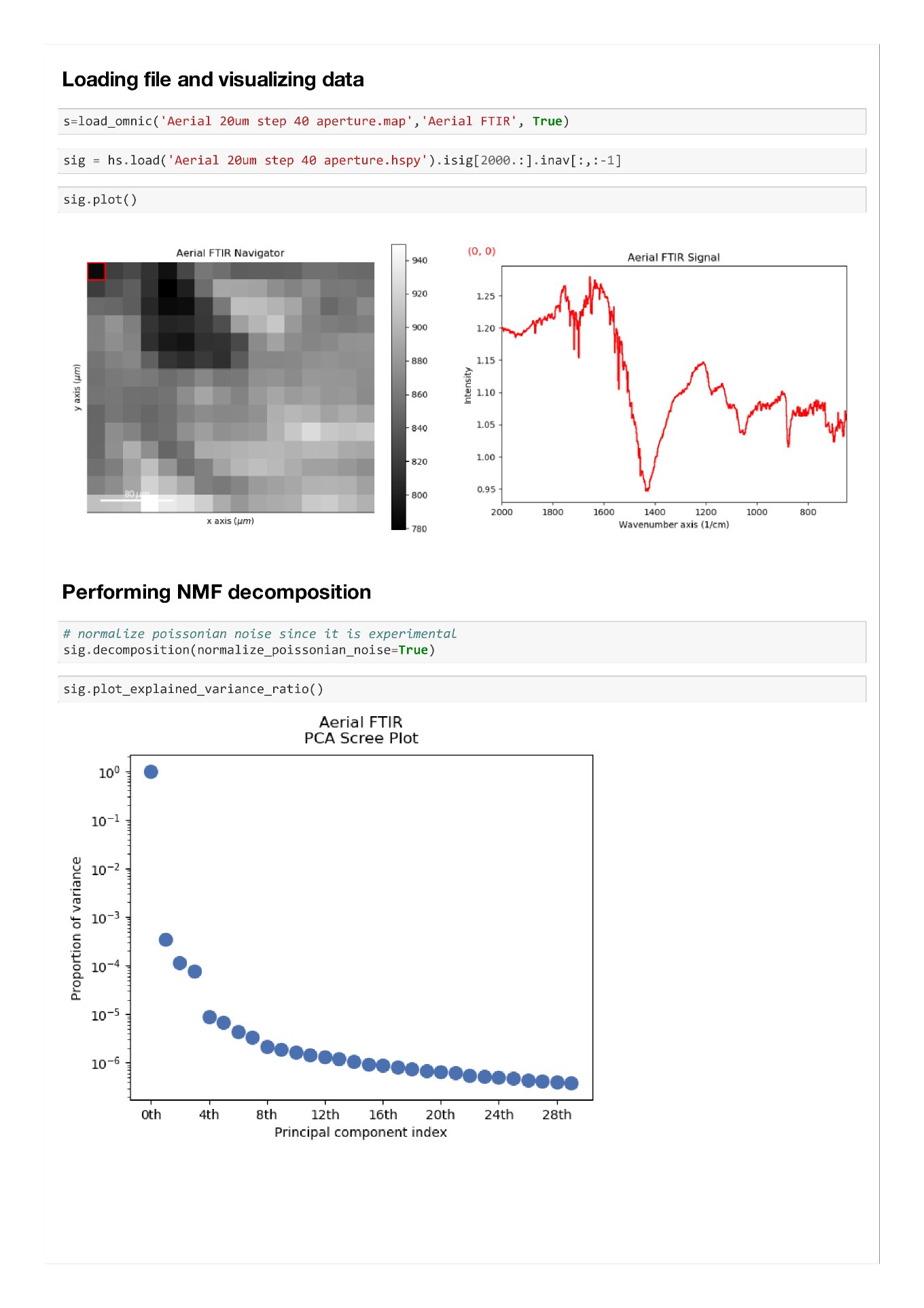
**return** sig

*# ## Loading file and visualizing data*

s=load\_omnic('Aerial 20um step 40 aperture.map','Aerial FTIR', True)

sig = hs.load('Aerial 20um step 40 aperture.hspy').isig[2000.:].inav[:,:-1]

sig.plot()

