KaraboData_Talk

January 11, 2019

```
In [1]: import warnings

        from IPython.display import HTML
        import matplotlib
        from matplotlib import pyplot as plt
        import numpy as np
        plt.rcParams['figure.figsize'] = (20.0, 10.0)
        matplotlib.rc('image', cmap='RdYlBu')

        warnings.filterwarnings("ignore")
        %matplotlib inline
```

1 Offline data analysis

```
<div class="intro-body">
    <div class="intro_h1"><h3>The karabo-data ecosystem</h3></div>
    <p><strong><span class="a">Martin Bergemann</span></strong><span class="b"></span><span>CAS
```

A Closer Look at Detector Data:
Data from Modular Detectors is stored across multiple files
Using standard hdf5 tools can be tricky and tedious

1.1 Solution: *karabo-data*

- Python Library that supports Analysis of EuXFEL Data.
- It is Open Source and available on the max-well cluster.

1.2 Aim:

Give you a glimpse of what is possible and provide a ground base for your own data analysis at EuXFEL

1.3 Scenario I: Take a look at some detector data

An image from a single module of the AGIP-D Detector at SPB

- How could this data be retrieved and visualized?

1.3.1 Reading single files (demo)

```
In [2]: #Live-demo
        import karabo_data as kd
        exmpl_file = '/gpfs/exfel/exp/XMPL/201750/p700000/proc/r0273/CORR-R0273-AGIPD03-S00000.h
        hdf5_file = kd.H5File(exmpl_file)
```

Data can be accessed by: * selecting trains id's * selecting train indexes * iteration over trains
Let's select data based on trains: (demo)

```
In [3]: #Live-demo (sel from train-id)
        train_id, train_data = hdf5_file.train_from_id(198425246)
        print(train_id, train_data.keys())

198425246 dict_keys(['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf'])
```

```
In [4]: #Live-demo (sel from index)
        train_id, train_data = hdf5_file.train_from_index(5)
        print(train_id, train_data.keys())

198425246 dict_keys(['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf'])
```

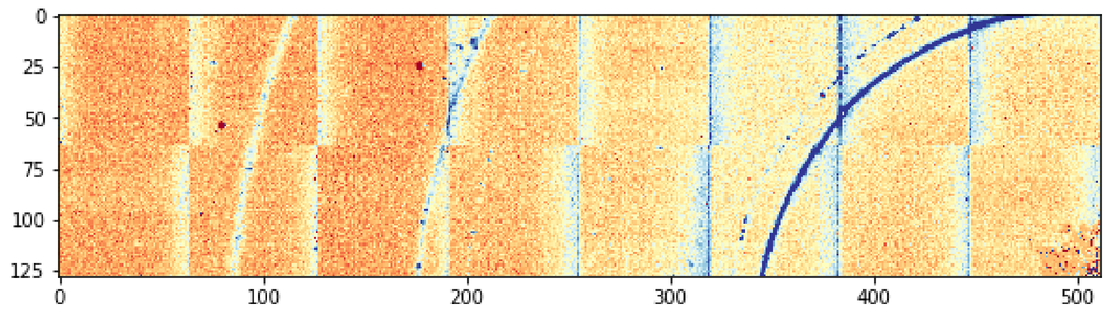
```
In [5]: #Live-demo (iteration)
        for train_id, train_data in hdf5_file.trains():
            print(train_id, train_data.keys())
            break

198425241 dict_keys(['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf'])
```

