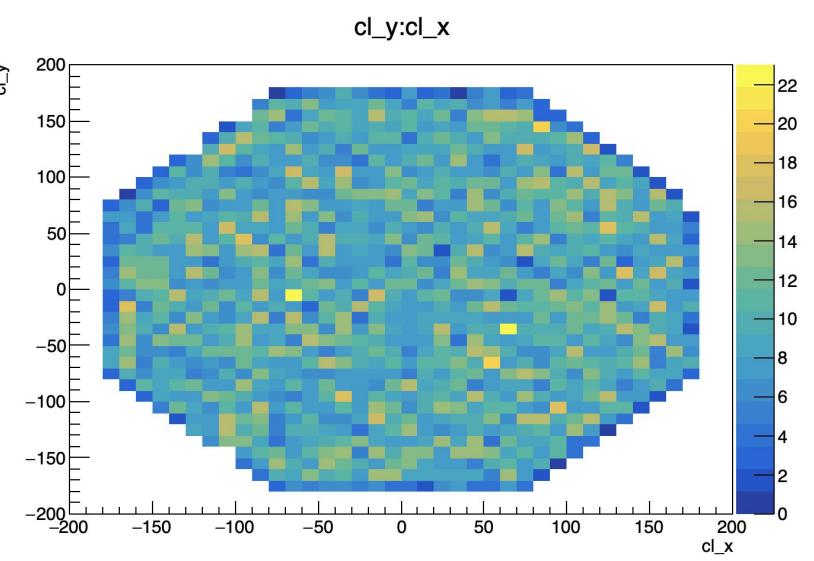
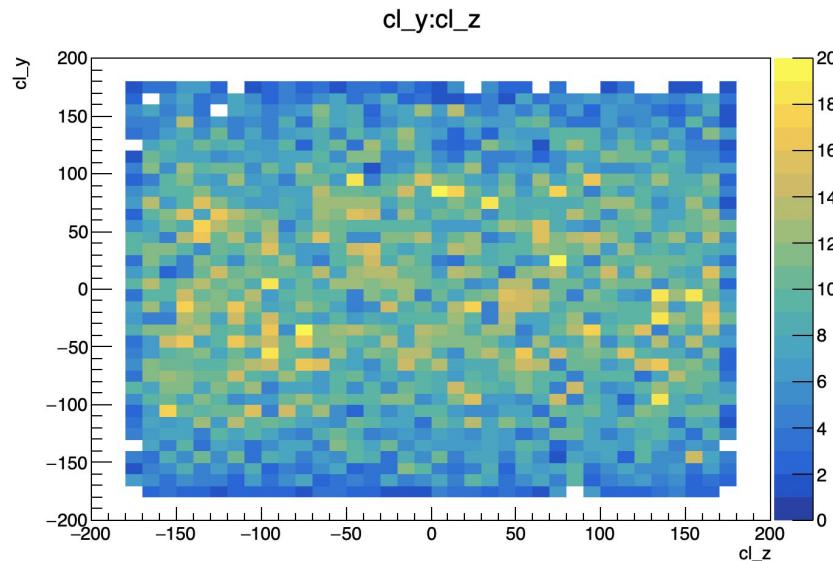


