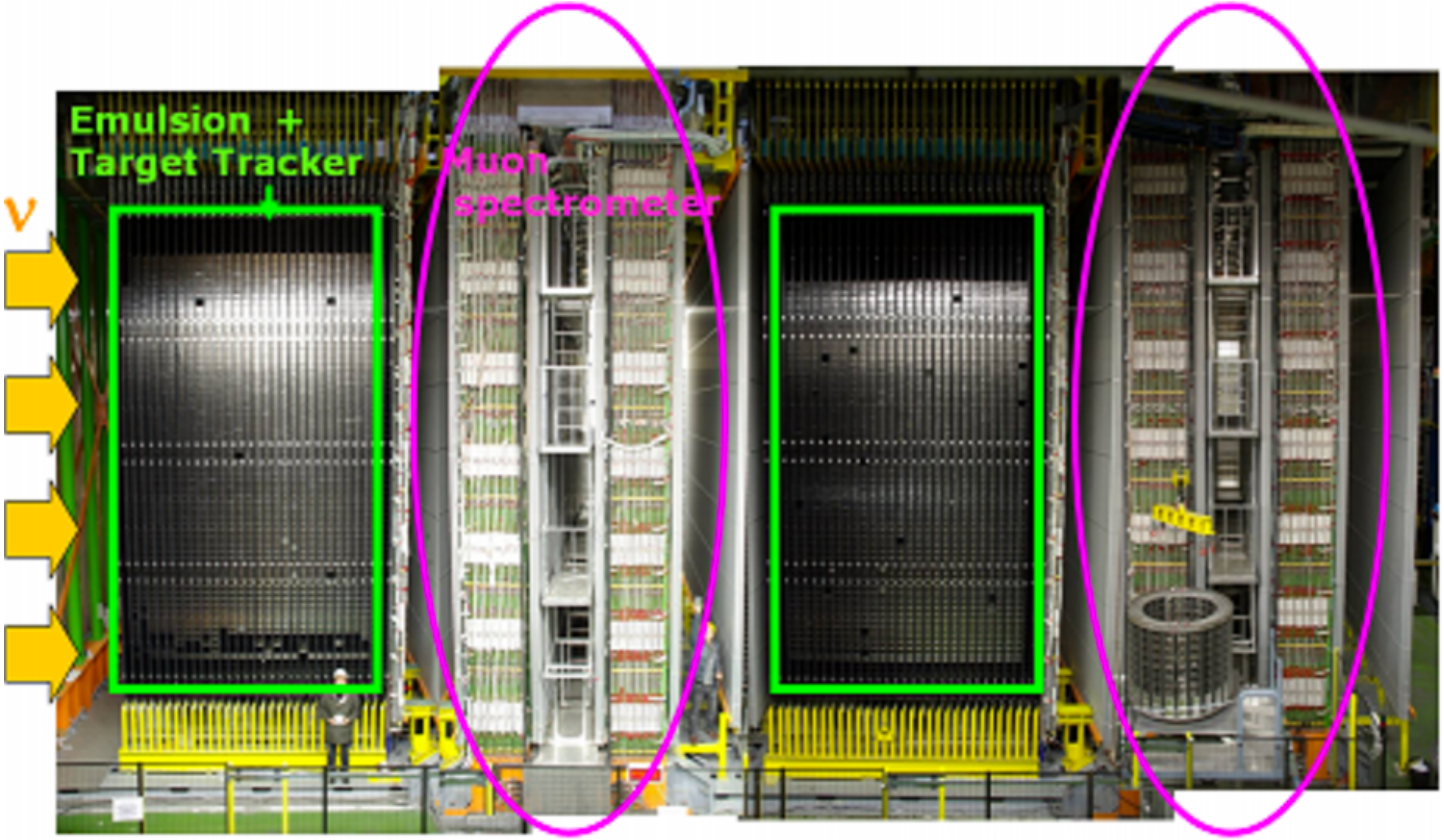
Yandex SCHOOL OF DATA ANALYSIS

Overview of electro-magnetic showers identification in OPERA. SHiP perspective.

A. Rogozhnikov, V. Belavin, A. Filatov, S. Shirobokov, Andrey Ustyuzhanin
Yandex School of Data Analysis
NRU Higher School of Economics
INFN, Napoli

Opera detector



OPERA ECC brick (similar to SHiP)

