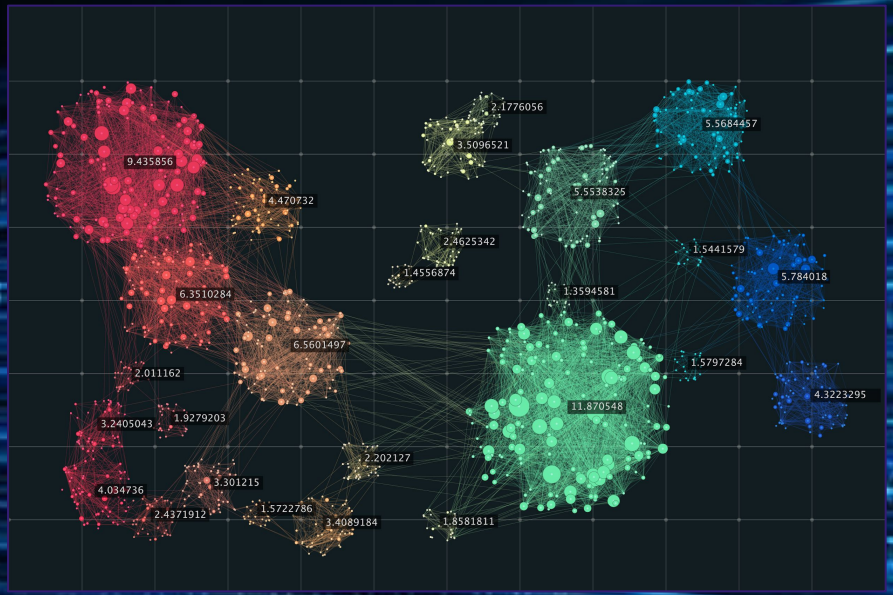
Unsupervised Clusterization



C. Fanelli

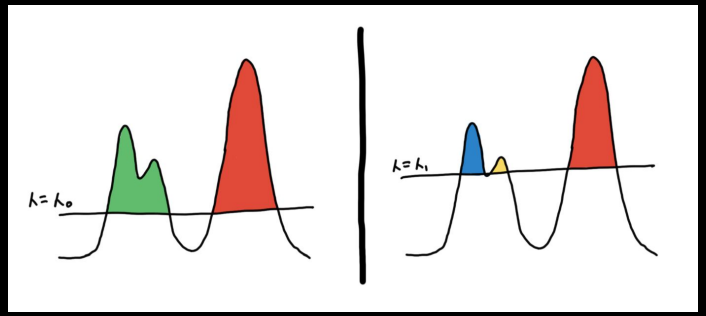
Jefferson Lab

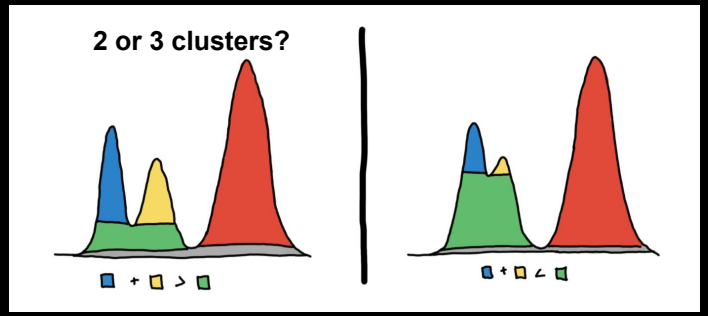
Motivation

- Clustering is used extensively in calorimetry to collect information at the hit level in order to reconstruct the energy deposited by particles.
- Typically the definition/formation of clusters depends on some cuts/selection criteria.
- During data taking detector response may change (e.g., need different calibrations) which can affect the quality of your online reconstruction.
- Ideally one would like to have a flexible algorithm that can be used under different experimental conditions (e.g., larger intensities, hit multiplicities etc.).
- Unsupervised clustering methods look at all the available information in the calorimeter at the hit-level, x, y, t, E, to cluster objects sharing common features; all you need to define is a metric in that space and tune some hyperparameters.
- Can handle arbitrarily shaped clusters, clusters with different sizes and densities, noise

