Test of Peak Finders - V2

Content

- Content
- Data
- V2 News
  - Peak selection parameters
  - Summary of peak selection parameters
  - Raw n-d array pre-processing before peak-finders
  - Peak list
  - Peak list processing
    - Peak pre-selection for histogramms
      - ARC region
      - EQU region
    - Peak selection for fit
      - ARC region
      - EQU region
  - References

Data

exp=cxif5315:run=169

V2 News

V2 is done for test of peak finders after revision r1.

See for details

- Hit/Peak Finding - description of algorithms
- ImgAlgos.PyAlgos - peak finders API
- PSAS-147 - details about revision 1

We work with peak finder versions v2r1, v3r1, v4r1.

Data processing and peak finding is done in cxif5315/proc-cxif5315-r0169-data-pfvn-2016-04-19.py

Peak selection parameters

- selection parameters were set with as minimal number of parameters as possible.
- selection parameters of different peak finders were adjusted to get about the same yield of peaks in the file.
from ImgAlgos import PyAlgos

alg_arc = PyAlgos(windows=winds_arc, mask=mask_arc, pbits=2)
#alg_arc.set_peak_selection_pars(npix_min=0, npix_max=1e6, amax_thr=0, atot_thr=500, son_min=6)  # for pfv2r1
alg_arc.set_peak_selection_pars(npix_min=0, npix_max=1e6, amax_thr=0, atot_thr=0, son_min=6)     # for pfv3r1, pfv4r1

alg_equ = ...  # the same

# in the event loop:

# run peakfinders and get list of peak records for each region
#peaks_arc = alg_arc.peak_finder_v2r1(nda, thr=30, r0=6, dr=0.5)
peaks_arc = alg_arc.peak_finder_v3r1(nda, rank=5, r0=6, dr=0.5)
#peaks_arc = alg_arc.peak_finder_v4r1(nda, thr_low=10, thr_high=150, rank=5, r0=6, dr=0.5)

#peaks_equ = alg_equ.peak_finder_v2r1(...)  # The same

Summary of peak selection parameters

Peak finder specific parameters for seed peak finding

- v2: thr=30
- v3: rank=5
- v4: thr_low=10, thr_high=150, rank=5

Use the same parameters for S/N calculation

- r0=6, dr=0.5

Peak selection in the list

- common: son_min=6
- v2: atot_thr=500 # to keep the same number of peaks in the list as for v3, v4

Raw n-d array pre-processing before peak-finders

- get raw data
- subtract pedestals
- subtract radial background to polarization corrected data
- apply status mask
from pyimgalgos.RadialBkgd import RadialBkgd, polarization_factor

nda_bkgd = det.bkgd(runnum) # pre-defined n-d array with averaged background from calib/.../pixel_bkgd/...
nda_smask = det.mask(runnum, calib=False, status=True, edges=True, central=True, unbond=True, unbondnbrs=True)

mask_bkgd = nda_smask # * mask_winds_tot
rb = RadialBkgd(Xarr, Yarr, mask=mask_bkgd, radedges=(5200, 80000), nradbins=200, nphibins=1)
pf = polarization_factor(rb.pixel_rad(), rb.pixel_phi(), DIST_STOD)

# in the event loop:
da_data = det.raw(evt)
if nda_data is not None:
    nda = np.array(nda_data, dtype=np.float32, copy=True)
    nda -= nda_peds

    #det.common_mode_apply(evt, nda, cmpars=(1,50,50,100))
    #nda = subtract_bkgd(nda, nda_bkgd, mask=nda_smask, winds=winds_bkgd, pbits=0)
    nda = rb.subtract_bkgd(nda.flatten() * pf)
    nda.shape = shape_cspad

    nda *= nda_smask

Common mode correction was tested before and after background subtraction.
For unknown reason it makes image visually worse...