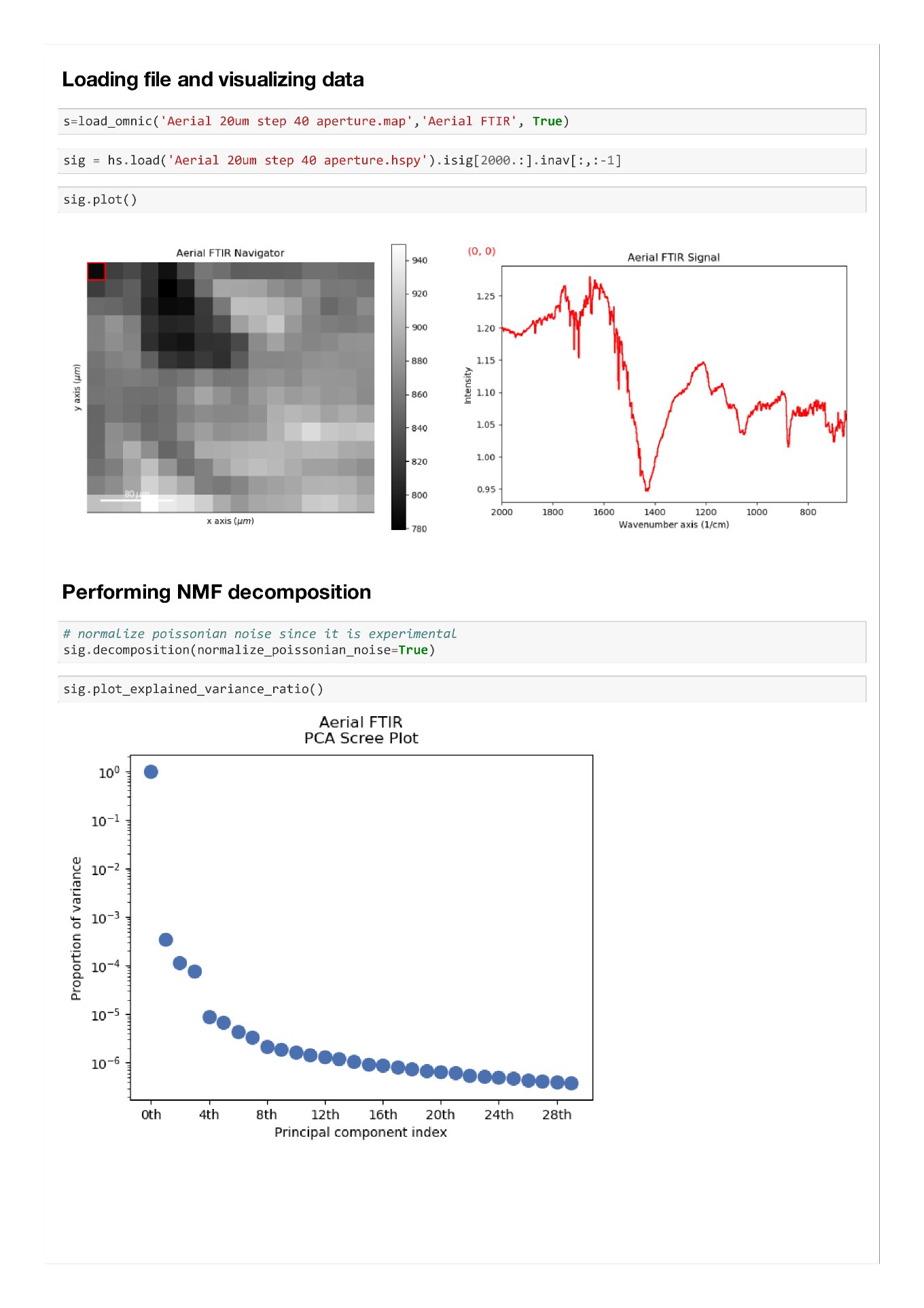
****

*# ## Performing NMF decomposition*

*# normalize poissonian noise since it is experimental*

sig.decomposition(normalize\_poissonian\_noise=True)

sig.plot\_explained\_variance\_ratio()

****

*# ## Performing NMF decomposition*

*# normalize poissonian noise since it is experimental*

sig.decomposition(normalize\_poissonian\_noise=True)

sig.plot\_explained\_variance\_ratio()

sig.decomposition(True, algorithm='nmf', output\_dimension=3)

loadings = sig.get\_decomposition\_loadings()