Hierarchical Clustering

Two different clusterings based on two different level-sets



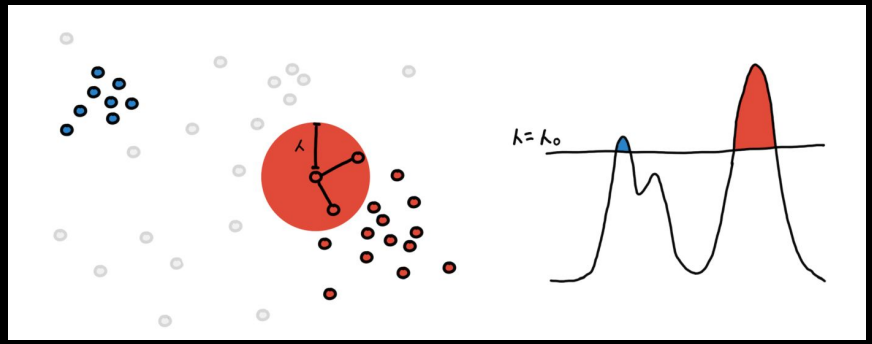


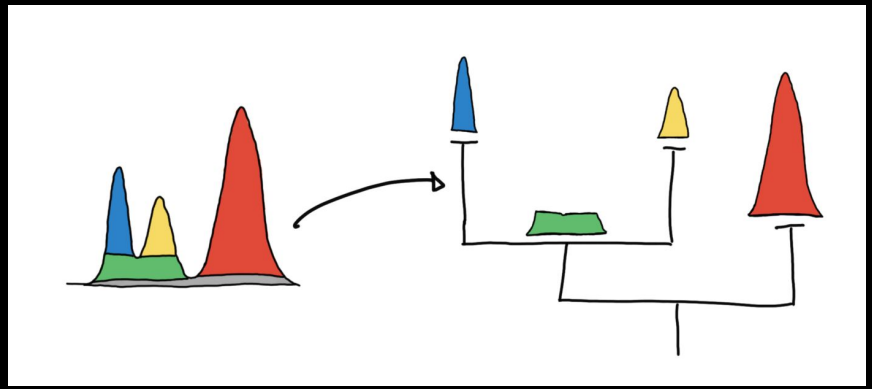
The area of the regions is the measure of “persistence”.

Maximize the persistence of the clusters under the constraint that they do not overlap.

Core distance (defined by a required # of neighbors) as estimate of density

Points have to be in a high density region and close to each other (“mutual reachability”)

