

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 histograms](#)
 - [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.PyAlgos      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: atol_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:
nda_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

# Exp	Run	Date	Time	time(sec)	time(nsec)	fiduc	Evnum	Reg	Seg	Row	Col	Npix	
Amaz	Atot	rcent	ccent	rsigma	csigma	bkgd	rms	son	imrow	imcol	x[um]	y[um]	r
[um]	phi[deg]												
cxif5315	169	2015-02-22	02:20:47	1424600447	478050876	104418		0	ARC	7	135	366	109
121.5	2327.8	134.6	365.6	2.71	2.93	-13.24	19.66	11.34	152	638		7357	46871
47445	81.08												
cxif5315	169	2015-02-22	02:20:47	1424600447	478050876	104418		0	ARC	8	21	170	91
133.1	1513.3	22.3	170.5	2.41	2.92	-9.00	20.34	7.80	149	434		-15104	47246
49601	107.73												
cxif5315	169	2015-02-22	02:20:47	1424600447	478050876	104418		0	ARC	8	76	240	84
113.2	1167.8	74.3	240.6	2.25	2.87	-7.49	20.67	6.17	204	361		-23143	41219
47272	119.31												
cxif5315	169	2015-02-22	02:20:47	1424600447	478050876	104418		0	EQU	1	162	30	88
247.7	1687.0	160.8	29.7	2.21	1.37	9.43	26.64	6.75	574	640		7582	550
7601	4.15												
cxif5315	169	2015-02-22	02:20:47	1424600447	478050876	104418		0	EQU	17	93	16	121
69.1	906.7	93.1	16.0	2.77	2.70	-5.91	11.99	6.88	552	423		-16334	2911
16592	169.89												
