Karlsruhe Institute of Technology

PIERRE AUGER OBSERVATORY

Machine learning triggers: feasibility study

status report, what did I do last month, 09.22

Topics today

- Training with random traces
  - Results seem promising so far but…
  - need to effectively tell single muons apart
  - lack (accurate) estimates for Q_{Peak} and Q_{Area}
- Extracting monitoring data from random traces
  - Online estimate finally accesible
  - Random traces aren‘t timestamped
  - Daily average most likely not good enough
- Doing the online/offline estimate myself
  - A bunch of problems
  - A lot of work so far

Training with random traces

- **Hardware triggers ~ 2 / 1000 false signals**
- **Neural networks ~ 2 / 5000 false signals**

} Rates depend on my injection, so take this with a grain of salt

```
DATASET
HardwareClassifier validation_data                                    -> Acc = 84.09%, TPR = 99.8375%

ENSEMBLES/minimal_conv2d_real_background/ensemble_1/ validation_data   -> Acc = 99.63%, TPR = 99.9708%
ENSEMBLES/minimal_conv2d_real_background/ensemble_2/ validation_data   -> Acc = 99.63%, TPR = 99.9490%
ENSEMBLES/minimal_conv2d_real_background/ensemble_3/ validation_data   -> Acc = 99.62%, TPR = 99.9465%
ENSEMBLES/minimal_conv2d_real_background/ensemble_4/ validation_data   -> Acc = 99.61%, TPR = 99.9808%
ENSEMBLES/minimal_conv2d_real_background/ensemble_5/ validation_data   -> Acc = 99.62%, TPR = 99.9736%
ENSEMBLES/minimal_conv2d_real_background/ensemble_6/ validation_data   -> Acc = 99.63%, TPR = 99.9493%
ENSEMBLES/minimal_conv2d_real_background/ensemble_7/ validation_data   -> Acc = 99.63%, TPR = 99.9485%
ENSEMBLES/minimal_conv2d_real_background/ensemble_8/ validation_data   -> Acc = 99.63%, TPR = 99.9386%
ENSEMBLES/minimal_conv2d_real_background/ensemble_9/ validation_data   -> Acc = 99.62%, TPR = 99.9730%
ENSEMBLES/minimal_conv2d_real_background/ensemble_10/ validation_data  -> Acc = 99.61%, TPR = 99.9683%
```

Training with random traces

- **Hardware triggers ~ 2 / 1000 false signals**
- **Neural networks ~ 2 / 5000 false signals**

} Rates depend on my injection, so take this with a grain of salt

```
DATASET
HardwareClassifier validation_data                                  -> Acc = 84.09%, TPR = 99.8375%

ENSEMBLES/minimal_conv2d_real_background/ensemble_1/ validation_data   -> Acc = 99.63%, TPR = 99.9708%
ENSEMBLES/minimal_conv2d_real_background/ensemble_2/ validation_data   -> Acc = 99.63%, TPR = 99.9490%
ENSEMBLES/minimal_conv2d_real_background/ensemble_3/ validation_data   -> Acc = 99.62%, TPR = 99.9465%
ENSEMBLES/minimal_conv2d_real_background/ensemble_4/ validation_data   -> Acc = 99.61%, TPR = 99.9808%
ENSEMBLES/minimal_conv2d_real_background/ensemble_5/ validation_data   -> Acc = 99.62%, TPR = 99.9736%
ENSEMBLES/minimal_conv2d_real_background/ensemble_6/ validation_data   -> Acc = 99.63%, TPR = 99.9493%
ENSEMBLES/minimal_conv2d_real_background/ensemble_7/ validation_data   -> Acc = 99.63%, TPR = 99.9485%
ENSEMBLES/minimal_conv2d_real_background/ensemble_8/ validation_data   -> Acc = 99.63%, TPR = 99.9386%
ENSEMBLES/minimal_conv2d_real_background/ensemble_9/ validation_data   -> Acc = 99.62%, TPR = 99.9730%
ENSEMBLES/minimal_conv2d_real_background/ensemble_10/ validation_data  -> Acc = 99.61%, TPR = 99.9683%
```

Use online daily avg. Q_{Peak}, Q_{Area} for calibration

Training with random traces

- **Hardware triggers ~ 2 / 1000 false signals**
- **Neural networks ~ 2 / 5000 false signals**

} Rates depend on my injection, so take this with a grain of salt

```
DATASET
HardwareClassifier validation_data                                  -> Acc = 84.09%, TPR = 99.8375%

ENSEMBLES/minimal_conv2d_real_background/ensemble_1/ validation_data   -> Acc = 99.63%, TPR = 99.9708%
ENSEMBLES/minimal_conv2d_real_background/ensemble_2/ validation_data   -> Acc = 99.63%, TPR = 99.9490%
ENSEMBLES/minimal_conv2d_real_background/ensemble_3/ validation_data   -> Acc = 99.62%, TPR = 99.9465%
ENSEMBLES/minimal_conv2d_real_background/ensemble_4/ validation_data   -> Acc = 99.61%, TPR = 99.9808%
ENSEMBLES/minimal_conv2d_real_background/ensemble_5/ validation_data   -> Acc = 99.62%, TPR = 99.9736%
ENSEMBLES/minimal_conv2d_real_background/ensemble_6/ validation_data   -> Acc = 99.63%, TPR = 99.9493%
ENSEMBLES/minimal_conv2d_real_background/ensemble_7/ validation_data   -> Acc = 99.63%, TPR = 99.9485%
ENSEMBLES/minimal_conv2d_real_background/ensemble_8/ validation_data   -> Acc = 99.63%, TPR = 99.9386%
ENSEMBLES/minimal_conv2d_real_background/ensemble_9/ validation_data   -> Acc = 99.62%, TPR = 99.9730%
ENSEMBLES/minimal_conv2d_real_background/ensemble_10/ validation_data  -> Acc = 99
```

