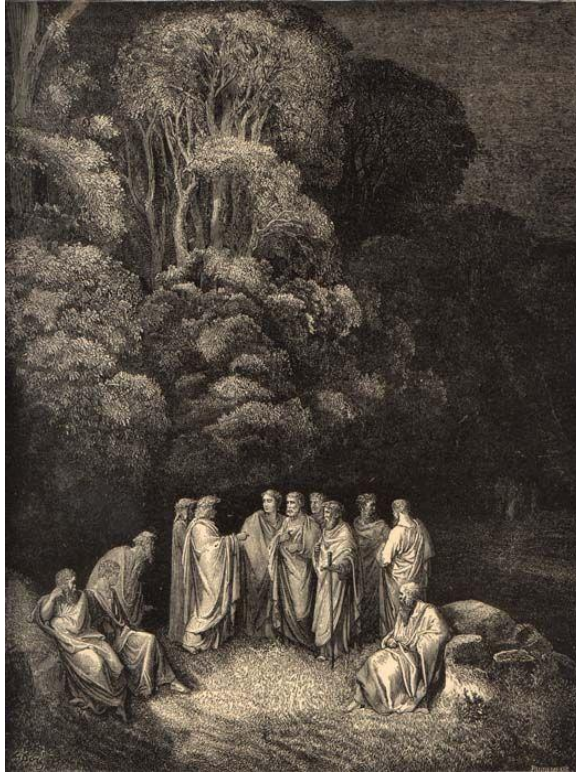# Straxferno

Circle 1: high-level processing

Today

Intro

- Homework for next week(s)!
- Outline of high-level plugins

Questions / discussion

- `straxen/plugins/peak_processing.py`
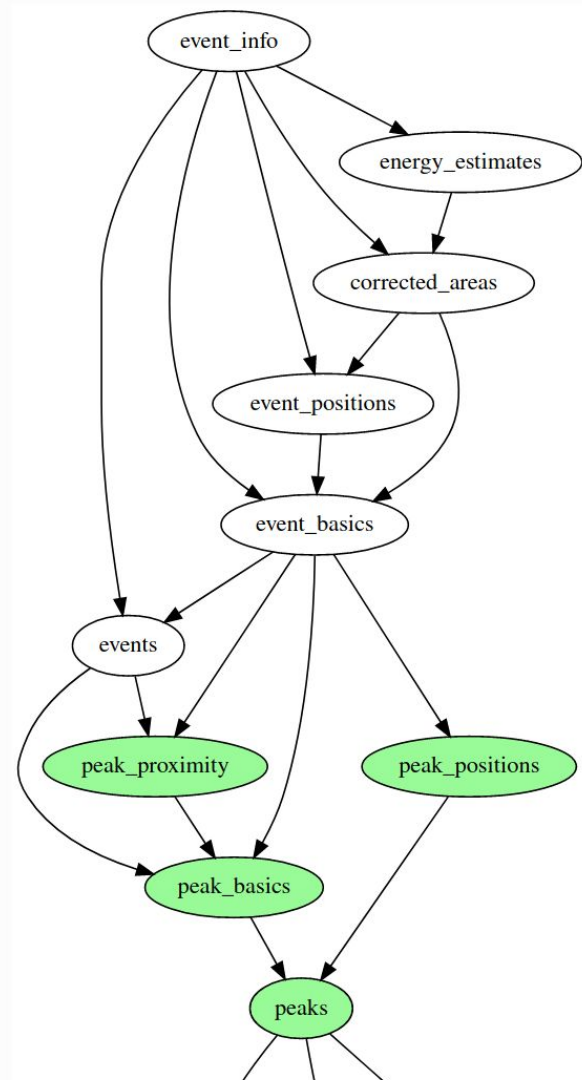- `straxen/plugins/event_processing.py`