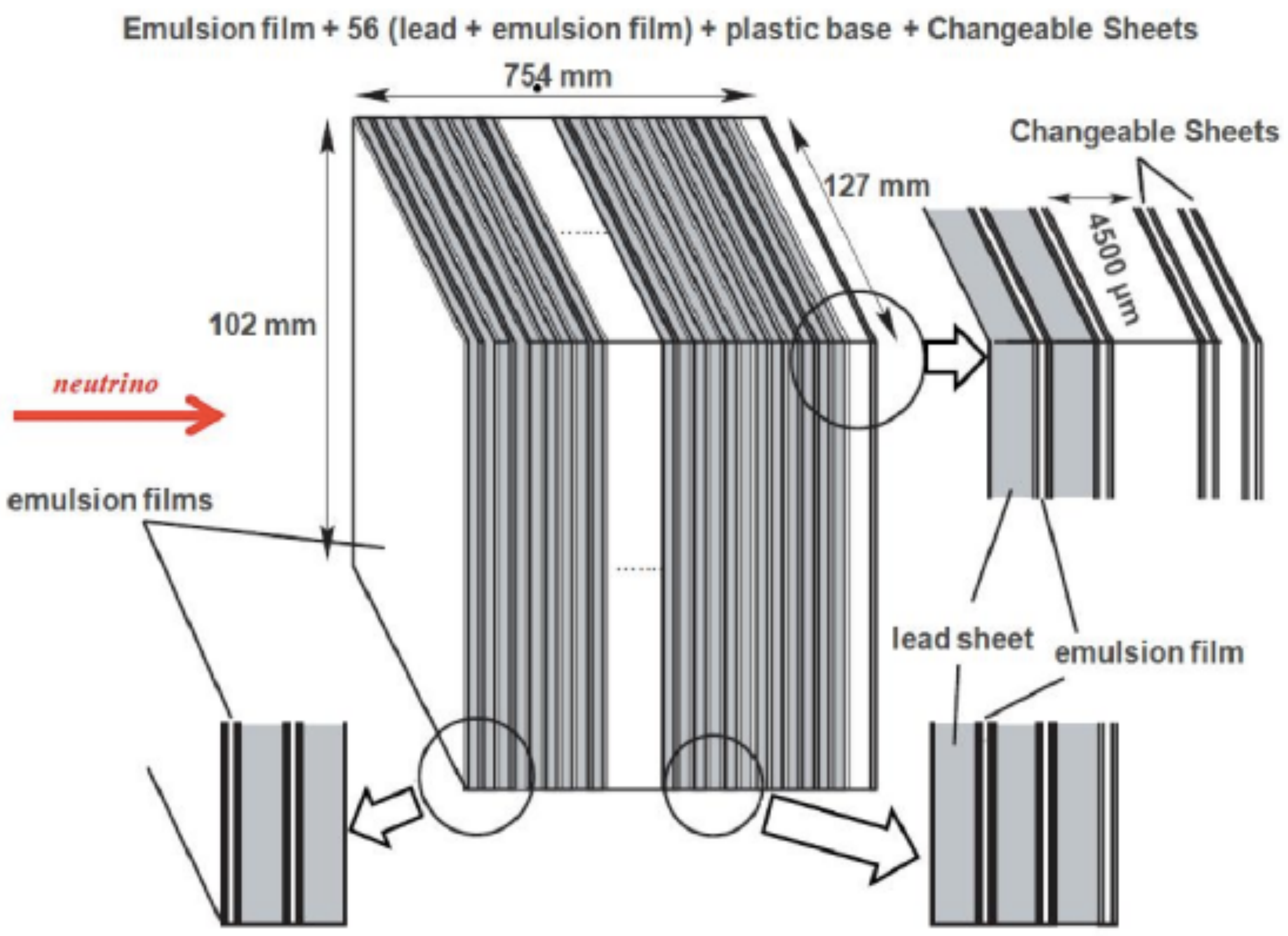
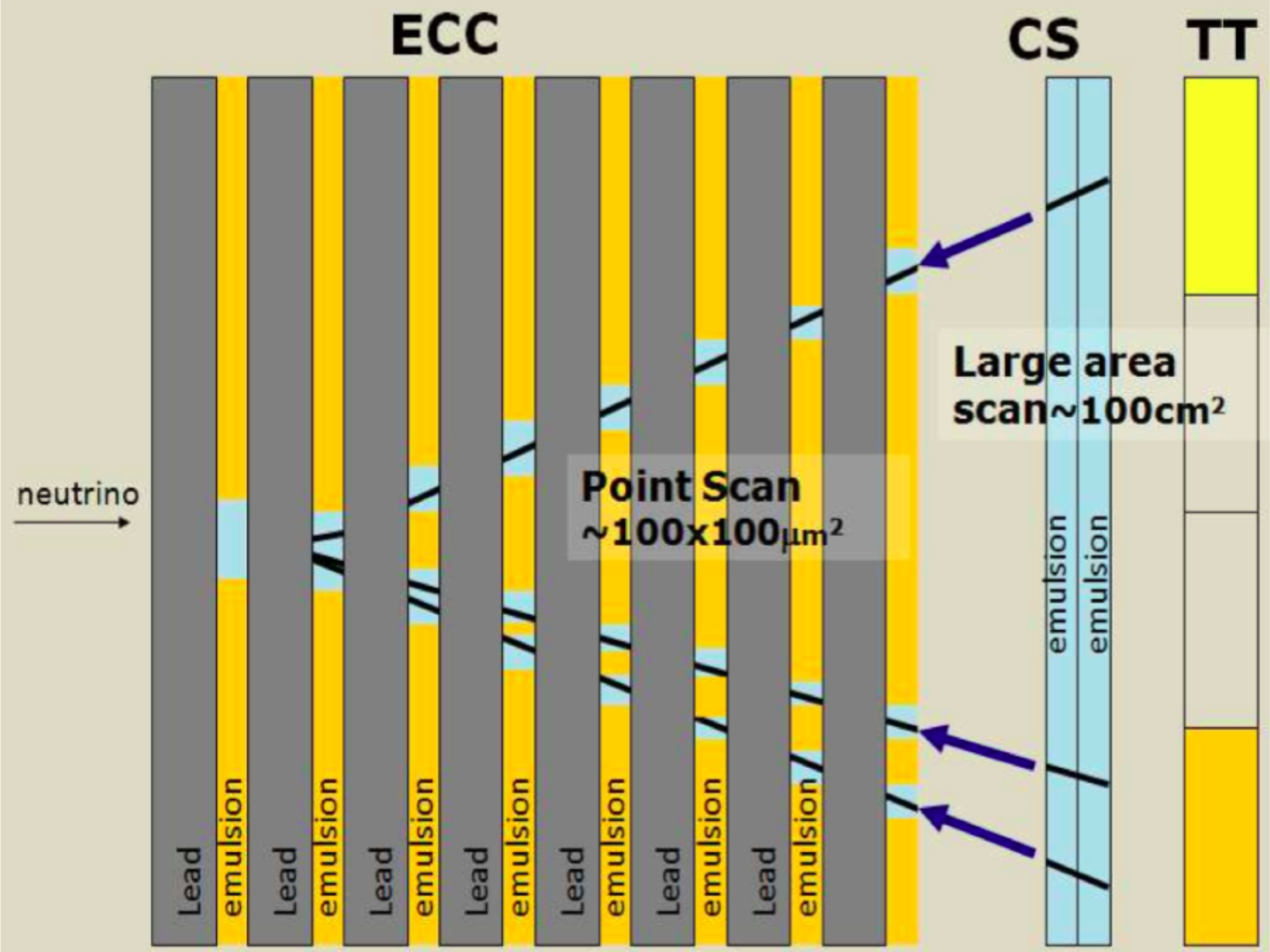


Figure 2.4 – Schematic structure of an ECC brick.