Train data is of type *dictionary*. Hence the data can be accessed by giving keys:

```
In [6]: train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']

Out[6]: {'detector.data': array([  0,  64, 114, ...,   0,   0,   0], dtype=uint8),
         'detector.trainId': 198425241,
         'header.dataId': 0,
         'header.linkId': 18446744069414584335,
         'header.magicNumberBegin': array([-50,  -6, -17, -66,  70,  68,  84,  88], dtype=int8),
         'header.majorTrainFormatVersion': 1,
         'header.minorTrainFormatVersion': 0,
         'header.pulseCount': 176,
         'header.reserved': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=uint8)
         'header.trainId': 198425241,
         'image.cellId': array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
                13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
                26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
                39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
         52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
         65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
         78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
         91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
        104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
        117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
        130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
        143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
        156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
        169, 170, 171, 172, 173, 174, 175], dtype=uint16),
 'image.data': array([[[ 2.86249847e+01,  2.84662476e+01,  2.94381580e+01, ...,
           3.97410750e+00,  1.53232050e+01,  2.65694761e+00],
         [ 3.76776047e+01,  1.66408386e+01,  1.25324402e+01, ...,
           3.53553891e+00, -7.76110840e+00,  5.06602287e+00],
         [ 2.33586578e+01,  1.33027906e+01,  1.29060669e+01, ...,
          -3.58545852e+00, -8.78809452e+00, -1.76954997e+00],
         ...,
         [ 2.67929325e+01,  3.82321243e+01,  1.69615307e+01, ...,
          -5.36543350e+01,  1.77964573e+01,  2.29760933e+01],
         [ 5.43048973e+01,  2.40606403e+01,  2.68685341e+01, ...,
           1.56021561e+02,  0.00000000e+00,  1.39021225e+02],
         [ 5.41300812e+01,  3.99643478e+01,  3.18235302e+01, ...,
           4.46845245e+00, -3.83902435e+02,  1.09092340e+01]],

        [[ 2.82328766e+02,  2.13707367e+02,  3.85529510e+02, ...,
           2.87144348e+02,  2.49376450e+02,  3.46322205e+02],
         [ 1.72267563e+02,  2.91269714e+02,  2.23787582e+02, ...,
           1.87312103e+02,  1.91055298e+02,  1.92247986e+02],
         [ 1.74026535e+02,  2.09151901e+02,  2.23993912e+02, ...,
           1.28322266e+02,  6.25818977e+01,  1.35687103e+02],
         ...,
         [ 9.67379822e+02,  4.64296753e+02,  1.03795129e+03, ...,
           4.12897301e+01,  2.44968739e+01,  1.36369019e+02],
         [ 1.12769836e+03,  8.79650452e+02,  6.07432495e+02, ...,
           3.33366852e+02,  2.61182800e+02,  4.13849525e+01],
         [ 1.37194507e+03,  1.16530359e+03,  9.33655640e+02, ...,
          -1.40187941e+01, -1.80979218e+02, -1.05465247e+03]],

        [[ 2.74763580e+02,  2.57296173e+02,  1.92696045e+02, ...,
           5.14890625e+02,  4.83632965e+02,  4.13015289e+02],
         [ 1.88023987e+02,  1.00196587e+02,  1.05712891e+02, ...,
           3.33557678e+02,  4.17891815e+02,  4.55165558e+02],
         [ 2.97341156e+02,  1.76079010e+02,  8.91075516e+01, ...,
           2.00919449e+02,  2.54712372e+02,  2.78320801e+02],
         ...,
         [ 1.41089551e+03,  1.34565967e+03,  1.08119263e+03, ...,
           7.16647387e+00,  2.45052700e+01,  5.89854355e+01],
         [ 1.80306104e+03,  1.42946960e+03,  9.75338379e+02, ...,
          1.60704651e+02,  2.65418762e+02,  1.36963837e+02],
        [ 2.03517444e+03,  1.21541614e+03,  9.28263855e+02, ...,
          9.67607212e+00, -2.76129272e+02, -5.88472351e+02]],

       ...,

       [[ 3.09872532e+00, -5.08774281e-01,  1.51738377e+01, ...,
          9.67595291e+01,  1.02846840e+02,  9.79162140e+01],
        [-2.02406406e+00, -8.11292076e+00, -1.02228737e+01, ...,
          1.00479095e+02,  1.04622551e+02,  1.17755341e+02],
        [ 1.53183994e+01,  9.43448830e+00, -9.73048973e+00, ...,
          9.80847092e+01,  9.90229797e+01,  9.15223618e+01],
        ...,
        [ 9.65496826e+01,  9.00805054e+01,  8.23390045e+01, ...,
          3.99469910e+01, -4.39730103e+02,  8.46993359e+03],
        [ 1.09837486e+02,  9.43753433e+01,  1.01021187e+02, ...,
         -7.42510234e+04, -1.19970758e+05, -7.43315469e+04],
        [ 1.11256569e+02,  9.82772827e+01,  1.06616699e+02, ...,
          7.79505127e+02, -1.85362646e+03, -1.54386367e+04]],

       [[ 2.27756405e+01,  2.25000038e+01,  2.32766819e+01, ...,
          1.05306572e+02,  1.19960114e+02,  1.06827599e+02],
        [ 1.22687225e+01,  1.11412373e+01, -3.10237098e+00, ...,
          1.11624672e+02,  9.99300003e+01,  1.20338463e+02],
        [ 2.03352380e+00,  3.13974047e+00,  1.52829742e+01, ...,
          1.17073769e+02,  1.08744743e+02,  1.05334351e+02],
        ...,
        [ 1.06949814e+02,  9.66776505e+01,  9.08718491e+01, ...,
         -1.01162968e+01, -8.50073535e+03, -1.30835596e+04],
        [ 1.24415733e+02,  9.19042053e+01,  9.73192291e+01, ...,
         -2.65518613e+04,  1.87262070e+04, -1.64417363e+04],
        [ 1.32533249e+02,  1.10008453e+02,  1.15432304e+02, ...,
          3.16230347e+02, -8.02921484e+03, -1.48106921e+03]],

       [[ 1.53365967e+02,  1.39493073e+02,  1.64242966e+02, ...,
          2.43825836e+02,  2.61614197e+02,  2.44643021e+02],
        [ 1.56373383e+02,  1.53281754e+02,  1.43108688e+02, ...,
          2.61418030e+02,  2.48436081e+02,  2.61712524e+02],
        [ 1.51726349e+02,  1.45641235e+02,  1.53762711e+02, ...,
          2.48970200e+02,  2.41733551e+02,  2.53827728e+02],
        ...,
        [ 2.72267700e+02,  2.46381104e+02,  2.55895477e+02, ...,
          1.01002051e+03, -2.68937812e+04, -6.32623047e+04],
        [ 2.82559753e+02,  2.68116241e+02,  2.57689056e+02, ...,
         -6.13691211e+03, -5.68883984e+03, -3.82133008e+03],
