

Peak Finding Module

Content

- [Content](#)
- [Example for module NDarrDropletFinder](#)
 - [Script functionality](#)
 - [Remarks about parameters](#)
 - [Running script with psana](#)
 - [Format of the ndarray of droplets](#)
 - [Access droplet data in python](#)
- [Examples for exp=cxif5315:run=165](#)
 - [Example 1](#)
 - [Example 2](#)
 - [Example 3](#)
- [Get latest version](#)
- [References](#)

Example for module NDarrDropletFinder

See description of the module in [Module ImgAlgos::NDarrDropletFinder](#)

This module is a more generic implementation of the [Module ImgAlgos::ImgPeakFinder](#) in case of data segmented for multiple sensors, like in cspad.

The minimal configuration file for psana may look like `psana-xppd7114-r0081-cspad2x2-NDarrDropletFinder.cfg`

```

[psana]
#calib-dir = /reg/d/psdm/XPP/xppd7114/calib - is used by default
files = exp=xppd7114:run=81
events = 10
modules = CSPadPixCoords.CSPad2x2NDArrProducer \
          ImgAlgos.NDArrCalib \
          CSPadPixCoords.CSPad2x2NDArrReshape \
          ImgAlgos.NDArrDropletFinder

[CSPadPixCoords.CSPad2x2NDArrProducer]
source = XppGon.0:Cspad2x2.0
inkey =
outkey = nda_raw
outtype = int16
print_bits = 3

[ImgAlgos.NDArrCalib]
source = XppGon.0:Cspad2x2.0
key_in = nda_raw
key_out = nda_clb:as_data
do_peds = yes
do_cmod = yes
do_stat = yes
do_mask = no
do_bkgd = no
do_gain = no
do_nrms = no
do_thre = no
fname_mask =
fname_bkgd =
masked_value = 0
threshold_nrms = 3
threshold = 100
below_thre_value = 0
bkgd_ind_min = 0
bkgd_ind_max = 1000
bkgd_ind_inc = 10
print_bits = 255

[CSPadPixCoords.CSPad2x2NDArrReshape]
source = XppGon.0:Cspad2x2.0
keys_in = nda_clb:as_data
print_bits = 0

[ImgAlgos.NDArrDropletFinder]
source = XppGon.0:Cspad2x2.0
key = nda_clb
key_droplets = nda_droplets
key_smeared = nda_sme
threshold_low = 1
threshold_high = 3
sigma = 1.5
smear_radius = 3
peak_radius = 3
low_value = -0.1
windows = 0 10 160 20 350 \
          1 10 160 10 150 \
          1 30 170 220 370
fname_prefix = test
print_bits = 511

```

which defines parameters of a few modules with functionality as follows.

Script functionality

- `CSPadPixCoords.CSPad2x2NDArrProducer` - get raw data for `XppGon.0:Cspad2x2.0` and put them in the event store as ndarray with key `nda_raw`
- `ImgAlgos.NDArrCalib` - get ndarray with key `nda_raw`, applies a bunch of intensity corrections, and saves in the event store the ndarray with key `nda_clb:as_data`
- `CSPadPixCoords.CSPad2x2NDArrReshape` - re-shuffle `cspad2x2` pixels shaped as data `[185,388,2]` to "natural" ndarray format `[2,185,388]`
- `ImgAlgos.NDArrDropletFinder` - gets calibrated and shaped as two 2-d segments ndarray `nda_clb`, applies smearing (if `sigma>0`), find droplets (wide peaks), and saves them in the event store and in the file (if `fname_prefix` is not empty).

