SD Machine learning triggers

Status update 11.04.23

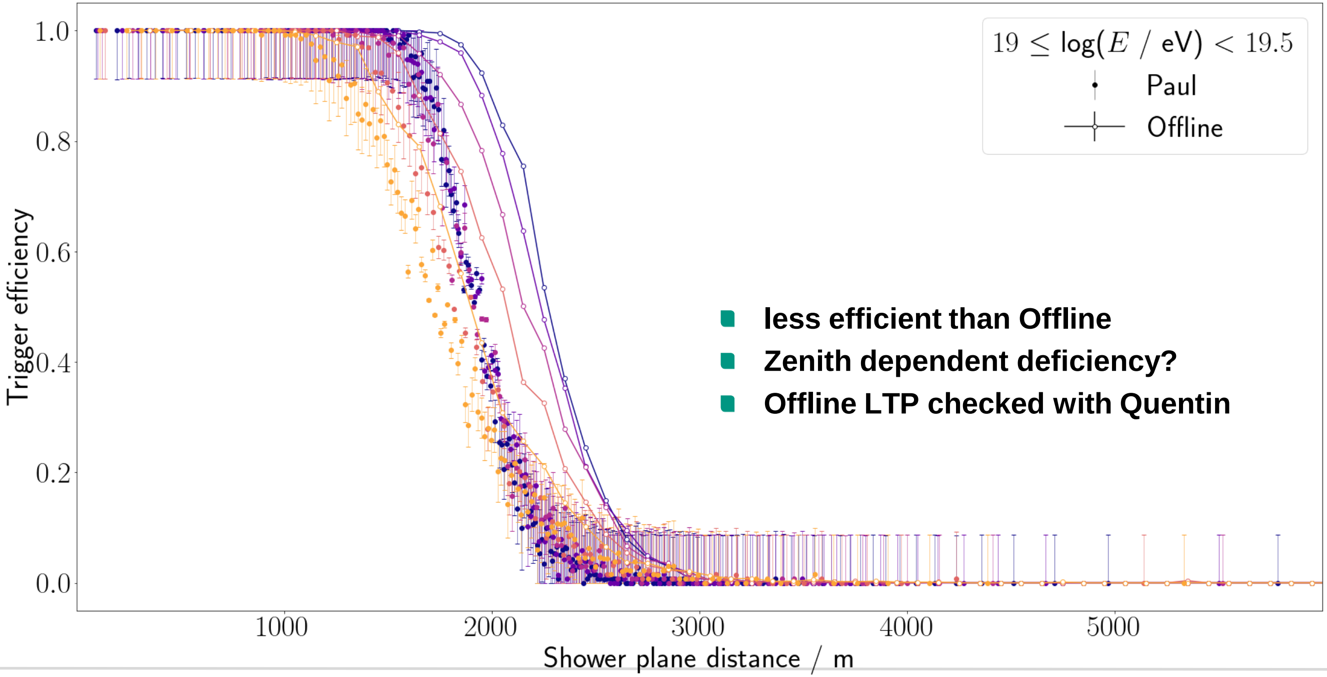
Topics today

- **Issue comparing Lateral Trigger Probabilities**
- **Possible error sources**
  - **Simulation**
  - **ADST readout**
  - **Trace building**
  - **Filtering and downsampling**
  - **Trigger algorithms**
  - **LTP calculation**

Analysis chain

Issue comparing Lateral Trigger Probability

- **My calculated LTPs (for my implementation) don't match Offline**



- **less efficient than Offline**
- **Zenith dependent deficiency?**
- **Offline LTP checked with Quentin**