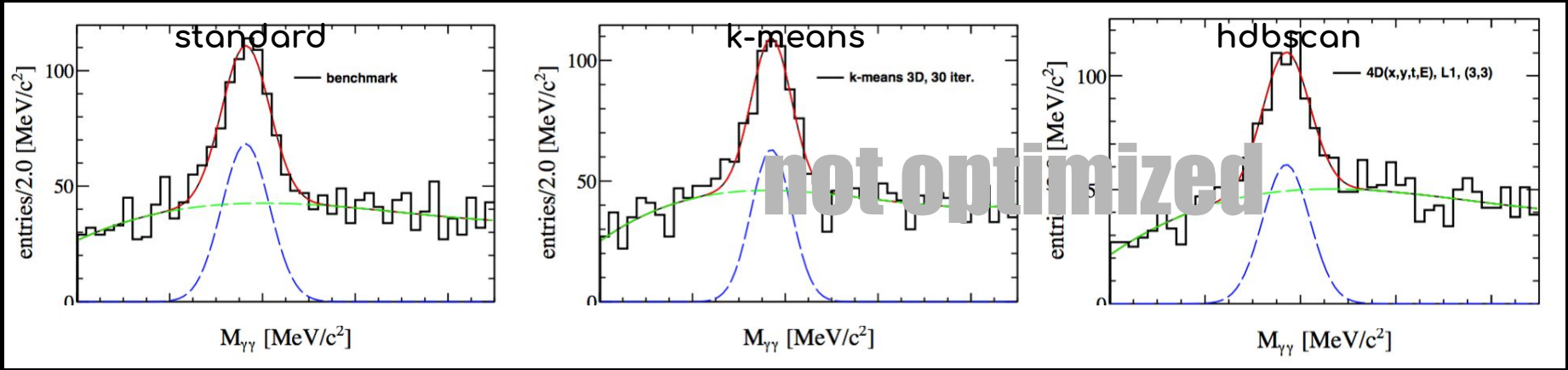




clusters are more likely regions separated by less likely regions -> densities

Offline tests on triggered data

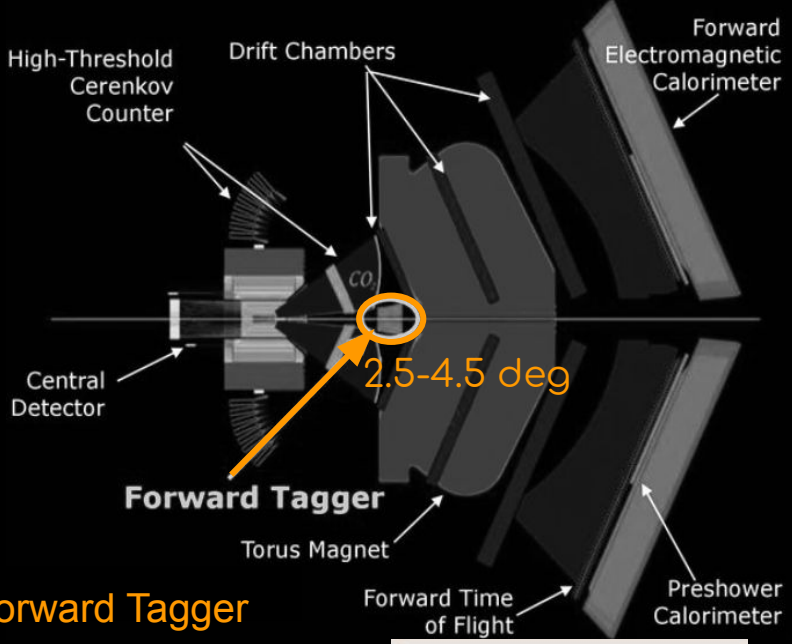
- Implementation of AI-algorithms as plugins in the JANA2 reconstruction framework tested on real data (reconstruction of π^0 peak): results below are not optimized.
- Main ingredients: define a metric (N-dim) for the distance and the hyperparameters (two main (only) for hdbscan).
  - For unsupervised clustering need only few hyperparameters and no other cuts.
  - How do you optimize? Look at ‘candles’ in the calorimeter.



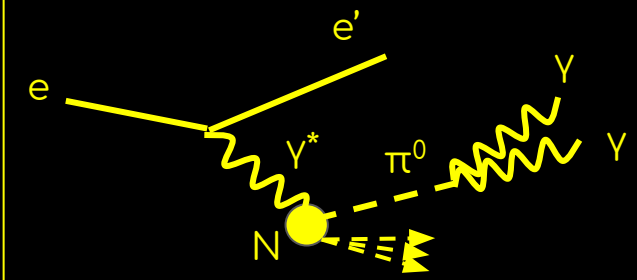
Streaming Readout

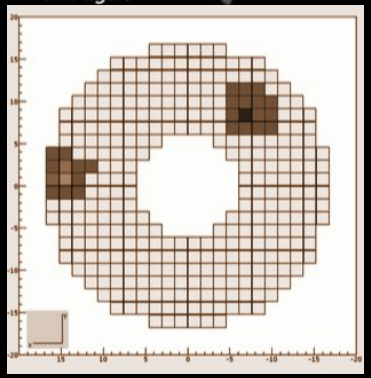
data read continuously from all channels

- CLAS12 SRO setup
- TriDAS SR back end
- JANA2 reconstruction framework
- Analysis of π^0 electro-production



