```
## PLOTING XES MAPS ON ENERGY LOSS SCALE ##
ELOSSMap = ELOSSMapper()

## LOADING/ADDING/SUBSTRACTING 2-D XES DATA FROM A FILE ##
## Loads 2-D XES scans data from HDF5 file
ELOSSMap.load(config,'filename', 'detector', arg, **kwargs)
## args = scan number to be loaded

## Loads and sums 2-D XES scans data from HDF5 file
ELOSSMap.add(config,'filename', 'detector', *args, **kwargs)
## *args = comma seperated list of scans to be plotted or added and then plotted

## Loads and subtracts 2-D XES scans data from HDF5 file
ELOSSMap.subtract(config,'filename', 'detector', *args, **kwargs)
## *args = s1, p1                        -> The data from p1 is subtracted from s1
## *args = [s1, ..., sn], [p1, ..., pn] -> The sum of p1..pn is subtracted from the sum of s1...sn

## Loads and stitches 2-D XES scans data from HDF5 file
ELOSSMap.stitch(config,'filename','detector', *args, **kwargs)
## *args = comma seperated list of scans to be stitched

## Loads and subtracts the background from each image column
ELOSSMap.background_1d(config,'filename','detector', *args, **kwargs)
## *args = s1 -> The scan to be subtracted from all previous load/add/subtract actions
## *args = [s1, ..., sn] -> The sum of scans s1..sn to be subtracted from all previous load/add/subtract

## Loads and subtracts the image the loaded image
ELOSSMap.background_2d(config,'filename','detector', *args, **kwargs)
## *args = s1 -> The scan to be subtracted from all previous load/add/subtract actions
## *args = [s1, ..., sn] -> The sum of scans s1..sn to be subtracted from all previous load/add/subtract

## REQUIRED VARIABLES ##
## config = RIXS -> RIXS Endstation
## config = RSXS -> RSXS Endstation
## filename = hdf5 file -> Extension .h5 not needed
## xes_stream -> MCA type detector
## NOTE: Simple math allowed with xes_stream with contstants and variables, i.e. +, -, /, *

## *kwargs options ##
## norm = True                      -> Scales the data such that its range is 0 to 1.
## xoffset = [(S1,P1),...,(SN,PN)] -> Adjusts x-axis scale to map SN to PN
## xcoffset = value                 -> Shifts x-axis scale by a constant value
## yoffset = [(S1,P1),...,(SN,PN)] -> Adjusts y-axis scale to map SN to PN
## ycoffset = value                 -> Shifts y-axis scale by a constant value
## grid_x = [start,stop,delta]      -> Change x-axis grid to be uniform
## grid_y = [start,stop,delta]      -> Change y-axis grid to be uniform
## binsize_x = bins                 -> Bins data along x-axis, specify the number of points (extra points removed)
## binsize_y = bins                 -> Bins data along y-axis, specify the number of points (extra points removed)

## SET RANGE OF Y and X VALUES ##
ELOSSMap.xlim(min, max)
ELOSSMap.ylim(min, max)

# PLOTTING SCAN DATA ##
ELOSSMap.plot(**kwargs)

## **kwargs ##
## title = 'New Title of plot' -> Replaces default title with user defined
## xlabel = 'x-axis label'     -> Replaces default x-axis label with user defined
## ylabel = 'y-axis label'     -> Replaces default y-axis label with user defined
## zlabel = 'colorscale label' -> Replaces default colorscale label with user defined
## plot_height = value         -> The plot height in points, default is 600
## plot_width = value          -> The plot width in points, default is 900
## norm = True                 -> Normalizes all the data between 0 and 1
## vmin = value                -> Sets the maximum value of the colorscale
## vmax = value                -> Sets the minimum value of the colorscale

## EXPORTING PLOT DATA ##
ELOSSMap.export('filename', **kwargs)

# REQUIRED VARIABLES ##
## filename = filename to be used for ASCII file, do not add extension
## NOTE: Data is exported as it displayed, only options in plotting methods are ignored.

## **kwargs ##
## split_files = True -> Saves each data stream with number appended to the filename
```

```
## ELOSS Map of hBN
ELOSSMap = ELOSSMapper()
yoffset = [(369.438, 370), (379.375, 380), (389.210, 390), (398.932, 400),(408.719, 410), (418.632, 420)]
ELOSSMap.load(RIXS, 'HDF5_Notebook' , 'XES', 25,yoffset = yoffset)
ELOSSMap.plot()
ELOSSMap.export('ELOSS_Map')
```