Use online daily avg. Q_{Peak}, Q_{Area} for calibration

- **Triggers at a rate of ≈ 30 kHz …**
- **Mostly noise, but also … Muons?**

```
random traces injected: 10000
summed traces duration: 0.17s
total T2 trigger found: 5714
**********************************
TRIGGER FREQUENCY = 33614.9285 Hz
```

Training with random traces

- **Single muons from shower library:** ~ $\mathcal{O}(1 \text{ in } 4 - 5 \text{ trace batches})$
- **Background muons in random traces:** ~$\mathcal{O}(1 \text{ in } 12 \text{ trace batches})$
- **How to account for this?**
  - **Cut on Q_{Area} (e.g. $1\ \text{VEM}_{\text{Charge}}$)**

```
Acc = 87.01%, TPR = 99.7535%
Acc = 87.15%, TPR = 99.8088%
Acc = 87.49%, TPR = 99.7859%
Acc = 86.94%, TPR = 99.7316%
Acc = 86.97%, TPR = 99.8168%
Acc = 87.02%, TPR = 99.7796%
Acc = 87.21%, TPR = 99.7249%
Acc = 87.19%,
Acc = 86.99%,
Acc = 87.14%,
random traces injected: 10000
summed traces duration: 0.17s
total T2 trigger found: 192
**********
TRIGGER FREQUENCY = 1129.5181 Hz
```

Training with random traces

- **Single muons from shower library:** ~ $\mathcal{O}(1 \text{ in } 4 - 5 \text{ trace batches})$
- **Background muons in random traces:** ~$\mathcal{O}(1 \text{ in } 12 \text{ trace batches})$

- **How to account for this?**
  - **Cut on Q_{Area} (e.g. $1\ \text{VEM}_{\text{Charge}}$)**
  - **Train with lower prior**

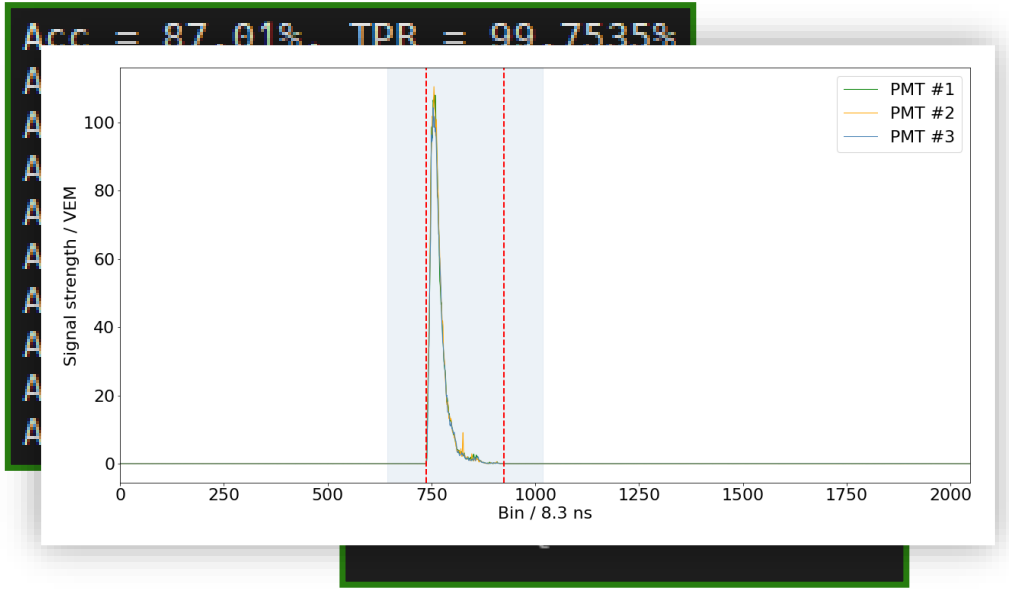
```
Acc = 87.01%, TPR = 99.7535%
Acc = 87.15%, TPR = 99.8088%
Acc = 87.49%, TPR = 99.7859%
Acc = 86.94%, TPR = 99.7316%
Acc = 86.97%, TPR = 99.8168%
Acc = 87.02%, TPR = 99.7796%
Acc = 87.21%, TPR = 99.7249%
Acc = 87.19%,
Acc = 86.99%,
Acc = 87.14%,
```

```
random traces injected: 10000
summed traces duration: 0.17s
total T2 trigger found: 192
**********
TRIGGER FREQUENCY = 1129.5181 Hz
```

Training with random traces

- **Single muons from shower library:** ~ $\mathcal{O}(1 \text{ in } 4 - 5 \text{ trace batches})$
- **Background muons in random traces:** ~$\mathcal{O}(1 \text{ in } 12 \text{ trace batches})$