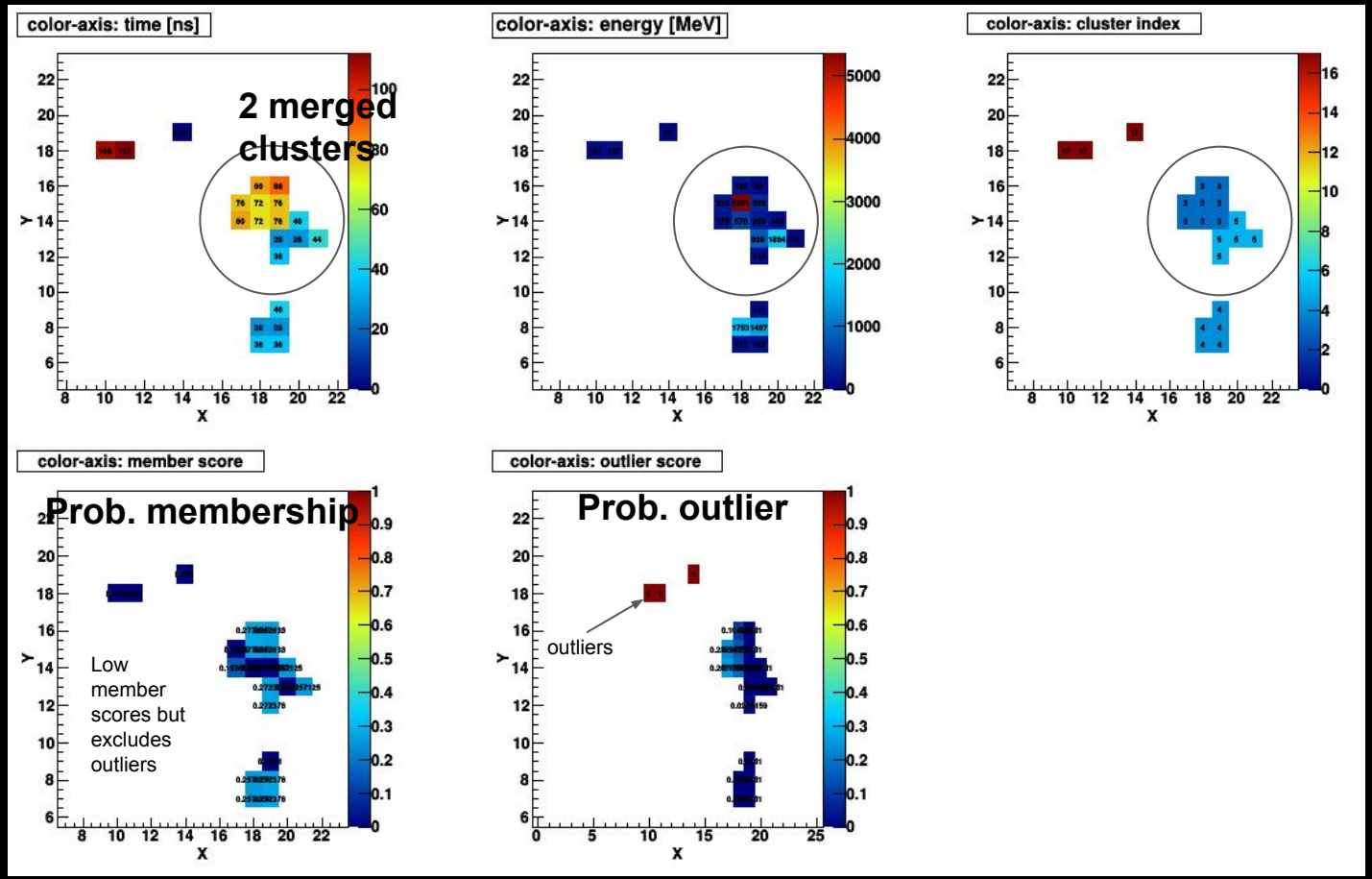
Forward Tagger





Probabilities

No need to identify seeds of clusters and associate other hits to them.



SRO Feb 2020
Evt 8575

4D metric

Cluster quality cuts in invariant mass histogram:

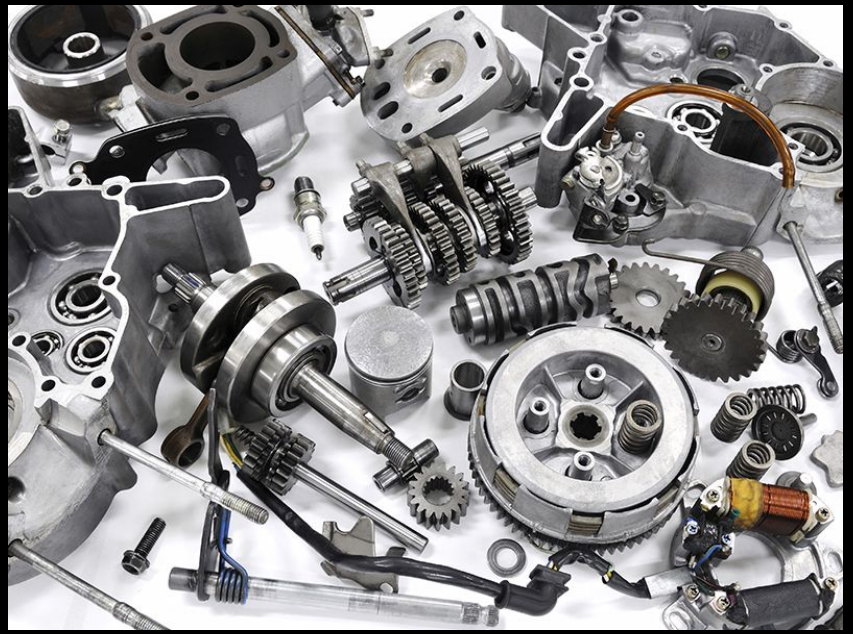
- >= 2 hits
- >= 3 GeV

$\Delta t(clu1,clu2)<50$ ns

Perspectives

- Everything shown can work for online reconstruction in SRO.
- For data exploration we want a clustering algorithm with as few assumptions as possible.
- Code development, test on real data and comparison ongoing.
- Benchmark performance against traditional algorithm underway:
  - Time performance;
  - Merged clusters;
  - Behavior with larger number of clusters in a time slice.