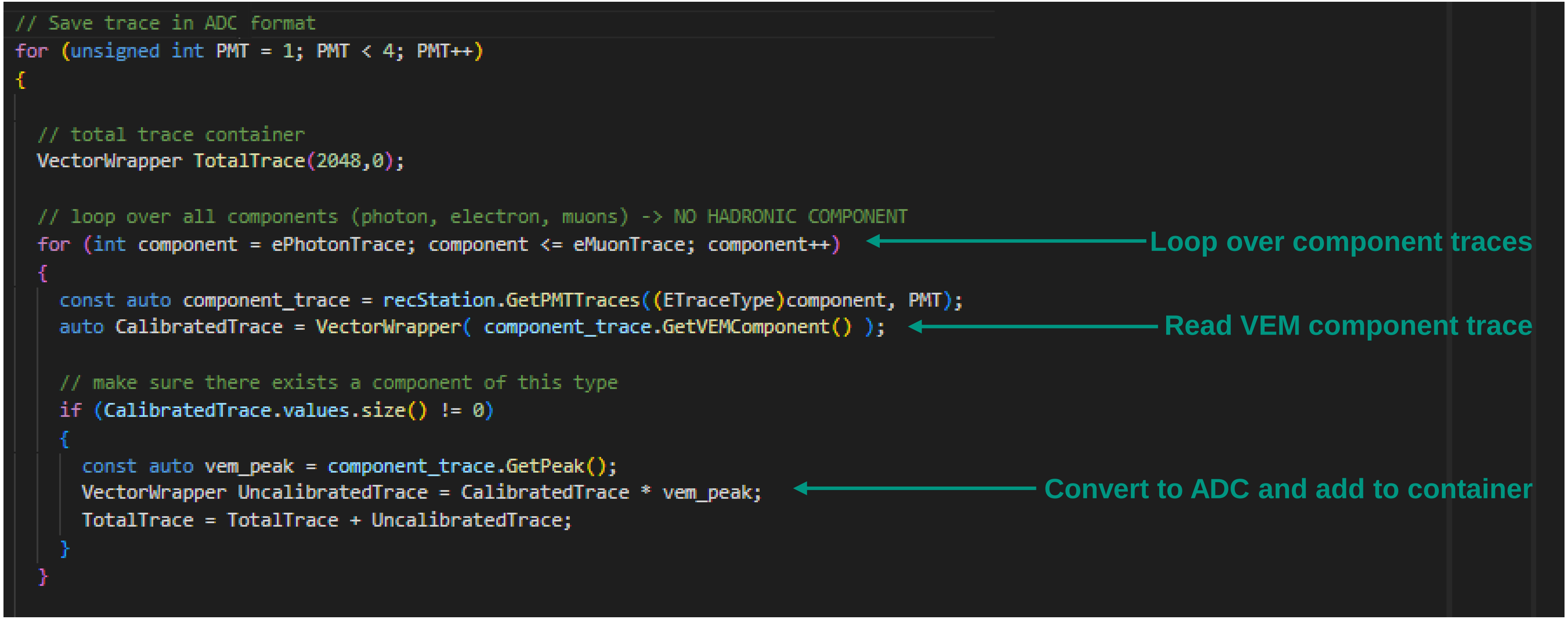
Possible error sources: Simulation

- **Source files from Napoli/Prague cont. library**
  - $16 \leq log(\, E \, / \, \mathrm{eV}) < 19.5$ primary energy protons
  - Hadronic interaction model QGSJET-II.04

- **Extract *all* signal from shower footprint by disabling triggers**
  - See section 6.2 of my thesis for more details

- **Same source files as for Offline calculation**
  - Only difference are trigger thresholds

Possible error sources: ADST readout



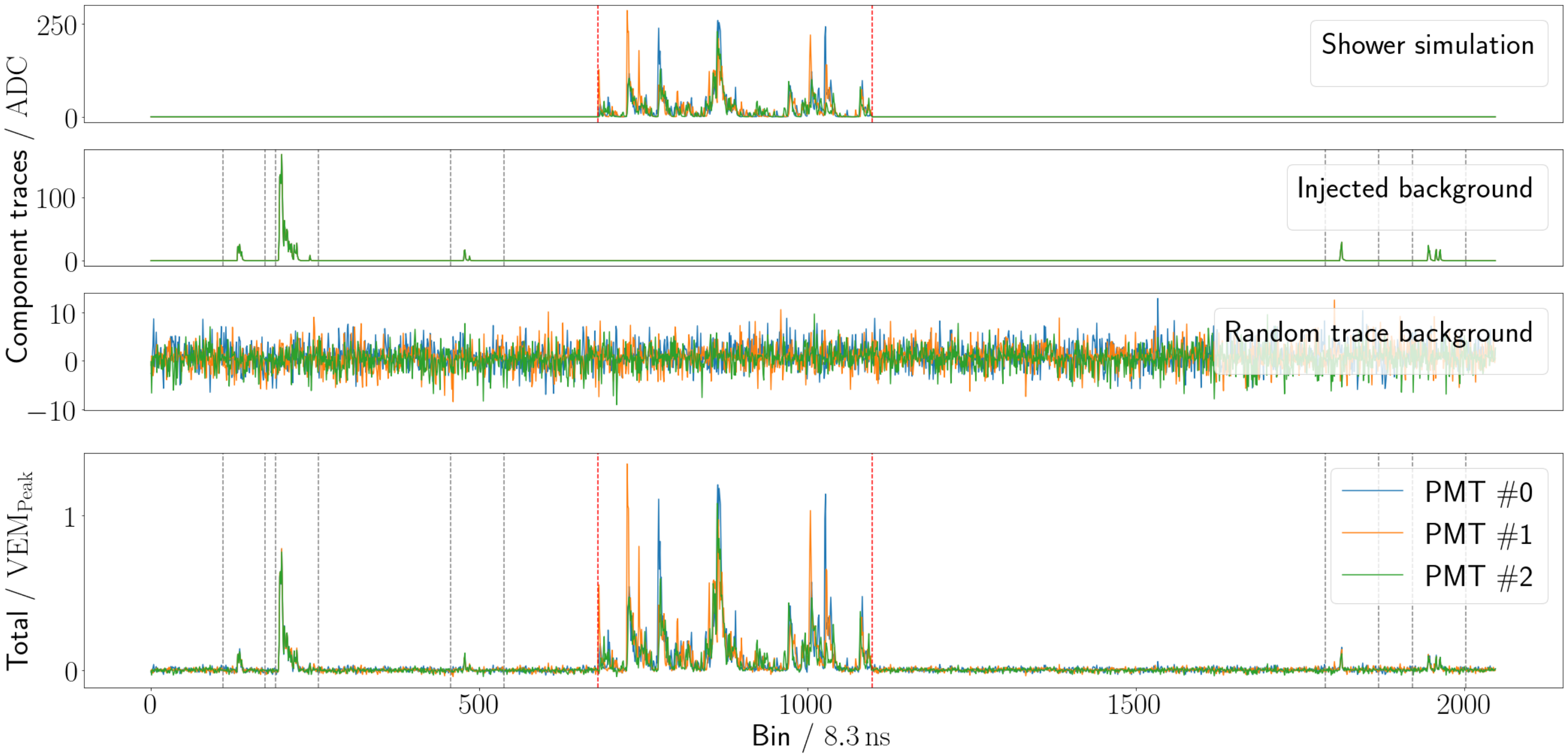
```
// Save trace in ADC format
for (unsigned int PMT = 1; PMT < 4; PMT++)
{

  // total trace container
  VectorWrapper TotalTrace(2048,0);

  // loop over all components (photon, electron, muons) -> NO HADRONIC COMPONENT
  for (int component = ePhotonTrace; component <= eMuonTrace; component++)
  {
    const auto component_trace = recStation.GetPMTTraces((ETraceType)component, PMT);
    auto CalibratedTrace = VectorWrapper( component_trace.GetVEMComponent() );

    // make sure there exists a component of this type
    if (CalibratedTrace.values.size() != 0)
    {
      const auto vem_peak = component_trace.GetPeak();
      VectorWrapper UncalibratedTrace = CalibratedTrace * vem_peak;
      TotalTrace = TotalTrace + UncalibratedTrace;
    }
  }
```