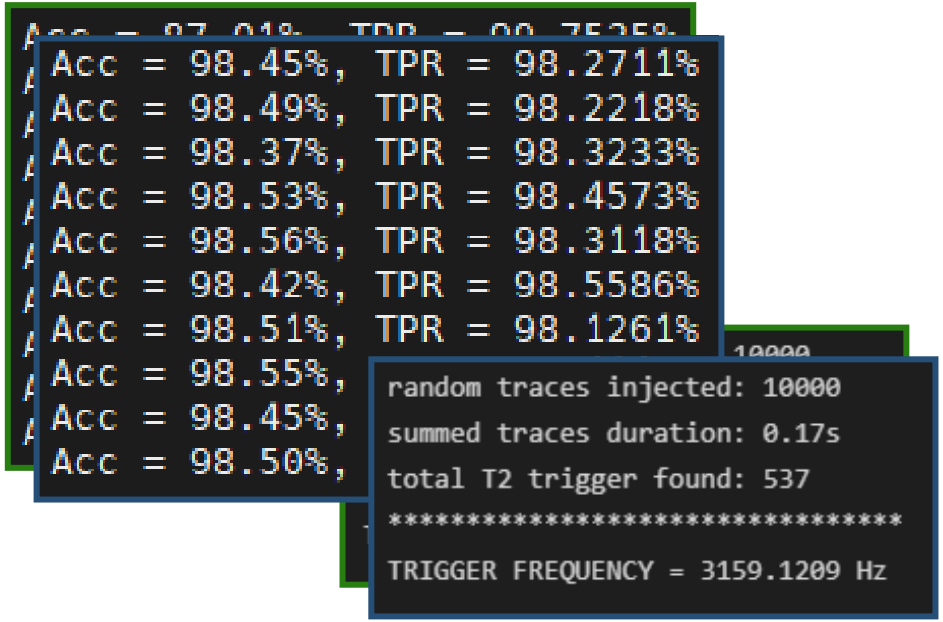
- **How to account for this?**
  - **Cut on Q_{Area} (e.g. $1\ \text{VEM}_{\text{Charge}}$)**
  - **Train with lower prior**



Training with random traces

- **Single muons from shower library:** ~ $\mathcal{O}(1 \text{ in } 4 - 5 \text{ trace batches})$
- **Background muons in random traces:** ~$\mathcal{O}(1 \text{ in } 12 \text{ trace batches})$

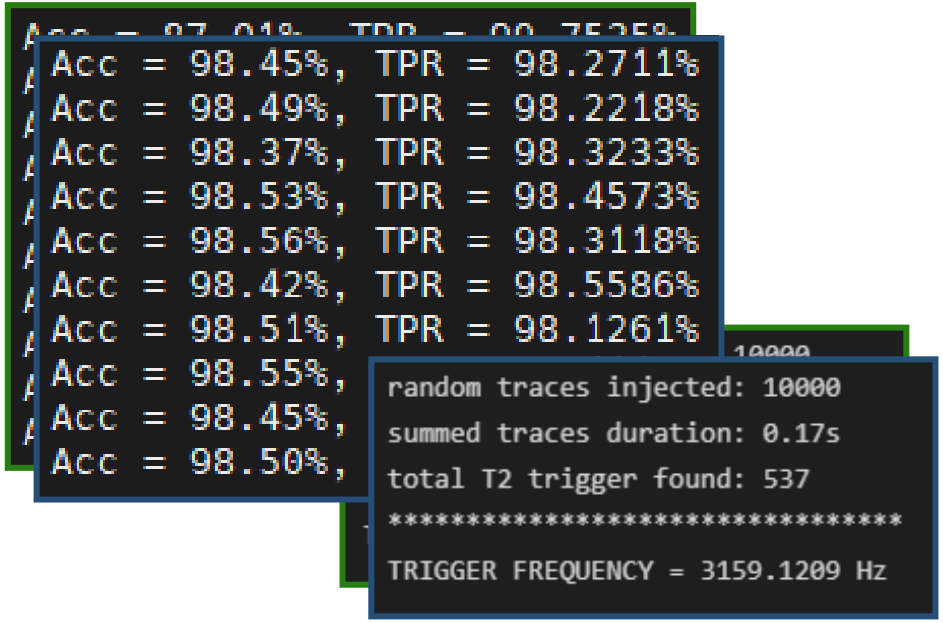
- **How to account for this?**
  - **Cut on Q_{Area} (e.g. $1\ \text{VEM}_{\text{Charge}}$)**
  - **Train with lower prior**



Training with random traces

- **Single muons from shower library:** ~ $\mathcal{O}(1 \text{ in } 4 - 5 \text{ trace batches})$
- **Background muons in random traces:** ~$\mathcal{O}(1 \text{ in } 12 \text{ trace batches})$

- **How to account for this?**
  - **Cut on Q_{Area} (e.g. $1\ \text{VEM}_{\text{Charge}}$)**
  - **Train with lower prior**
  - **TODO: Cut + low prior**