SPARES

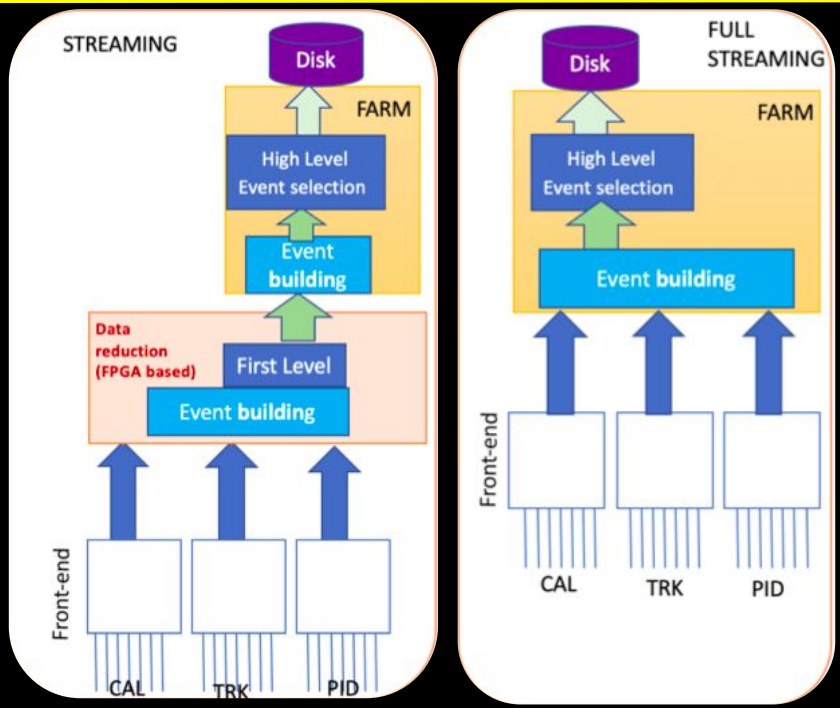


Streaming Readout

G. Heyes, JLab12 streaming readout and future EIC
Y. Furletova, ML for particle identification

data read continuously from all channels

- Validation checks at source reject noise and suppress empty channels.
- Data then flows unimpeded in parallel channels to storage or a local compute resource.
- Data organized in multi-dimensions by channel and time.



Different streaming pipelines:
FPGA based (w/ data reduction) and full streaming.
ML naturally suited for online data reduction or high level physics event selection/trigger

Semi-supervised Clustering: e.g., K-means

STEP 1: Choose the number K of clusters

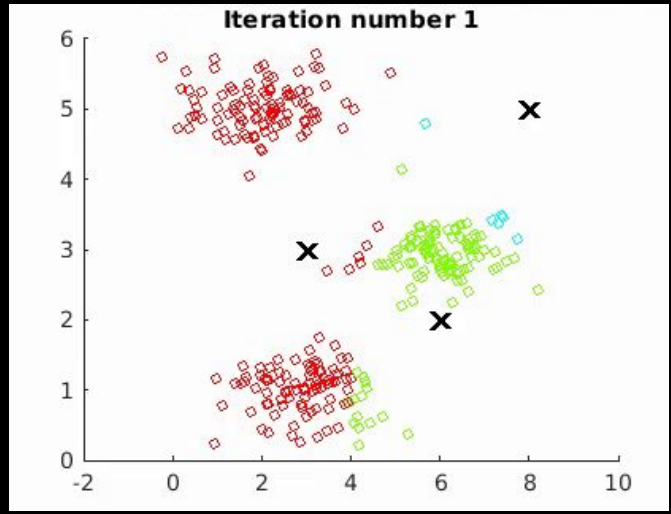
STEP 2: Select at random K points, the centroids
(not necessarily from your dataset)

STEP 3: Assign each data point to the closest centroid
(That forms K clusters)

STEP 4: Compute and place the new centroid of each cluster

STEP 5: Reassign each data point to the new closest centroid
If any reassignment took place, go to STEP 4,
otherwise go to FIN.

Your Model is Ready



Hyperparameters and metrics

Table 2. The different metrics used for k-means.

metric	description
$(X_{hit} - X_{mean})^2 + (Y_{hit} - Y_{mean})^2$	squared 2D space distance
$\frac{(X_{hit}-X_{mean})^2}{L_{cell}^2} + \frac{(Y_{hit}-Y_{mean})^2}{L_{cell}^2} + \frac{(t_{hit}-t_{mean})^2}{(50\ ns)^2}$	squared 3D space-time distance
$\frac{(X_{hit}-X_{mean})^2}{L_{cell}^2} + \frac{(Y_{hit}-Y_{mean})^2}{L_{cell}^2} + \frac{(t_{hit}-t_{mean})^2}{(50\ ns)^2} + (E_{hit} - E_{mean})^2$	squared 4D space-time-energy distance