cxif5315	169	2015-02-22	02:20:47	1424600447	486382070	104421		1	ARC	8	6	65	121
123.2	1536.9	5.7	64.5	2.90	3.13	-7.68	22.41	6.23	134	539		-3559	48867
48996	94.17												
cxif5315	169	2015-02-22	02:20:47	1424600447	486382070	104421		1	ARC	8	9	20	121
97.4	2003.4	8.7	19.7	2.75	2.93	-9.76	20.46	8.90	137	584		1387	48525
48545	88.36												
cxif5315	169	2015-02-22	02:20:47	1424600447	486382070	104421		1	ARC	8	19	157	111
83.1	1616.3	19.1	156.5	2.89	2.93	-8.70	16.42	9.34	147	447		-13675	47462
49393	106.07												
cxif5315	169	2015-02-22	02:20:47	1424600447	486382070	104421		1	EQU	16	103	235	121
88.7	1923.3	102.9	235.0	3.18	2.88	-13.77	10.90	16.04	774	228		-37664	-21466
43351	-150.32												
cxif5315	169	2015-02-22	02:20:47	1424600447	503058551	104427		3	ARC	8	6	33	121
111.8	2222.4	5.3	32.7	3.19	3.05	-7.85	27.39	7.38	134	571		-41	48858
48858	90.05												
cxif5315	169	2015-02-22	02:20:47	1424600447	503058551	104427		3	ARC	8	43	195	99
120.1	1640.6	43.7	195.0	2.86	3.36	-10.60	24.00	6.87	171	406		-18188	44835
48383	112.08												
cxif5315	169	2015-02-22	02:20:47	1424600447	503058551	104427		3	EQU	17	64	130	121
88.0	1342.3	64.0	129.7	3.08	2.78	-5.90	14.02	8.71	666	393		-19542	-9614
21779	-153.80												
...													

Peak list processing

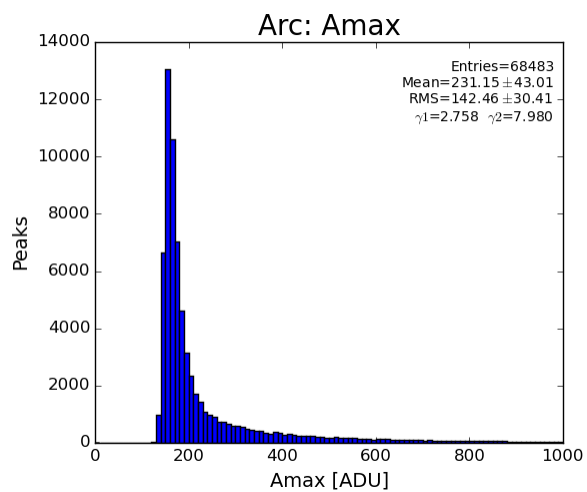
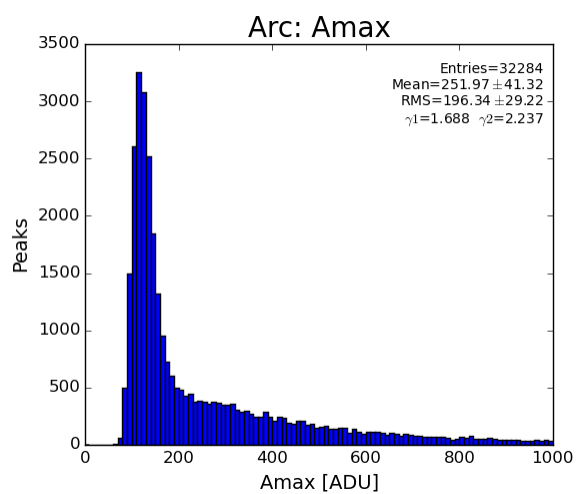
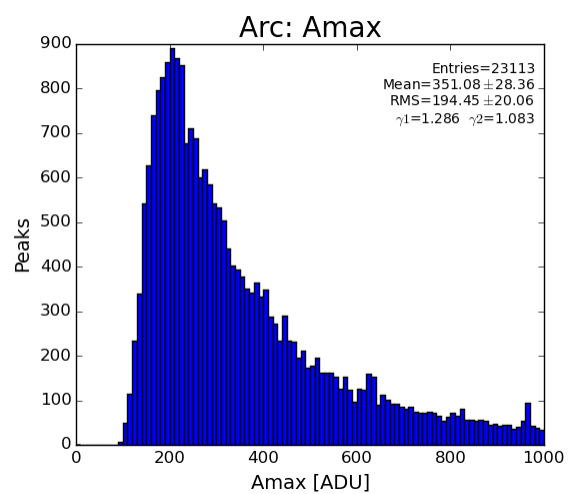
For peak list processing we use script:

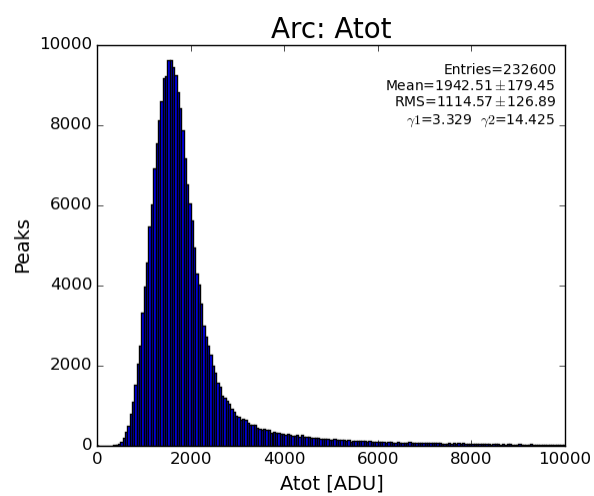
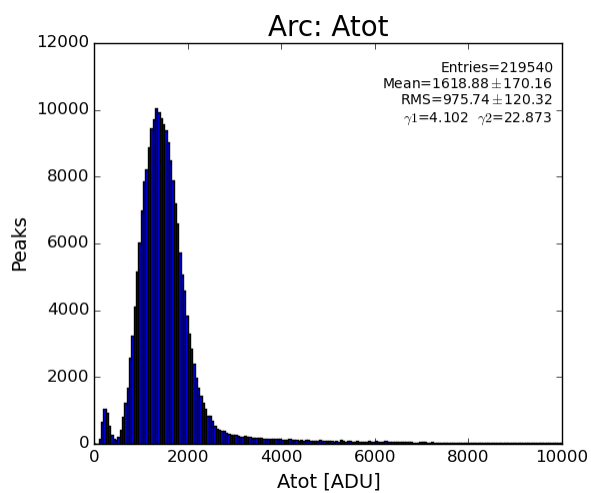
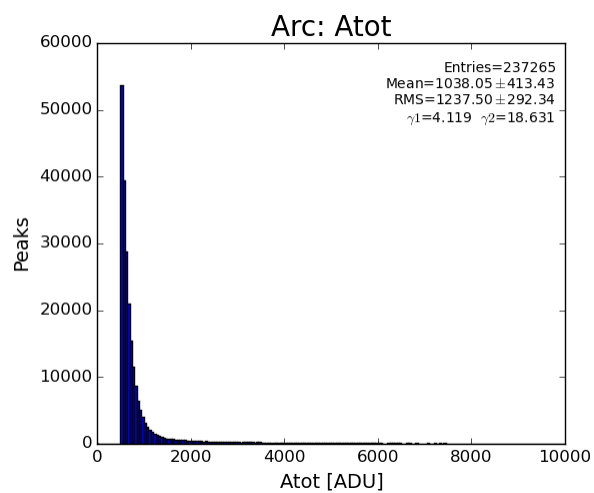
```
cxif5315/proc-cxif5315-r0169-peaks-from-file-v6.py
```

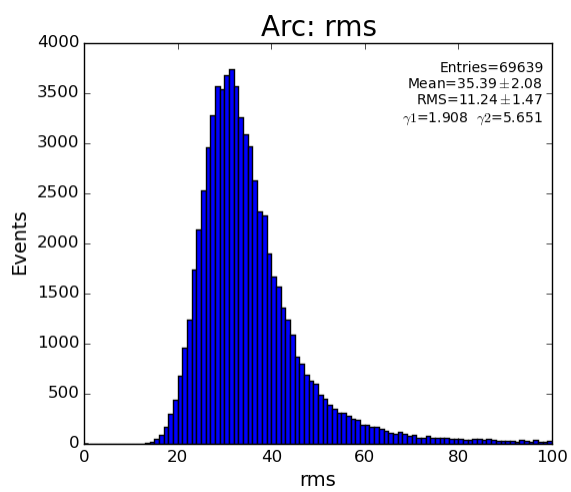
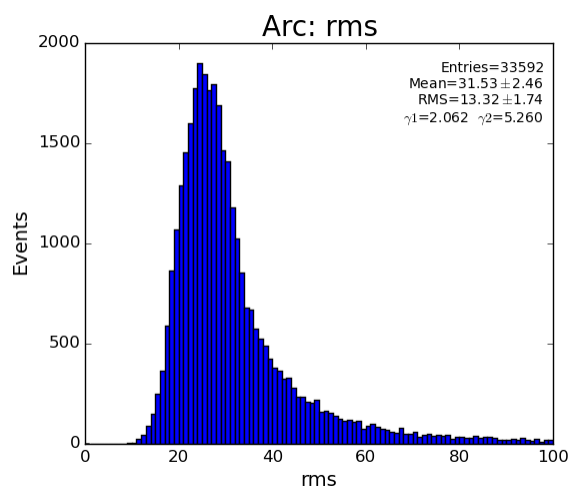
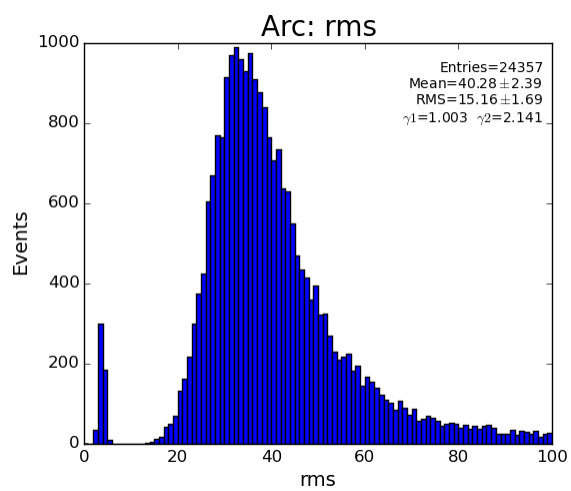
Peak pre-selection for histograms

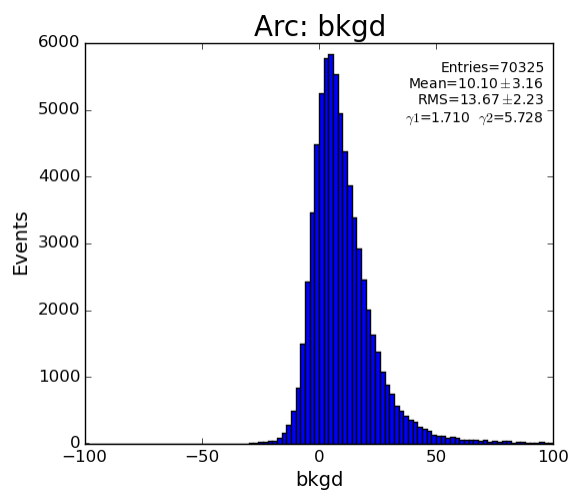
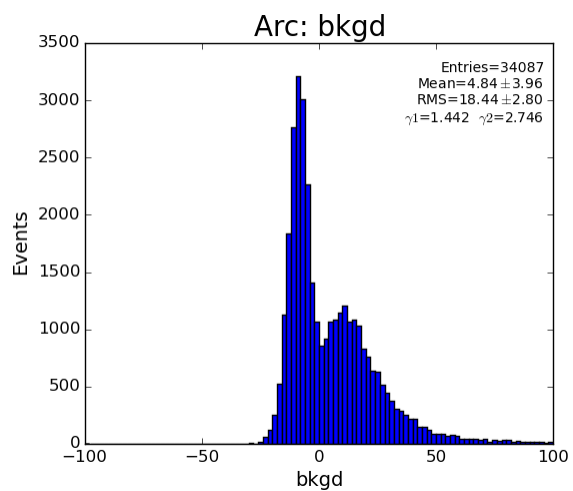
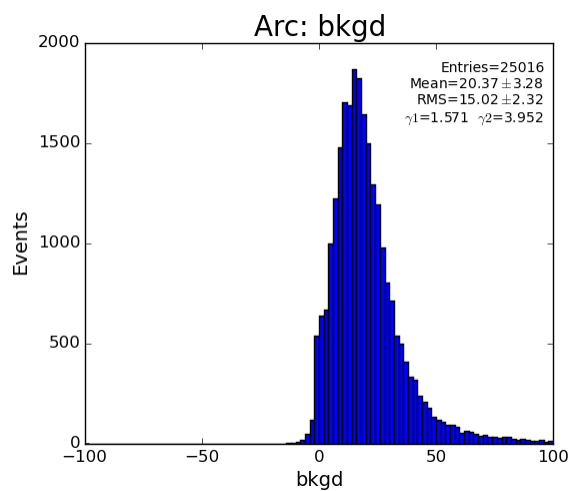
ARC region

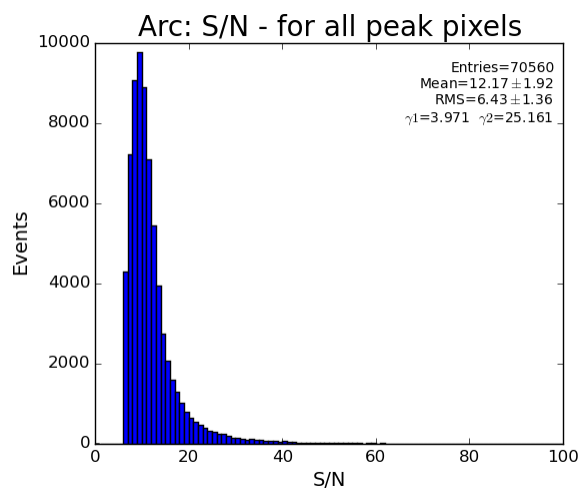
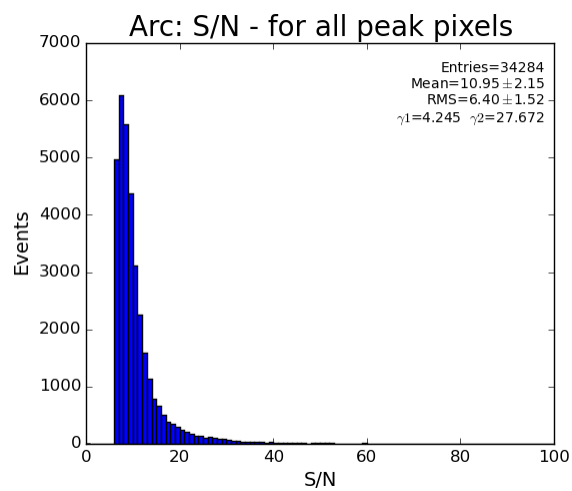
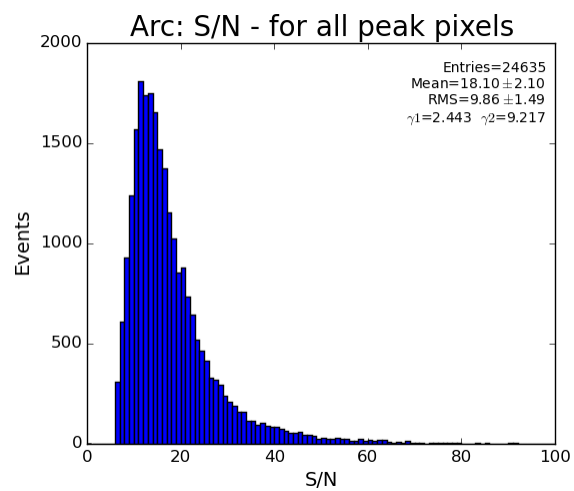
```
def procPeakDataArc(pk) :
    """ Process peak for ARC region; accumulate peak statistics in histogram arrays.
    """
    #=====
    # discard from all histograms except its own
    sp.lst_arc_atot.append(pk.atot)
    if pk.atot<2000 : return
    #=====
    sp.lst_arc_amax.append(pk.amax)
    sp.lst_arc_npix.append(pk.npix)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

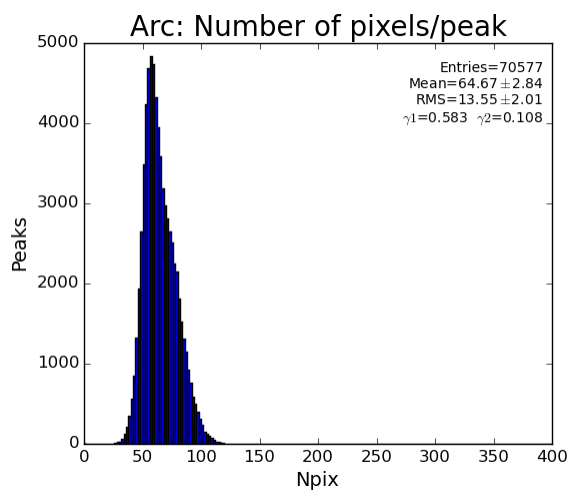
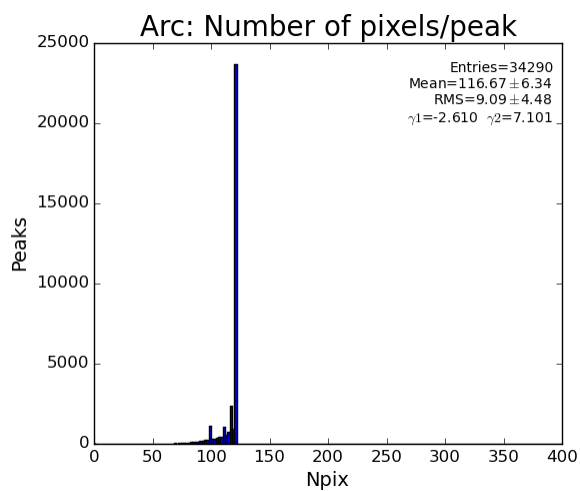
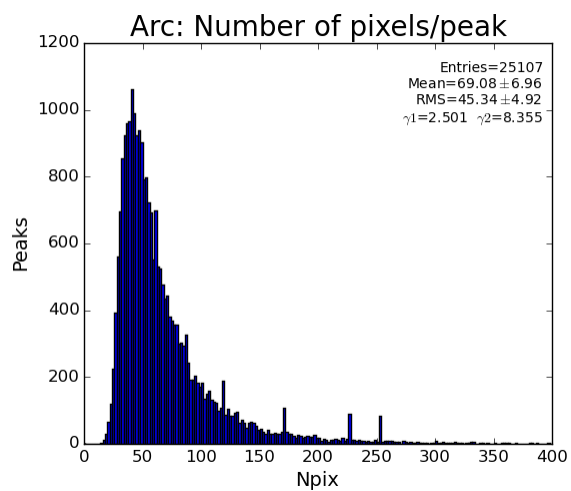


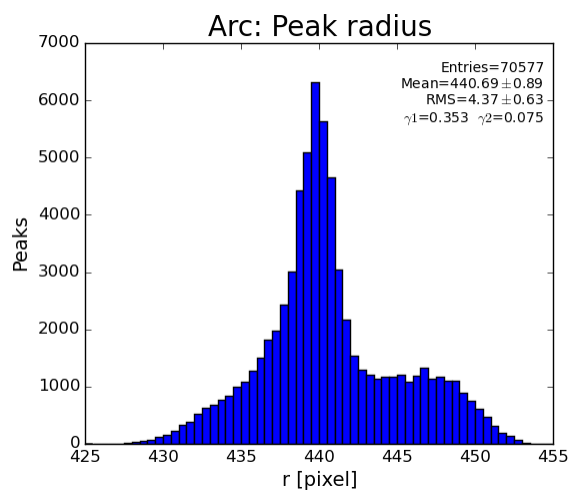
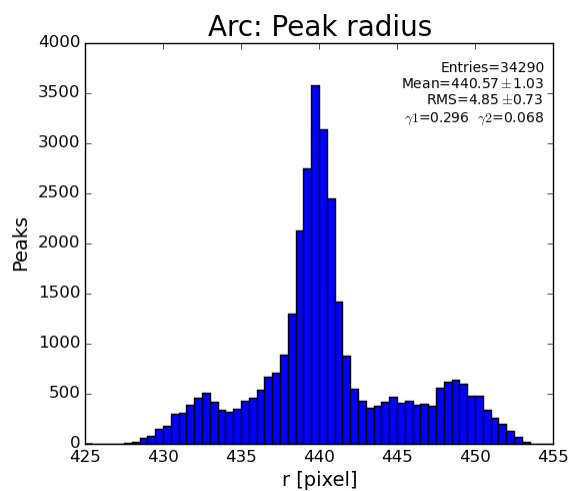
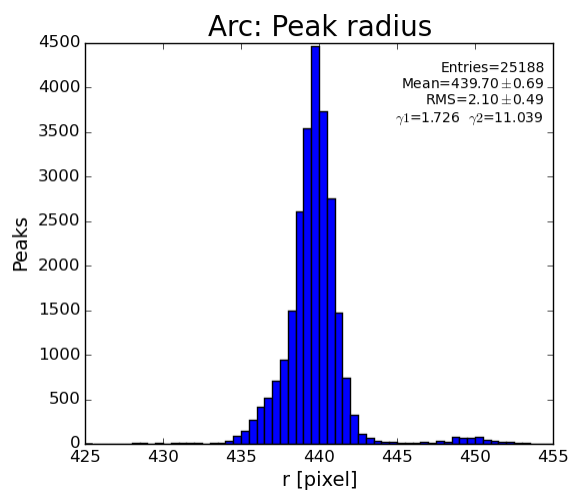


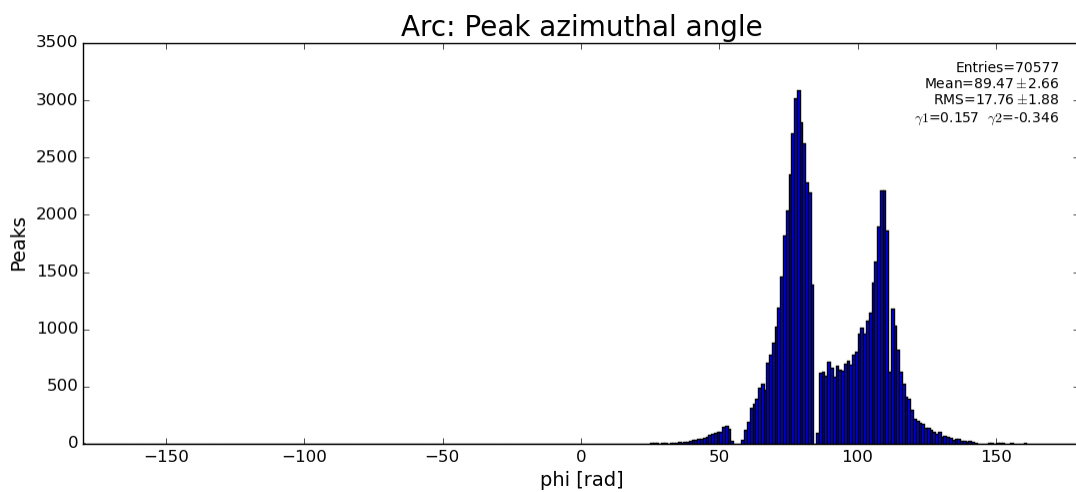
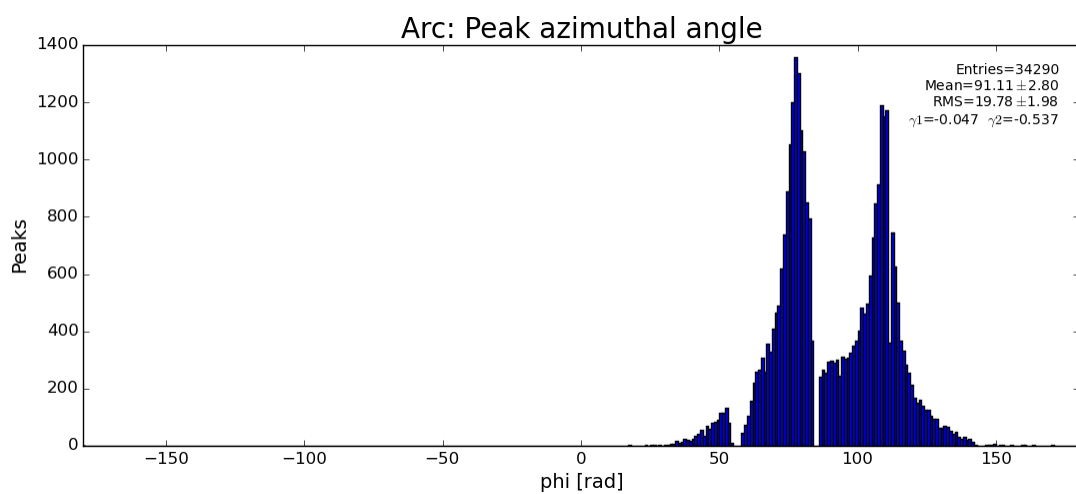
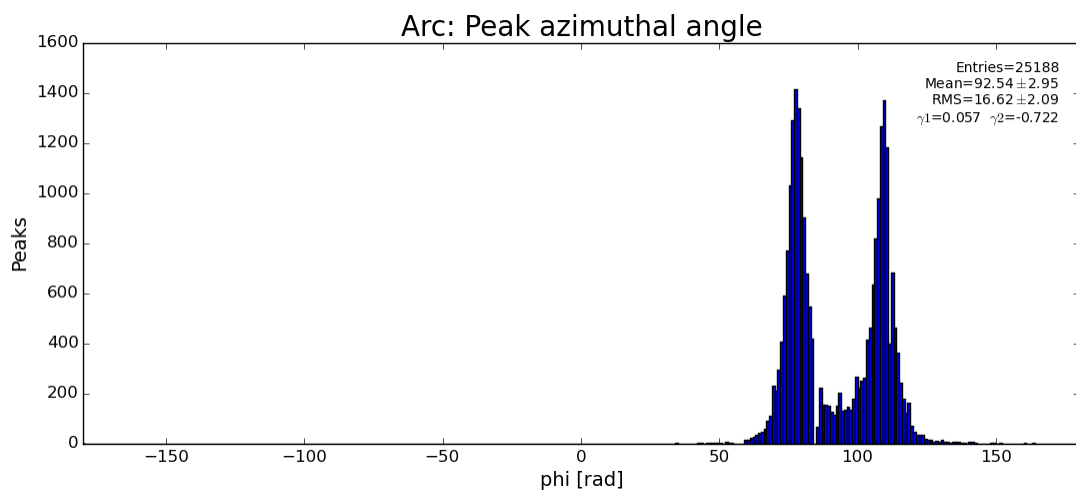