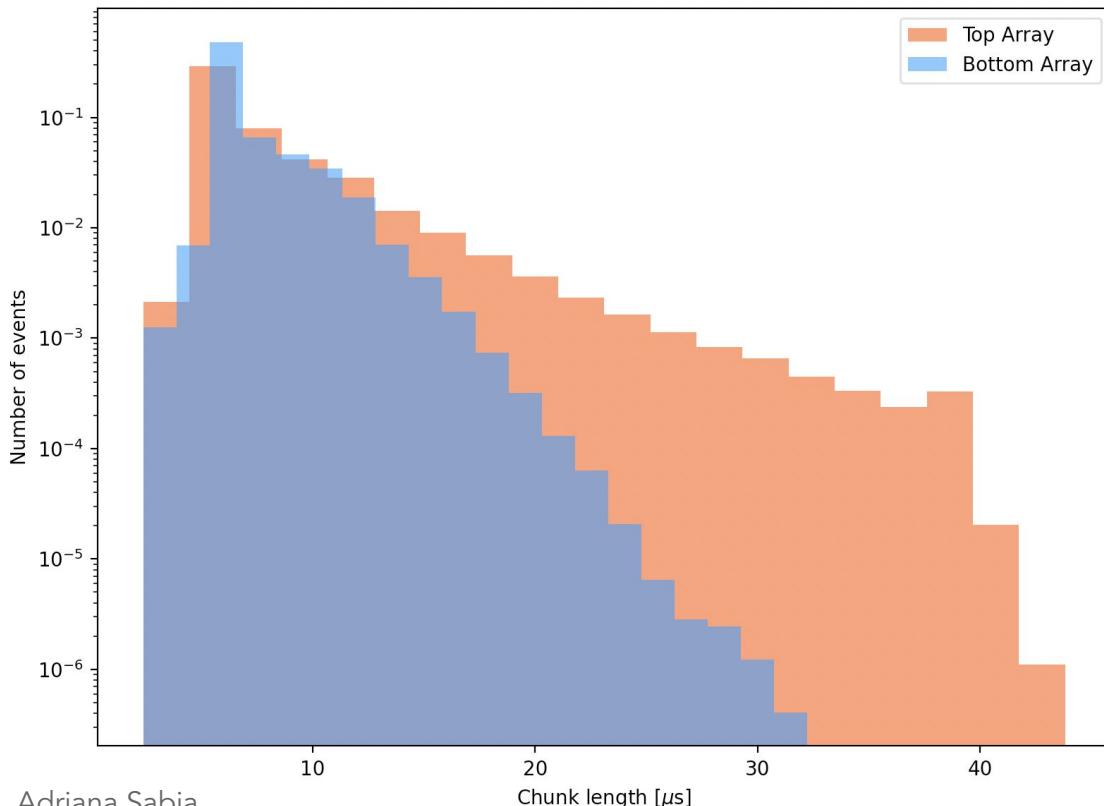
DAQ-Simulation PyReco Output + Event size

Sample G:

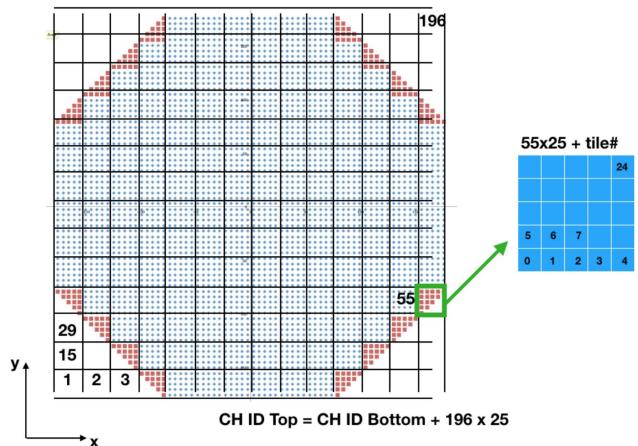
S2 [from 100 to 1E4] PE in the full ACTIVE volume (1 cm distance between the SiPM planes and the anode window)



Chunk Length [μs] Top vs Bottom array

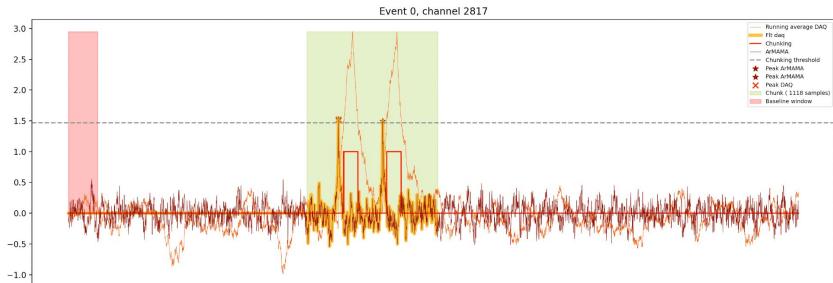


1000 entries
Bottom array $\rightarrow 0:196 \times 25$
Top array $\rightarrow 196 \times 25 + 1:8280$