Loop over component traces

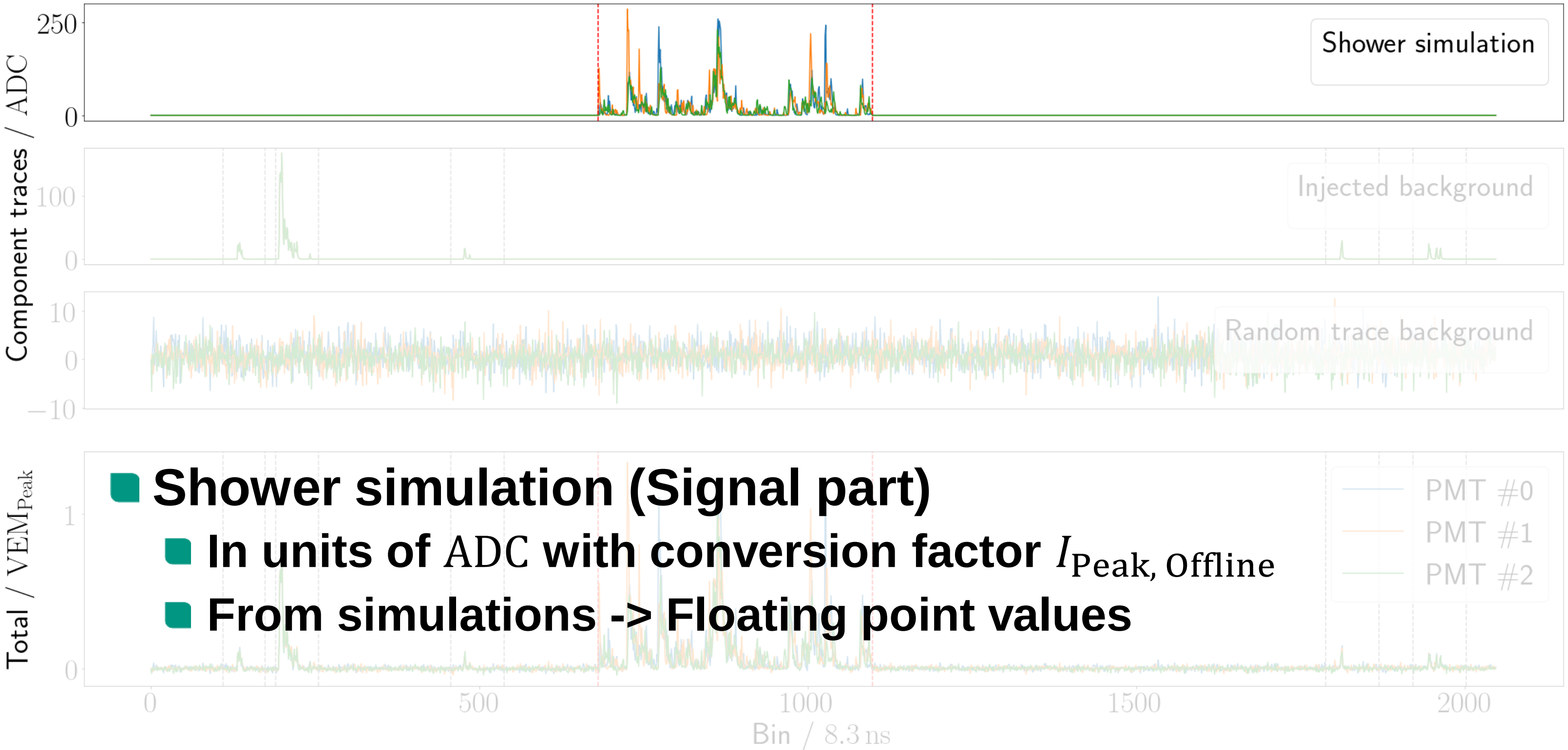
Read VEM component trace

Convert to ADC and add to container

Possible error sources: Trace Building

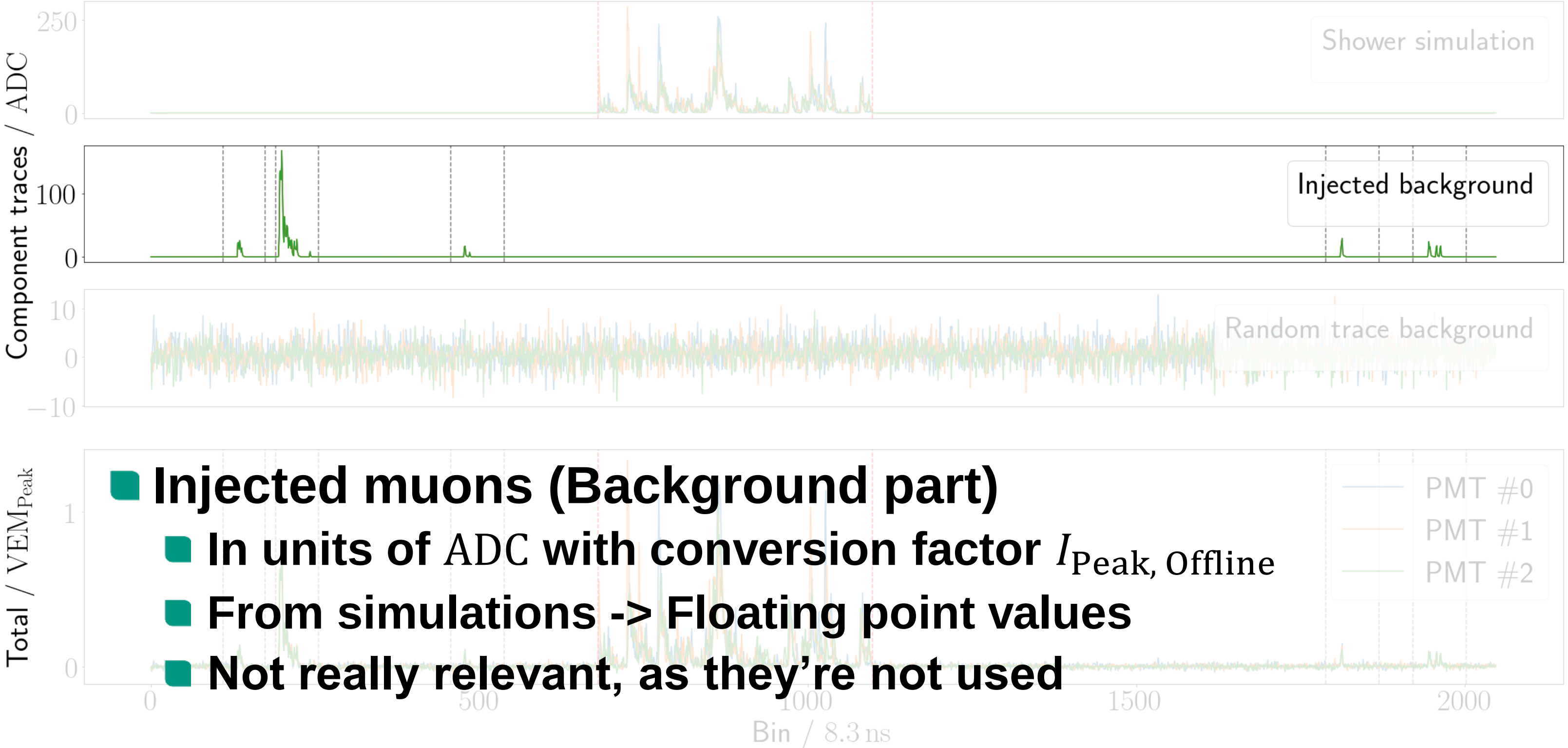