Given

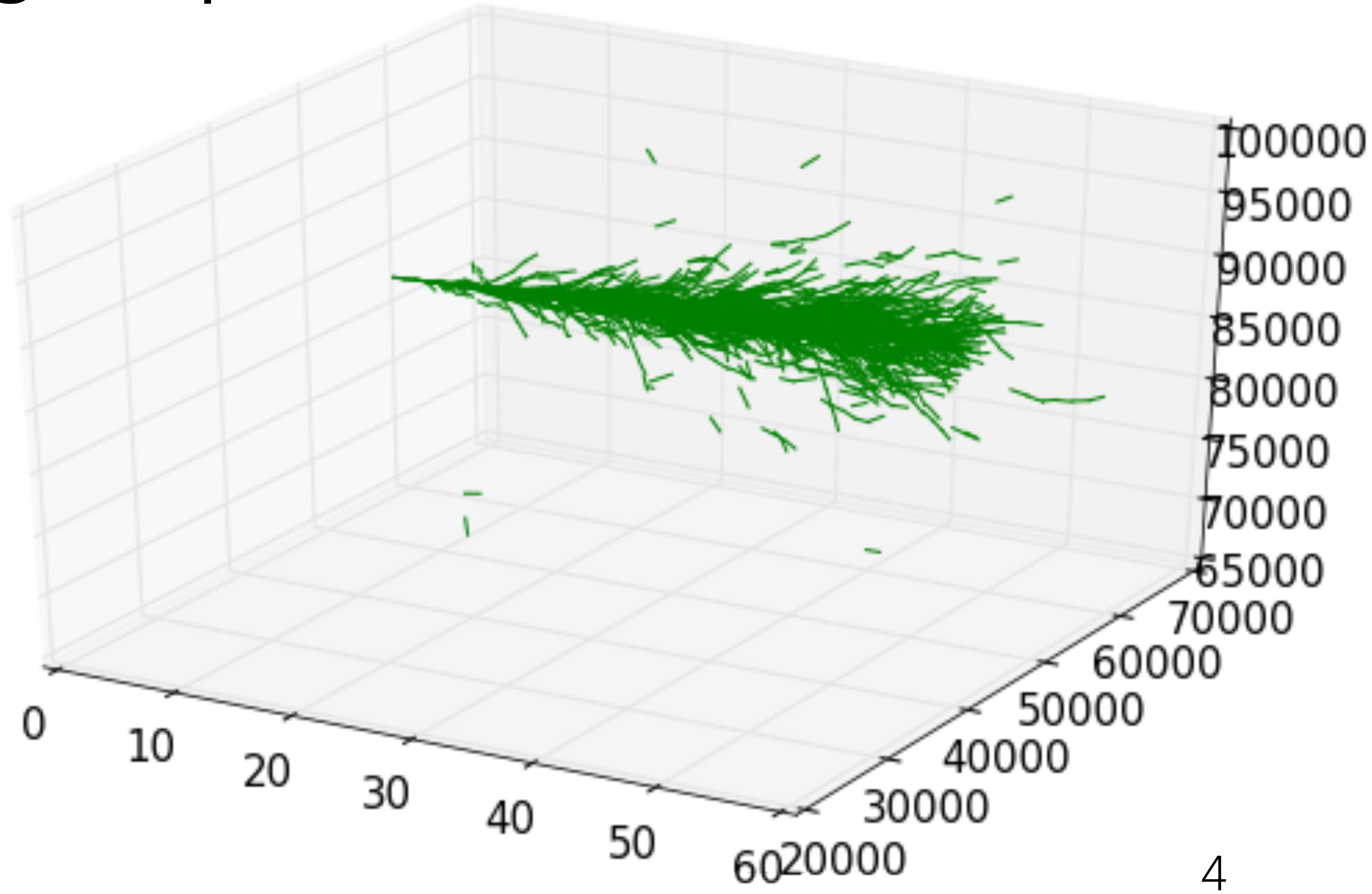
Data Background: 8 layers from one brick (48-56) ~ 4 * 10^6 base tracks

- **background is generated by subsampling given part of the brick**

MC Signal: simulation of pure EM showers (~6000 events)

- **~ 3 * 10^6 base tracks in total**

Algorithm for reconstruction of shower given its origin and direction



Research goal

Develop algorithm that can

- detect e-m shower basetracks within a brick basetracks
- identify shower origin
- estimate shower energy

… with only assumption that there is no more than 1 shower in the brick

- A) start from 10 the most downstream plates
- B) using whole brick volume (similar to SHiP)