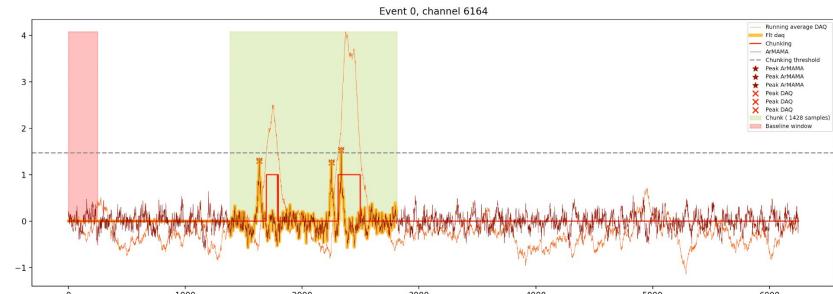


How to deal with long and populated chunks?

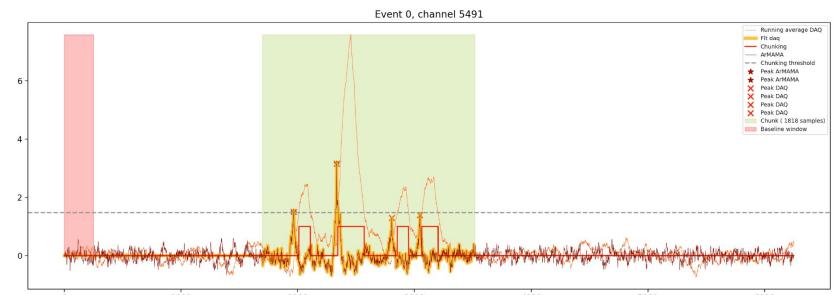
→ Number of peaks = 2



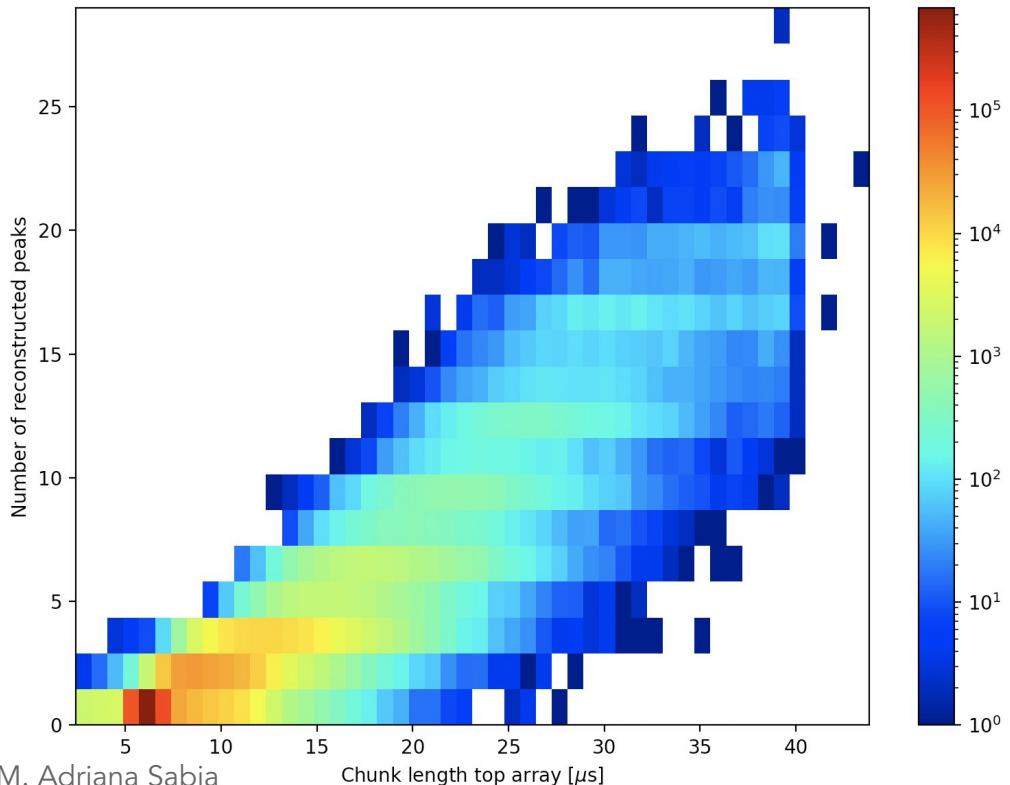
→ Number of peaks = 3



→ Number of peaks = 4



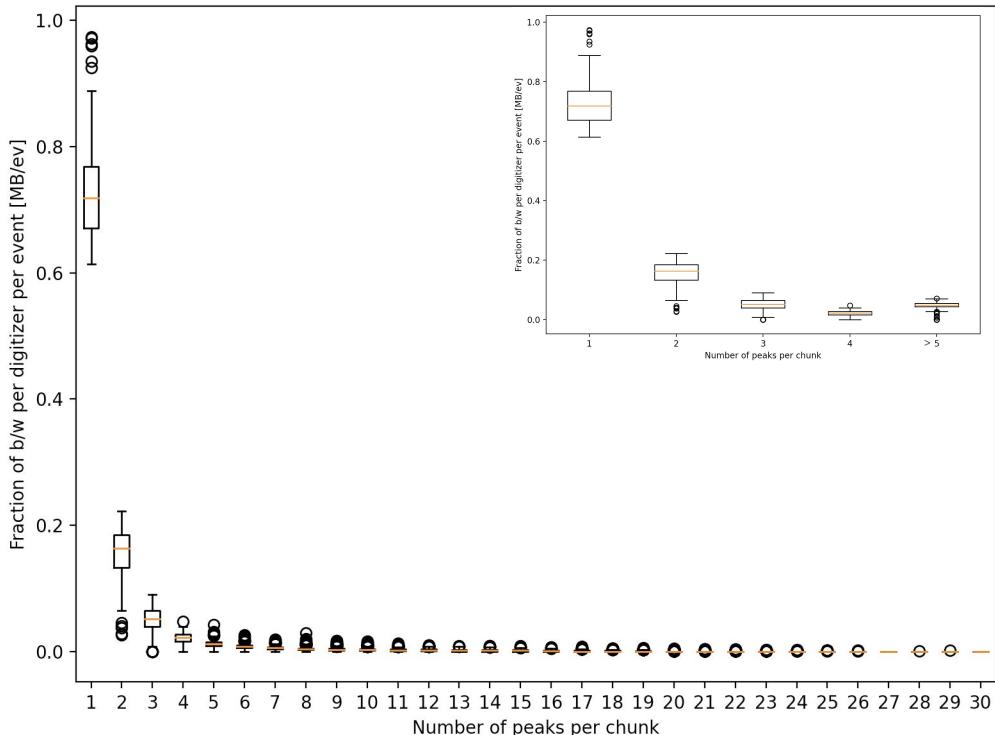
Chunk length recorded in top array vs number of reconstructed peaks in the same chunk



General trend:
long chunks \leftrightarrow populated chunks

Fraction of bandwidth per event per digitizer vs number of reconstructed peaks per chunk

(normalized to the total b/w per event)



Strategy:

- 1) Fix the number N of peaks per chunk
- 2) Sum all the chunk length yielding N peaks
- 3) Convert to MB/ev

1 peaks chunks $\sim 70\%$ of the bandwidth

BACKUP

Output event structure

- chunk_ch = chunk channel
- chunk_t0 = chunk start time
- chunk_len = chunk length
- chunk_hit_t = lista hit_time
- chunk_hit_p = lista hit_prom
- pk_p = prominence
- pk_t = peak time
- pk_ch = peak channel

→ ex. [ch1, ch78, ch84, ...]

→ ex. [$t_0^1, t_0^2, t_0^3, \dots$]

→ ex. [l_1, l_2, l_3, \dots]

→ ex. [[h_1, h_2, h_3]₁, [h_1, h_2, h_3]₂, ...]