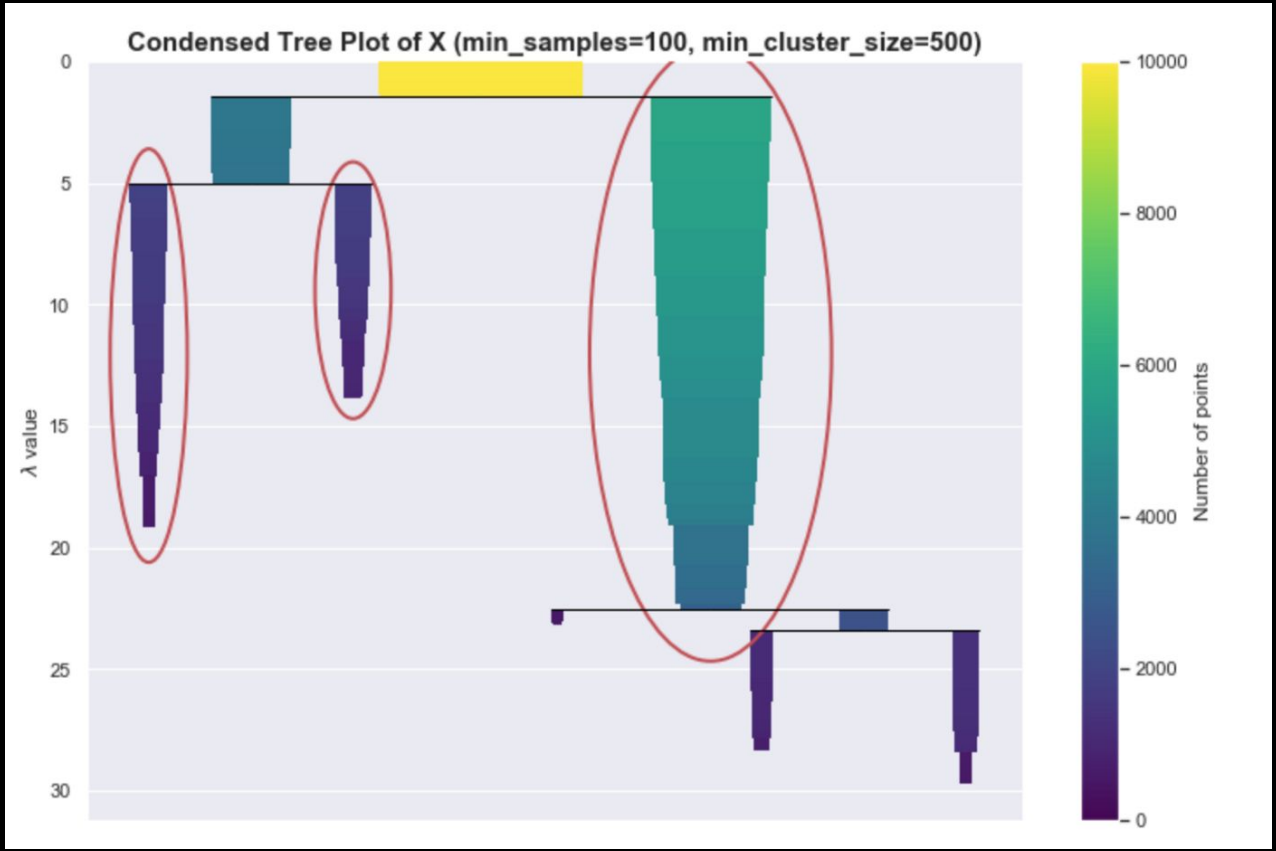
Table 3. The main parameters of the k-means algorithm are described and their values reported. For each parameter, the last column shows when it intervenes, either if in the pre-processing or in the clustering phase.

parameter	description	value [units]	phase
t threshold	minimum time of hits	0. ns	preprocessing
E threshold	minimum energy of hits	0. GeV	preprocessing
time_window	time difference between hits	50 ns	preprocessing
count_cells	active neighbor cells for each hit	≥ 1	preprocessing
iterations	k-means updates	10 (30)	clustering
bad_distance	max distance hit-cluster	not used	clustering
bad_time	max time difference hit-cluster	not used	clustering
norm_space	normalization space distance hit-cluster	L_cell (cell length, see Tab. 2)	clustering
norm_time	normalization time difference hit-cluster	50 ns (see Tab. 2)	clustering
norm_ene	normalization energy difference hit-cluster	not used	clustering

$$bool = \Delta t < 50\ ns\ \&\&\ \Delta X \leq 1\ \&\&\ \Delta Y \leq 1\ \&\&\ (\Delta X + \Delta Y) > 0 \qquad (3.1)$$

For K-means we need to make some assumptions, in particular we need to provide the seeds.