hs.plot.plot\_images(loadings, cmap='viridis',

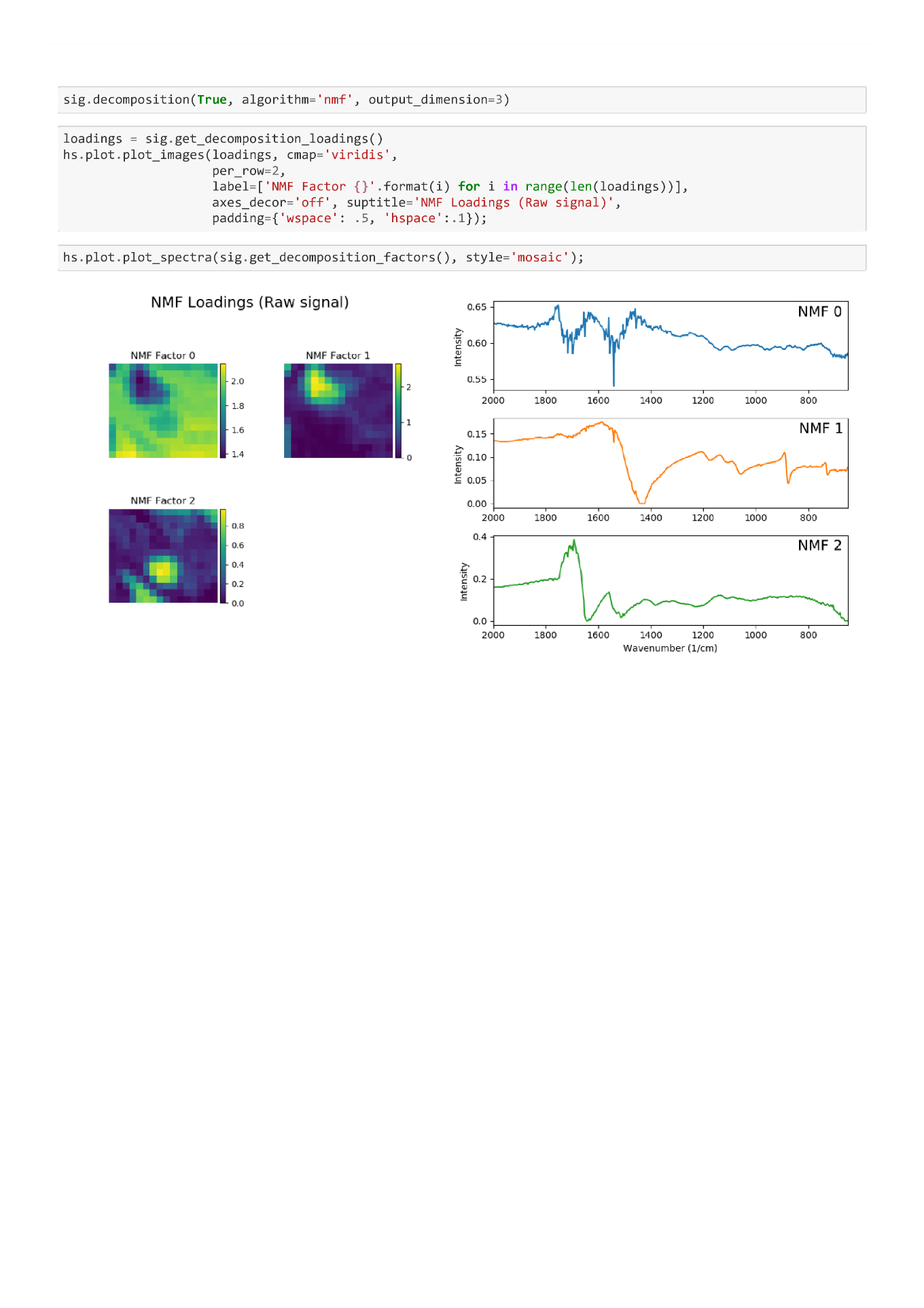
per\_row=2,

label=['NMF Factor {}'.format(i) **for** i **in** range(len(loadings))],

axes\_decor='off', suptitle='NMF Loadings (Raw signal)',

padding={'wspace': .5, 'hspace':.1})

hs.plot.plot\_spectra(sig.get\_decomposition\_factors(), style='mosaic')

****

**Supplemental Figure 2.** Hyperspy Code for FTIR Cross-sectional View

*# # HyperSpy Code for FTIR Cross-Sectional View*

*# ## Loading tools and libraries*

get\_ipython().run\_line\_magic('matplotlib', 'qt')

**import** **hyperspy.api** **as** **hs**

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

*# ## Reading Thermo OMNIC<sup>®</sup> FTIR data files*

**def** load\_omnic(filename, signal\_title, save=False):

**import** **os**

**import** **hyperspy.api** **as** **hs**

**from** **PyMca5.PyMcaIO** **import** OmnicMap

omnic\_file = OmnicMap.OmnicMap(filename)

reversed\_data = omnic\_file.data[::-1,:,::-1] *# Need to reverse y and energy dims*

sig = hs.signals.Signal1D(reversed\_data)

sig.axes\_manager[0].name = 'x'

sig.axes\_manager[0].offset = 0

sig.axes\_manager[0].scale = omnic\_file.info['OmnicInfo']['Mapping stage X step size']

sig.axes\_manager[0].units = '${\mu}m$'

sig.axes\_manager[1].name = 'y'

sig.axes\_manager[1].offset = 0

sig.axes\_manager[1].scale = omnic\_file.info['OmnicInfo']['Mapping stage Y step size']

sig.axes\_manager[1].units = '${\mu}m$'

sig.axes\_manager[2].name = 'Wavenumber'

sig.axes\_manager[2].offset = omnic\_file.info['OmnicInfo']['Last X value']

sig.axes\_manager[2].scale = -1 \* omnic\_file.info['OmnicInfo']['Data spacing']

sig.axes\_manager[2].units = '1/cm'

sig.metadata.General.title = signal\_title

**if** save:

sig.save(os.path.splitext(filename)[0] + '.hspy')