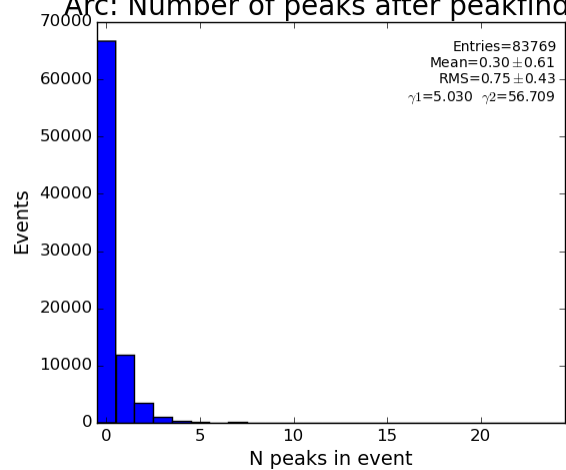




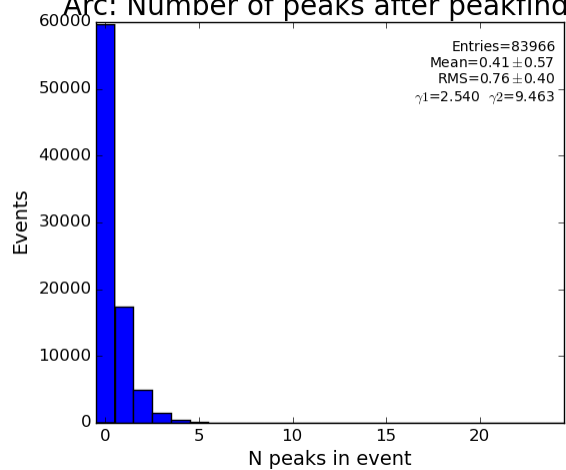




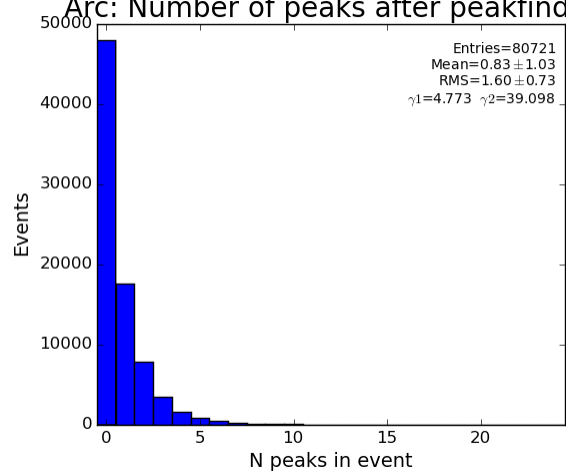
Arc: Number of peaks after peakfinder

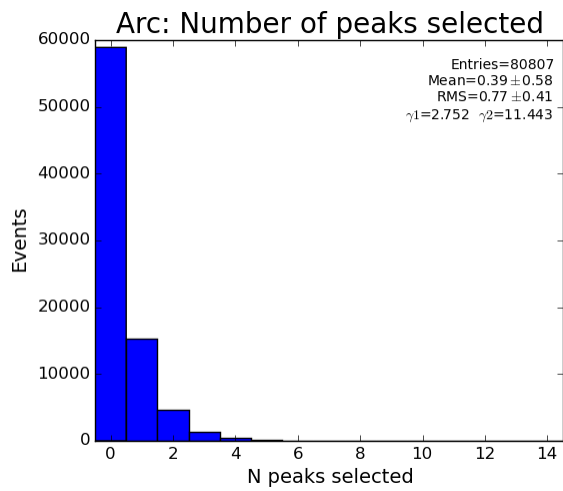
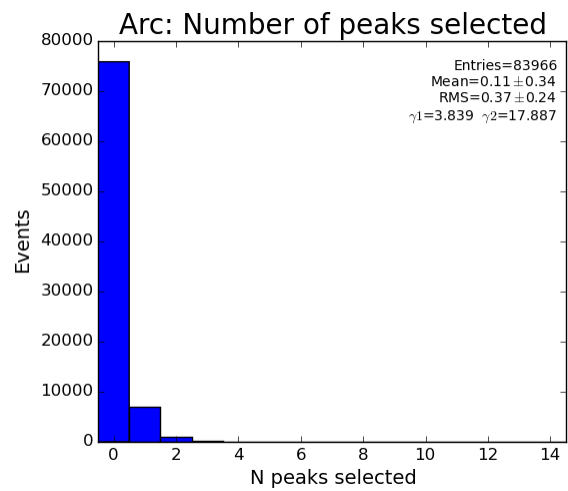
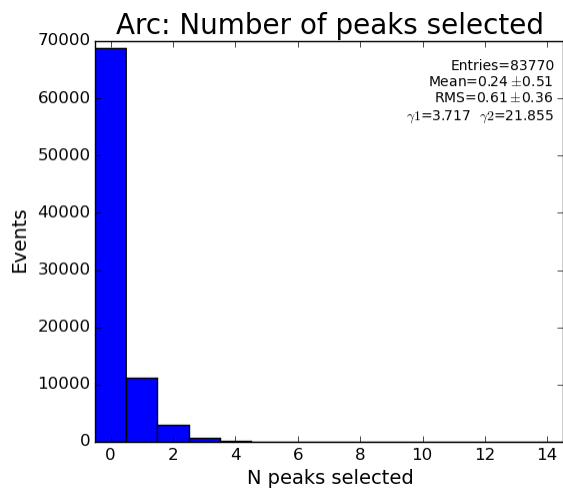


Arc: Number of peaks after peakfinder



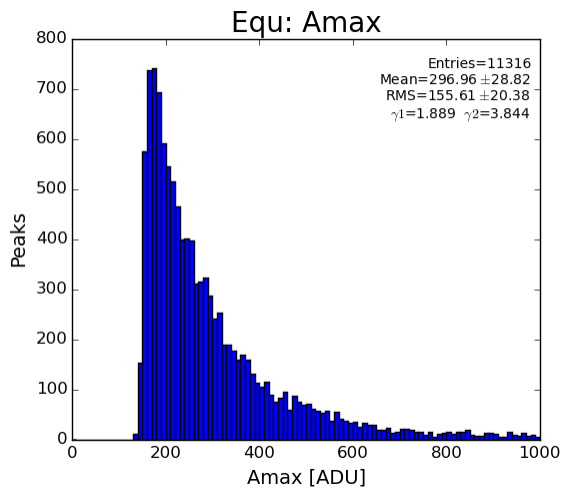
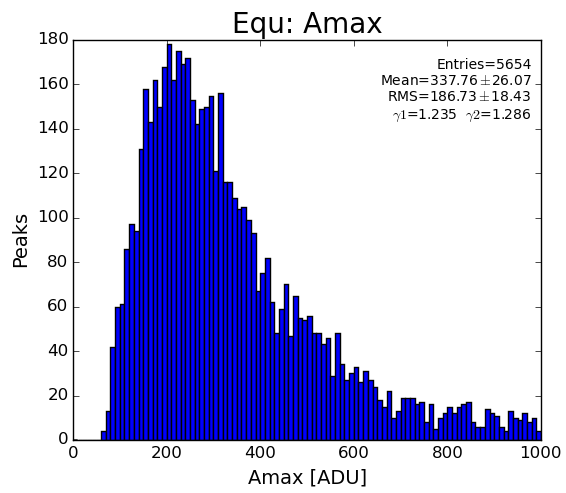
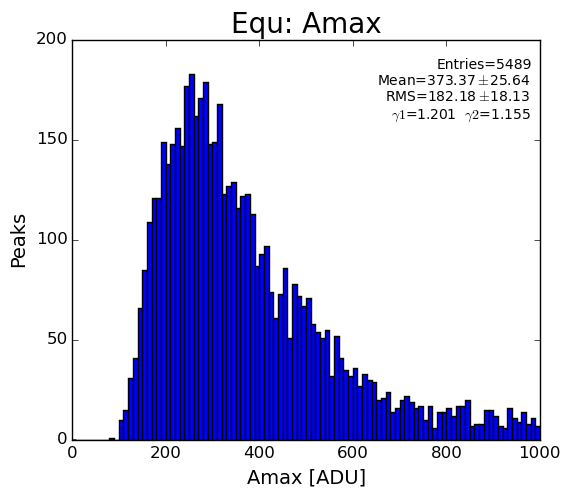
Arc: Number of peaks after peakfinder

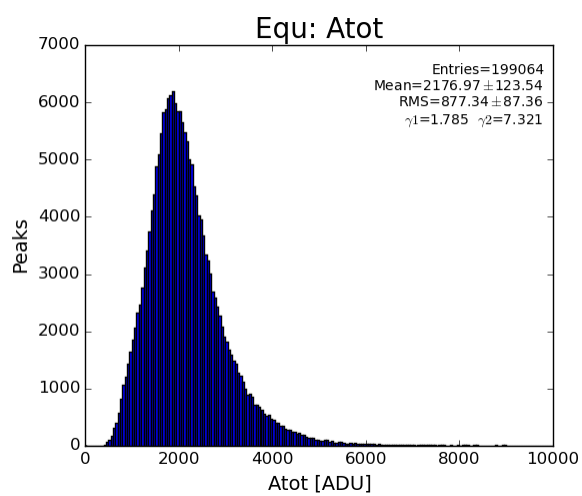
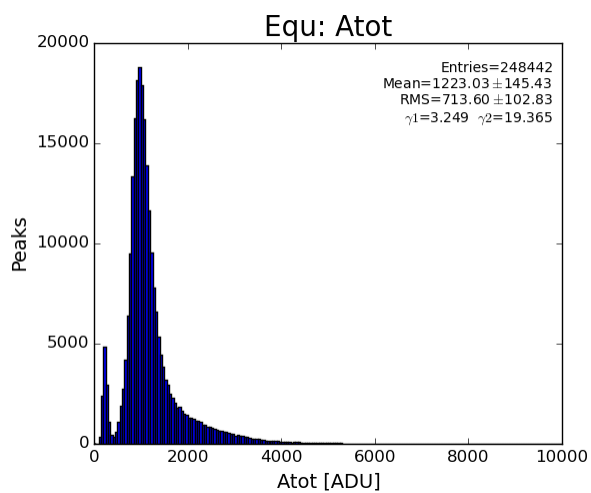
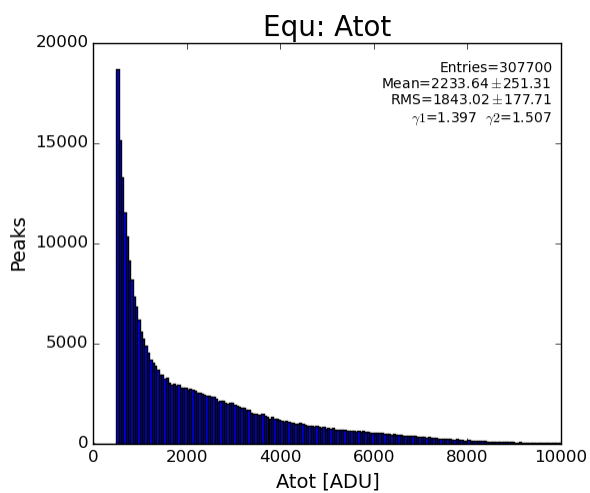


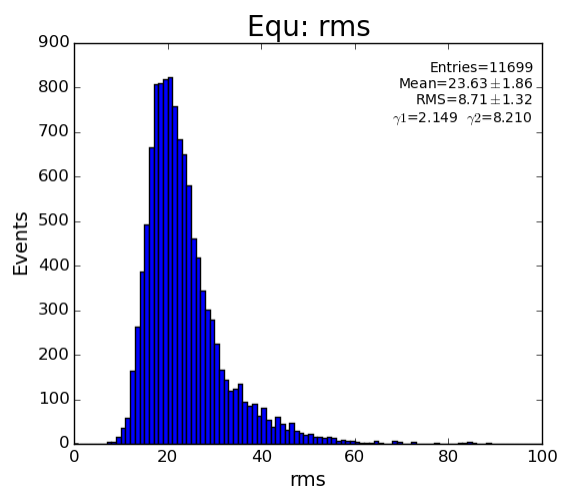
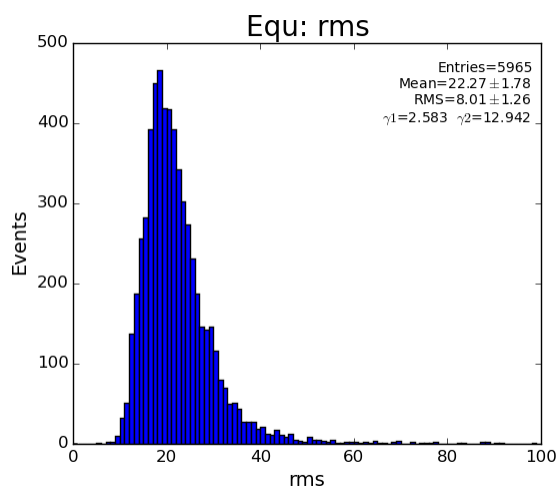
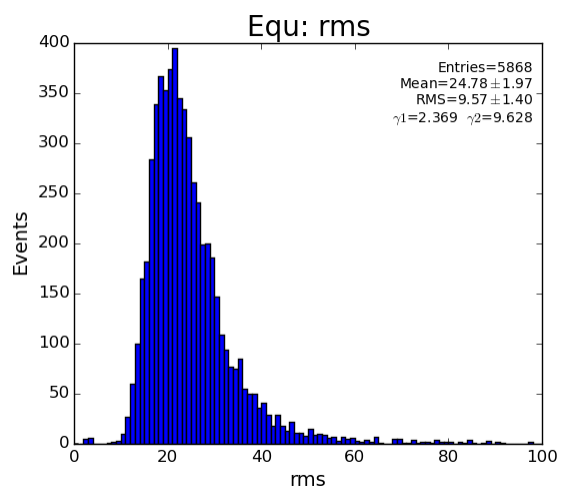


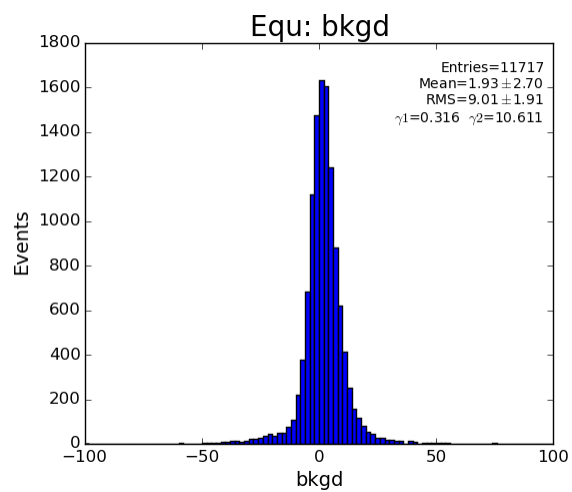
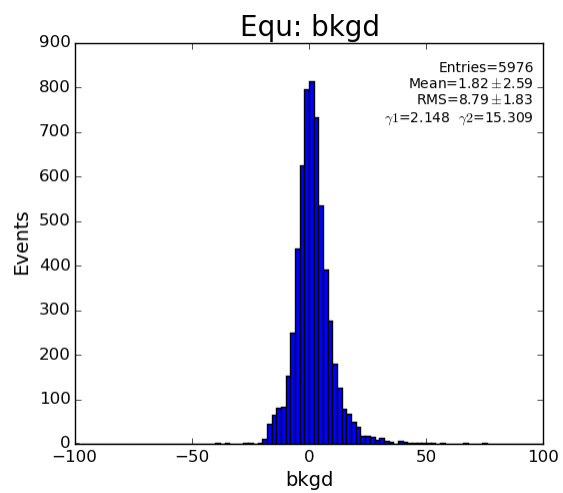
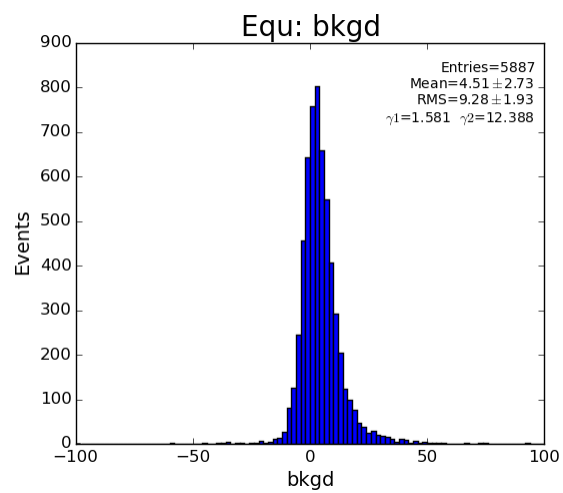
EQU region

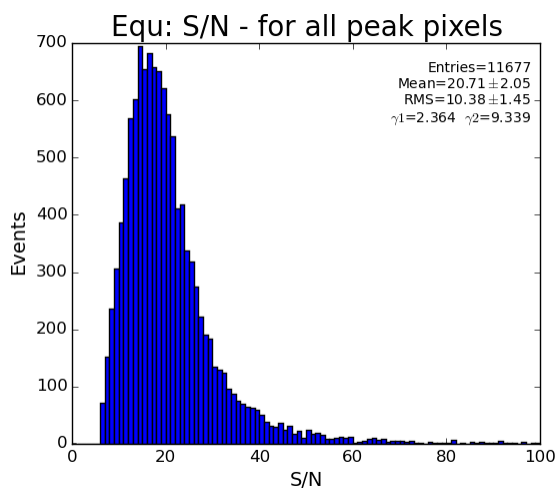
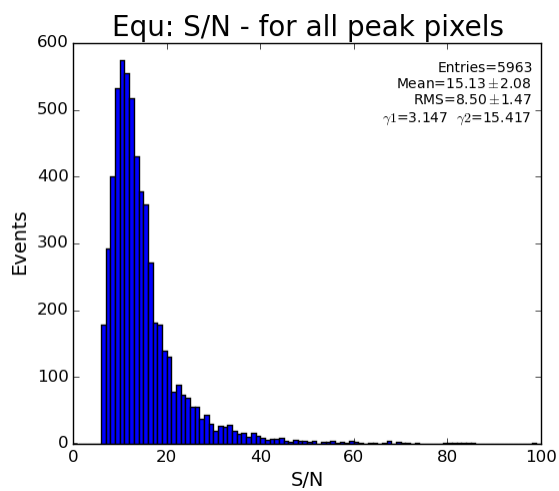
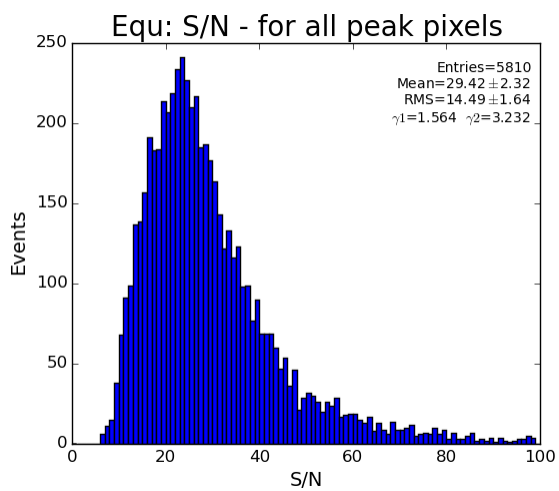
```
def procPeakDataEqu(pk) :
    """ Process peak for EQU region; accumulate peak data
    """
    #=====
    # discard from all histograms except its own
    sp.lst_equ_atot.append(pk.atot)
    if pk.atot<2000 : return
    sp.lst_equ_r_raw.append(pk.r)
    if pk.r<100      : return
    #=====
    sp.lst_equ_r      .append(pk.r)
    sp.lst_equ_amax.append(pk.amax)
    sp.lst_equ_npix.append(pk.npix)
    ...