Peak list
In revision 1 four parameters col_min, col_max, row_min, row_max were discarded.
For each peak finder we created list of peak parameters, beginning as
Peak list processing

For peak list processing we use script:

cxif5315/proc-cxif5315-r0169-peaks-from-file-v6.py

Peak pre-selection for histogramms

ARC region
def procPeakDataArc(pk) :
    # Process peak for ARC region; accumulate peak statistics in histogram arrays.
    #
    #===================
    # discard from all histograms except its own
    sp.lst_arc_atot.append(pk.atot)
    if pk.atot<2000 : return
    #===================
    sp.lst_arc_amax.append(pk.amax)
    sp.lst_arc_npix.append(pk.npix)
    sp.lst_arc_r.append(pk.r)
    ...

Arc: Amax

Entries: 23113
Mean= 351.08 ± 28.36
RMS= 194.45 ± 20.06
γ= 1.286 ; ν= 1.083

Arc: Amax

Entries: 32284
Mean= 251.97 ± 41.12
RMS= 196.34 ± 29.22
γ= 1.668 ; ν= 2.237
Arc: Peak radius

Entries=70577
Mean=440.69 ± 0.89
RMS=4.37 ± 0.63
γ1=0.353 γ2=0.075

Arc: Peak azimuthal angle

Entries=25188
Mean=92.54 ± 2.95
RMS=16.62 ± 2.09
γ1=0.057 γ2=0.722

Arc: Peak azimuthal angle

Entries=34290
Mean=91.11 ± 2.80
RMS=19.78 ± 1.96
γ1=0.047 γ2=0.537
Arc: Number of peaks after peakfinder

Entries=80221
Mean=0.83 ± 1.03
RMS=1.60 ± 0.73
γ=4.773 χ²=39.098

Arc: Number of peaks selected

Entries=83770
Mean=0.24 ± 0.51
RMS=0.61 ± 0.36
γ=3.717 χ²=21.855

Arc: Number of peaks selected

Entries=83366
Mean=0.11 ± 0.34
RMS=0.37 ± 0.24
γ=3.839 χ²=17.887
def procPeakDataEqu(pk):
    """ Process peak for EQU region; accumulate peak data """
    #===================
    # discard from all histograms except its own
    sp.lst_equ_atot.append(pk.atot)
    if pk.atot<2000 : return
    sp.lst_equ_r_raw.append(pk.r)
    if pk.r<100 : return
    #===================
    sp.lst_equ_r.append(pk.r)
    sp.lst_equ_amax.append(pk.amax)
    sp.lst_equ_npix.append(pk.npix)
    ...

EQU region

Equ: Amax
def peakIsSelectedArc(pk):
    # Apply peak selection criteria to each peak from file
    if pk.son<9     : return False
    if pk.amax<150  : return False
    if pk.atot<2000 : return False
    if pk.npix>500  : return False
    if pk.r<435     : return False
    if pk.r>443     : return False
    if pk.rms>80    : return False
    if pk.bkgd<-20  : return False
    if pk.bkgd>50   : return False
    return True
Arc: dr between 2 peaks

Entries: 2924
Mean: 0.43 ± 0.71
RMS: 1.64 ± 0.50
\(\chi^2 = 0.132\) \(\nu=0.617\)

Entries: 1031
Mean: 0.33 ± 0.69
RMS: 1.49 ± 0.48
\(\chi^2 = 0.329\) \(\nu=1.304\)

Entries: 4607
Mean: 0.17 ± 0.83
RMS: 2.25 ± 0.58
\(\chi^2 = 0.203\) \(\nu=0.375\)
To fit peaks we use `funcy_l1_v0(x, phi_deg, bet_deg, DoR=433/sp.DETD, sgnrt=-1.)`
def peakIsSelectedEqu(pk) :
    """Apply peak selection criteria to each peak from file
    """
    if pk.son<9     : return False
    if pk.amax<150  : return False
    if pk.atot<2000 : return False
    if pk.npix>500  : return False
    if pk.r<100     : return False
    if pk.r>454     : return False
    if pk.rms>80    : return False
    if math.fabs(pk.bkgd)>20 : return False
    return True

To fit peaks we use *funcy_l0* which automatically select solution depending on sign of parameter B.
References

- Hit/Peak Finding Details - description of algorithms
- ImgAlgos.PyAlgos - interface methods
- PSAS-147 - details about revision 1
- Radial Background Subtraction Algorithm