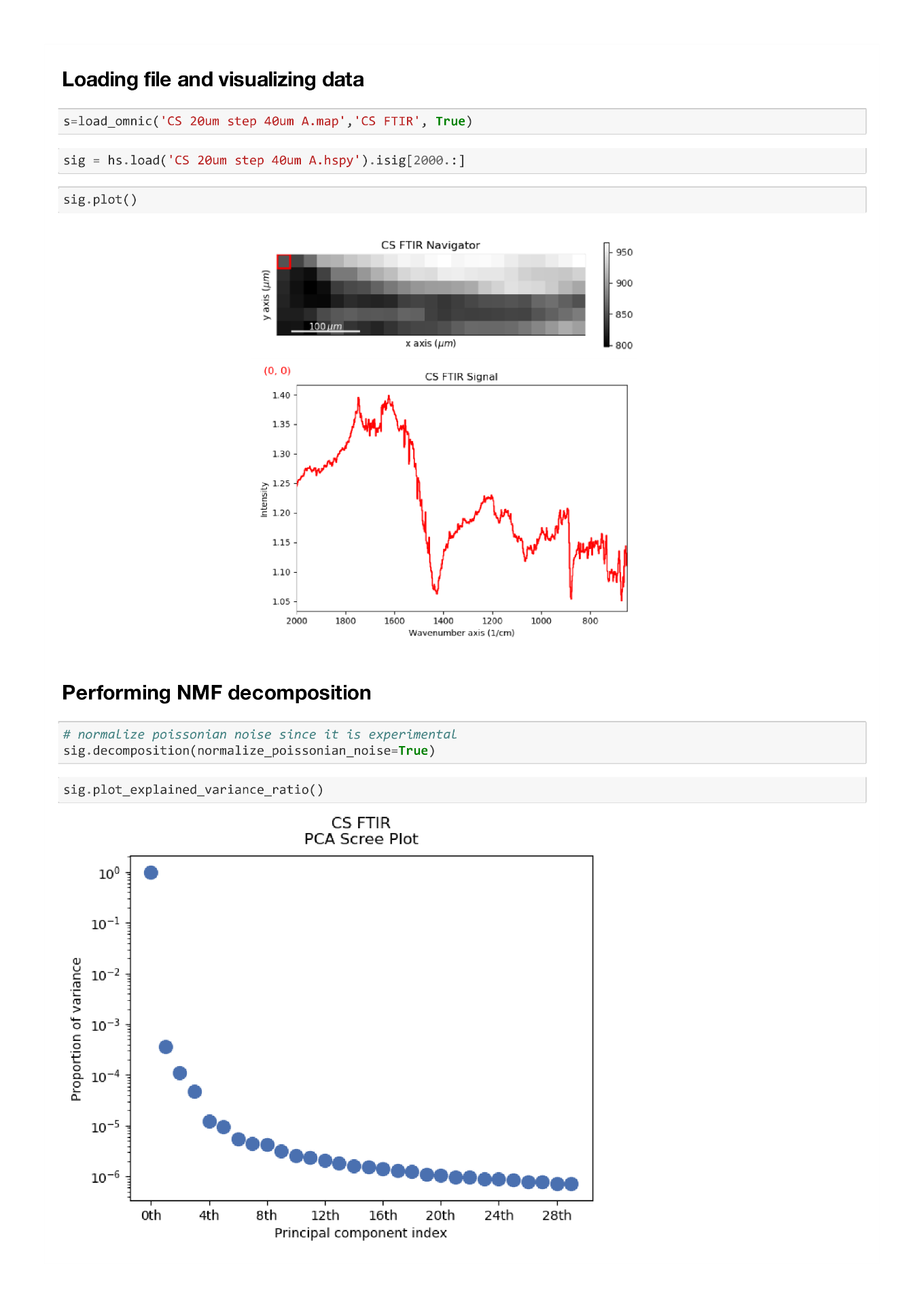
**return** sig

*# ## Loading file and visualizing data*

s=load\_omnic('CS 20um step 40um A.map','CS FTIR', True)

sig = hs.load('CS 20um step 40um A.hspy').isig[2000.:]

sig.plot()

