Visualization Tools

In psana-python we support three types of visualization: the standard python matplotlib, a more interactive form of plotting that can also be used for real-time monitoring called "psmon" (see https://github.com/lcls-psana/psmon) and the graphical analysis tool "psocake" which is described here.

A simple matplotlib script can be seen in this example. A corresponding version in psmon (which allows much more "interaction" with the plot) is in /reg/g/psdm/tutorials/examplePython3/psmonLocal.py:

```
from psana import *
ds = DataSource('exp=xpptut15:run=54:smd')
det = Detector('cspad')
for nevent,evt in enumerate(ds.events()):
    img = det.image(evt)
    y = img.sum(axis=0)
    break
from psmon.plots import Image,XYPlot
from psmon import publish
publish.local = True
publish.plot_opts.aspect = 1 # needed to get images with correct 1:1 aspect ratio
plotimg = Image(0,"CsPad",img)
publish.send('IMAGE',plotimg)
plotxy = XYPlot(0,"Y vs. X",range(len(y)),y)
publish.send('XY',plotxy)
```

The above example demonstrates the two most popular styles of plots: Image (with x,y,z coordinates) and XYPlot (with x,y coordinates).

psmon is also able to send plots over the network, which is useful for real-time monitoring. To do this, **do not set** publish.local=True (see script in /reg/g/psdm/tutorials/examplePython3/psmonRemote.py):

```
from psana import *
from psmon.plots import Image
from psmon import publish
ds = DataSource('exp=xpptut15:run=54:smd')
det = Detector('cspad')
for nevent,evt in enumerate(ds.events()):
    img = det.image(evt)
    plotimg = Image(0,"CsPad",img)
    publish.send('IMAGE',plotimg)
    raw_input('Hit <CR> for next event')
    if nevent>=2: break
```

Then this plot can then be viewed (by multiple people, if desired) using:

```
psplot -s <hostname> IMAGE
```

The "-s <hostname>" argument does not need to be included if "psplot" is run on the same machine as the script that is providing the plot.

There are more sophisticated psmon examples demonstrating features like Multiplots (multiple plots that are updated at the same time), plot overlays etc. here: https://github.com/lcls-psana/psmon/tree/master/examples. The "formats" keyword to an XYPlot can be used to select plot markers (square, circle, etc.) following the matplotlib pattern here.