Preview of next week

Homework: deep dive on one algorithm

Detailed description on the wiki

1. **Sign up** for a strax function / algorithm here
2. **Study** the function:
    a. Docstring
    b. Where is it used?
    c. What goes in, what goes out?
    d. How is it tested, or how could you test it?
    e. Implementation. Anything unusual?
3. **Choose one**:
    a. Add one slide to this presentation
    b. Make a pull request to improve something

Suggested deadline: before next session or session after that
Will feature (some) PRs / slides in next sessions

| datatype	depth	n_files	filesize_mb
event_info_double	-1	0	0.0
event_info	0	1	4.7
energy_estimates	1	1	0.5
corrected_areas	2	1	0.6
event_positions	3	1	0.9
event_basics	4	1	3.7
events	5	1	0.4
peak_positions	5	1	70.5
peak_proximity	5	2	122.7
peak_basics	6	3	183.7
peaks	7	0	0.0
merged_s2s	8	3	63.3
peaklet_classification	9	1	38.9
peaklets	10	72	1052.4
lone_hits	10	28	1112.2
veto_regions	11	721	0.4
records	11	721	12094.0
pulse_counts	11	721	3.1
raw_records	12	721	33055.3
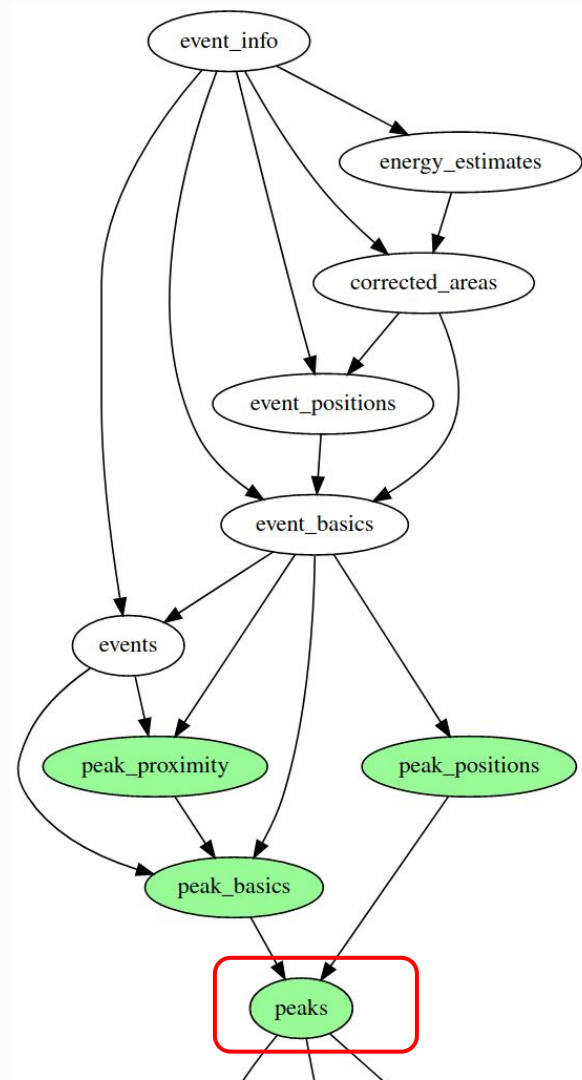
Peaks: boundary between high- and low-level processing.
O(GB / hour)
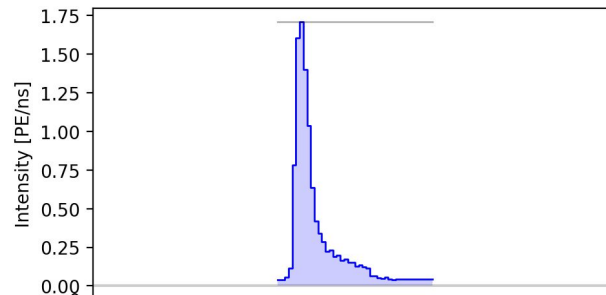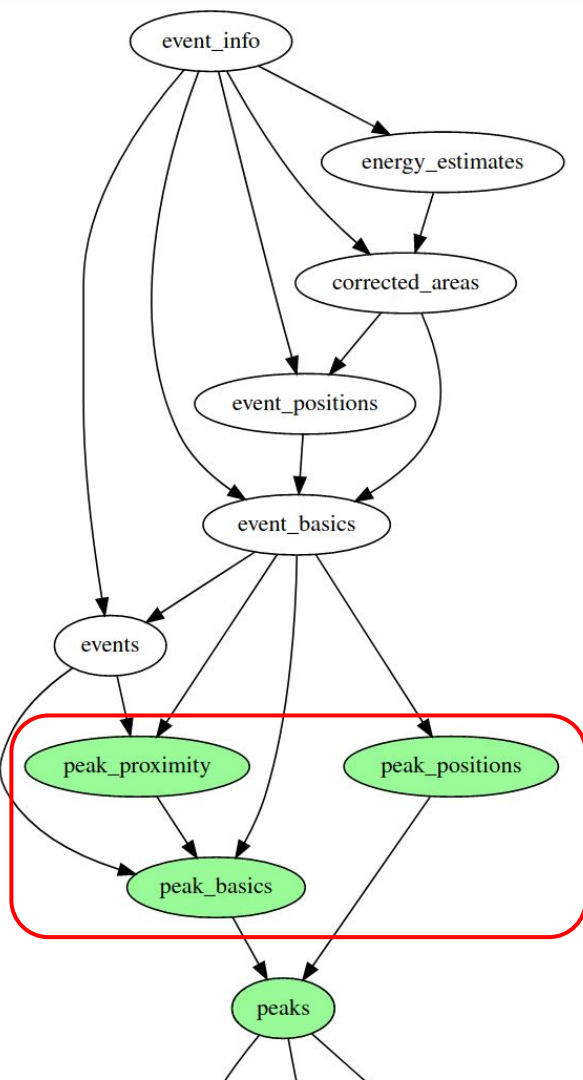