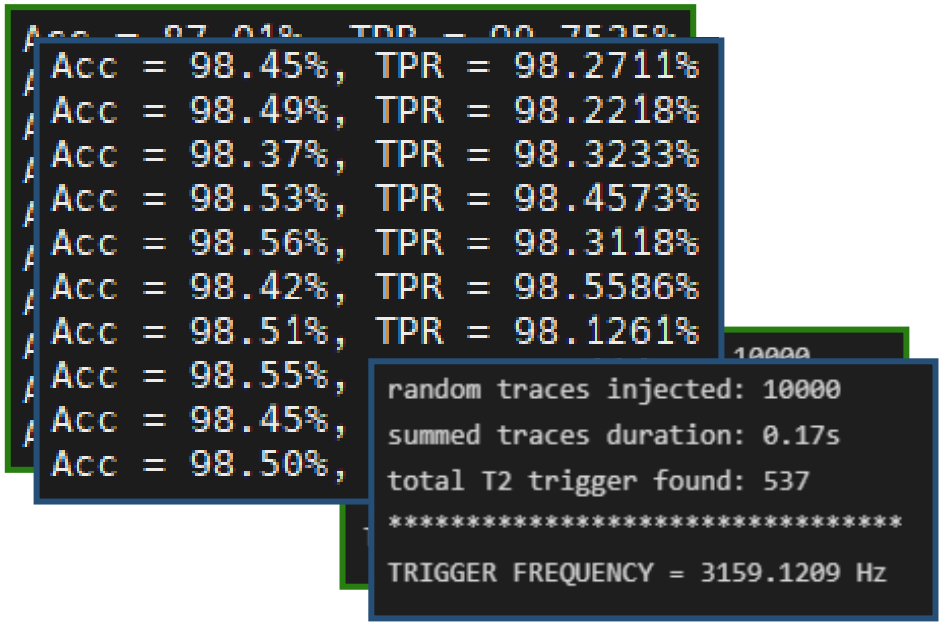
Training with random traces

- **Single muons from shower library:** ~ $\mathcal{O}(1 \text{ in } 4 - 5 \text{ trace batches})$
- **Background muons in random traces:** ~$\mathcal{O}(1 \text{ in } 12 \text{ trace batches})$

- **How to account for this?**
  - **Cut on Q_{Area} (e.g. $1\ \text{VEM}_{\text{Charge}}$)**
  - **Train with lower prior**
  - **TODO: Cut + low prior**

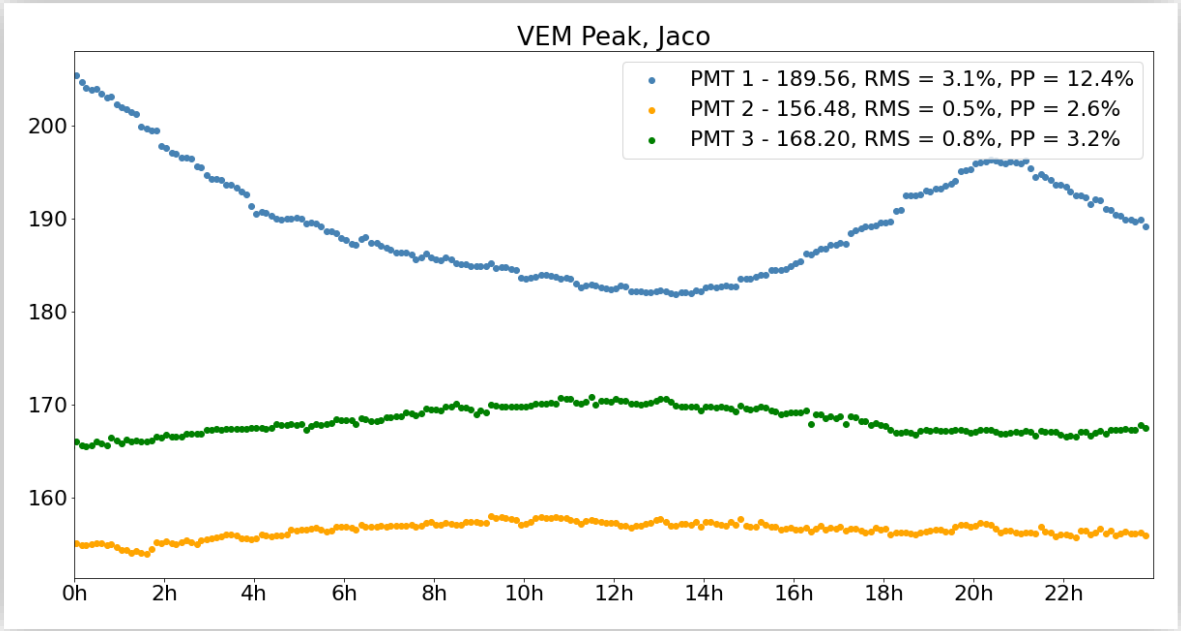
  - **Flag single muon showers as background somehow**

  ➡ **Will have to redo simulations!**



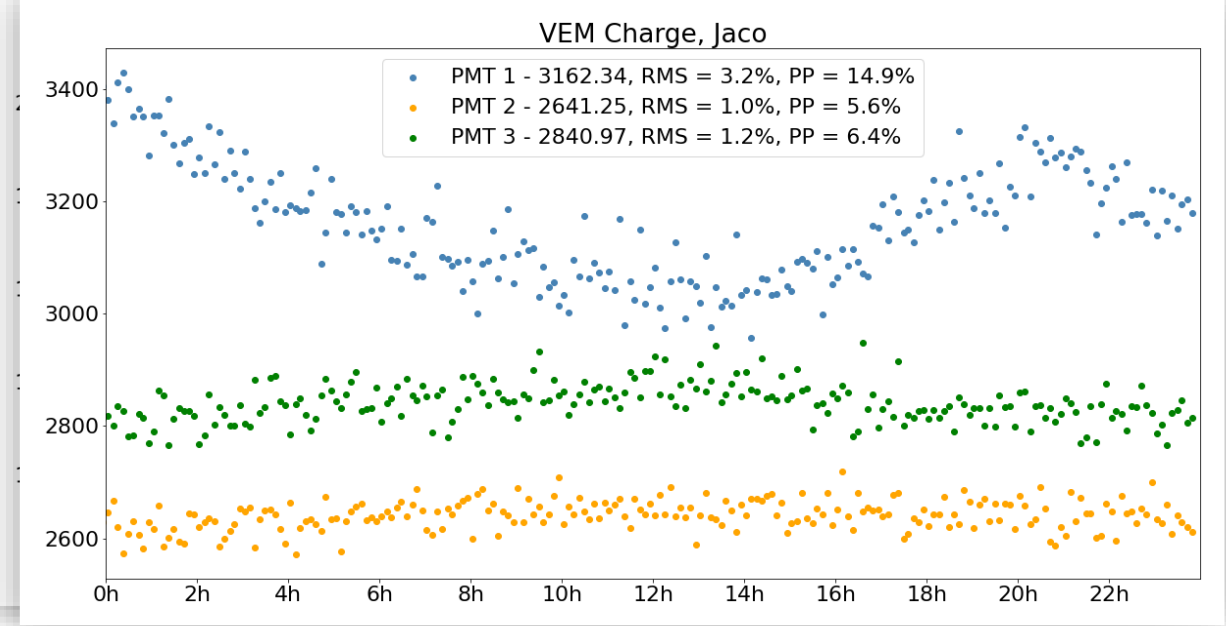
Extracting monitoring data from random traces