Possible error sources: Trace Building



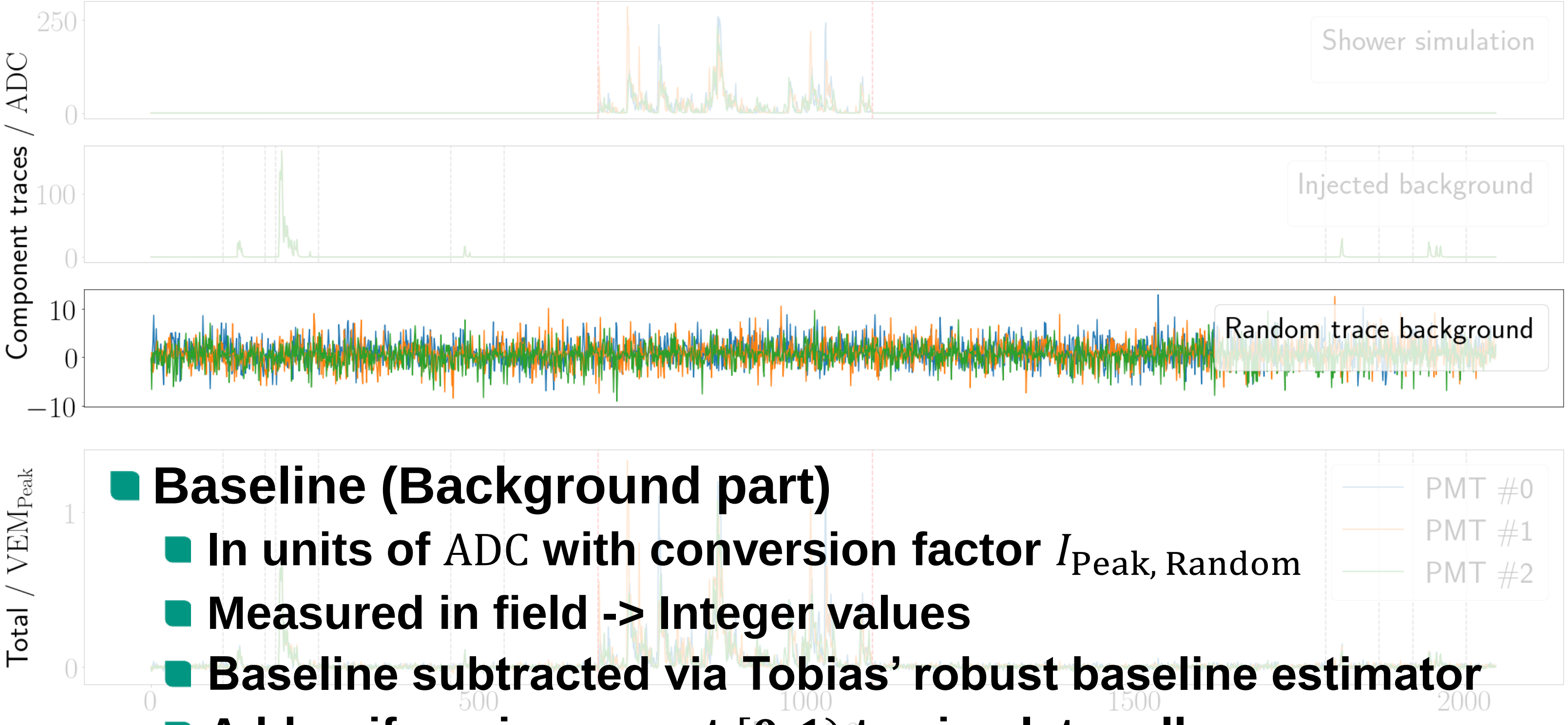
- **Shower simulation (Signal part)**
  - **In units of ADC with conversion factor $I_{\mathrm{Peak,\,Offline}}$**
  - **From simulations -> Floating point values**

