

Image processing algorithms for scikit

- [Peak-finders](#)
 - [Interface](#)
 - [Input](#)
 - [Output](#)
 - [Peak-finding algorithms](#)
- [References](#)

Peak-finders

A set of peak-finding algorithms developed for analysis of data from LCLS pixel array detectors.

Interface

```
from ... import imagealgos

peaks = imagealgos.peaks_adapt_thr(data, mask, rank, nsigm, base_r, base_dr, do_base_sub=False, npix_min=None,
npix_max=None, thr_atot=None, son_min=None)
peaks = imagealgos.peaks_two_thr(data, mask, rank, thr_low, thr_high, base_r, base_dr, do_base_sub=False,
npix_min=None, npix_max=None, thr_atot=None, son_min=None)
```

Input

arguments:

data :

- `numpy.array, ndim=2` - 2-d data numpy array for processing
- `[numpy.array, ndim=2]` - list of 2-d data numpy arrays for processing
- `numpy.array, ndim>2` - a set of 2-d data numpy arrays for processing; last two indexes are used as 2-d row and column, higher indexes are compacted to the 1-dim segment index

mask : `numpy.array (dtype=np.uint16)` | `None` - array should have the same structure as data or `None` if non-applicable - pixel mask with 0/1 for bad/good pixel

rank : `int` - radial size of the pixel region ($2 \times \text{rank} + 1$ rows and columns) around central pixel

thr_low : `float` - low threshold on pixel intensity

thr_high : `float` - high threshold on pixel intensity

nsigm : `float` - threshold in number of noise rms

base_r : `float` - internal radius of the ring for evaluation of base level mean and rms (noise)

base_dr : `float` - width of the ring for evaluation of base level mean and rms (noise)

keyword arguments :

do_base_sub : `bool` - enable subtraction of local base level from signal

npix_min : `int` - minimum number of pixels in the peak

npix_max : `int` - maximum number of pixels in the peak

thr_atot : `float` - threshold on total peak intensity

son_min : `float` - threshold on signal over noise (S/N) value

Output

peaks : `dict` - dictionary of arrays/lists - most natural order to work with a single value for many peaks

dict keywords:

- `row` : array of `int` - row of the central pixel with maximal intensity in the square region defined by `rank`
- `col` : array of `int` - column of the central pixel with maximal intensity in the square region defined by `rank`
- `seg` : array of `int` - segment index for `data.ndim>2`, e.g. in CSPAD consisting of 32 segments this index should be in the range from 0 to 31
- `npix` : array of `int` - number of pixels accounted in the peak
- `amp_max` : array of `float` - maximal intensity among peak pixels
- `amp_total` : array of `float` - total intensity of pixels accounted in the peak
- `row_cgmean` : array of `float` - "center of gravity" row coordinate evaluated for pixels accounted in the peak using their intensities as weights
- `col_cgmean` : array of `float` - "center of gravity" column coordinate evaluated for pixels accounted in the peak using their intensities as weights
- `row_cgrms` : array of `float` - "center of gravity" row coordinate rms
- `col_cgrms` : array of `float` - "center of gravity" column coordinate rms
- `base` : array of `float` - per pixel base level mean estimated for pixels in the ring region (parameters `base_r`, `base_dr`) around peak center
- `noise` : array of `float` - per pixel base level rms estimated for pixels in the ring region (parameters `base_r`, `base_dr`) around peak center
- `son` : array of `float` - signal over noise ratio estimated as `amp_total / (noise * sqrt(npix))`
- `peakinds` : list of `numpy.array(ndim=2)` - list of the peak-pixel `row,col` indexes. Shape of the array for each peak is `(npix, 2)`.
- `peakmap` : array shaped as `data` of `int` - map with pixels marked by peak numbers

Peak-finding algorithms

Each peak-finding algorithm works in a few stages. First stage is a search for peak candidates or "seed" peaks and this is a most distinctive part between algorithms:

- **peaks_two_thr** - is a two-threshold peak-finding algorithm searching for groups of connected pixels with intensity above `thr_low` in the square region restricted by the parameter `rank` around central pixel with intensity exceeding `thr_high`. In the loop over all pixels each pixel with intensity greater or equal `thr_high` is considered as a peak candidate. For each candidate recursive algorithm is launched and searches for a group of connected pixels with intensity greater or equal `thr_low` in the square region with dimensions `2*rank+1` pixel in rows and columns. Pixels of the group are marked as busy and are not used in other groups. Central pixel of the peak candidate should have maximal intensity in the group of connected pixels, otherwise recursion is terminated and all group pixels released for further search. Two neighbor pixels with intensity above `thr_low` having common boarder are assumed connected. Two pixels having a single common vertex are assumed disconnected. Base level mean, `base`, and rms, `noise`, values are evaluated for pixels in the ring with internal radius `base_r` and width `base_dr` around central pixel. Pixels with intensity above `thr_low` are excluded from evaluation of the base level parameters.
- **peaks_adapt_thr** - is an adaptive-threshold peak-finding algorithm searching peak candidates as group of connected pixels with intensity above dynamically evaluated threshold around local intensity maximums of specified `rank`. Maps of local maximums and minimums of specified `rank` are generated for `data` array. Parameter `rank` defines a square region with dimensions `2*rank+1` pixels in rows and columns around central pixel. Each local maximum of specified `rank` is considered as a peak candidate. Its base level mean, `base`, and rms, `noise`, values are evaluated using surrounding pixels in the ring with internal radius `base_r` and width `base_dr`. Pixels marked as local maximums and minimums (for symmetry) of specified `rank` in rows and columns are excluded in evaluation of the base level parameters. Then, for each peak candidate local adaptive-threshold parameter `thr=base+nsigm*noise` is used. Recursive algorithm searches for the group of connected pixels with intensity above local threshold `thr` in the `rank`-restricted region.

NOTE: due to excluded extremes the base level rms is under-estimated in this algorithm, so, set parameter `nsigm` wisely.

Other stages of all peak-finding algorithms are about the same.

- returned values of `amp_max` and `amp_total` are base level corrected by subtracting `base` and `base*npix`, respectively, if `do_base_sub=True`.
- signal over noise `son` is evaluated as `amp_total / (noise * sqrt(npix))`,
- output dictionary of peak parameters is generated from the list of seed peaks complying with selection parameters `npix_min`, `npix_max`, `amax_thr`, `atot_thr`, `son_min`.

References

- [Peak Finding Details](#)
- [Hit and Peak Finding Algorithms](#)
- [ImgAlgos auto-doc](#)