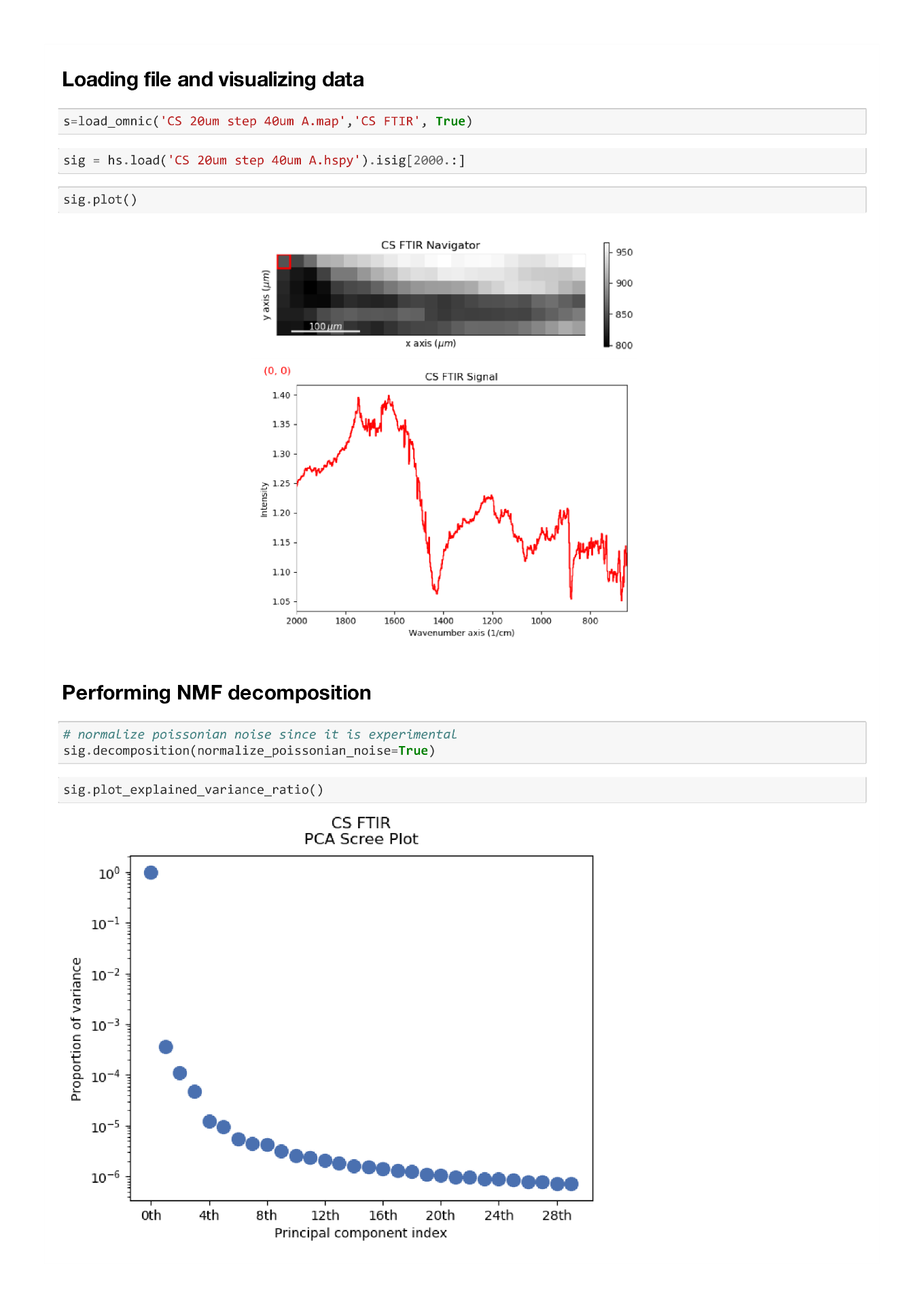
****

*# ## Performing NMF decomposition*

*# normalize poissonian noise since it is experimental*

sig.decomposition(normalize\_poissonian\_noise=True)

sig.plot\_explained\_variance\_ratio()

****

sig.decomposition(True, algorithm='nmf', output\_dimension=3)

loadings = sig.get\_decomposition\_loadings()