```

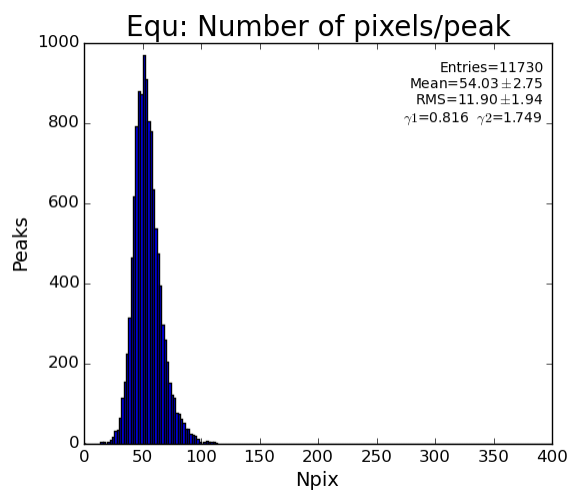
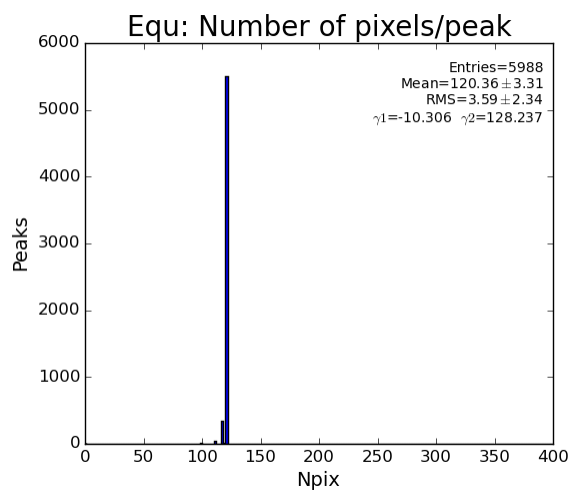
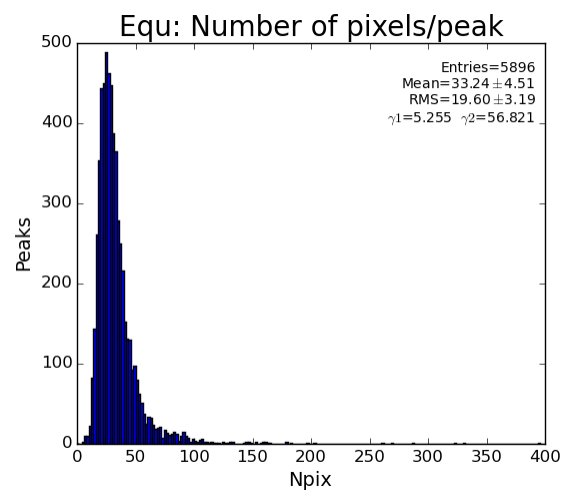


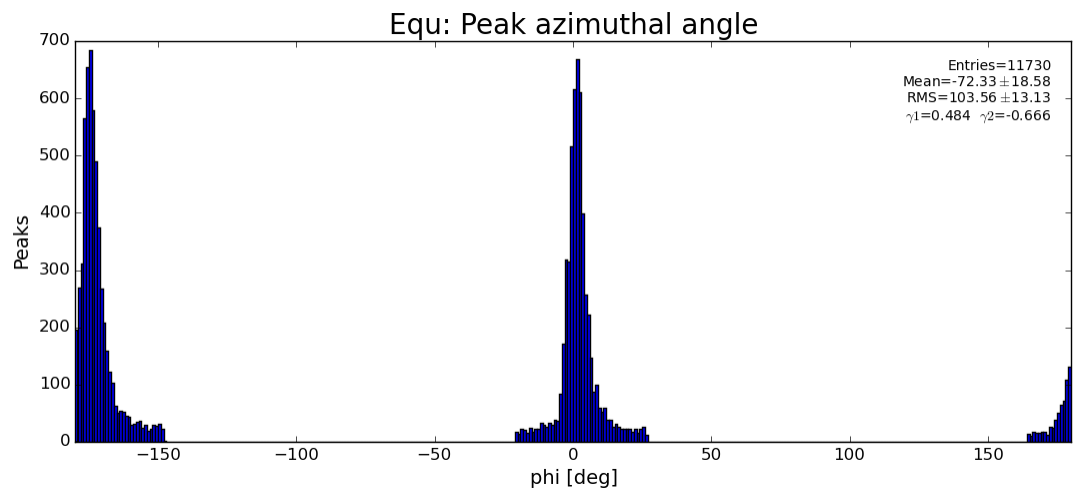
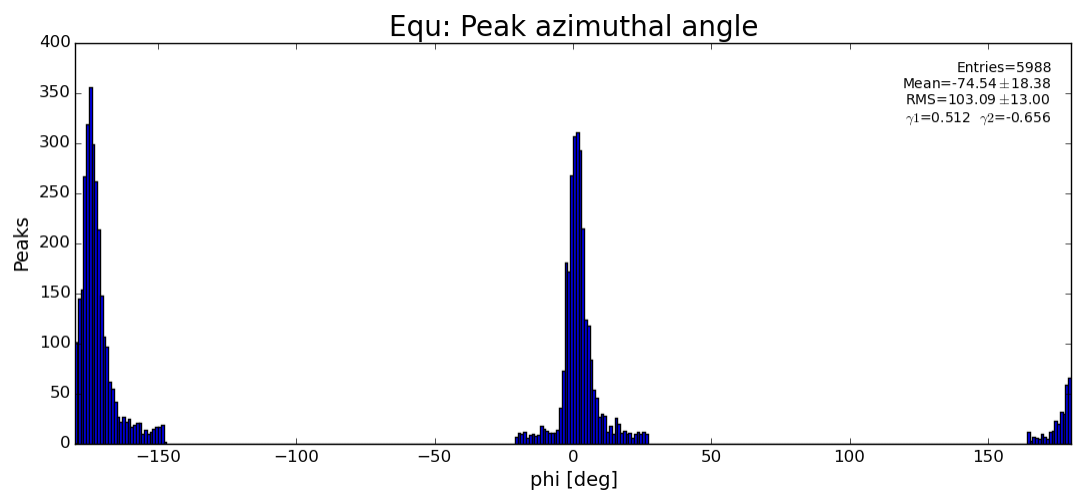
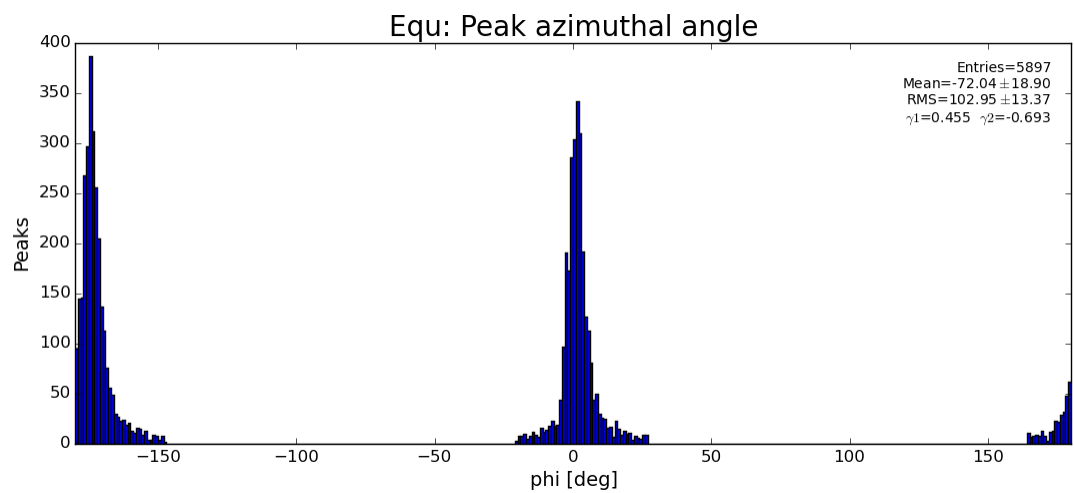


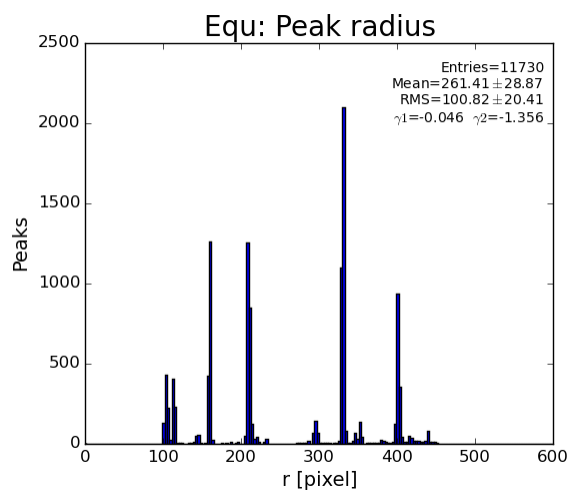
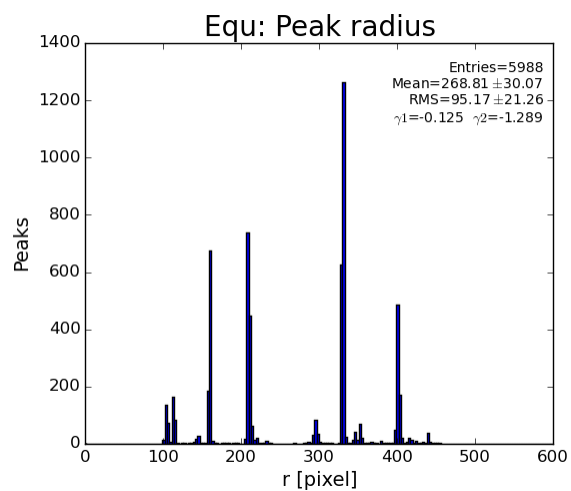
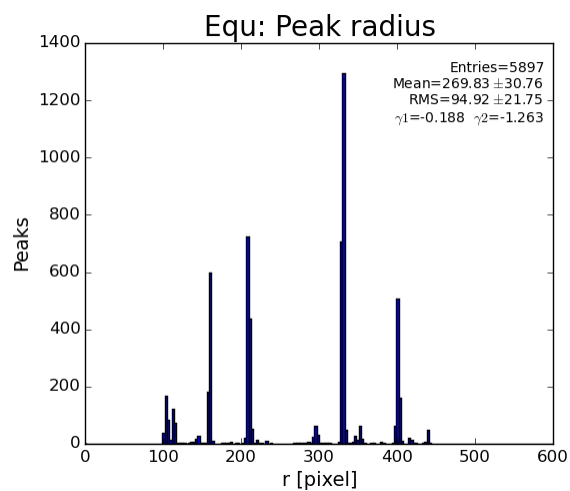




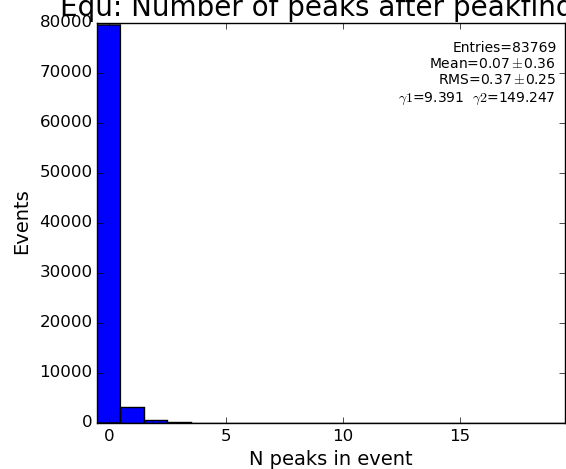




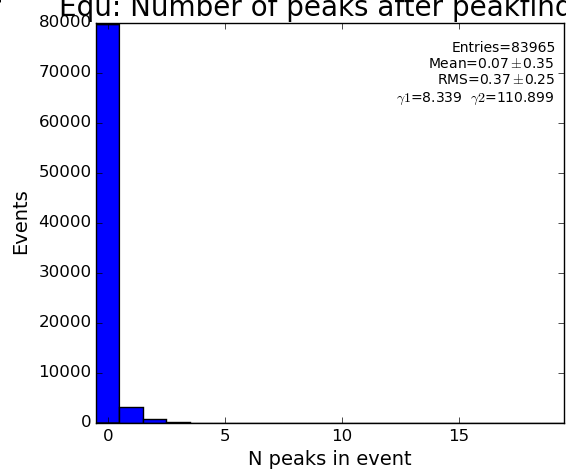




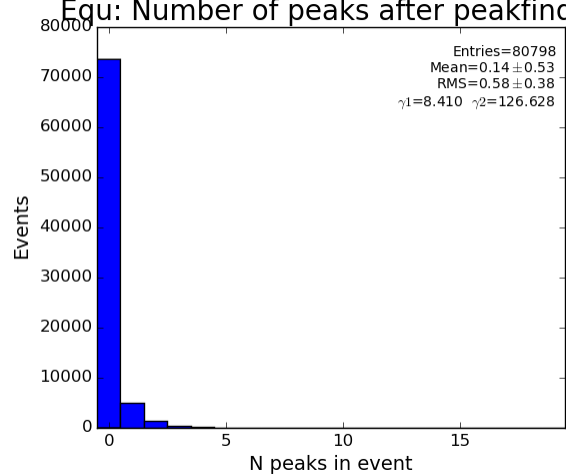
Egu: Number of peaks after peakfinder

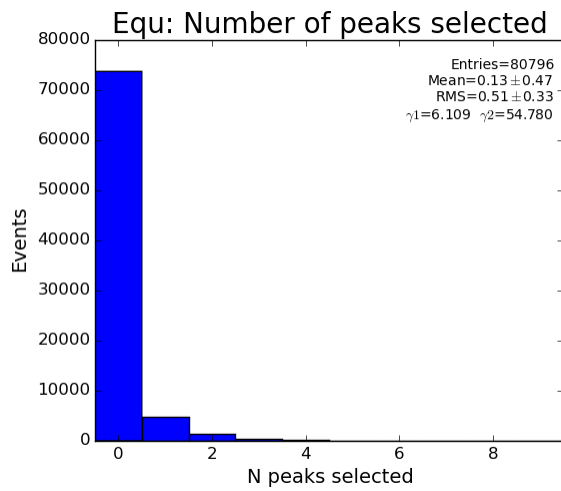
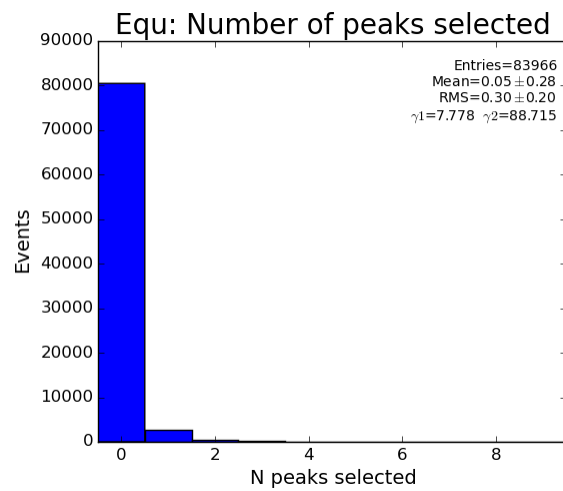
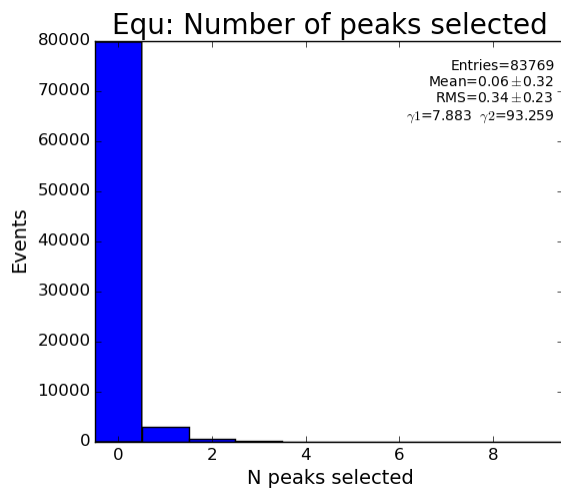


Egu: Number of peaks after peakfinder



Egu: Number of peaks after peakfinder

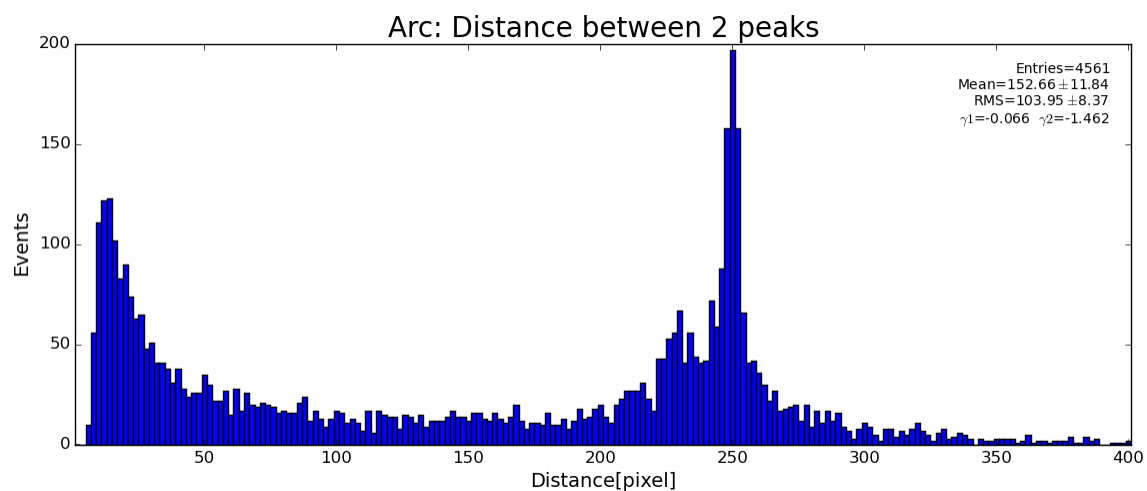