        [ 2.84443420e+02,  2.49587311e+02,  2.78960602e+02, ...,
          3.70983740e+03,  9.68601562e+03,  1.14233716e+03]]],
      dtype=float32),
'image.gain': array([[[0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         ...,
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 1, 0],
         [0, 0, 0, ..., 0, 0, 0]],

        [[0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         ...,
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 1]],

        [[0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         ...,
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 1]],

        ...,

        [[0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         ...,
         [0, 0, 0, ..., 1, 1, 1],
         [0, 0, 0, ..., 2, 2, 2],
         [0, 0, 0, ..., 1, 1, 2]],

        [[0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         ...,
         [0, 0, 0, ..., 1, 2, 2],
         [0, 0, 0, ..., 2, 2, 2],
         [0, 0, 0, ..., 1, 1, 2]],

        [[0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         [0, 0, 0, ..., 0, 0, 0],
         ...,
         [0, 0, 0, ..., 1, 2, 2],
         [0, 0, 0, ..., 2, 2, 2],
        [0, 0, 0, ..., 1, 2, 2]]], dtype=uint8),
 'image.length': array([262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144,
        262144, 262144, 262144, 262144, 262144, 262144, 262144, 262144],
       dtype=uint32),
 'image.mask': array([[[     0,      0,      0, ...,      0,      0,      0],
        [     0,      0,      0, ...,      0,      0,      0],
        [     0,      0,      0, ...,      0,      0,      0],
        ...,
        [     0,      0,      0, ...,    515,    515,    515],
        [     0,      0,      0, ...,    515,    515,    515],
        [     0,      0,      0, ...,    515,    515,    515]],

       [[     0,      0,      0, ...,      0,      0,      0],
        [     0,      0,      0, ...,      0,      0,      0],
        [     0,      0,      0, ...,      0,      0,      0],
        ...,
        [     0,      0,      0, ...,    515,    515,    515],
        [     0,      0,      0, ...,    515,    515,    515],
        [     0,      0,      0, ...,    515,    515, 131587]],

       [[     0,      0,      0, ...,      0,      0,      0],
        [     0,      0,      0, ...,      0,      0,      0],
        [     0,      0,      0, ...,      0,      0,      0],
        ...,
        [     0,      0,      0, ...,    515,    515,    515],
        [     0,      0,      0, ...,    515,    515,    515],
        [     0,      0,      0, ...,    515,    515,    515]],
           ...,

           [[   512,    512,    512, ...,    512,    512,    512],
            [   512,    512,    512, ...,    512,    512,    512],
            [   512,    512,    512, ...,    512,    512,    512],
            ...,
            [   512,  16896,    512, ...,    515,    515,    515],
            [   512,    512,    512, ...,    513,    513,    515],
            [   512,    512,    512, ...,  16899,    515,    515]],

           [[   512,    512,    512, ...,    512,    512,    512],
            [   512,    512,    512, ...,    512,    512,    512],
            [   512,    512,    512, ...,    512,    512,    512],
            ...,
            [   512,    512,    512, ...,    515,  16897,    515],
            [   512,    512,    512, ...,  16899,    513,    515],
            [ 16896,    512,    512, ...,    515,    515,  16899]],

           [[   512,    512,    512, ...,    512,    512,    512],
            [   512,    512,    512, ...,    512,    512,    512],
            [   512,    512,    512, ...,    512,    512,    512],
            ...,
            [   512,    512,    512, ..., 131587,    515,  16899],
            [   512,    512,    512, ...,    515,    513,    515],
            [   512,    512,    512, ..., 131587,    513,    515]]],
          dtype=uint32),
    'image.pulseId': array([  0,   4,   8,  12,  16,  20,  24,  28,  32,  36,  40,  44,  48
            52,  56,  60,  64,  68,  72,  76,  80,  84,  88,  92,  96, 100,
           104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 148, 152,
           156, 160, 164, 168, 172, 176, 180, 184, 188, 192, 196, 200, 204,
           208, 212, 216, 220, 224, 228, 232, 236, 240, 244, 248, 252, 256,
           260, 264, 268, 272, 276, 280, 284, 288, 292, 296, 300, 304, 308,
           312, 316, 320, 324, 328, 332, 336, 340, 344, 348, 352, 356, 360,
           364, 368, 372, 376, 380, 384, 388, 392, 396, 400, 404, 408, 412,
           416, 420, 424, 428, 432, 436, 440, 444, 448, 452, 456, 460, 464,
           468, 472, 476, 480, 484, 488, 492, 496, 500, 504, 508, 512, 516,
           520, 524, 528, 532, 536, 540, 544, 548, 552, 556, 560, 564, 568,
           572, 576, 580, 584, 588, 592, 596, 600, 604, 608, 612, 616, 620,
           624, 628, 632, 636, 640, 644, 648, 652, 656, 660, 664, 668, 672,
           676, 680, 684, 688, 692, 696, 700], dtype=uint64),
    'image.status': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
           0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
              dtype=uint16),
         'image.trainId': array([198425241, 198425241, 198425241, 198425241, 198425241, 19842524
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241, 198425241, 198425241, 198425241, 198425241,
                198425241, 198425241], dtype=uint64),
         'metadata': {'source': 'SPB_DET_AGIPD1M-1/DET/3CH0:xtdf',
          'timestamp.tid': 198425241},
         'trailer.checksum': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int8)
         'trailer.magicNumberEnd': array([-51, -85, -83, -34,  70,  68,  84,  88], dtype=int8),
         'trailer.status': 0,
         'trailer.trainId': 198425241}
```

```
In [9]: train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']['image.data'].shape

Out[9]: (176, 512, 128)

In [11]: #Live-demo
         type(train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']['image.data'])

Out[11]: numpy.ndarray

In [12]: #Live-demo
         from matplotlib import pyplot as plt
```

```
fig = plt.figure(figsize=(10,10))
_ = plt.imshow(train_data['SPB_DET_AGIPD1M-1/DET/3CH0:xtdf']['image.data'][0].T,
               vmin=-50, vmax=100)
```