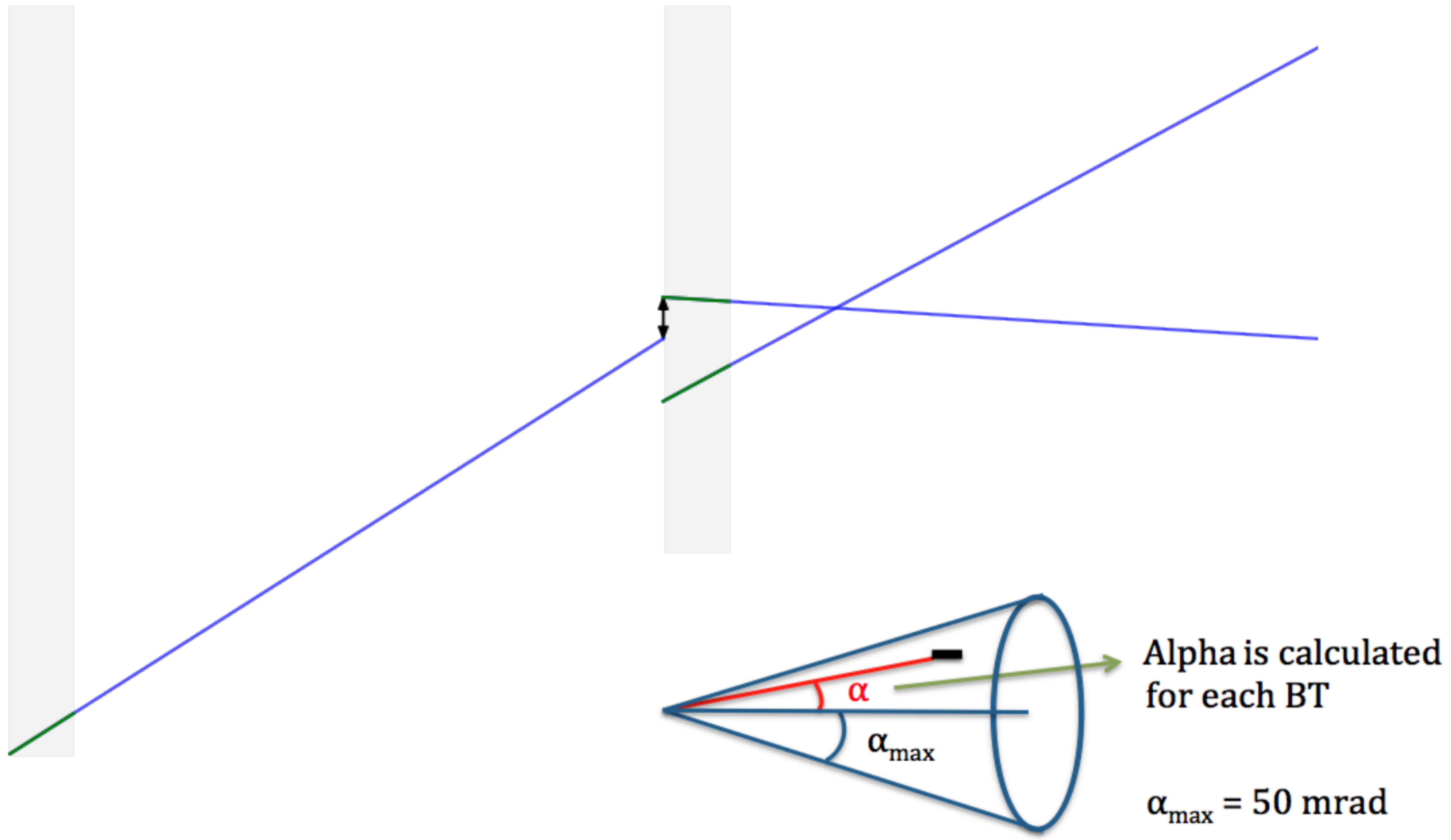
Generic approach

1. Mix signal and background to create event configuration
2. Describe every basetrack (BT) by set of features
3. Train classifier X to discriminate signal BT from background BT
4. Apply a cut on the classifier output
5. Topology filter Y to identify shower among selected BTs
6. Estimate quality/energy resolution etc.

Basetrack features examples

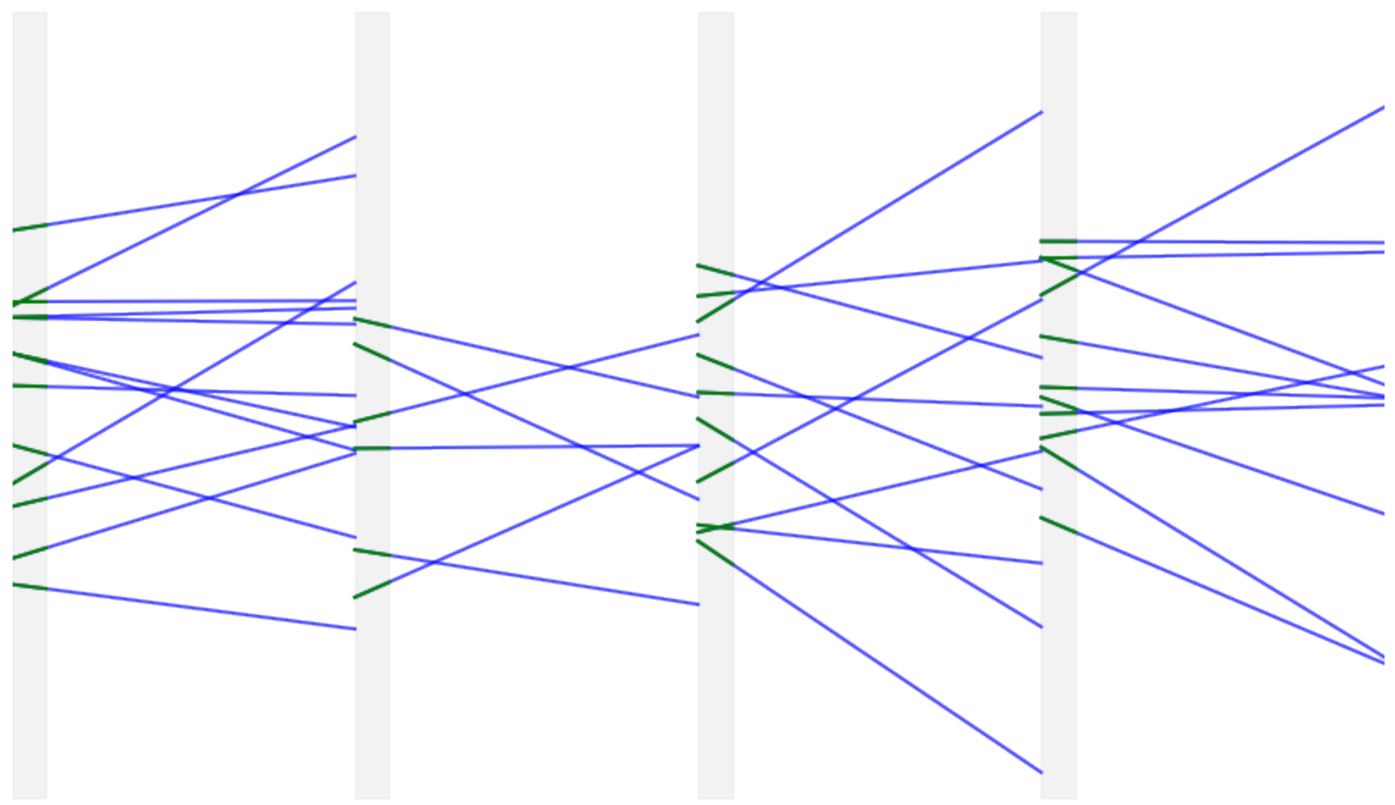
For every basetrack select a cone 50mrad, and compute