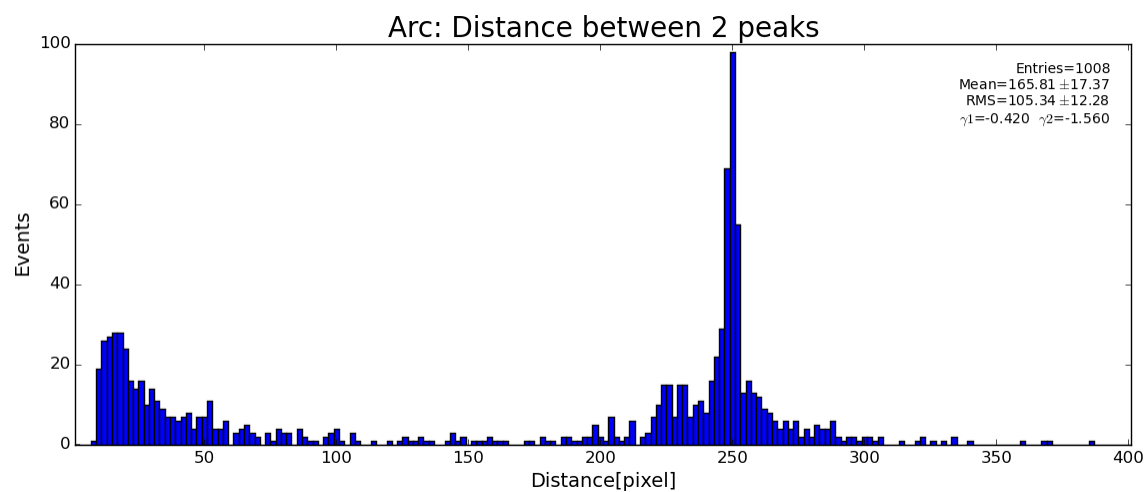
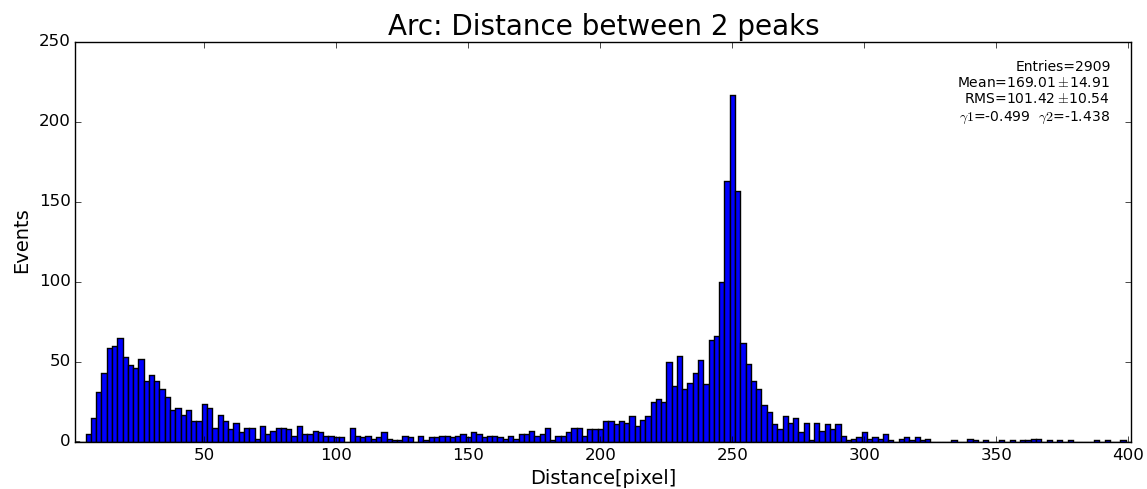


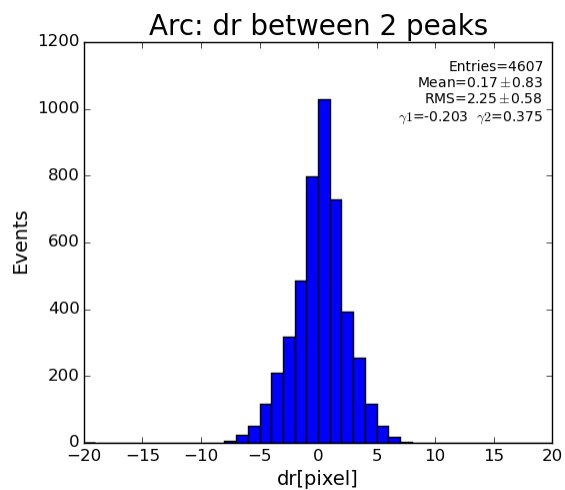
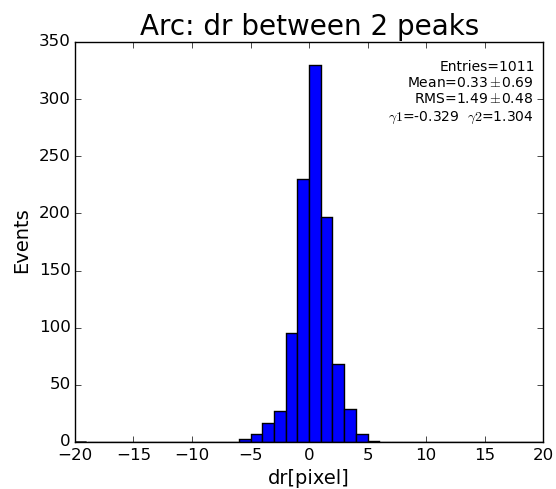
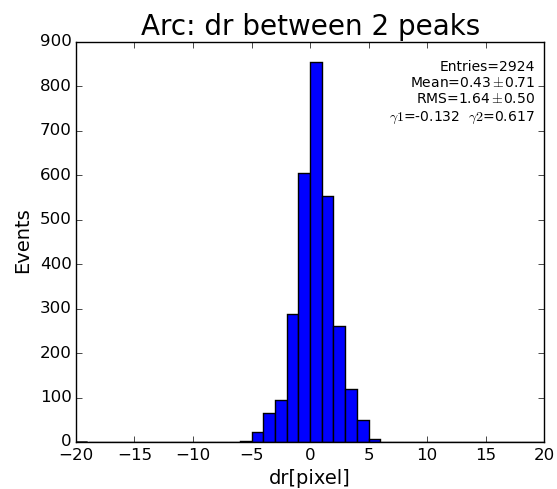


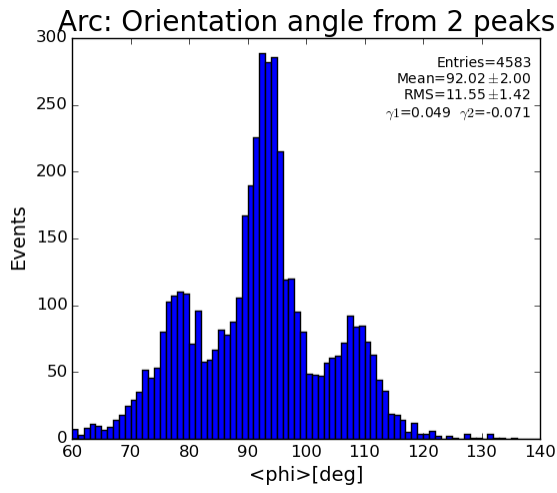
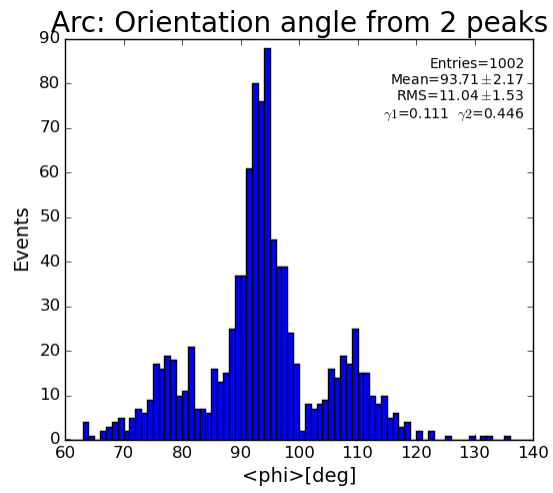
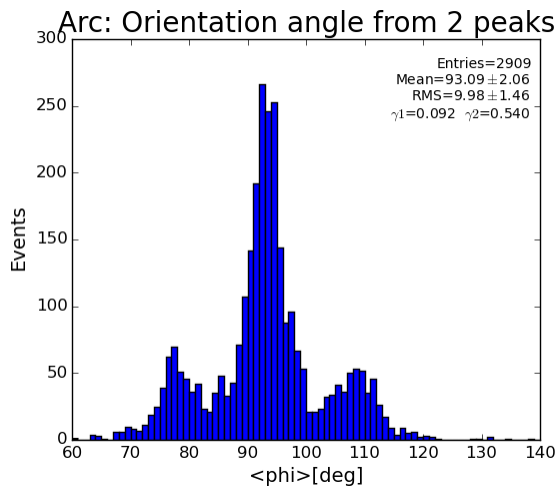
Peak selection for fit

ARC region

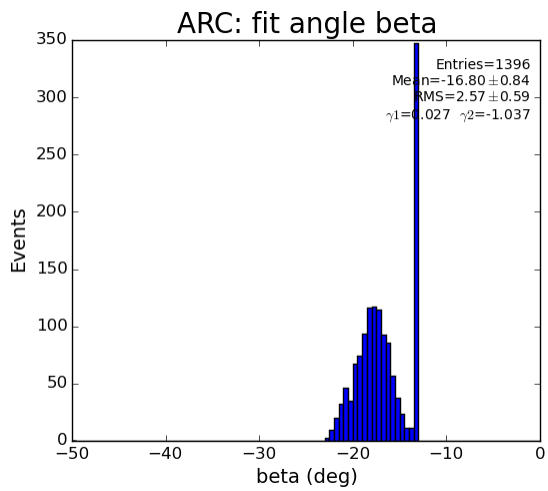
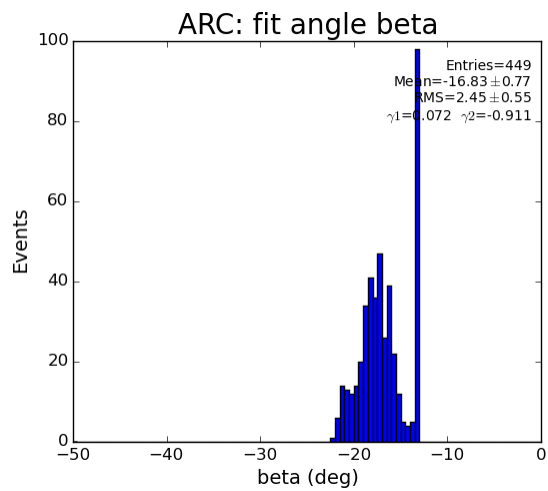
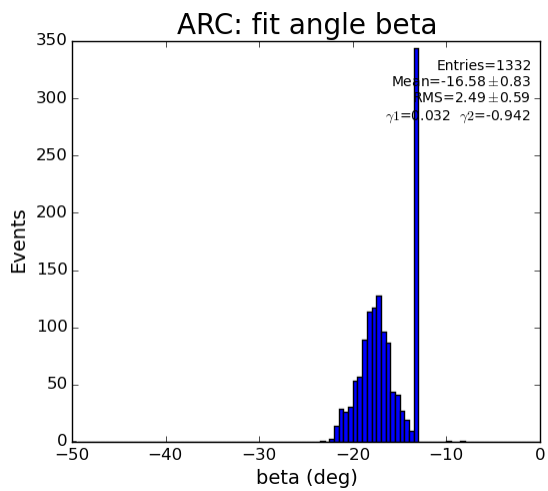
```
def peakIsSelectedArc(pk) :
    """Apply peak selection criteria to each peak from file
    """
    if pk.son<9      : return False
    if pk.amax<150   : return False
    if pk.atot<2000  : return False
    if pk.npix>500    : return False
    if pk.r<435       : return False
    if pk.r>443       : return False