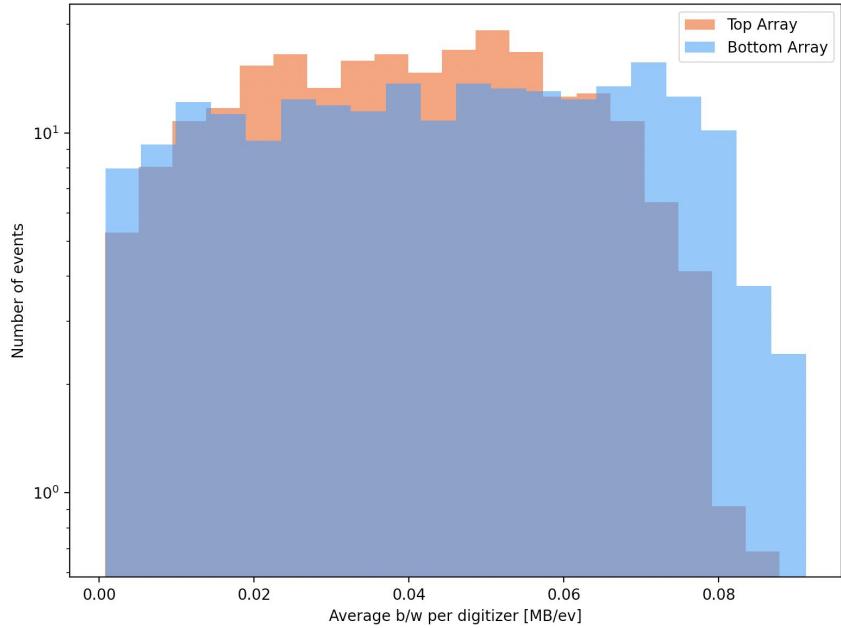
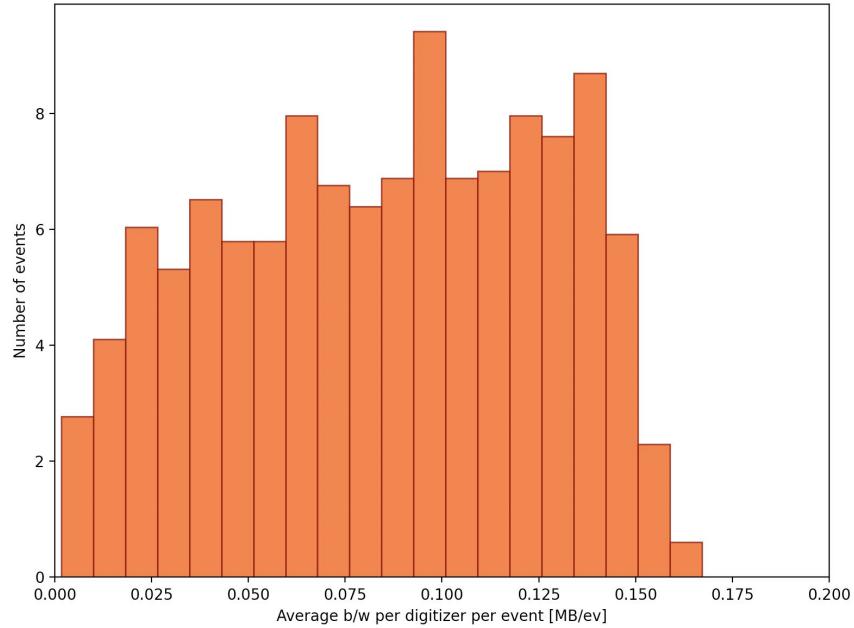
→ ex. [[p_1, p_2, p_3]₁, [p_1, p_2, p_3]₂, ...]

DAQ
simulation

Full waveform
simulation

Average Bandwidth per event per digitizer

(top an bottom channels distributed in the same board)



Channel Mapping

Strategy:

- 1) Loop over the reco channels
- 2) Find the PDU number ([1:196] → bottom array [197:406] → top array)
- 3) Find the channel coordinates (x,y)
- 4) Find nearest neighbors

20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4