Remarks about parameters

- For each new set of smearing parameters (if applied, `sigma>0`) the droplet-finder parameters need to be re-tuned.
- Number of windows per sensor is unlimited but they should not overlap each other in order to get rid of duplication of found peaks in the overlapping regions.
- Parameter `low_value = 0.1` is set for demonstration purpose - in order to distinguish on plot sensors' pixels from image bins.
- Parameter `fname_prefix = test` is set in demonstration purpose in order to save image and peak data files for browser

Running script with psana

This file can be executed by the command

```
psana -c psana-xppd7114-r0081-cspad2x2-NDArrDropletFinder.cfg
```

This job generates a bunch of files for all events with names like `test-xppd7114-r0081-e000006-<suffix>.txt`, where `<suffix>` stands for raw (data), smeared (if requested), and peaks. Using browser `ImgAlgos/data/PlotNDArrayAndPeaks.py` launched by the command (from release directory):

```
ImgAlgos/data/PlotNDArrayAndPeaks.py -i test-xppd7114-r0081-e000009-smeared.txt
### or ###
ImgAlgos/data/PlotNDArrayAndPeaks.py -i test-xppd7114-r0081-e000009-raw.txt
```

one may get images like:

Fig.1: Image of `cspad2x2`, containing two 2x1 segments. Droplet-finder is applied to the calibrated and smeared data in three windows.

Format of the ndarray of droplets

Content of the record for each droplet is defined by the structure of six components

```
struct Droplet{
    unsigned seg;    // segment index (in 3-d ndarray of data) of the found droplet
    double row;      // row index of the droplet center
    double col;      // columnindex of the droplet center
    double ampmax;    // amplitude in the droplet maximum
    double amptot;    // total amplitude in the range of peak_radius
    unsigned npix;    // number of pixels in the range of peak_radius
};
```

The shape of the ndarray with table of droplets is [`<number-of-droplets>`, 6].

File with droplets data is a text table of the same parameters, for example

```
0      127      137      5.01175  60.1614  20
0      129      166      4.55816  52.7998  20
0      155      131      4.09197  48.5101  20
1      105      17       3.65102  17.5217   6
1      145      49       3.87695  30.5501  15
1      116      249      3.07719  20.5181  11
1      139      325      3.60641  60.2324  27
...
```

Access droplet data in python

Example of the python script `psana-xppd7114-r0081-cspad2x2-NDArrDropletFinder.py`:

```
#!/usr/bin/env python

import sys
import numpy as np
import psana

psana.setConfigFile('psana-xppd7114-r0081-cspad2x2-NDArrDropletFinder.cfg')

dsname = 'exp=xppd7114:run=81'
print ""Data source: %s"" % dsname
ds = psana.DataSource(dsname)

evnum_max = 50

#-----

for evnum, evt in enumerate(ds.events()) :
    evtid = evt.get(psana.EventId)
    if evnum > evnum_max : break

    nda_droplets = evt.get(psana.ndarray_float32_2, psana.Source('DetInfo(XppGon.0:Cspad2x2.0)'),
'nda_droplets')
    nda_smeared = evt.get(psana.ndarray_float64_3, psana.Source('DetInfo(XppGon.0:Cspad2x2.0)'), 'nda_sme')
    nda_calibrated = evt.get(psana.ndarray_float64_3, psana.Source('DetInfo(XppGon.0:Cspad2x2.0)'), 'nda_clb')

    print 50*','=', '\nEvent: %d' % evnum
    if (nda_smeared is not None) : print 'nda_smeared.shape = ', nda_smeared.shape

    if (nda_calibrated is not None) : print 'nda_calibrated.shape = ', nda_calibrated.shape

    if (nda_droplets is not None) :
        print 'nda_droplets.shape = ', nda_droplets.shape
        for droplet in nda_droplets :
            seg, row, col, amax, atot, npix = droplet
            print '    seg:%2d row:%3d col:%3d amax:%8.1f atot:%8.1f npix:%2d' % \
                (seg, row, col, amax, atot, npix)
```

For each event in the data set this script executes `psana-xppd7114-r0081-cspad2x2-NDArrDropletFinder.cfg` then gets its results as numpy arrays `nda_droplets` and `nda_smeared` and prints them.

This script can be executed by the command

```
./psana-xppd7114-r0081-cspad2x2-NDArrDropletFinder.py
```

In order to eliminate too excessive printout from psana modules, set parameters `print_bits = 0` or `1` in the configuration file.

Examples for exp=cxif5315:run=165

Example 1

Shows how to get ndarray with CSPAD data in psana, calibrate it, find peaks (droplets) in imaging data and plot them offline.