Sum waveform
can be downsampled
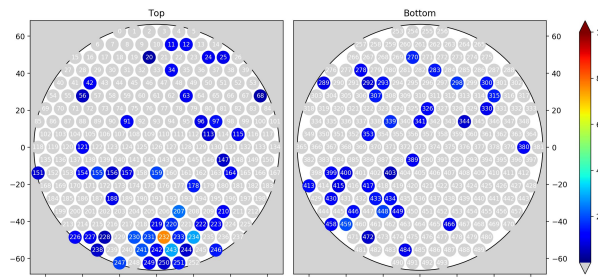
Per-PMT hitpattern
NOT its time-dependence!

Also saturated samples / channel, tight coin, n_hits (if not split)

Not stored on disk!
Auto-merged from peaklets and merged_s2s (next time)

Peak properties, O(100 MB / hour)

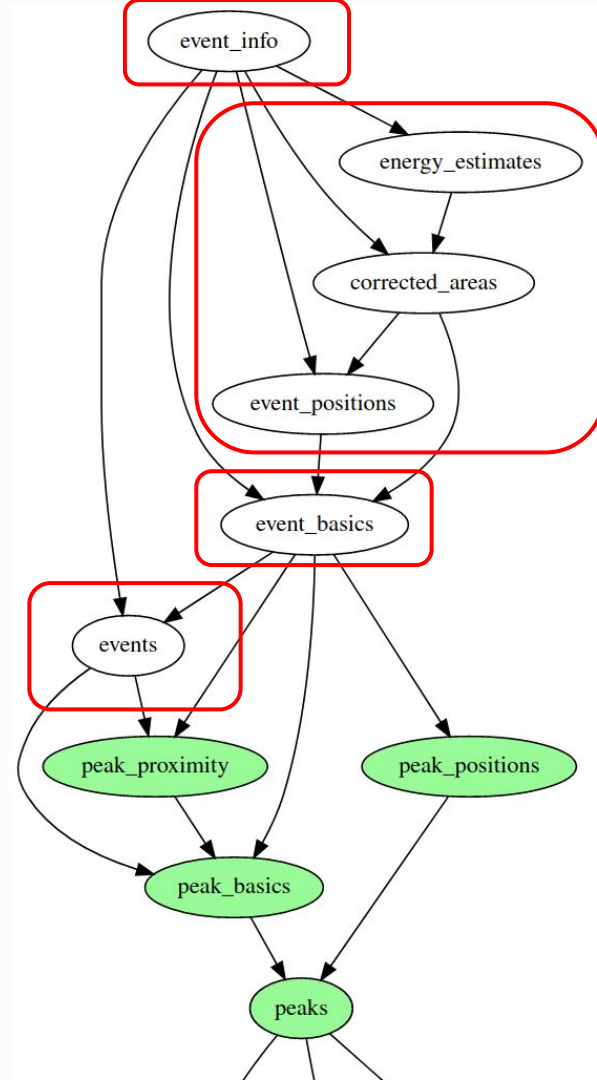Trivial to compute but useful to alias:

- N contributing channels
- Area fraction top
- Max PMT, area in it

Hard to compute, want to version separately:

- Reconstructed positions (NN)

Need to look at other peaks:

- Time to next / prev peak
- N nearby similar peaks

Event info, O(few MB / hour), just merges all event datatypes

Other event properties, O(few MB / hour)

Z from drift time, field distortion, area to energy via g1, g2, etc.

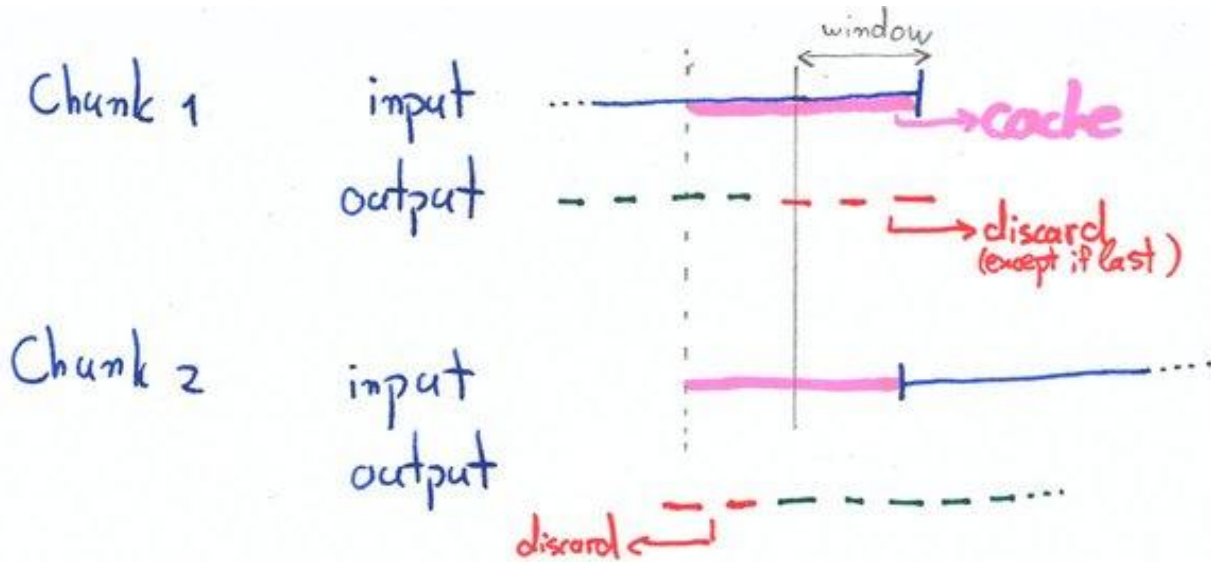**Event basics,** O(few MB / hour)

Decide main and *alternate* S1, S2; store their peak properties

Events, O(MB / hour)

Decide where events happen and how long they are

OverlapWindowPlugin

You can assume you have **window** on both sides of any data, strax takes care of the rest.

More details in the documentation:
https://strax.readthedocs.io/en/latest/developer/overlaps.html#overlap-window-plugins