- **Online values for Q_{Peak} and Q_{Area} available @ 5 min. resolution**



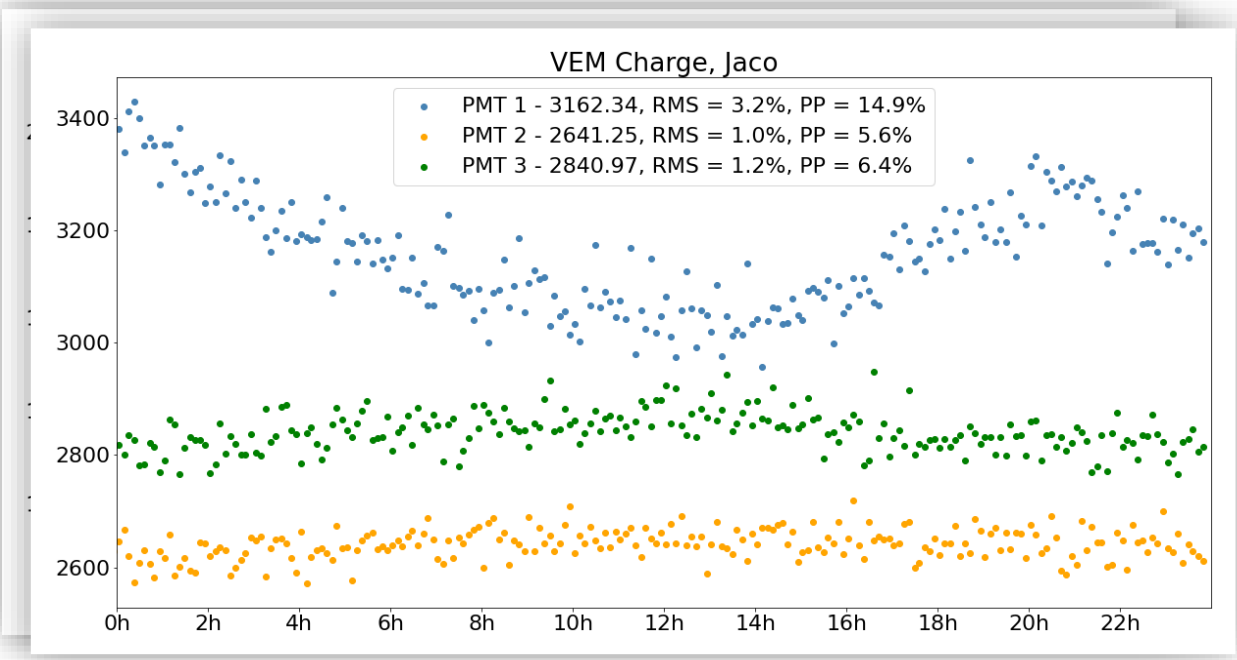
Extracting monitoring data from random traces

- **Online values for Q_{Peak} and Q_{Area} available @ 5 min. resolution**



Extracting monitoring data from random traces

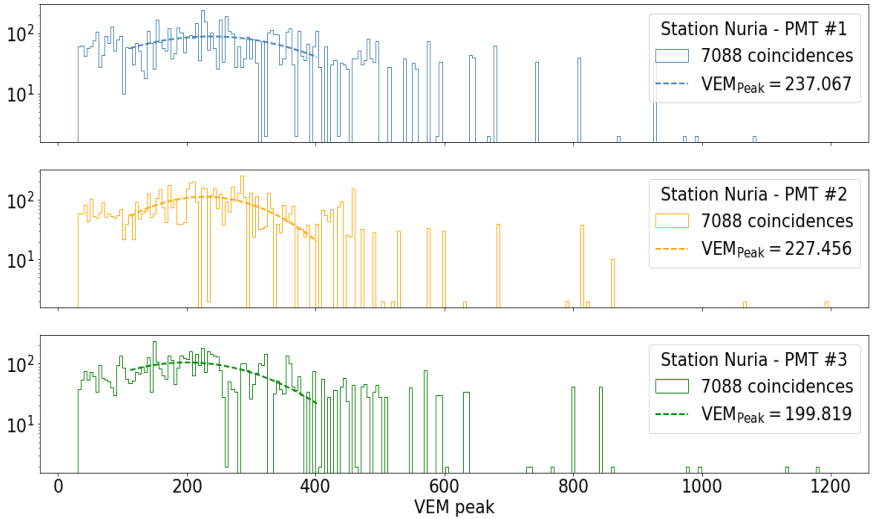
- **Online values for Q_{Peak} and Q_{Area} available @ 5 min. resolution**