Configuration file: [psana-cxif5315-r0165-cspad-ds2-NDArrDropletFinder.cfg](#)

ROI Mask: [roi_mask_nda.txt](#)

Command to run job:

```
psana -c psana-cxif5315-r0165-cspad-ds2-NDArrDropletFinder.cfg
```

Command to see results:

```
ImgAlgos/data/PlotNDArrayAndPeaks.py -g /reg/d/psdm/CXI/cxif5315/calib/CsPad::CalibV1/CxiDs2.0:Cspad.0/geometry
/l-end.data -i test-cxif5315-r0165-e000001-raw.txt
```

Example 2

Shows how to get ndarray with CSPAD data in psana, calibrate it, prepare and apply mask, find peaks (droplets) in imaging data and plot them offline.

Configuration file: [psana-cxif5315-r0165-cspad-ds2-NDArrDropletFinder-v2.cfg](#)

ROI Mask: [roi_mask_nda-v2.txt](#) is produced in [calibman](#) (tab ROI):

image without mask for event 10: (dark is subtracted).

mask production stages in the mask editor:

Use this mask-ndarray [roi_mask_nda-v2.txt](#) in the NDArrDropletFinder with additional list of segment windows:

```
windows = 0 0 185 0 388 \
           1 0 185 0 388 \
           8 0 185 0 388 \
           9 0 185 0 388 \
          16 0 185 0 388 \
          17 0 185 0 388 \
          24 0 185 0 388 \
          25 0 185 0 388
```

which contains 8 closest to the beam 2x1 sensors.

Command:

```
psana -c psana-cxif5315-r0165-cspad-ds2-NDArrDropletFinder-v2.cfg
```

produces files with image and list of peaks for all requested events, which can be viewed by the command

```
ImgAlgos/data/PlotNDArrayAndPeaks.py -g /reg/d/psdm/CXI/cxif5315/calib/CsPad::CalibV1/CxiDs2.0:Cspad.0/geometry
/l-end.data -i test-cxif5315-r0165-e000001-raw.txt
```

This viewer shows images like:

Example 3

Example 3 has the same functionality as Example 2. Also this example shows how to

- gets raw ndarray with CSPAD data, subtract pedestals and background, apply common mode correction, apply mask, find peaks,
- access psana data structures from the python script,
- reconstruct CSPAD image,
- apply selection to the list of droplets and make the list of "bold" beaks,
- skip events without "bold" peaks,
- and interactively plot images with overlayed peaks.

file with mask: [roi_mask_nda_v3.txt](#)

file with background: [bkgd-cxif5315-r0165.dat](#) obtained by averaging calibrated ndarray with module `ImgAlgos.NDArrAverage` over 1000 events:

psana configuration file: [psana-cxif5315-r0165-cspad-ds2-NDArrDropletFinder-v3.cfg](#) - [download](#)

python script for interactive psana: [psana-cxif5315-r0165-cspad-ds2-NDArrDropletFinder-v3.py.txt](#) - [download](#)

Interactive job can be launched by the command:

```
python psana-cxif5315-r0165-cspad-ds2-NDArrDropletFinder-v3.py
```

- which does all work and plots images with peaks like background is subtracted

- background is not subtracted, or

-

Get latest version

Release ana-current (ana-0.13.18 or later) has the latest version of all necessary packages.

In case of any packages updates relative to the current release, the new version of the code can be tested from local release directory (test_release_directory):

```
ssh psana
kinit
cd <one-of-your-directories>
newrel ana-current test_release_directory
cd test_release_directory
sit_setup
addpkg ImgAlgos HEAD
addpkg <other-package-name> HEAD
scons

...
psana -c psana-xppd7114-r0081-cspad2x2-NDArrDropletFinder.cfg
```



Packages PSCalib and CSPadPixCoords need to be added because they have been modified on the top of ana-0.13.13 release. They need to be compiled together due to dependency between packages.

References

- [psana - Module Catalog](#)
- [psana - Module Examples](#)
- [CSPAD2x2 Alignment](#)