    if pk.rms>80      : return False
    if pk.bkgd<-20    : return False
    if pk.bkgd>50     : return False
    return True
```

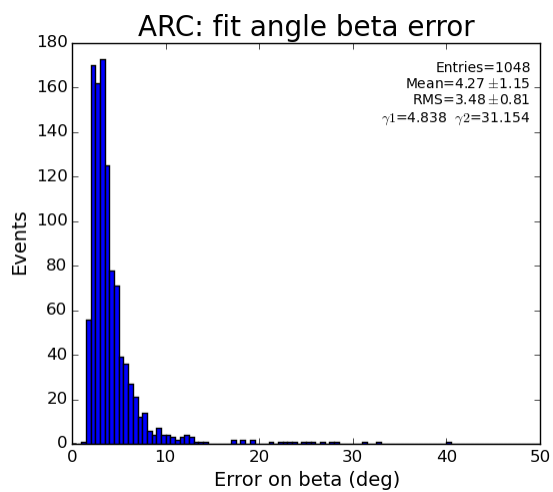
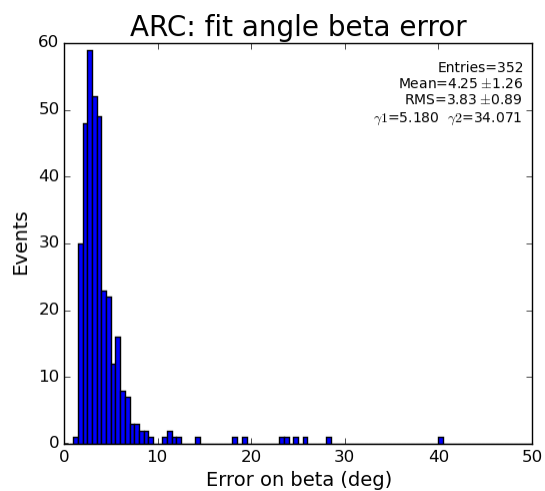
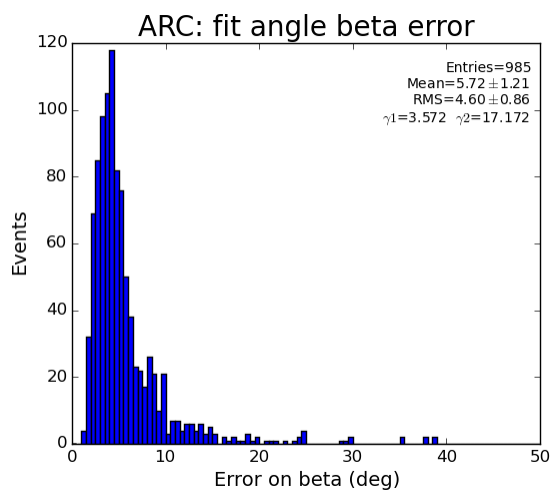


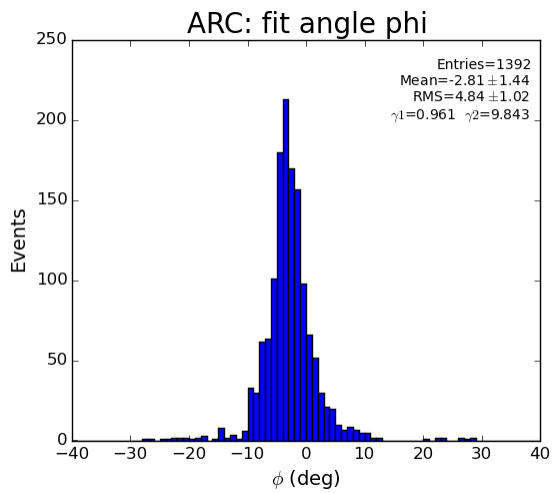
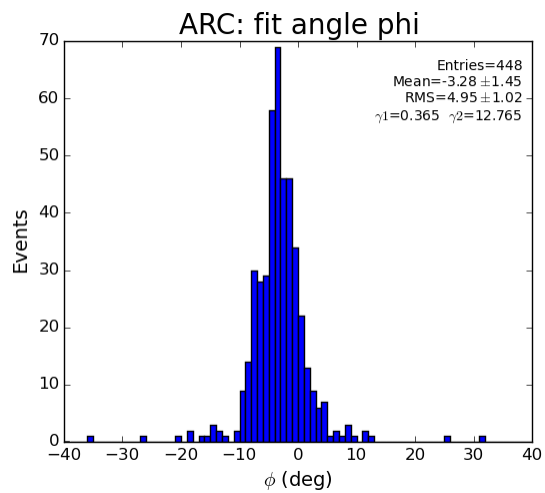
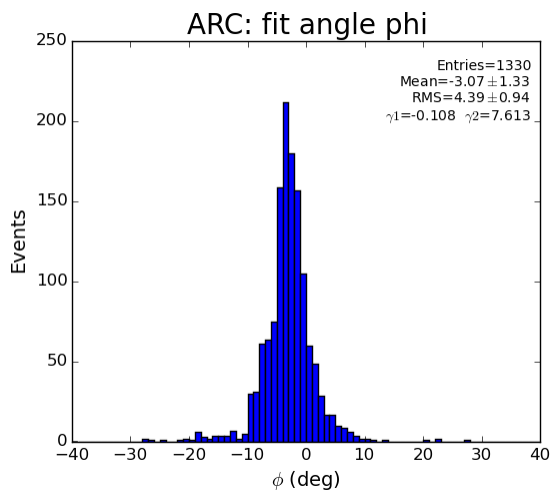


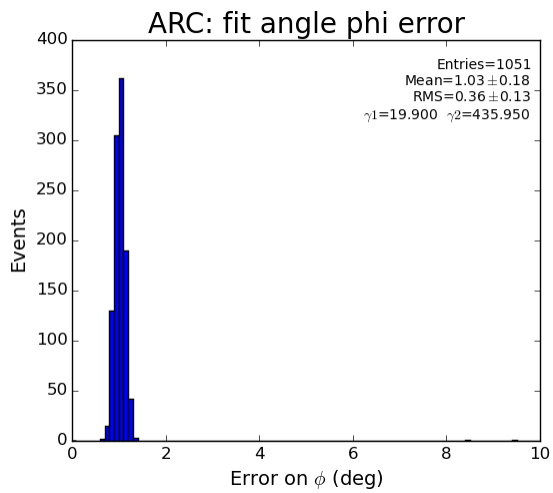
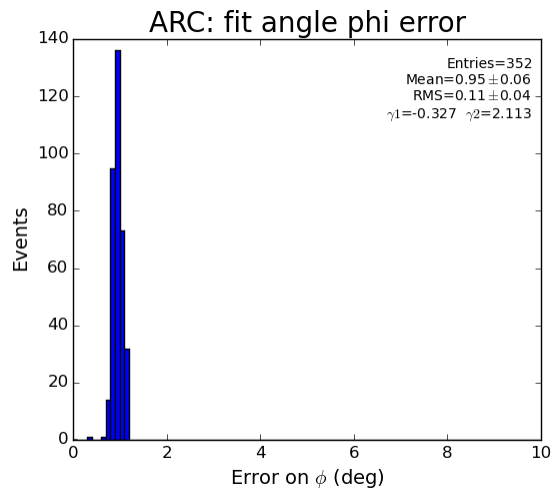
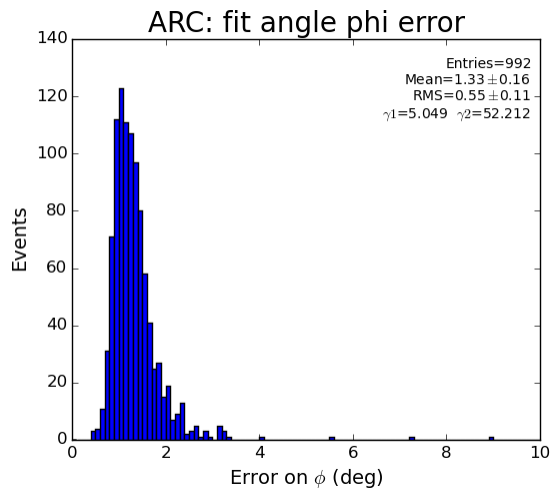


To fit peaks we use `fancy_l1_v0(x, phi_deg, bet_deg, DoR=433/sp.DETD, sgnrt=-1.)`





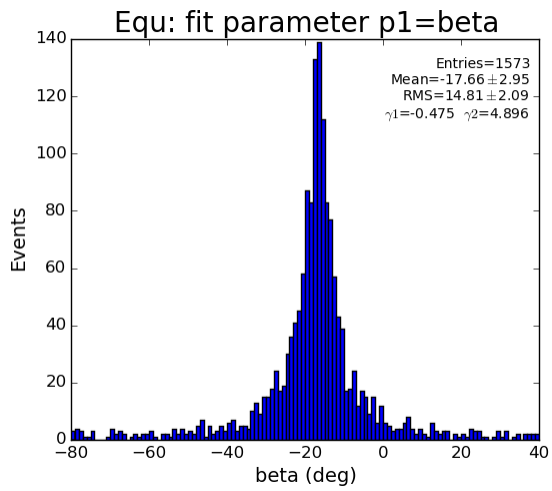
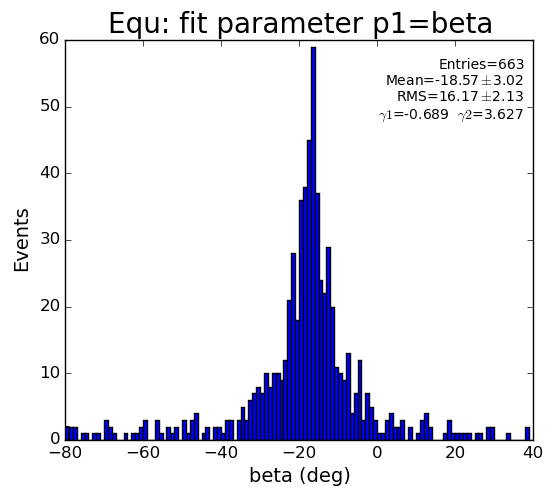
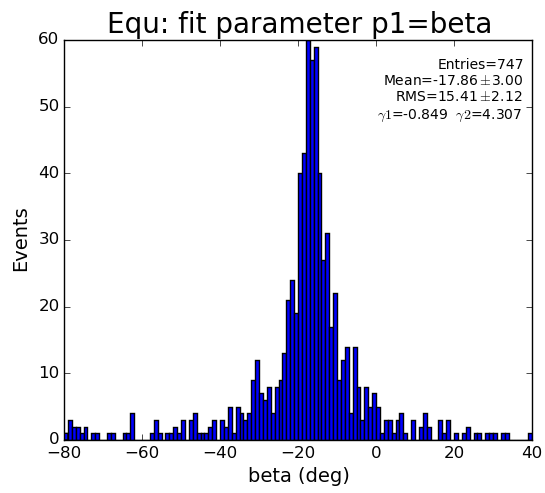


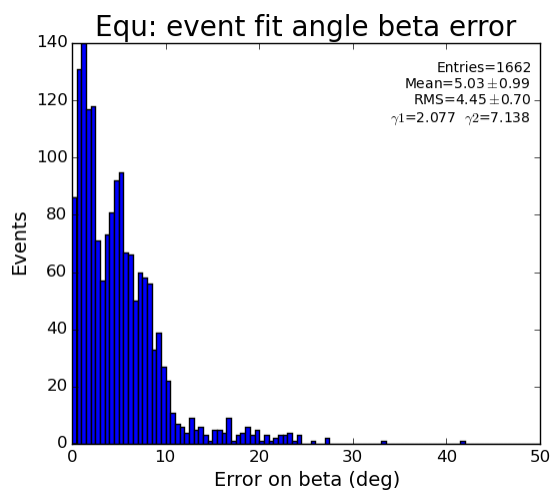
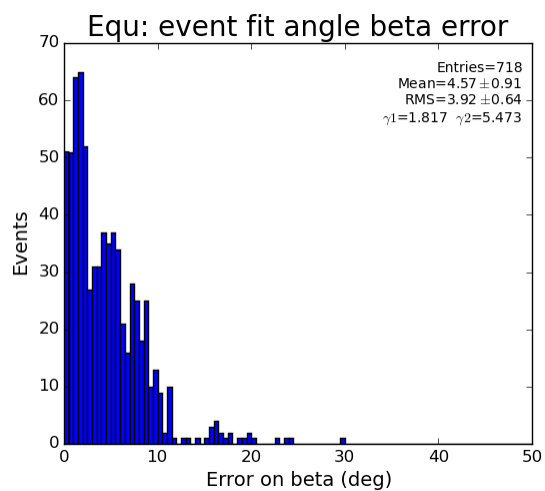
