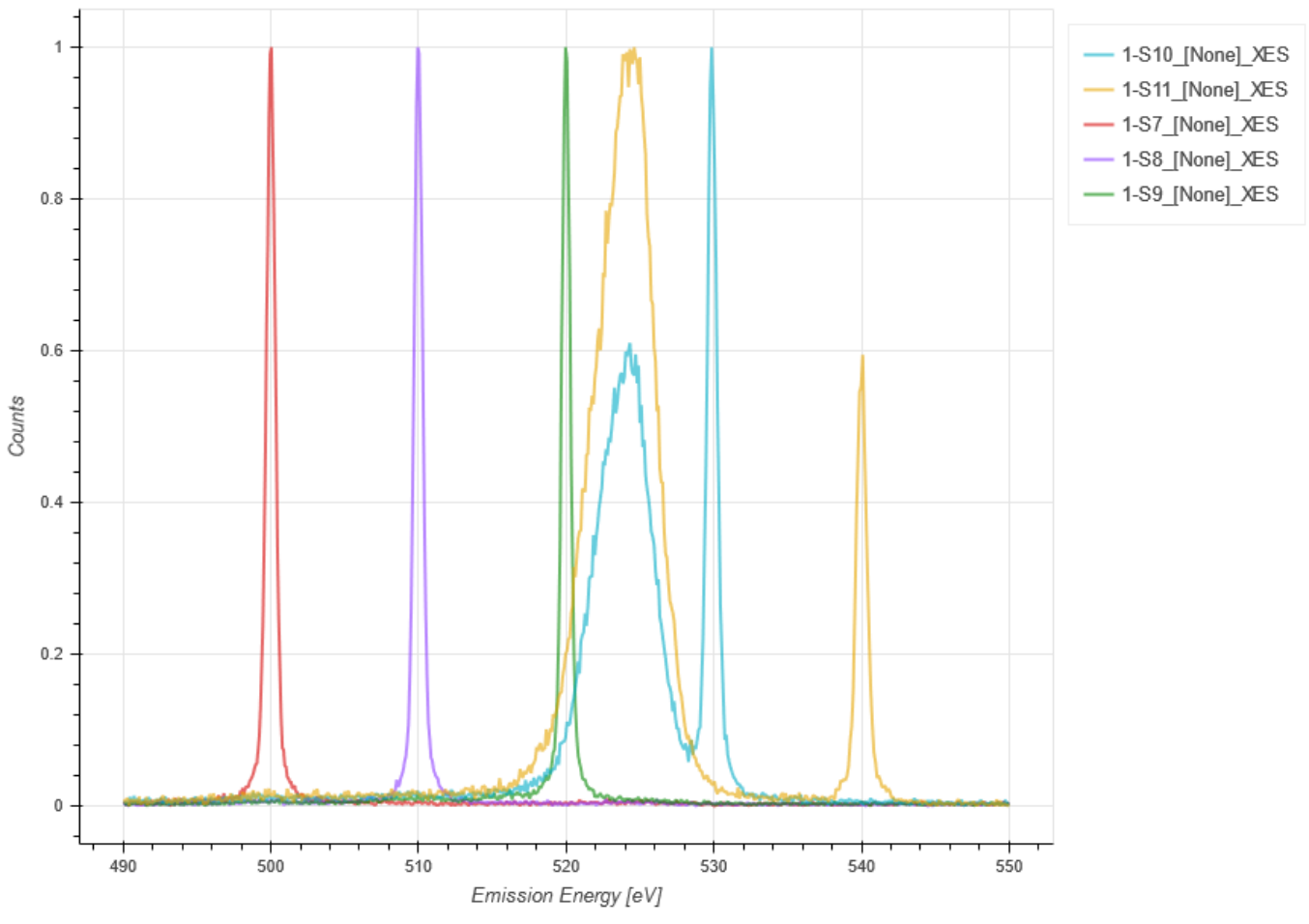
```
## PLOTING X-RAY EMISSION SPECTROSCOPY DATA ##
## Creates an object named XES to load XES DATA, EITHER TOTAL OR SUMMED OVER SPECIFIC ENERGIES
XES = XESLoader()

## LOADING/ADDING/SUBSTRACTING 1-D/REDUCED DATA FROM A FILE ##
## Loads XES scans data from HDF5 file
XES.load(config,'filename', 'detector', *args, **kwargs)
## *args = comma seperated list of scans to be plotted or added and then plotted

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

## Loads and subtratcs XES scans data from HDF5 file
XES.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 sub. from the sum s1...sn

## Loads and stitches non-overlapping regions
XES.stitch(config,'filename', 'detector', *args, **kwargs)
## *args = comma seperated list of scans to be stitched
## NOTE: The the scans are appended in order, overlap discared

## Loads and subtract scan from all previously loaded scans
XES.background(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 actions

## REQUIRED VARIABLES ##
## config = RIXS         -> RIXS Endstation
## config = RSXS         -> RSXS Endstation
## filename = hdf5 file  -> Extension .h5 not needed
## detector            -> sums all data from MCA type detector
## detector[Start:End] -> sums all MCA data within excitation energy range
## NOTE: Simple math allowed with xes_stream with contstants and variables, i.e. +, -, /, *

## **kwargs ##
## norm = True                          -> Scales the data such that its range is 0 to 1.
## twin_y = True                        -> Adds these plots to a secondary scale
## 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 = [start,stop,delta]          -> Change x-axis grid to be uniform
## savgol = (wind len, poly ord, deru) -> Smooths and takes derivative
## binsize = bins                       -> Bins data, specify the number of points (extra points removed)


## SET RANGE OF Y and X VALUES ##
XES.xlim(min, max)
XES.ylim(min, max)
## NOTE: These ranges will be preserved in the data export

## PLOTTING SCAN DATA ##
XES.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
## 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
## waterfall = offset          -> Normalizes as above and shifts each by the offset

## EXPORTING PLOT DATA ##
XES.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
```

```
## Plot elastic peaks for calibration
## Add offset for scale mapping, apply to all subsequent scans
XES_Calib = XESLoader()
xoffset = [(498.442, 500), (508.254, 510), (517.910, 520), (527.574, 530),(537.059, 540)]
XES_Calib.load(RIXS,'HDF5_Notebook.h5','XES',7,8,9,10,11,xoffset= xoffset)
XES_Calib.xlim(490,550)
XES_Calib.plot(norm = True)
XES_Calib.export('XES_Calib')
```




```
## Sum specific excitation energies of the scan and compare
## Bin the data to improve statisitcs
BGO_XES = XESLoader()
xoffset = [(498.442, 500), (508.254, 510), (517.910, 520), (527.574, 530),(537.059, 540)]
BGO_XES.load(RIXS,'HDF5_Notebook.h5','XES',14,xoffset = xoffset)
BGO_XES.load(RIXS,'HDF5_Notebook.h5','XES[530:535]',14,xoffset = xoffset)
BGO_XES.load(RIXS,'HDF5_Notebook.h5','XES[530:535]',14,binsize = 4,xoffset = xoffset)
BGO_XES.load(RIXS,'HDF5_Notebook.h5','XES',15,xoffset = xoffset)
BGO_XES.xlim(490,550)
BGO_XES.plot(waterfall = 0.2)
BGO_XES.export('BGO_XES')
```