- **Nuria, ID: 0056**
  - Q_{Peak}: $\sigma = 0.5\% \mid 2.2\%$
  - Q_{Area}: $\sigma = 1.2\% \mid 6.9\%$
- **Peru, ID: 1737**
  - Q_{Peak}: $\sigma = 3.1\% \mid 15.3\%$
  - Q_{Area}: $\sigma = 3.4\% \mid 18.5\%$
- **(Le Qui Don)**
  - **not analysed due to low statistics in random trace files**

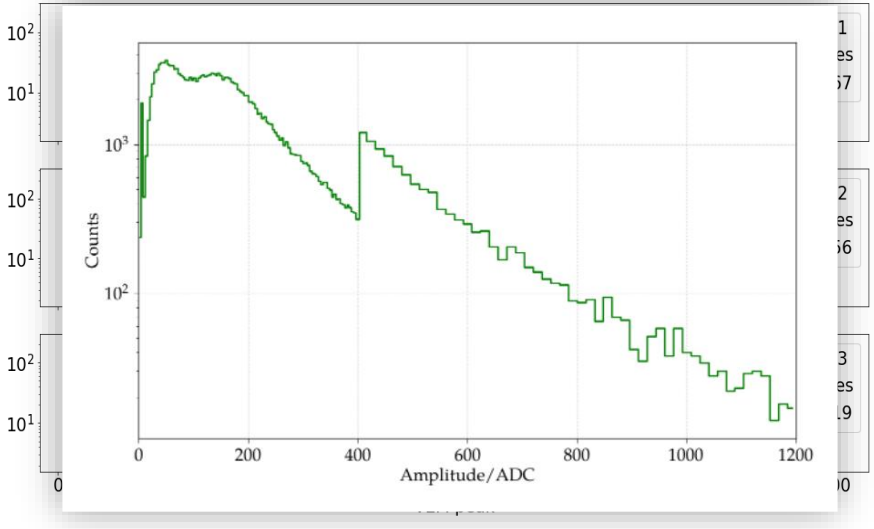
Doing the online/offline estimate myself

- **Offline estimation of Q_{Peak} and Q_{Area} - GAP2020-0??**
- **Consider only traces that satisfy calibration trigger:**
  - Extract 20 bins before and 49 bins after latch bin
  - Latch bin $\geq$ Baseline + 30 ADC
  - Multiplicity = 1
- **Q_{Peak} → Histogram max of each trace**
- **Q_{Area} → Histogram sum of each trace**
- **Fit parabola to histogram**
- **Maximum = Offline estimate**



Doing the online/offline estimate myself

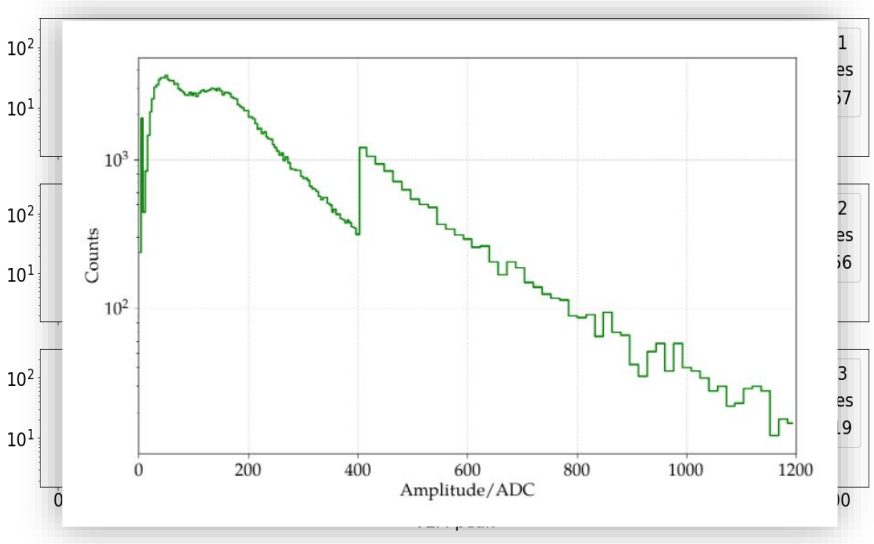
- **Offline estimation of Q_{Peak} and Q_{Area} - GAP2020-0??**
- **Consider only traces that satisfy calibration trigger:**
  - Extract 20 bins before and 49 bins after latch bin
  - Latch bin $\geq$ Baseline + 30 ADC
  - Multiplicity = 1
- **Q_{Peak} → Histogram max of each trace**
- **Q_{Area} → Histogram sum of each trace**
- **Fit parabola to histogram**
- **Maximum = Offline estimate**