Hierarchical clustering



$O(n^2)$

A hierarchy of multiple level-sets is obtained by varying the density threshold

Visualization of the tree top-down as in the literature

hdbscan vs K-means

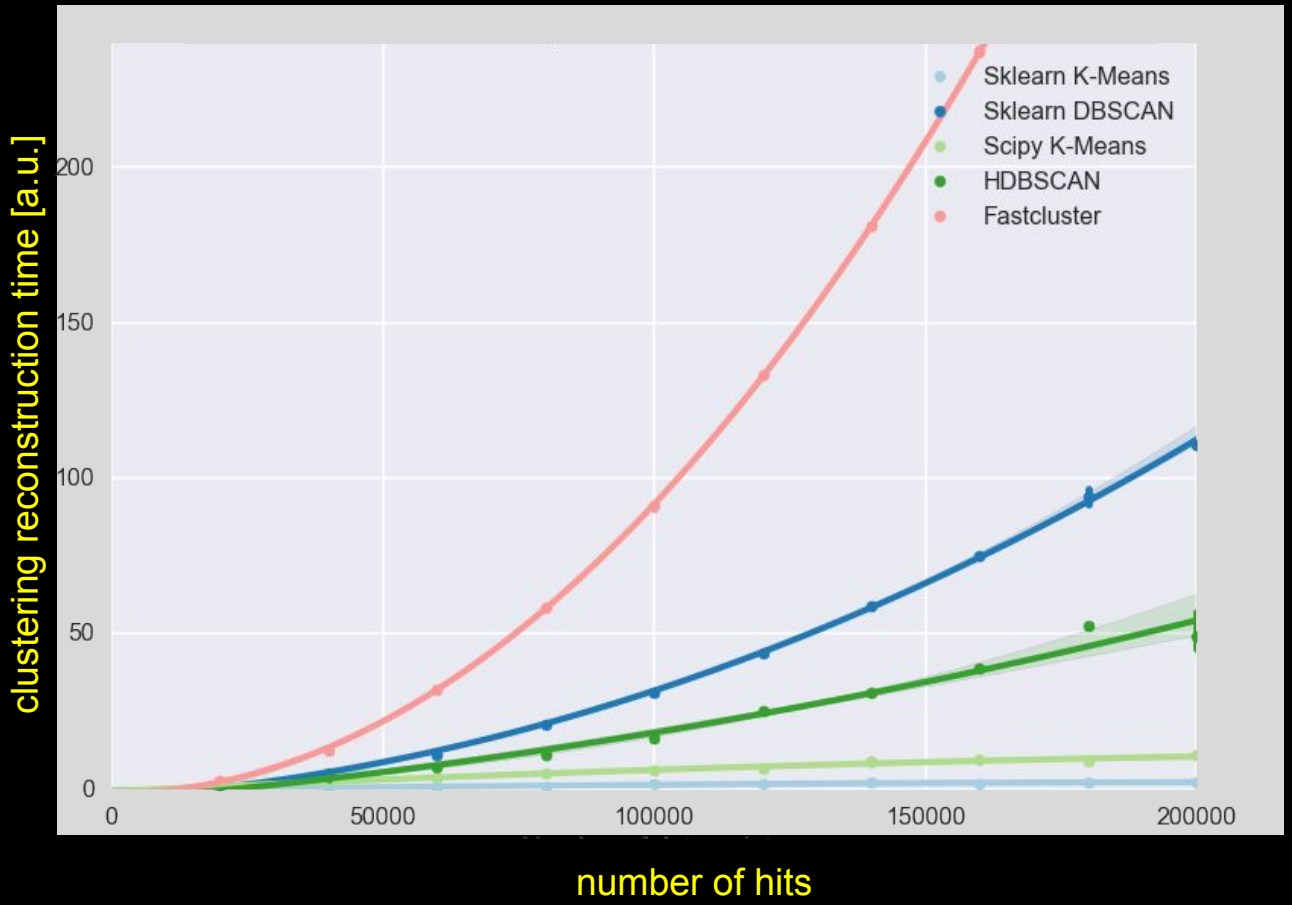
K-means is a semi-supervised parametric algorithm parameterized by the *K cluster centroids* (aka K seeds). Can perform if the underlying assumptions on the shape of the clusters are not met. Clusters have to be:

- “round” or “spherical”
- equally sized, dense
- typically most dense in the center
- not contaminated by noise and outliers

Hdbscan on the other hand is an unsupervised hierarchical clustering which excels when data has:

- arbitrarily shaped clusters
- clusters with different sizes and densities
- noise