hs.plot.plot\_images(loadings, cmap='viridis',

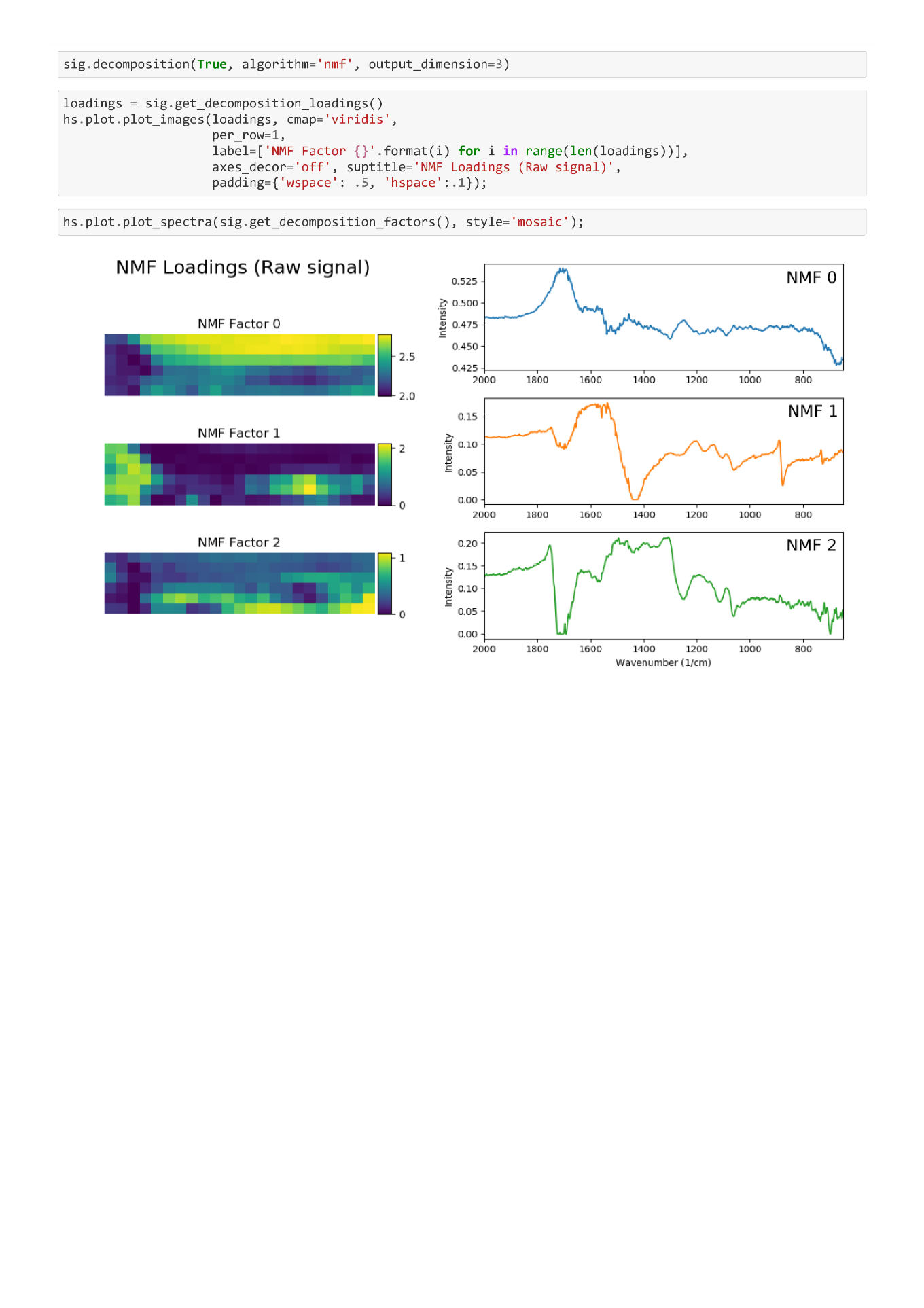
per\_row=1,

label=['NMF Factor {}'.format(i) **for** i **in** range(len(loadings))],

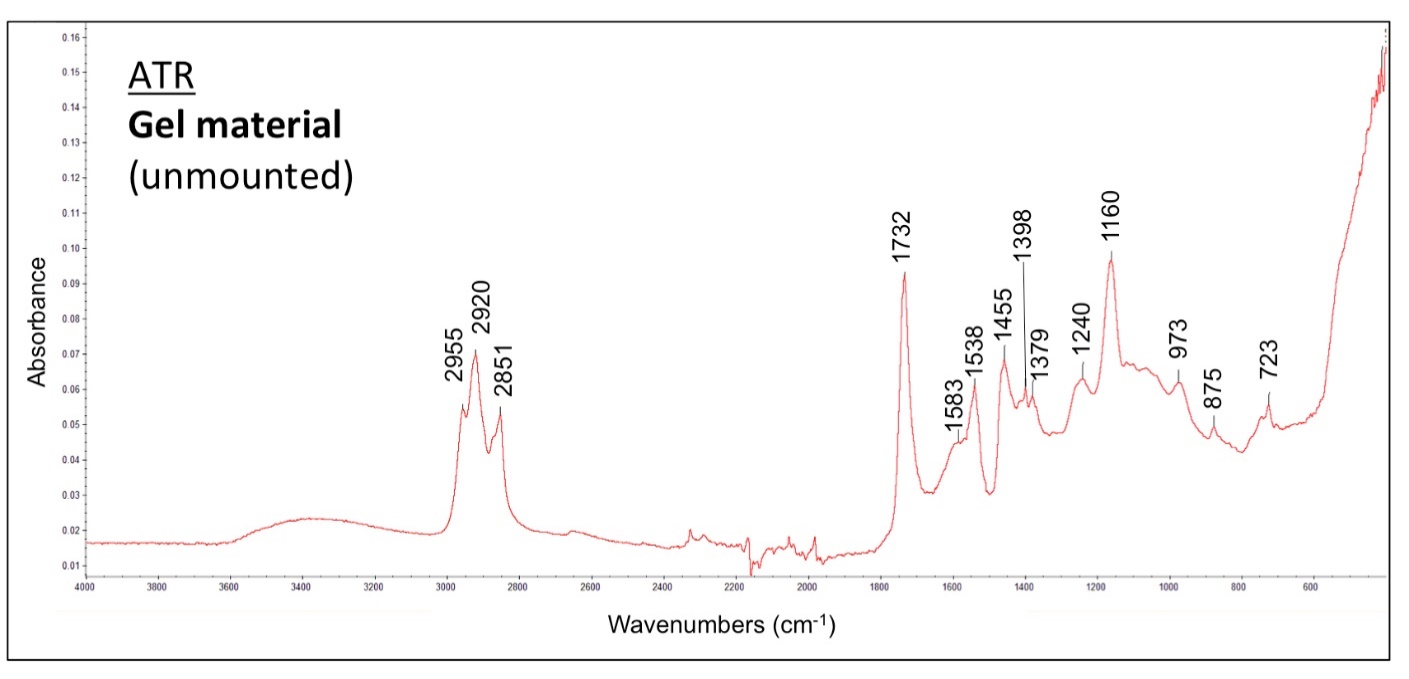
axes\_decor='off', suptitle='NMF Loadings (Raw signal)',

padding={'wspace': .5, 'hspace':.1})