- \Delta - distance between tracks
- \Alpha (see figure on the right)
- \Theta (angle between basetracks)
- SX, SY (slope difference)
- IP– Impact Parameter
- \Chi2
- …
- Use distance/angles computed from/to plate-after-the-next



Alpha is calculated for each BT

α_{max} = 50 mrad



Algorithm 1

- Use 10 the most downstream plates

- Features: for every basetrack select square 3mm by 3mm on the next plate and for every base track of **the same kind** compute:

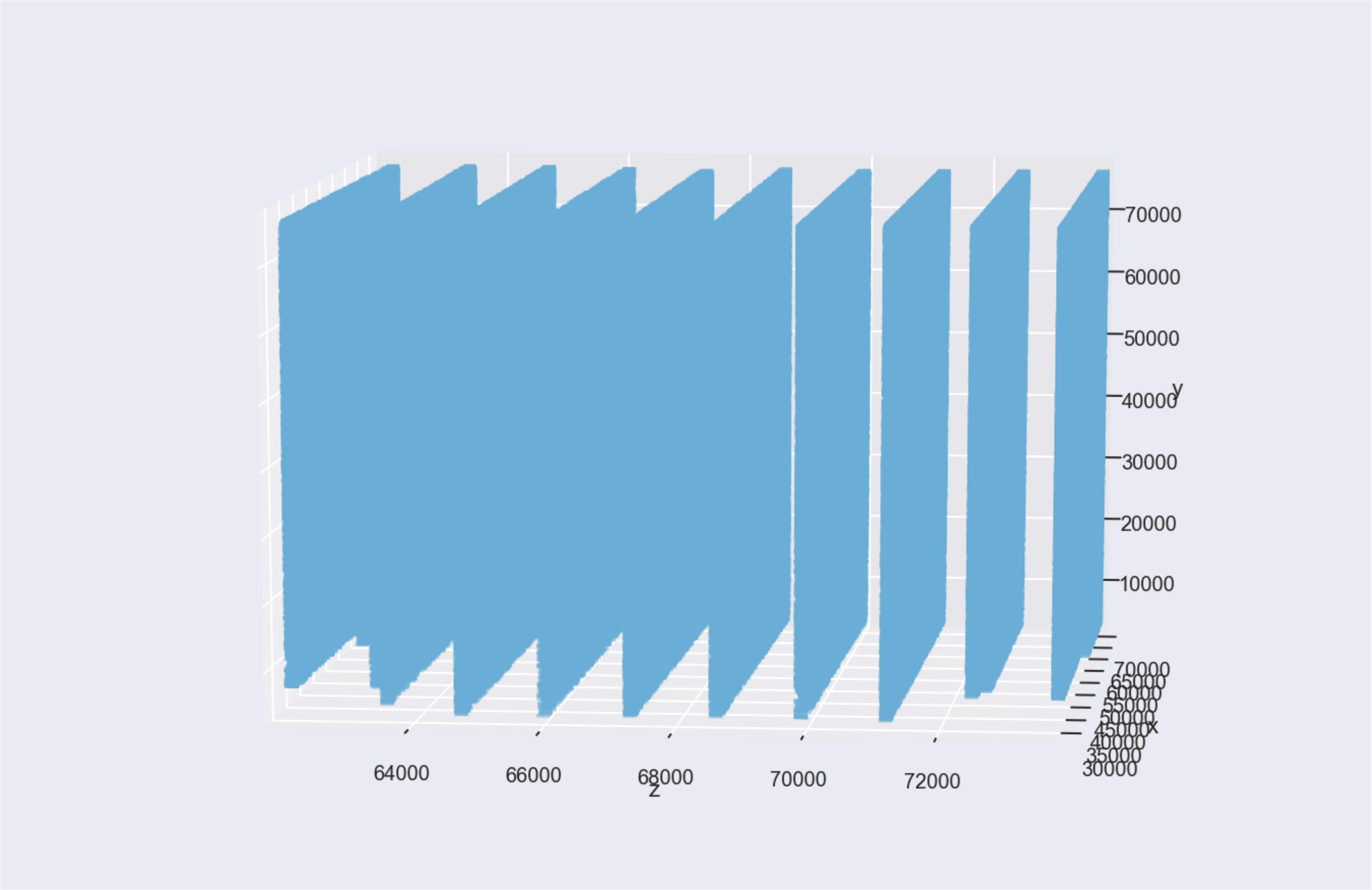
- **IP to both directions,**
- **Euclidean distance between tracks**
- **Tangent of angle between tracks**
- **Angles with respect to z direction**