Possible error sources: Trace Building



- **Injected muons (Background part)**
  - **In units of ADC with conversion factor $I_{\text{Peak, Offline}}$**
  - **From simulations -> Floating point values**
  - **Not really relevant, as they're not used**

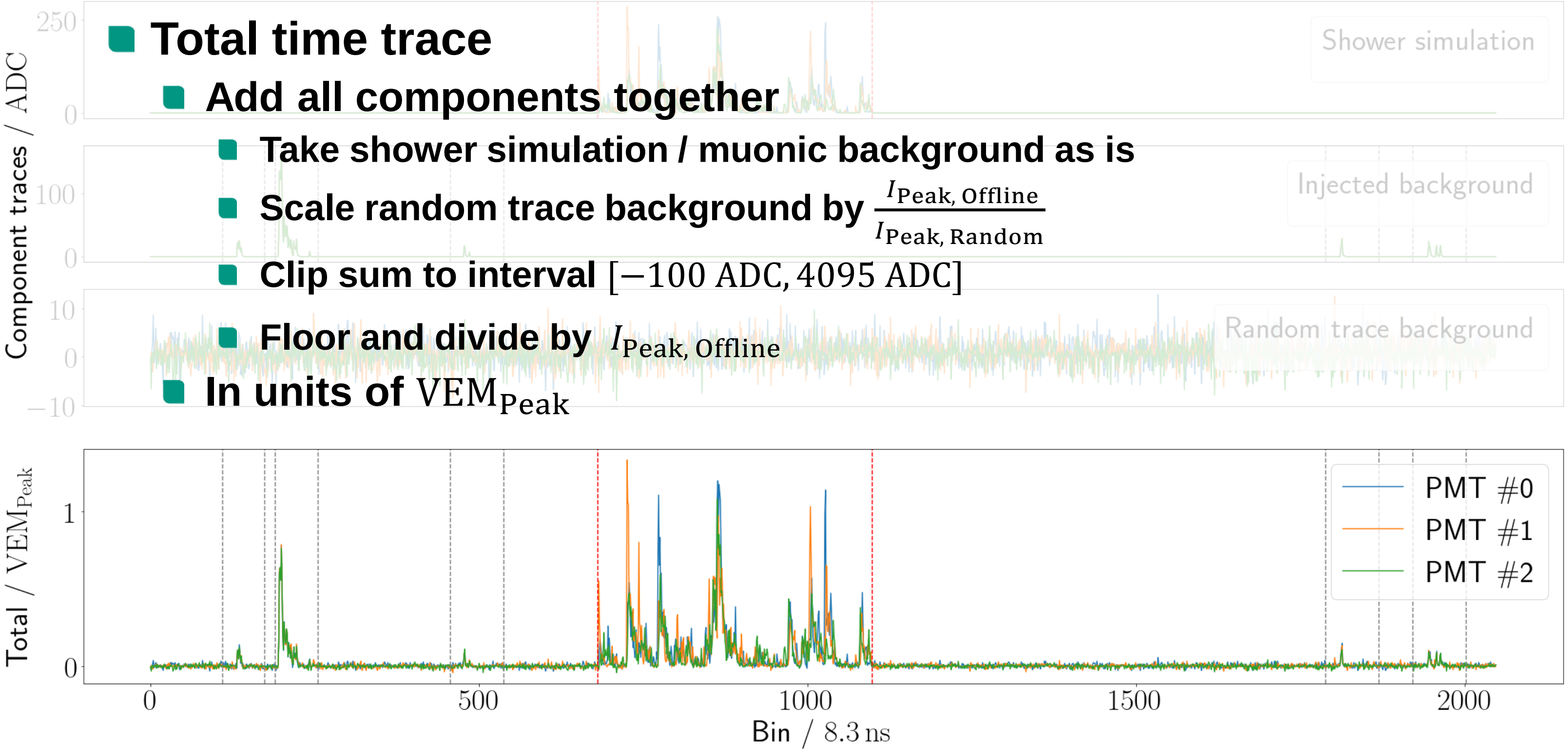
Possible error sources: Trace Building



- **Baseline (Background part)**
  - **In units of ADC with conversion factor $I_\mathrm{Peak,\,Random}$**
  - **Measured in field -> Integer values**
  - **Baseline subtracted via Tobias' robust baseline estimator**
  - **Add uniform increment $[0, 1)$ to simulate rollover**