1.4 Reading whole Runs:

The main advantage of karabo-data is that not only single files at a time but whole runs can be read:

```
In [13]: #Live-demo
         run_folder = '/gpfs/exfel/exp/XMPL/201750/p700000/raw/r0273'
         run_dir = kd.RunDirectory(run_folder)
         run_dir.info()
```

```
# of trains:    156
Duration:       0:00:15.500000
First train ID: 198425241
Last train ID:  198425396

16 detector modules (SPB_DET_AGIPD1M-1)
  e.g. module SPB_DET_AGIPD1M-1 0 : 512 x 128 pixels
  176 frames per train, 27456 total frames

2 instrument sources (excluding detectors):
  - SA1_XTD2_XGM/XGM/DOOCS:output
  - SPB_XTD9_XGM/XGM/DOOCS:output

13 control sources:
  - ACC_SYS_DOOCS/CTRL/BEAMCONDITIONS
  - SA1_XTD2_XGM/XGM/DOOCS
  - SPB_IRU_AGIPD1M/PSC/HV
  - SPB_IRU_AGIPD1M/TSENS/H1_T_EXTHOUS
  - SPB_IRU_AGIPD1M/TSENS/H2_T_EXTHOUS
  - SPB_IRU_AGIPD1M/TSENS/Q1_T_BLOCK
  - SPB_IRU_AGIPD1M/TSENS/Q2_T_BLOCK
- SPB_IRU_AGIPD1M/TSENS/Q3_T_BLOCK
- SPB_IRU_AGIPD1M/TSENS/Q4_T_BLOCK
- SPB_IRU_AGIPD1M1/CTRL/MC1
- SPB_IRU_AGIPD1M1/CTRL/MC2
- SPB_IRU_VAC/GAUGE/GAUGE_FR_6
- SPB_XTD9_XGM/XGM/DOOCS
```