Relative Performance k-means and hdbscan

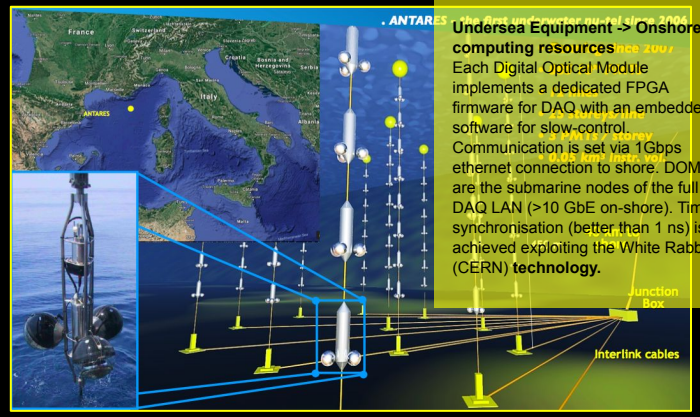


TriDAS

- TRIggerless Data acquiSition

- Developed to acquire and filter the data stream from the KM3NeT (underwater neutrino telescope) detector

- Framework ported to JLab physics to develop the future EIC Streaming Readout: does streaming work with our luminosities/rates?



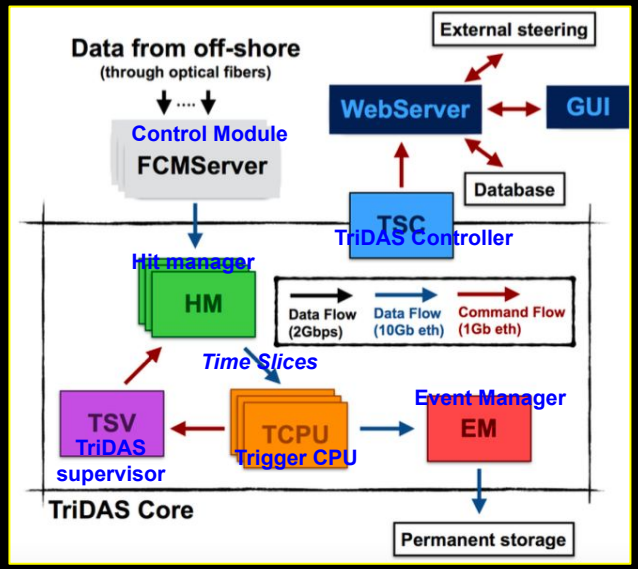
Undersea Equipment -> Onshore computing resources
Each Digital Optical Module implements a dedicated FPGA firmware for DAQ with an embedded software for slow-control. Communication is set via 1Gbps ethernet connection to shore. DOMs are the submarine nodes of the full DAQ LAN (>10 GbE on-shore). Time synchronisation (better than 1 ns) is achieved exploiting the White Rabbit (CERN) **technology.**

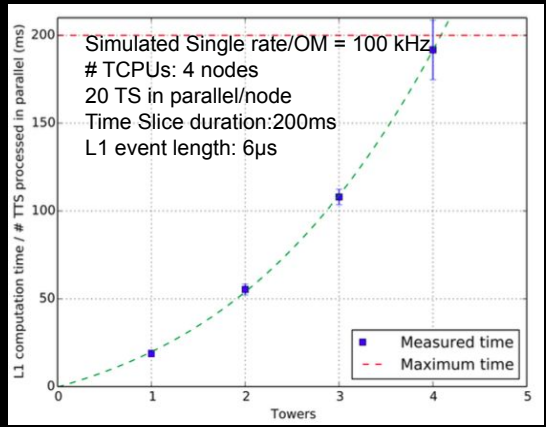
Each Hit Manager (HM):

receives data from a specific portion of detector called "Sector"

slices the data stream into "Time Slices" (TS) of fixed time duration

creates the so called "SectorTimeSlices" (STSs)





"add TCPU as much as it suffices without affecting the DAQ design"

Streaming Readout

- In view of the Electron Ion Collider physics, we want to demonstrate at JLab that the streaming of readout paradigm works.
- The general idea is to move the trigger decision from frontend to, e.g., CPU (GPU, FPGA etc. to validate)
- One can begin with a simple set of channels. The Event Builder in the Streaming paradigm is central.
- TriDAS SR provides:
  - Flexibility (add channels to TriDAS), Scalability (add CPUs if needed), Reproducibility (event selection logic pipeline is transparent), High Level Applications (e.g. online calibration becomes possible)
- Networking needs deterministic **switch with fixed latency** e.g. white rabbit switch for synchronization
- TriDAS so far worked with low rate events with large noise. The Jefferson Lab high-luminosity is challenging.
- At Jefferson Lab we are testing this system for the FT in CLAS12 in Hall B. We plan to test this for the calorimeter in Hall D too in the next months.