- Pre-selection algorithm (X) - SVM (Support Vector Machine)

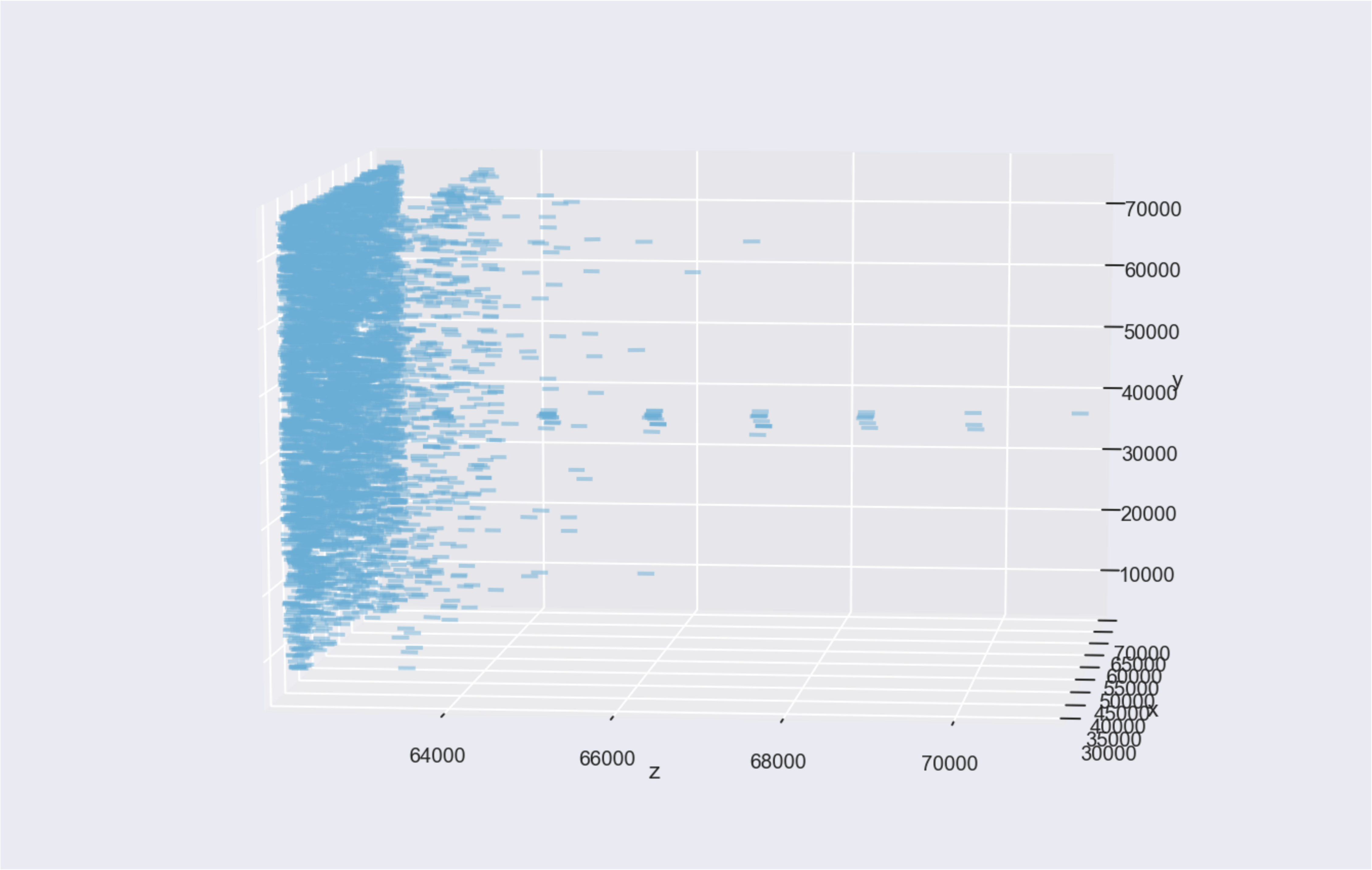
- Topology filtering algorithm (Y): Conditional Random Field classifier