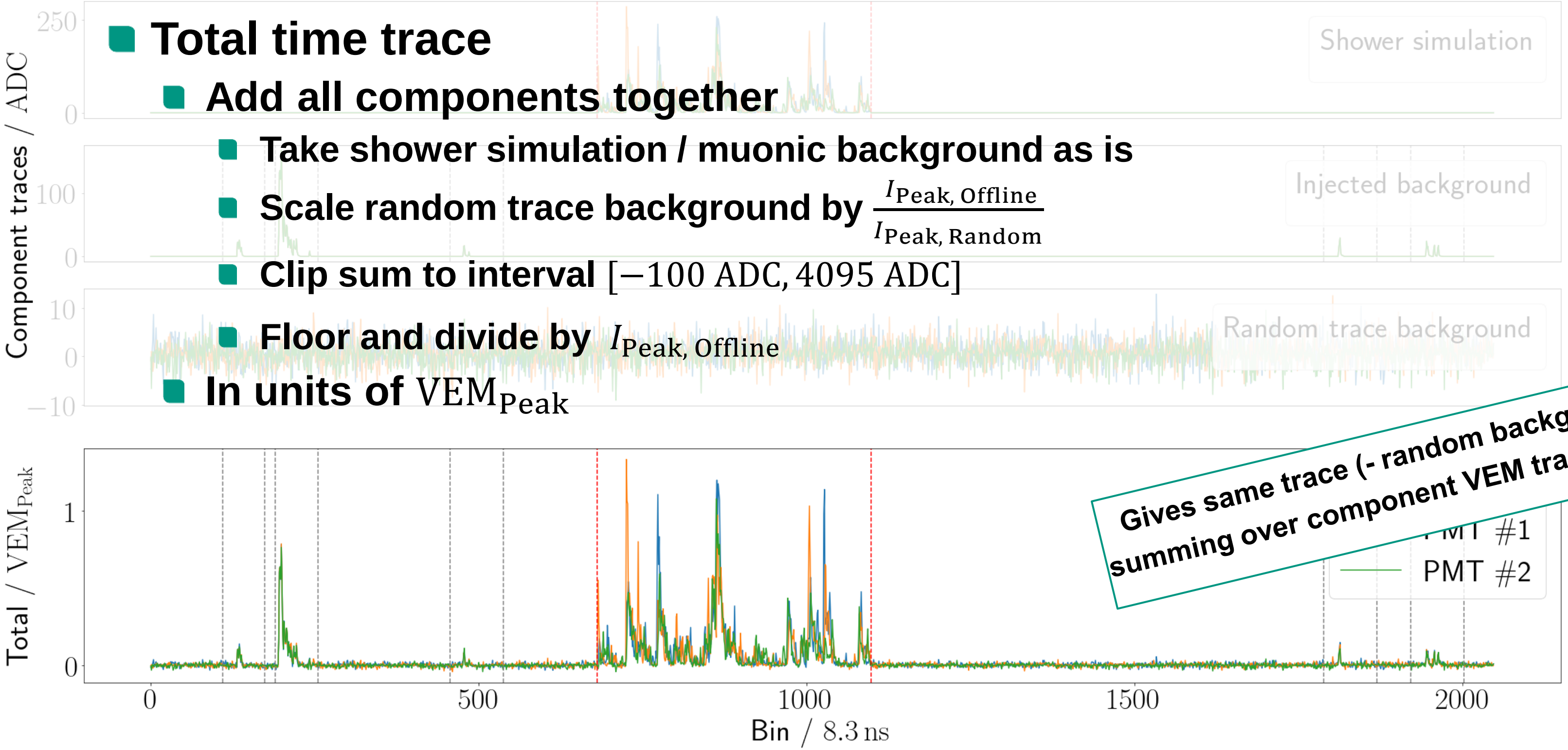
Possible error sources: Trace Building

- **Total time trace**
  - **Add all components together**
    - **Take shower simulation / muonic background as is**
    - **Scale random trace background by** $\frac{I_{\text{Peak, Offline}}}{I_{\text{Peak, Random}}}$
    - **Clip sum to interval** $[-100\ \text{ADC}, 4095\ \text{ADC}]$
    - **Floor and divide by** $I_{\text{Peak, Offline}}$
  - **In units of** VEM_{Peak}



Possible error sources: Trace Building

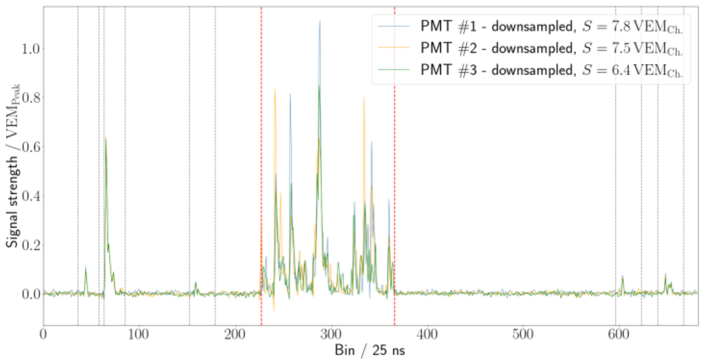
- Total time trace
  - Add all components together
    - Take shower simulation / muonic background as is
    - Scale random trace background by $\frac{I_{\text{Peak, Offline}}}{I_{\text{Peak, Random}}}$
    - Clip sum to interval $[-100\ \text{ADC}, 4095\ \text{ADC}]$
    - Floor and divide by $I_{\text{Peak, Offline}}$
  - In units of VEM_{Peak}

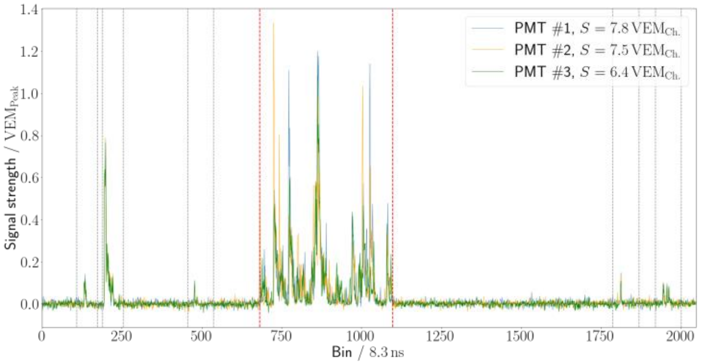


Gives same trace (- random background) as summing over component VEM traces in ADST