1.5 Data subsets

Yet another powerful tool is the ***select*** method. It selects a subset of sources and keys from the run directory:

```
In [14]: #Live-demo with glob pattern
         sel = run_dir.select('*/DET/*', 'image.*')
         sel.all_sources
```

```
Out[14]: frozenset({'SPB_DET_AGIPD1M-1/DET/0CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/10CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/11CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/12CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/13CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/14CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/15CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/1CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/2CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/3CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/4CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/5CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/6CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/7CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/8CH0:xtdf',
                    'SPB_DET_AGIPD1M-1/DET/9CH0:xtdf'})
```

Data can be access just like *previously* mentioned:

```
In [16]: #Live-demo (iterator example)
         for train_id, train_data in sel.trains():
             #print(train_id, train_data.keys())
             break
         train_id, train_data = sel.train_from_index(5)
         train_id, train_data.keys()
```

```
Out[16]: (198425246,
          dict_keys(['SPB_DET_AGIPD1M-1/DET/14CH0:xtdf', 'SPB_DET_AGIPD1M-1/DET/13CH0:xtdf', 'SP
```

1.6 What if I don't want to fire up a python console just to quickly check something?

The command `lsxfel` is a tool that provides some basic functionality using only the command line.

```
$: lsxfel /gpfs/exfel/exp/XMPL/201750/p700000/raw/r0273/
r0273 : Run directory

# of trains:    156
Duration:       0:00:15.500000
First train ID: 198425241
Last train ID:  198425396

16 detector modules (SPB_DET_AGIPD1M-1)
  e.g. module SPB_DET_AGIPD1M-1 0 : 512 x 128 pixels
  176 frames per train, 27456 total frames

2 instrument sources (excluding detectors):
  - SA1_XTD2_XGM/XGM/DOOCS:output
  - SPB_XTD9_XGM/XGM/DOOCS:output

13 control sources:
  - ACC_SYS_DOOCS/CTRL/BEAMCONDITIONS
  - SA1_XTD2_XGM/XGM/DOOCS
  - SPB_IRU_AGIPD1M/PSC/HV
  - SPB_IRU_AGIPD1M/TSENS/H1_T_EXTHOUS
  - SPB_IRU_AGIPD1M/TSENS/H2_T_EXTHOUS
  - SPB_IRU_AGIPD1M/TSENS/Q1_T_BLOCK
  - SPB_IRU_AGIPD1M/TSENS/Q2_T_BLOCK
  - SPB_IRU_AGIPD1M/TSENS/Q3_T_BLOCK
  - SPB_IRU_AGIPD1M/TSENS/Q4_T_BLOCK
  - SPB_IRU_AGIPD1M1/CTRL/MC1
  - SPB_IRU_AGIPD1M1/CTRL/MC2
  - SPB_IRU_VAC/GAUGE/GAUGE_FR_6
  - SPB_XTD9_XGM/XGM/DOOCS
```

1.7 If something goes wrong

A common source of errors is an invalid structure of the created data files. Hence a useful starting point to debug any errors is to check for valid file structures using the `karabo-data-validate` command:

```
$: karabo-data-validate /gpfs/exfel/exp/XMPL/201750/p700000/raw/r0273
Checking run directory: /gpfs/exfel/exp/XMPL/201750/p700000/raw/r0273
No problems found
```

The command checks if:

- All .h5 files in a run can be opened, and the run contains at least one usable file.