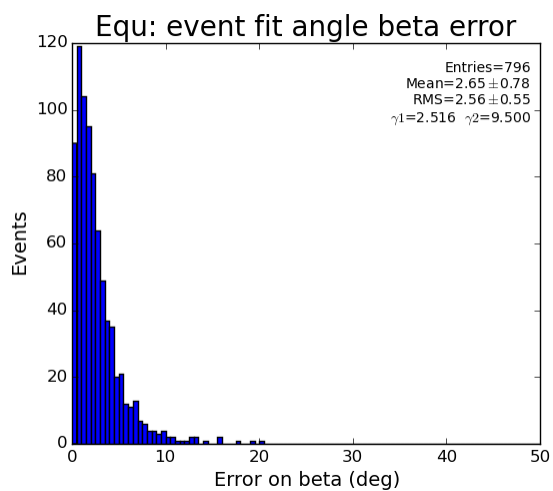


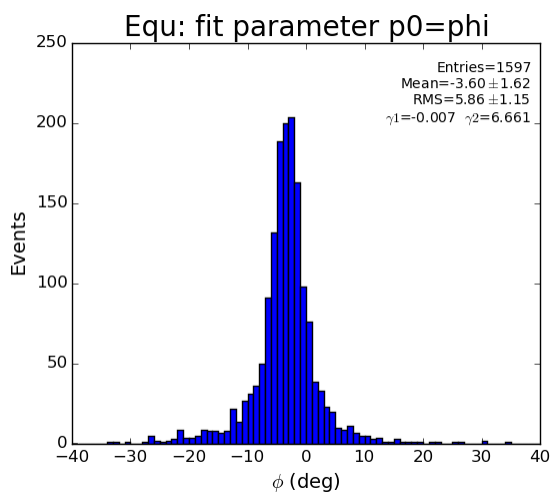
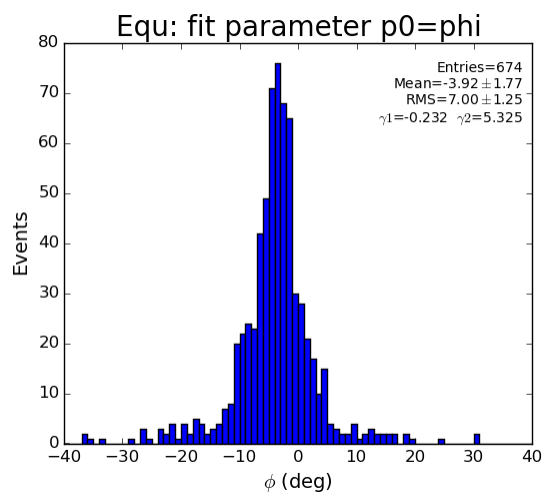
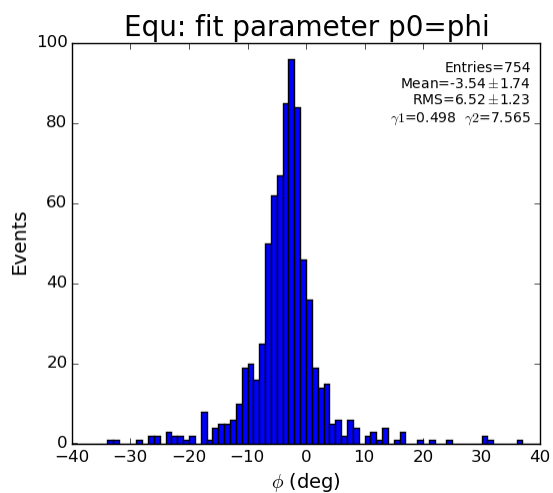
EQU region

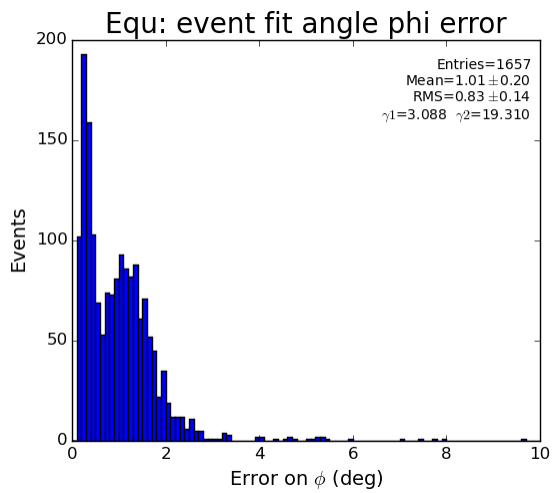
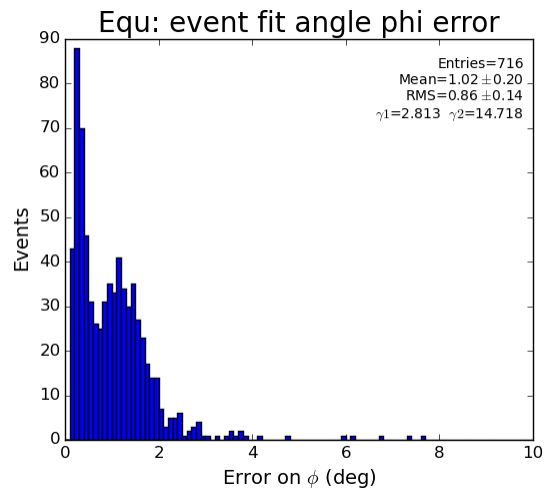
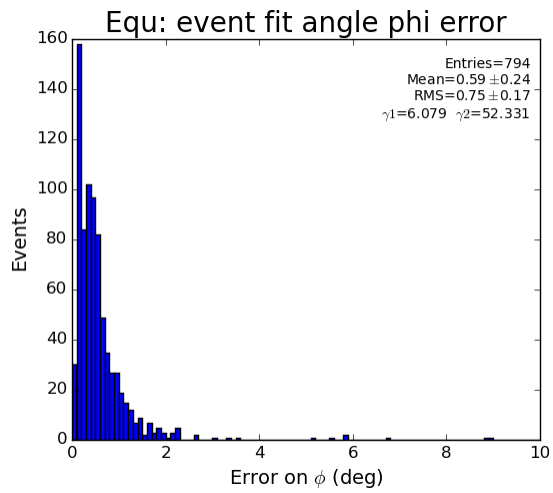
```
def peakIsSelectedEqu(pk) :
    """Apply peak selection criteria to each peak from file
    """
    if pk.son<9 : return False
    if pk.amax<150 : return False
    if pk.atot<2000 : return False
    if pk.npix>500 : return False
    if pk.r<100 : return False
    if pk.r>454 : return False
    if pk.rms>80 : return False
    if math.fabs(pk.bkgd)>20 : return False
    return True
```

To fit peaks we use `fancy_10` 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](#)