Possible error sources: Filter and Downsample

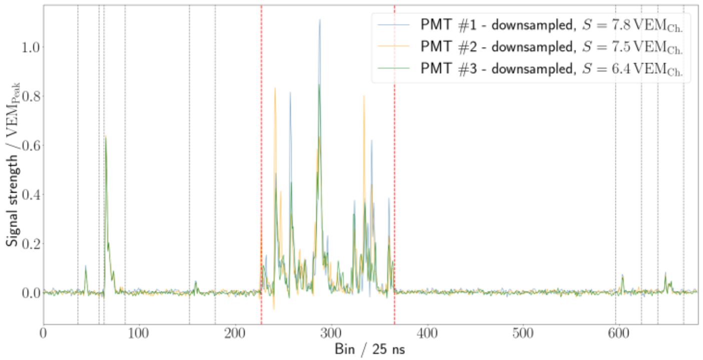
- **In compatibility mode**
  - Filter and downsample after components have been added but before converting from ADC to $\mathrm{VEM}_{\mathrm{Peak}}$
  - Same algorithm as `trunk/Framework/SDetector/UUBDownsampleFilter.h`
  - Conversion factor changes from $I_{\mathrm{Peak,\,Offline}}$ to $I_{\mathrm{Peak,\,Compat.}}$

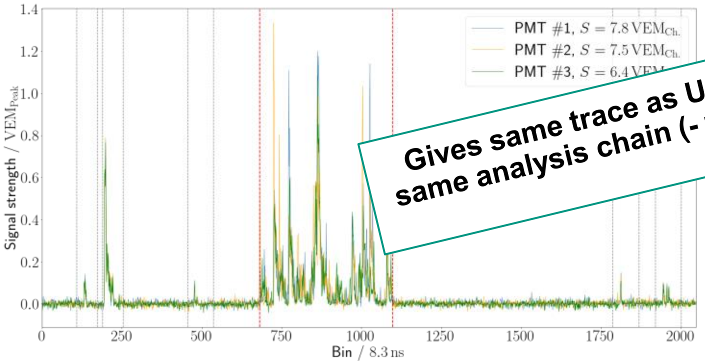




Possible error sources: Filter and Downsample

- **In compatibility mode**
  - Filter and downsample after components have been added but before converting from ADC to $\mathrm{VEM}_{\mathrm{Peak}}$
  - Same algorithm as `trunk/Framework/SDetector/UUBDownsampleFilter.h`





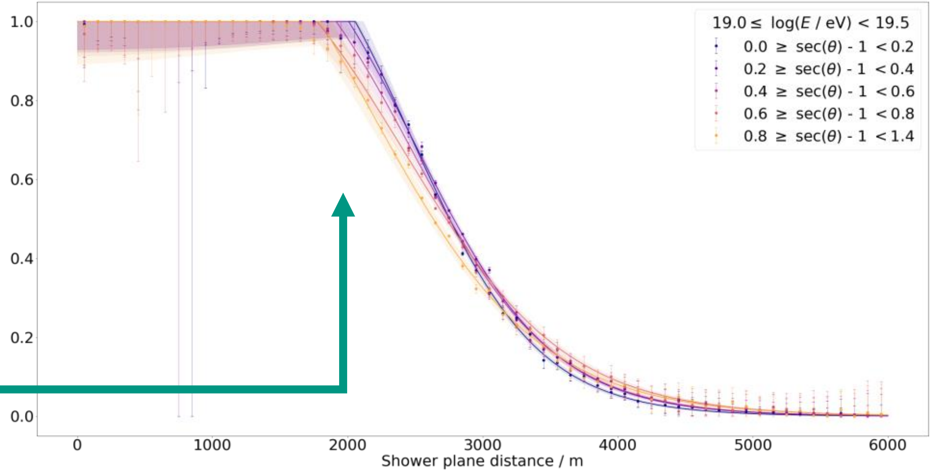
Gives same trace as UB sim that went through same analysis chain (- filtering & downsampling)

Possible error sources: Trigger algorithms

- **Th, ToT, ToTd**
  - Implemented to the best of my understanding of `Framework/SDetector/StationTriggerAlgorithm.h`
- **MoPS**
  - Compatibility version implemented but unused due to sketchy integral check
  - MoPS trigger rate cannot explain such a large discrepancy
- **Comparison to Offline per Trigger rate**
  - ≈ constant offset for Th for all angles/energies → calibration error?
  - ToT, ToTd high discrepancy for vertical showers