- The list of train IDs in a file has no zeros except for padding at the end.
- Each train ID in a file is greater than the one before it.
- The indexes do not point to data beyond the end of a dataset.
- The indexes point to the start of the dataset, and then to successive chunks for successive trains, without gaps or overlaps between them.

1.8 Scenario II: Extracting 1D Data

Photon flux *time-series* (by trainID)

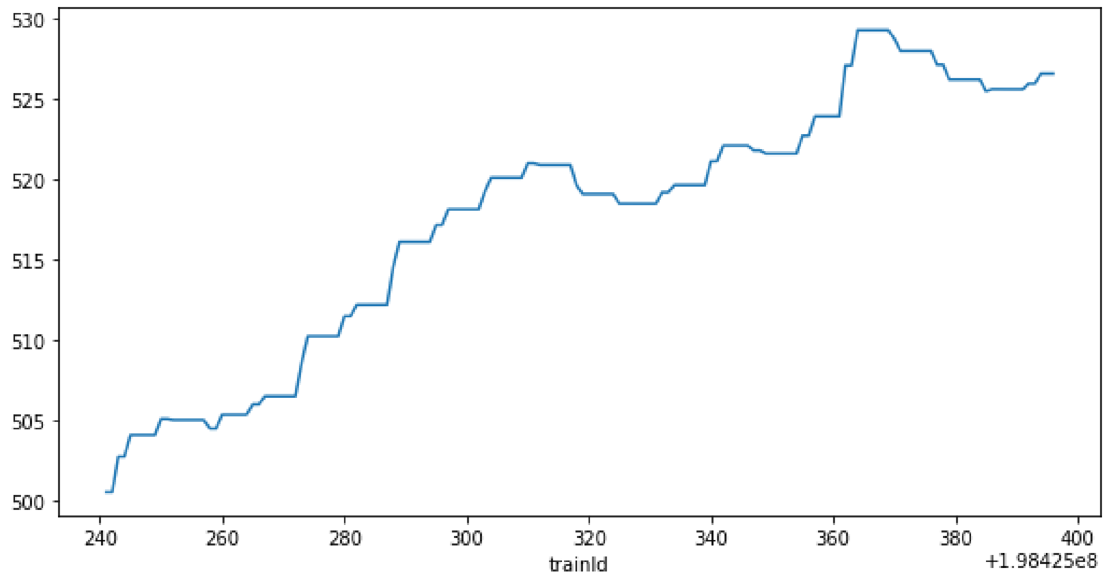
- How can 1D data be extracted and plotted?

The *get_series* method can extract a series across trainID's for a given device and property:

```
In [18]: #Live-demo
         ph_flux = run_dir.get_series('SA1_XTD2_XGM/XGM/DOOCS', 'pulseEnergy.photonFlux.value')
         ph_flux.head()
```

```
Out[18]: trainId
         198425241    500.519470
         198425242    500.519470
         198425243    502.727203
         198425244    502.727203
         198425245    504.070953
         Name: SA1_XTD2_XGM/XGM/DOOCS/pulseEnergy.photonFlux, dtype: float32
```

```
In [19]: #Live-demo
         fig = plt.figure(figsize=(10,5))
         _ = ph_flux.plot()
```