Doing the online/offline estimate myself

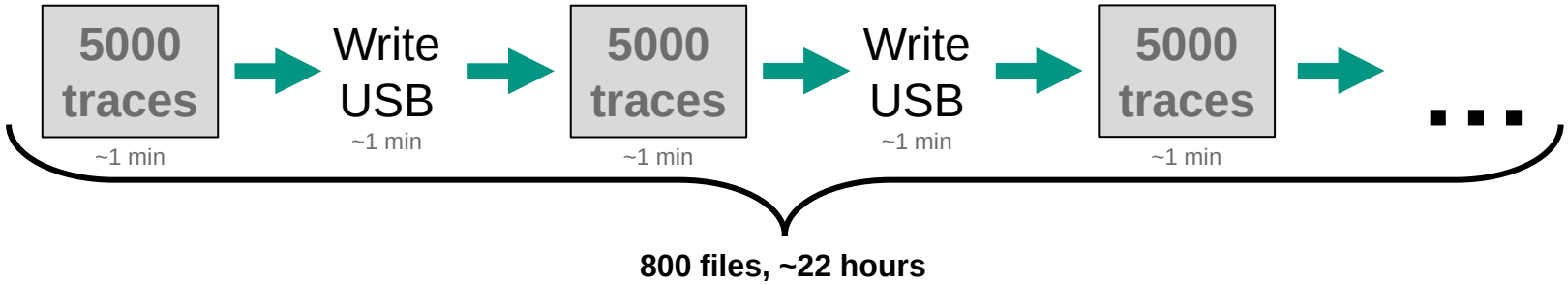
- **Offline estimation of Q_{Peak} and Q_{Area} - GAP2020-0??**
- **Consider only traces that satisfy calibration trigger:**
  - Extract 20 bins before and 49 bins after latch bin
  - Latch bin $\geq$ Baseline + 30 ADC
  - Multiplicity = 1
- **Q_{Peak} → Histogram max of each trace**
- **Q_{Area} → Histogram sum of each trace**
- **Fit parabola to histogram**
- **Maximum = Offline estimate**

→ **Too few traces to fill histogram**



Doing the online/offline estimate myself

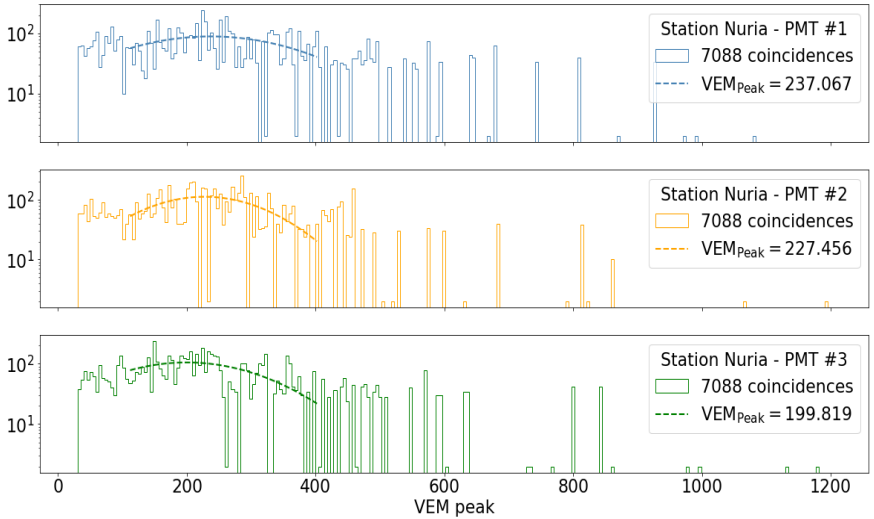
- **Random traces taken locally at 4 stations**
- **Fill buffer with data, then write it to USB**



- **Increasing statistics by combining 3 files (~5 minutes) should be okay**

Doing the online/offline estimate myself

- **Offline estimation of Q_{Peak} and Q_{Area} - GAP2020-0??**
- **Consider only traces that satisfy calibration trigger:**
  - Extract 20 bins before and 49 bins after latch bin
  - Latch bin $\geq$ Baseline + 30 ADC
  - Multiplicity = 1
- **Q_{Peak} → Histogram max of each trace**
- **Q_{Area} → Histogram sum of each trace**
- **Fit parabola to histogram**
- **Maximum = Offline estimate**



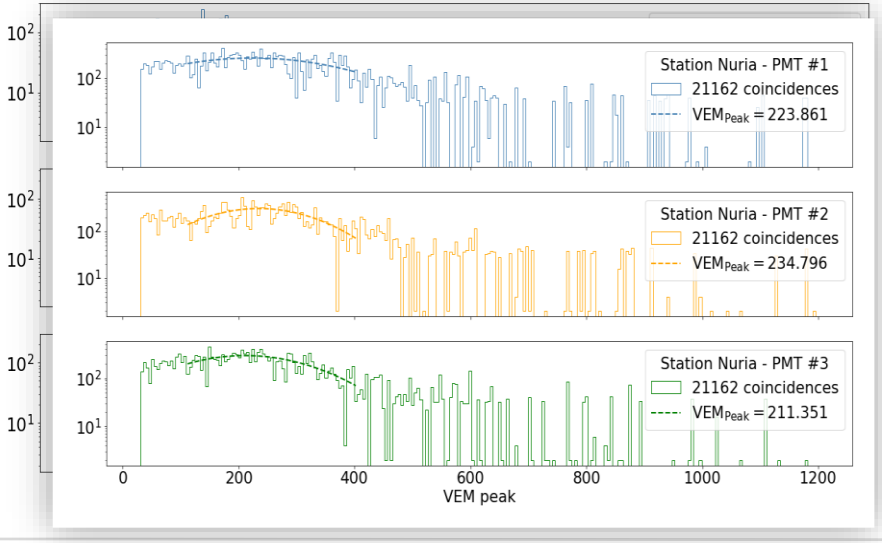
Doing the online/offline estimate myself

- **Offline estimation of Q_{Peak} and Q_{Area} - GAP2020-0??**
- **Consider only traces that satisfy calibration trigger:**
  - Extract 20 bins before and 49 bins after latch bin
  - Latch bin $\geq$ Baseline + 30 ADC
  - Multiplicity = 1

- **Q_{Peak} → Histogram max of each trace**
- **Q_{Area} → Histogram sum of each trace**