Possible error sources: LTP calculation

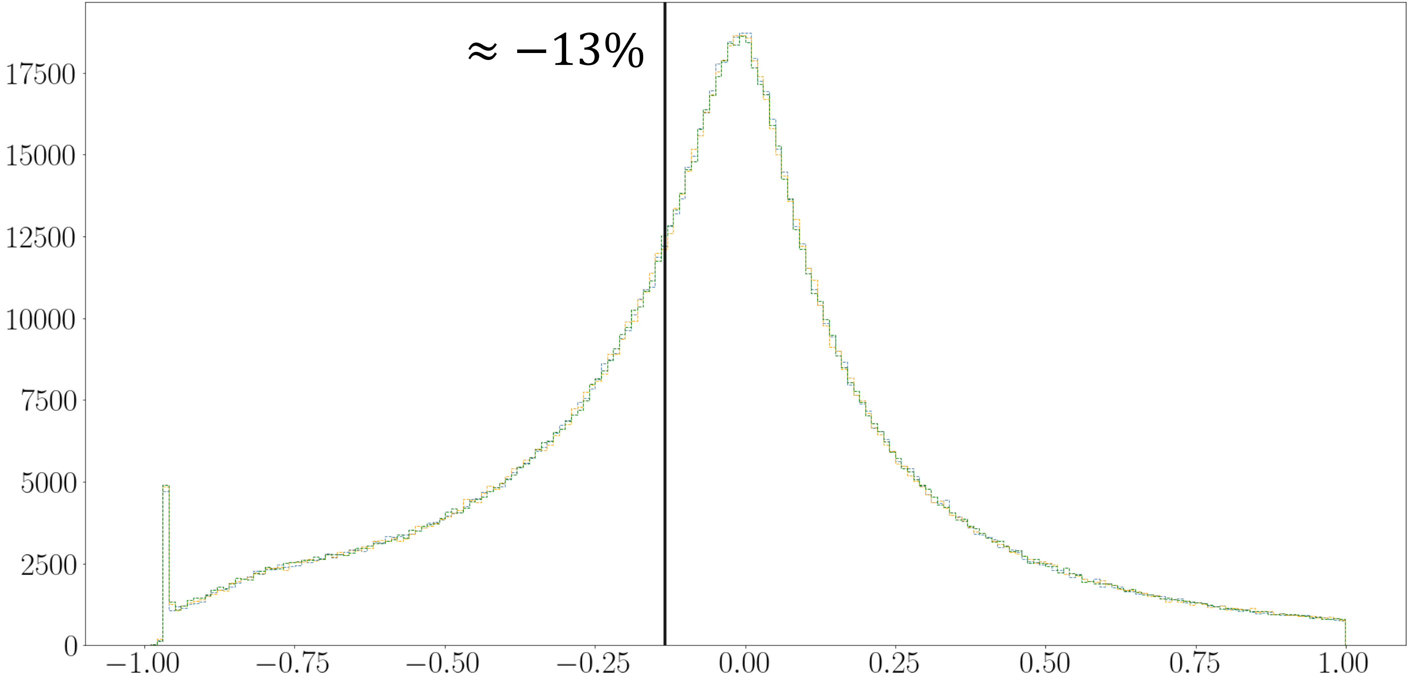
- **Evaluate trigger response on trace**
  - If trigger in signal region: count as success
  - If no trigger/not in signal region: count as fail
- $P(\mathrm{T2} \mid \text{Signal in tank}) = \dfrac{n_{\textbf{Success}}}{n_{\textbf{Sucess}} + n_{\textbf{Fail}}}$
- **Fold with** $P(\text{Signal in tank})$
- $P(\text{Signal in tank})$ **calculated the same way as Offline LTP, but from simulations where triggers are disabled**



Possible solutions

- **Random phase for filtering and downsampling:**
  - Downsampling algorithm chooses a random phase out of $[1, 2, 3]$
  - Trigger response only evaluated for a single phase
  - When evaluated over all phases, zenith dependency vanishes
  - Constant offset (as for Th trigger) remains for all energies/angles
  - Triggers are **more efficient** than Offline triggers using this correction
- **Bias from using component traces:**
  - Component traces are simulated separately from PMT trace
  - Component trace does not sum to total trace!
  - Component trace overestimates total trace

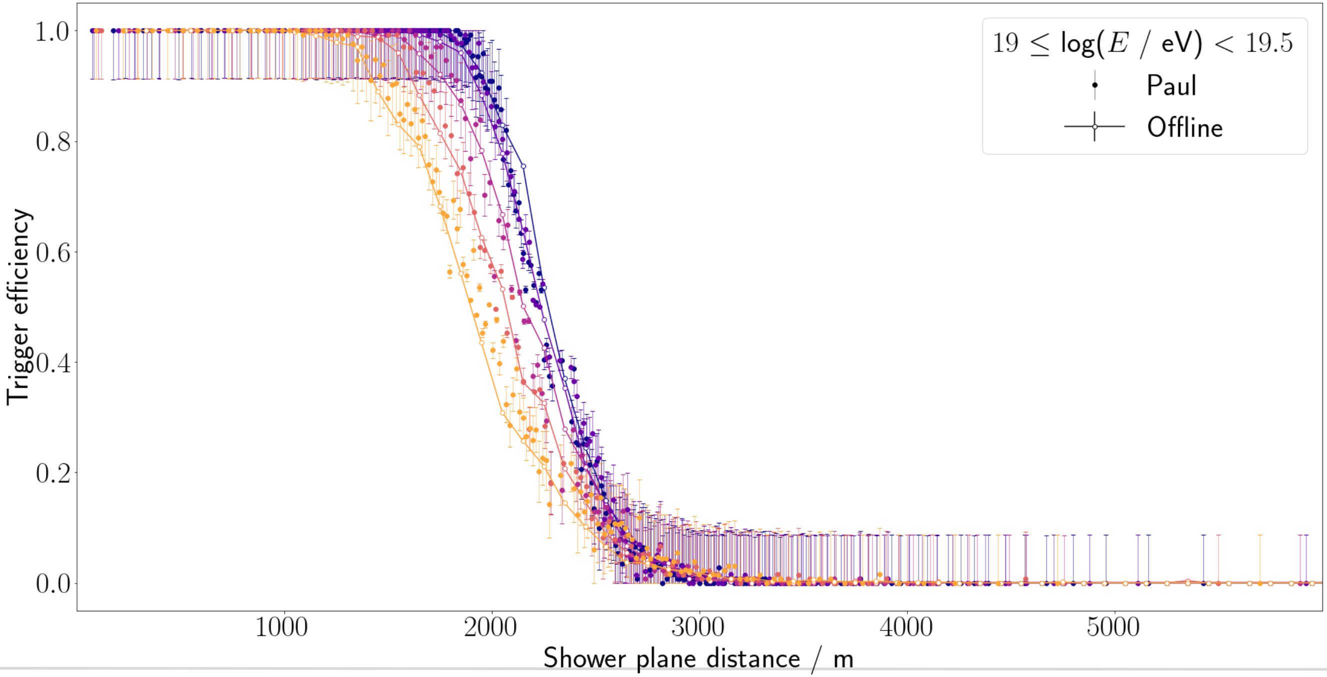
Possible solutions



$\text{recStation.GetVEMTrace()} \;/\; \Sigma\, \text{recStation.GetPMTTrace(component)} - 1$

Issue comparing Lateral Trigger Probability

- **My calculated LTPs (for my implementation) do match Offline?**


$19 \leq \log(E\ /\ \mathrm{eV}) < 19.5$
Paul
Offline
Trigger efficiency
Shower plane distance / m