- Multiple devices and properties can be can be combined using the *get_dataframe* method:

```
In [20]: #Live-demo
         fluxes_pos = run_dir.get_dataframe(fields=[("*_XGM/*", "*.photonFlux"),
                                                    ("*_XGM/*", "*.i[xy]Pos")])
         fluxes_pos.head(10)
```

```
Out[20]:            SPB_XTD9_XGM/XGM/DOOCS/pulseEnergy.photonFlux  \
         trainId
         198425241                                     404.392822
         198425242                                     404.392822
         198425243                                     404.392822
         198425244                                     404.392822
         198425245                                     404.392822
         198425246                                     404.392822
         198425247                                     404.665680
         198425248                                     404.665680
         198425249                                     406.852539
         198425250                                     406.852539

                    SPB_XTD9_XGM/XGM/DOOCS/beamPosition.iyPos  \
         trainId
         198425241                                  -3.121433
         198425242                                  -3.121433
         198425243                                  -3.121433
         198425244                                  -3.090523
         198425245                                  -3.090523
         198425246                                  -3.090523
         198425247                                  -3.090523
         198425248                                  -3.090523
         198425249                                  -3.090523
         198425250                                  -3.090523

                    SPB_XTD9_XGM/XGM/DOOCS/beamPosition.ixPos  \
         trainId
         198425241                                   5.512009
         198425242                                   5.512009
         198425243                                   5.512009
         198425244                                   5.528512
         198425245                                   5.528512
         198425246                                   5.528512
         198425247                                   5.528512
         198425248                                   5.528512
         198425249                                   5.528512
         198425250                                   5.528512

                    SA1_XTD2_XGM/XGM/DOOCS/pulseEnergy.photonFlux  \
         trainId
```

```
198425241                                          500.519470
198425242                                          500.519470
198425243                                          502.727203
198425244                                          502.727203
198425245                                          504.070953
198425246                                          504.070953
198425247                                          504.070953
198425248                                          504.070953
198425249                                          504.070953
198425250                                          505.071930

           SA1_XTD2_XGM/XGM/DOOCS/beamPosition.iyPos  \
trainId
198425241                                        0.315761
198425242                                        0.315761
198425243                                        0.315761
198425244                                        0.341187
198425245                                        0.341187
198425246                                        0.341187
198425247                                        0.341187
198425248                                        0.341187
198425249                                        0.341187
198425250                                        0.341187

           SA1_XTD2_XGM/XGM/DOOCS/beamPosition.ixPos
trainId
198425241                                        1.293711
198425242                                        1.293711
198425243                                        1.293711
198425244                                        1.336566
198425245                                        1.336566
198425246                                        1.336566
198425247                                        1.336566
198425248                                        1.336566
198425249                                        1.336566
198425250                                        1.336566
```

1.8.1 What is *pandas*

get_dataframe and *get_series* return **pandas** objects which are extremely useful for extensive data analysis tasks. More information is available under https://pandas.pydata.org.

1.9 Scenario III: Getting data with multiple values per train

XGM intensity data is pulse resolved

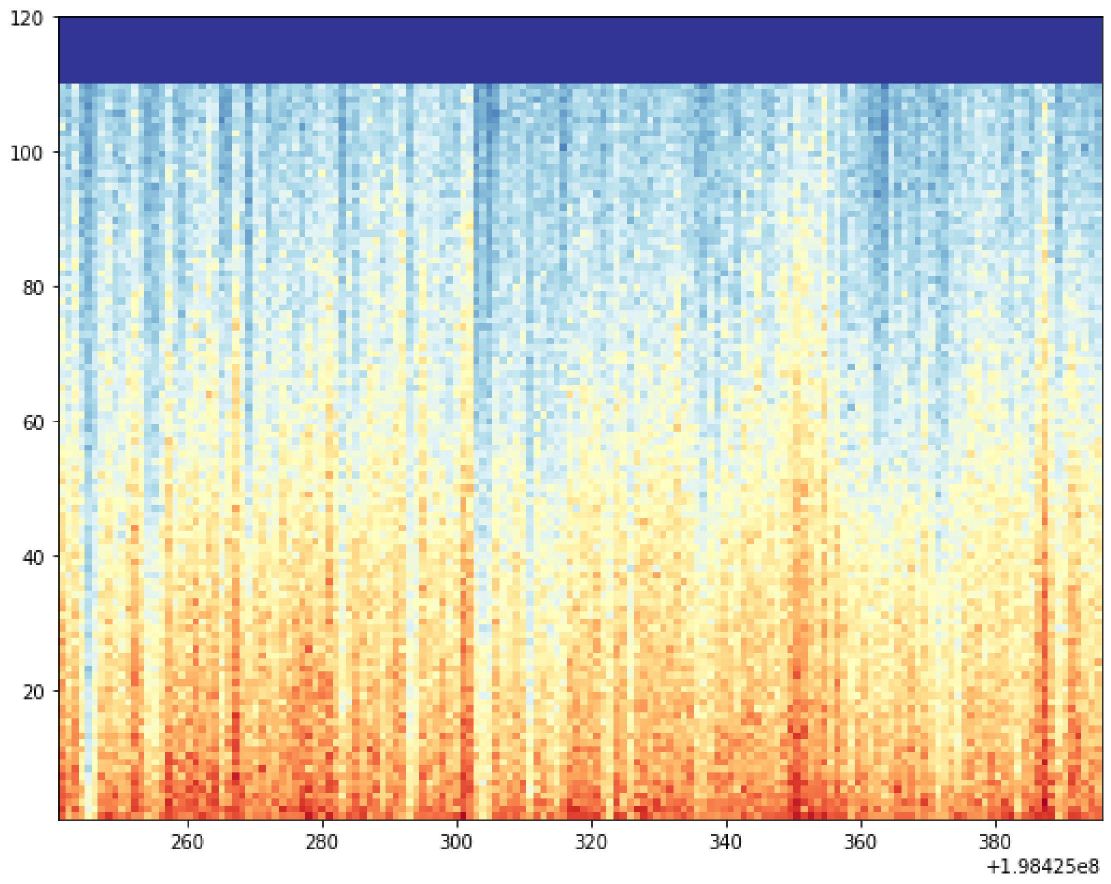
- How can this 2D data be extracted and plotted?