SVM basetrack preprocessing

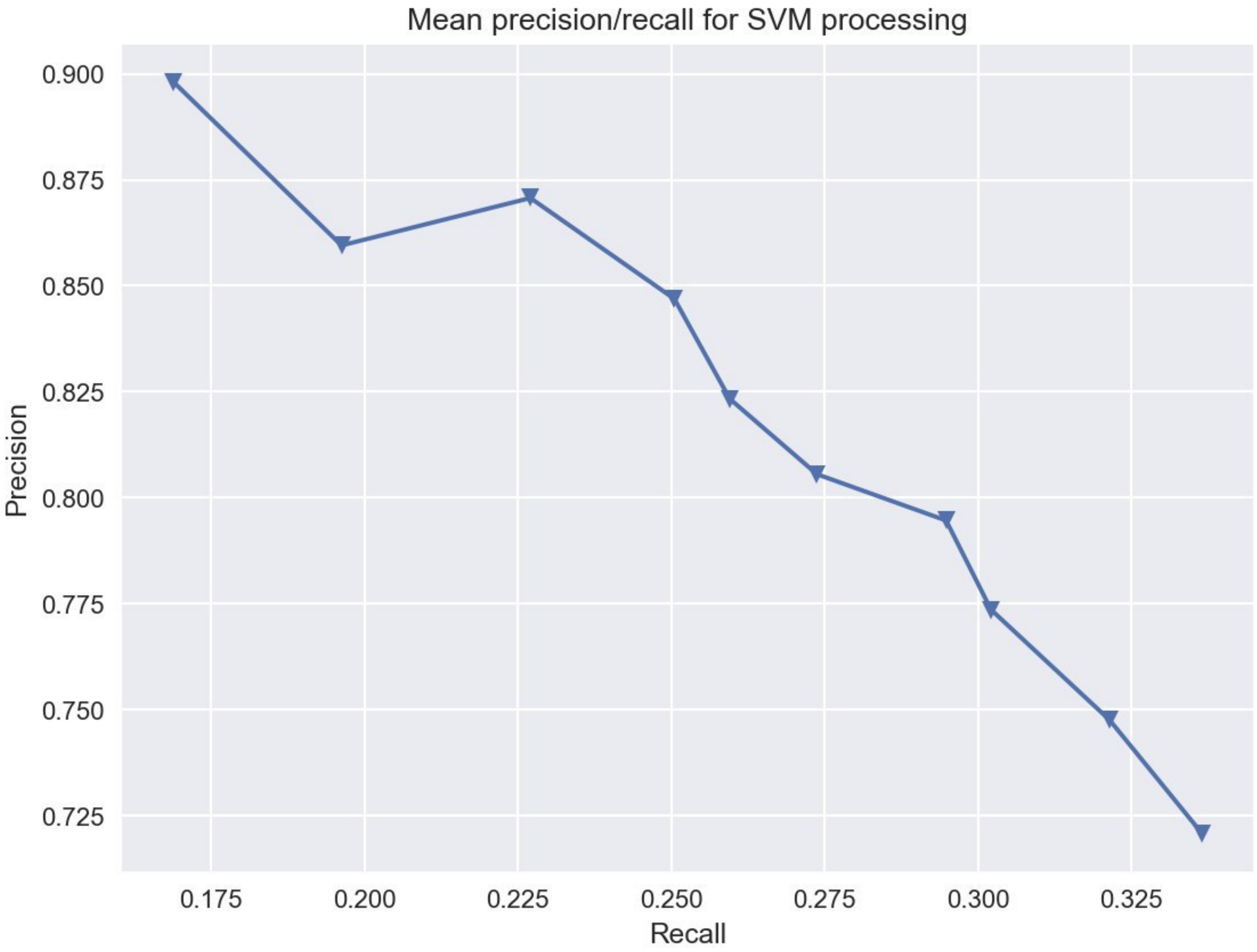
Background: ~5 *10^5
Signal: 308 base-tracks



Background: ~5 * 10^3
Signal: 77 base-tracks



Precision / Recall Curve for SVM step



Recall - signal efficiency
Precision - signal purity

signal purity is calculated in the 50mrad cone from the shower origin.

Andrey Ustyuzhanin

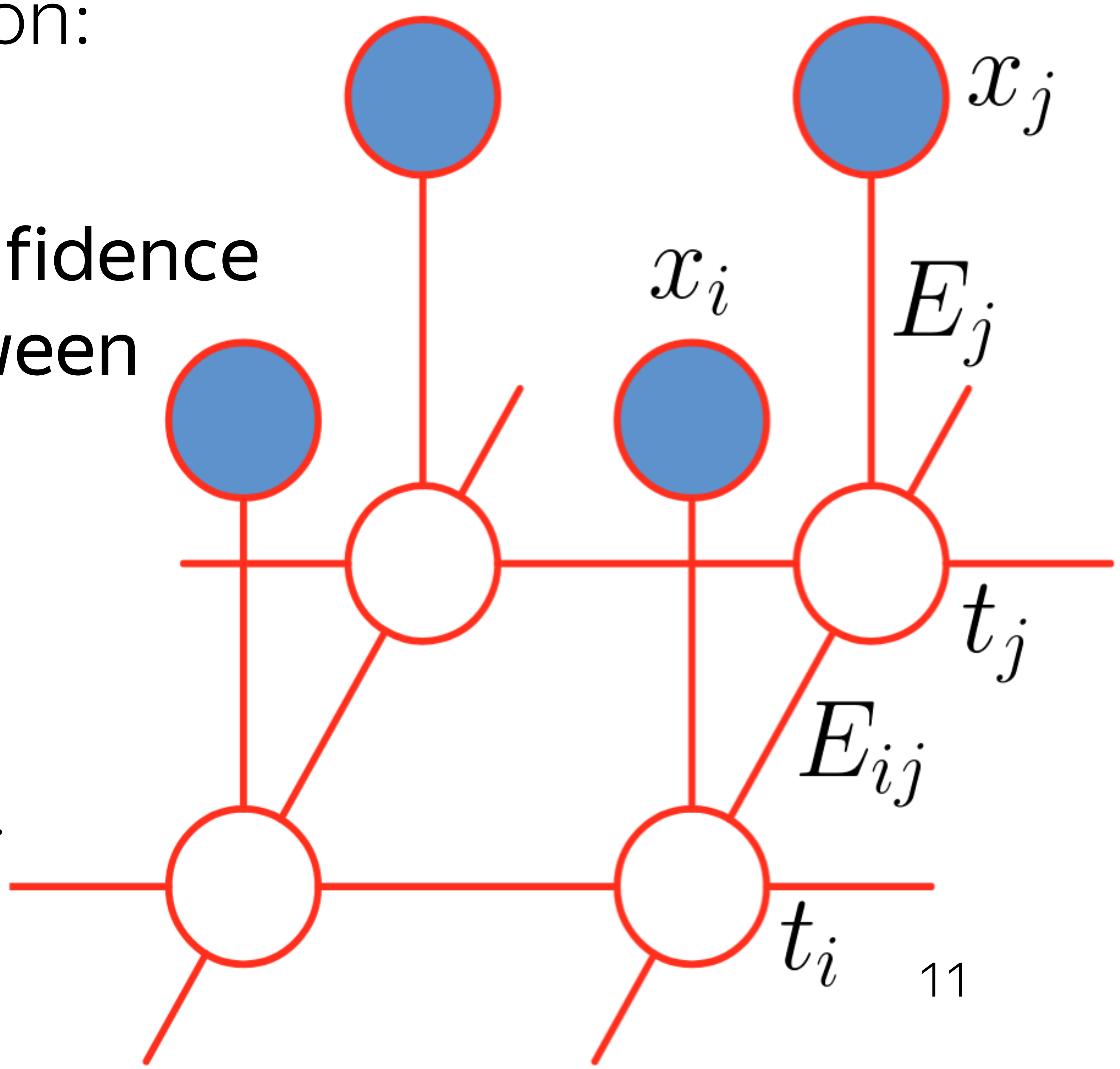
Topology filter: Conditional Random Field