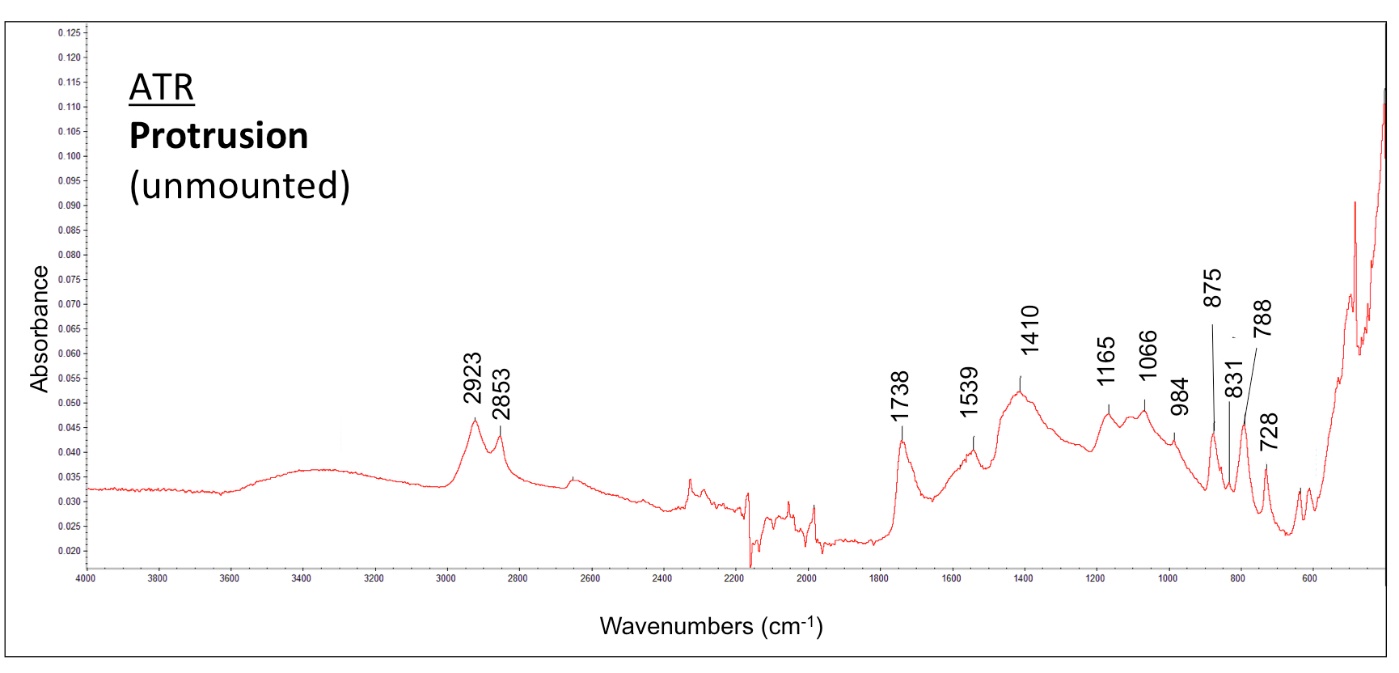
hs.plot.plot\_spectra(sig.get\_decomposition\_factors(), style='mosaic')