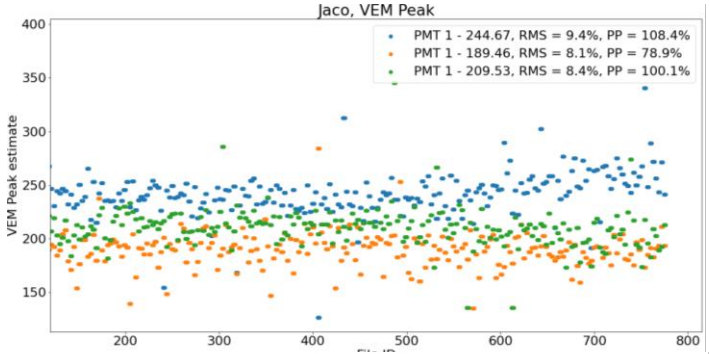
- **Fit parabola to histogram**
- **Maximum = Offline estimate**

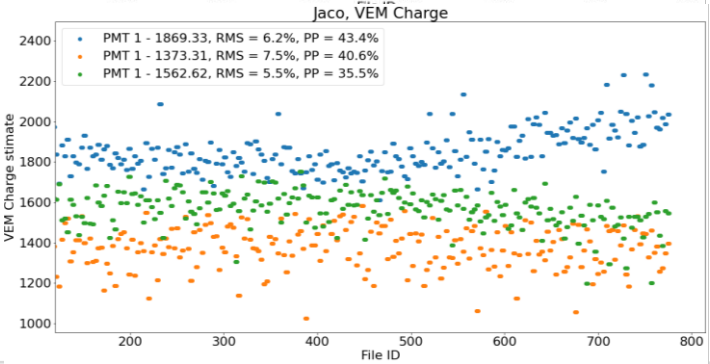
→ **Improves statistics, but not shape**



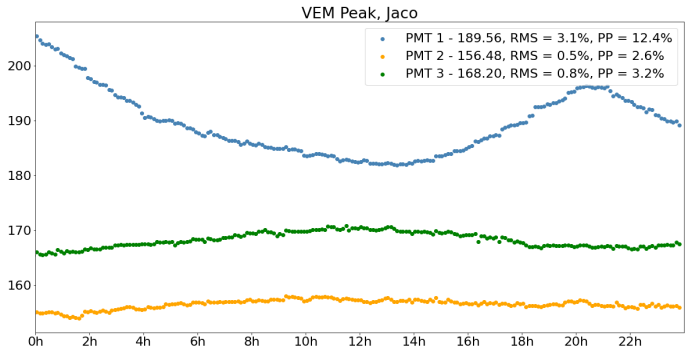
Doing the online/offline estimate myself

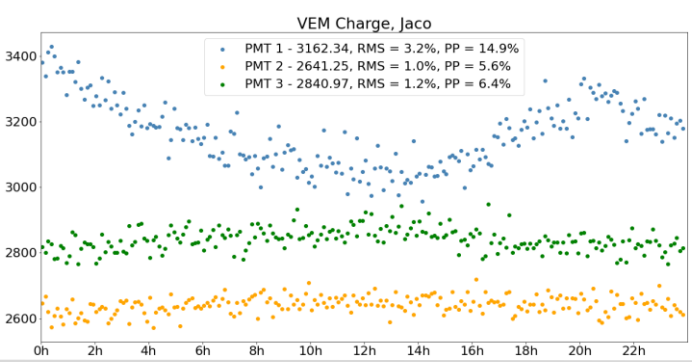
Pauls‘ offline estimate





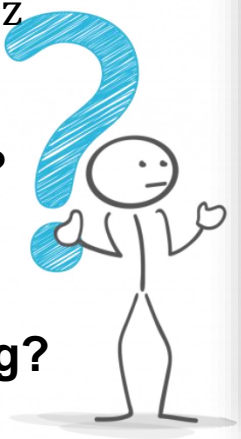
Online monitoring





Doing the online/offline estimate myself

- **Online estimation of Q_{Peak} and Q_{Area} -** https://doi.org/10.1016/j.nima.2006.07.066
- **Calibration trigger with threshold** $1.75 \cdot Q_{\mathrm{Peak}}\big|_{\mathrm{M=3}}$ && $2.5 \cdot Q_{\mathrm{Peak}}\big|_{\mathrm{M=1}}$ **:**
  - Set Q_{Peak} / Q_{Peak} **s.t. trigger rate** $=$ **70** Hz
- **Should result in Th2 rate of 20** Hz
- **Error in my implementation?**
- **Why not set thresholds with Th?**
- **Is this reasonable what I'm doing?**



```
pmt_has_converged = 0
n_triggers = np.zeros(3)

for pmt1, pmt2, pmt3 in zip(station[:,0], station[:,1], station[:,2]):

    for bins in zip(pmt1, pmt2, pmt3):

        # assume all PMTs are working, k70 = 1.75
        # check the actual trigger rate
        if np.all(bins >= 1.75 * I_vem):

            for i in range(3):
                if bins[i] >= 2.5 * I_vem[i]: n_triggers[i] += 1

trigger_rates = n_triggers / (len(Buffer._these_traces) * 2048 * 8.33e-8)

print(f"VEM_peak estimate: {I_vem} ===> {trigger_rates} Hz")

for i in range(3):
    if trigger_rates[i] <= 60: I_vem[i] -= 10
    elif 60 < trigger_rates[i] <= 68: I_vem[i] -= 1
    elif 68 < trigger_rates[i] <= 72: pmt_has_converged += 1
    elif 72 < trigger_rates[i] <= 80: I_vem[i] += 1
    elif 80 < trigger_rates[i]: I_vem[i] += 10

if pmt_has_converged >= 3: break
```