the *get_array* method returns a data array that contains more than one value per train. XGM intensity data is pulse resolved and serves as an example.

```
In [21]: xgm_intensity = run_dir.get_array('SA1_XTD2_XGM/XGM/DOOCS:output', 'data.intensityTD',
                                            extra_dims=['pulseId'])
         xgm_intensity
```

```
Out[21]: <xarray.DataArray (trainId: 156, pulseId: 1000)>
         array([[ 957.0532 , 1026.0005 ,  949.8755 , ...,    0.     ,    0.     ,
                    0.     ],
                [ 763.8806 ,  794.2738 ,  868.2455 , ...,    0.     ,    0.     ,
                    0.     ],
                [ 859.37   ,  995.1641 ,  838.5669 , ...,    0.     ,    0.     ,
                    0.     ],
                ...,
                [ 945.2731 ,  812.4336 ,  839.45654, ...,    0.     ,    0.     ,
                    0.     ],
                [ 903.26855,  940.15125,  953.9436 , ...,    0.     ,    0.     ,
                    0.     ],
                [ 944.08386,  949.549  ,  861.7509 , ...,    0.     ,    0.     ,
                    0.     ]], dtype=float32)
         Coordinates:
           * trainId  (trainId) uint64 198425241 198425242 ... 198425395 198425396
         Dimensions without coordinates: pulseId
```

```
In [22]: #Live-demo
         fig = plt.figure(figsize=(10,8))
         _= plt.imshow(xgm_intensity[:,:120].T,
                       extent=(xgm_intensity.trainId[0], xgm_intensity.trainId[-1], 1, 120),
                       origin='lower', cmap='RdYlBu_r')
```



a labeled array (**xarray**) is returned. More information on labeled arrays can be found on http://xarray.pydata.org

1.10 Scenario IV: Combine data and save new dataset to individual files

Suppose you want to compare data across different runs or combine various selections and then save the combined dataset to a new file.

This task could be realized with the **combine** and **write** method. This involves two different steps:

- combining
- file creation

```
In [20]: #Live-demo file combining
         xgm_intensities = run_dir.select('*/XGM/*','data.intensityTD')
         xgm_pos = run_dir.select('*/XGM/*', '*Pos')
         xgm_union = xgm_pos.union(xgm_intensities)
```

Yet another convenient way of showing the content of a selection is the **selection** method:

```
In [21]: xgm_union.selection
```

```
Out[21]: {'SA1_XTD2_XGM/XGM/DOOCS': {'beamPosition.ixPos.value',
           'beamPosition.iyPos.value'},
          'SA1_XTD2_XGM/XGM/DOOCS:output': {'data.intensityTD'},
          'SPB_XTD9_XGM/XGM/DOOCS': {'beamPosition.ixPos.value',
           'beamPosition.iyPos.value'},
          'SPB_XTD9_XGM/XGM/DOOCS:output': {'data.intensityTD'}}
```

```
In [ ]: xgm_union.write('/gpfs/exfel/data/scratch/xmpl/xmpl_xgm_subset.hdf5')
```

```
$: file /gpfs/exfel/data/scratch/xmpl/xmpl_xgm_subset.hdf5
gpfs/exfel/data/scratch/xmpl/xmpl_xgm_subset.hdf5: Hierarchical Data Format (version 5) data
```

1.11 How to install/get karabo-data?

Karabo-data is available on GitHub and there are multiple ways to install it: * it is already available on the max-well cluster nothing has to be done * it can be install using *pip* &rarr `pip install (--user) karabo_data` (preferred way) * the latest version could be downloaded from GitHub &rarr `git clone https://github.com/European-XFEL/karabo_data.git`

```
In [ ]:
```