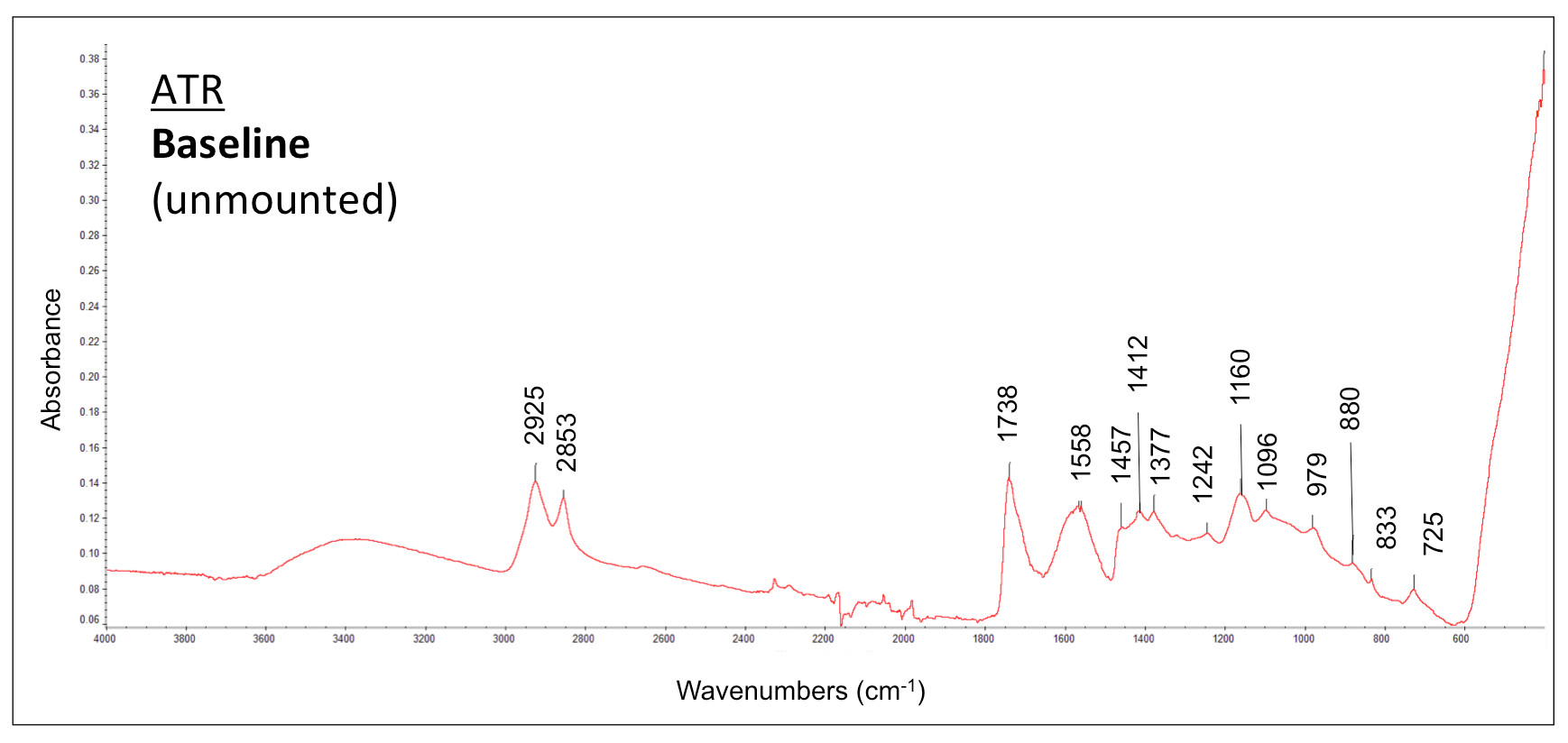
****

**Supplemental Figure 3.** ATR-FTIR spectra for Gel material



**Supplemental Figure 4.** ATR-FTIR Spectrum for Protrusion

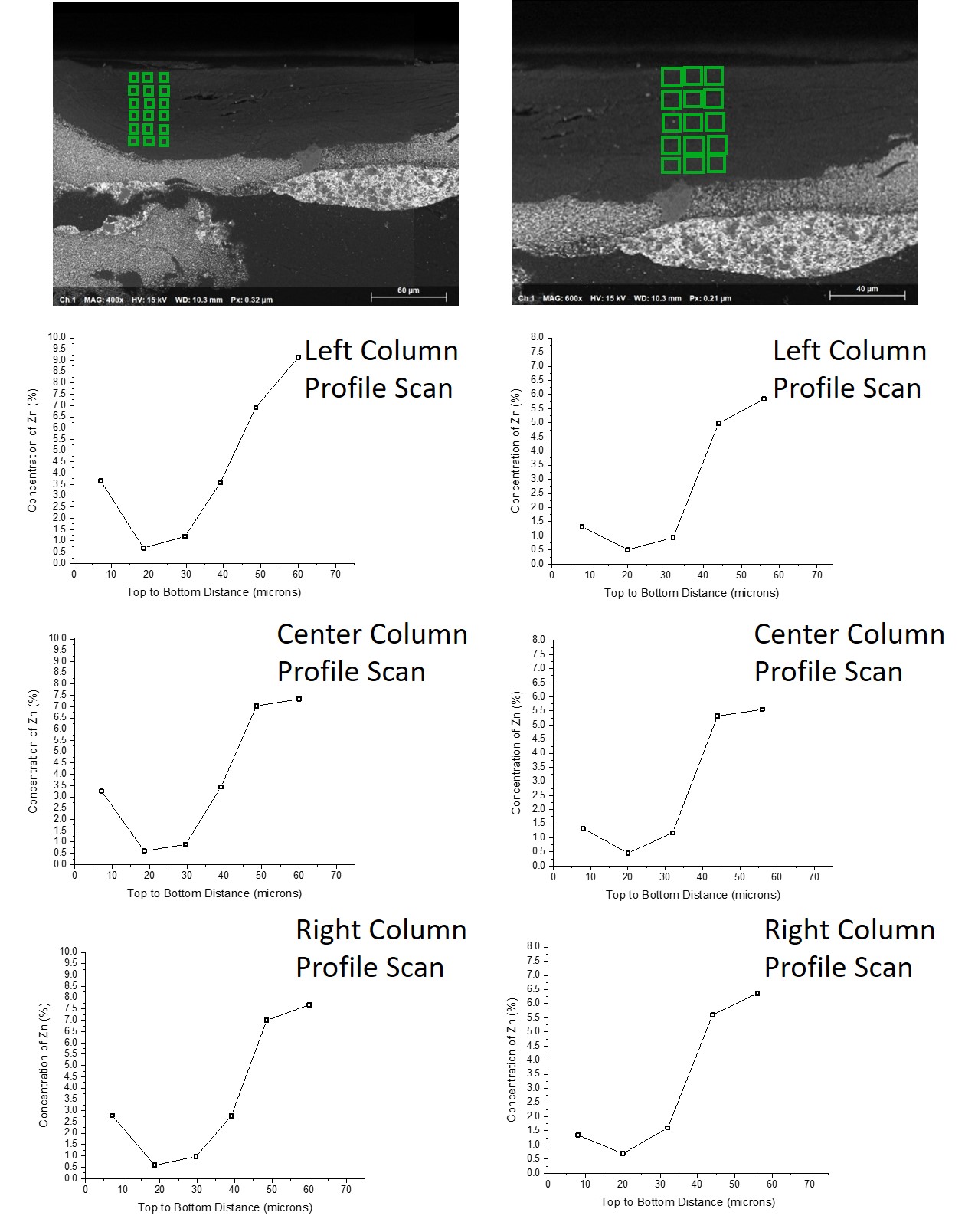
****

**Supplemental Figure 5.** ATR-FTIR Spectrum for Baseline Paint 

**Supplemental Table I. Mass fraction of Elemental Concentrations in the Gel, Bio-Plastic®, and Zinc White with ± standard of error**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **C (%)** | **O (%)** | | **S (%)** | | **Cl (%)** | | **Zn (%)** | |
| **Gel** | 74.71 ± 1.59 | | 16.12 ± 1.57 | | 0.75 ± 0.04 | | 1.04 ± 0.10 | | 6.75 ± 2.12 | |
| **Bio-Plastic** | 83.93± 0.71 | 15.67 ± 0.74 | | < 0.5 | | < 0.5 | | < 0.5 | |
| **Zinc white** | 47.93 ± 2.80 | 13.76 ± 0.18 | | < 0.5 | | < 0.5 | | 38.16 ± 2.88 | |

**Supplemental Figure 6.** Profile scans of 33 regions of the gel material in cross-sectional view with semi-quantified EDX mass fraction for Zn concentration.

****