Conditional Random Fields are a probabilistic framework for labeling and segmenting structured data, such as sequences or trees. Underlying idea of CRF is energy minimization:

- x_i –graph node, represented by some variables
- $E_i = E(x_i, t_i)$ – unary potential, prior knowledge/confidence
- $E_{ij}(t_i, t_j)$ – pairwise potential, shows similarity between nodes x_i, x_j
- *Goal: find* t_i *so that* $E = \sum E_{ij}(t_i, t_j) + \sum E_i \rightarrow min$

Inference done with min-cut/max-flow algorithm

- Algorithm output: distribution of $t_i \in \{0, 1\}$ over x_i



Conditional Random Field, details

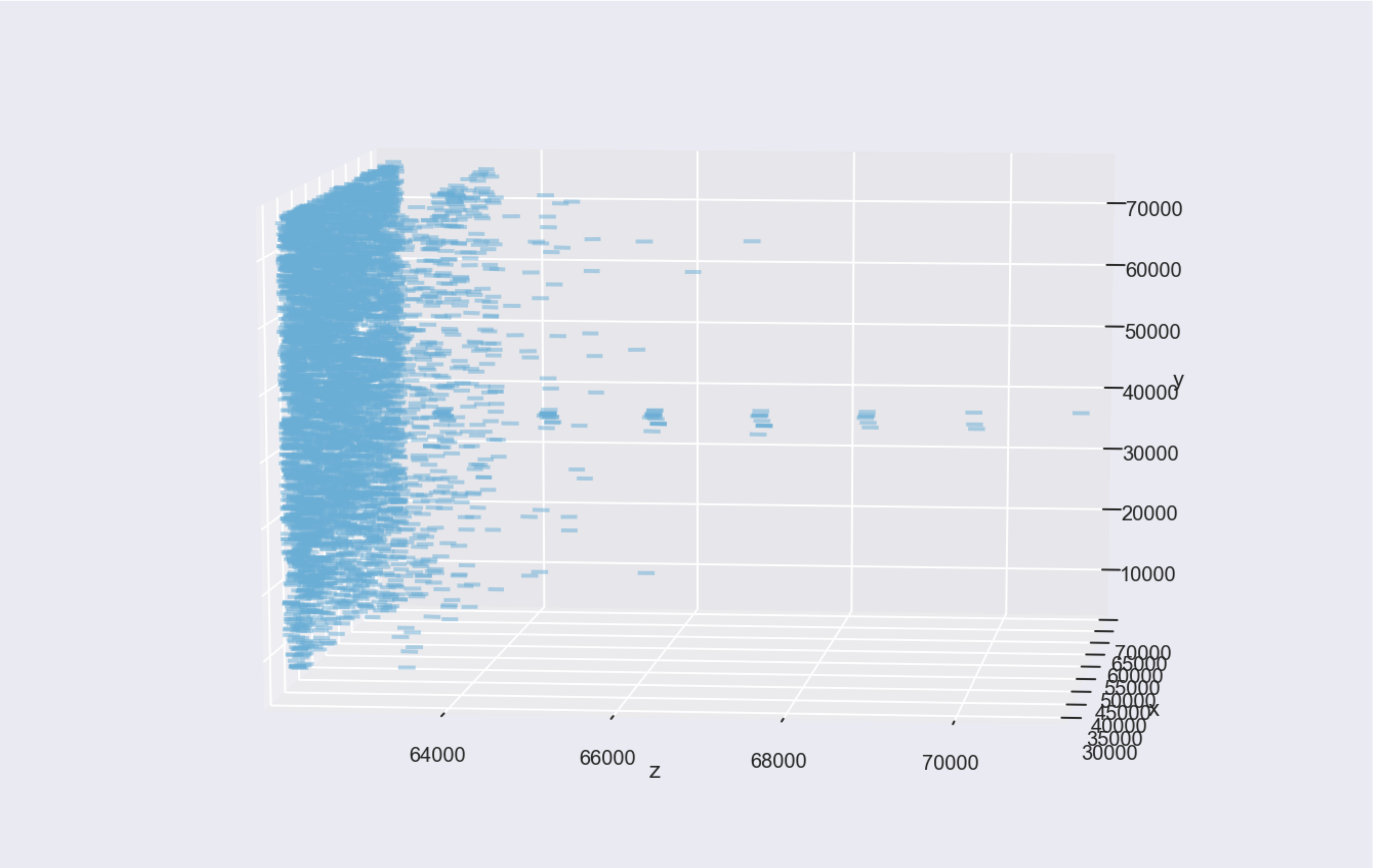
Graph construction:

- nodes of graph x_i - basetracks
- edges connect only sequential basetracks (from the 3mm x 3mm square)
- E_i is log(p), where p is SVM output
- E_{ij} represent basetracks similarity (e.g. E_{ij}= |\Theta$_i$ - *Theta*$_j$|)

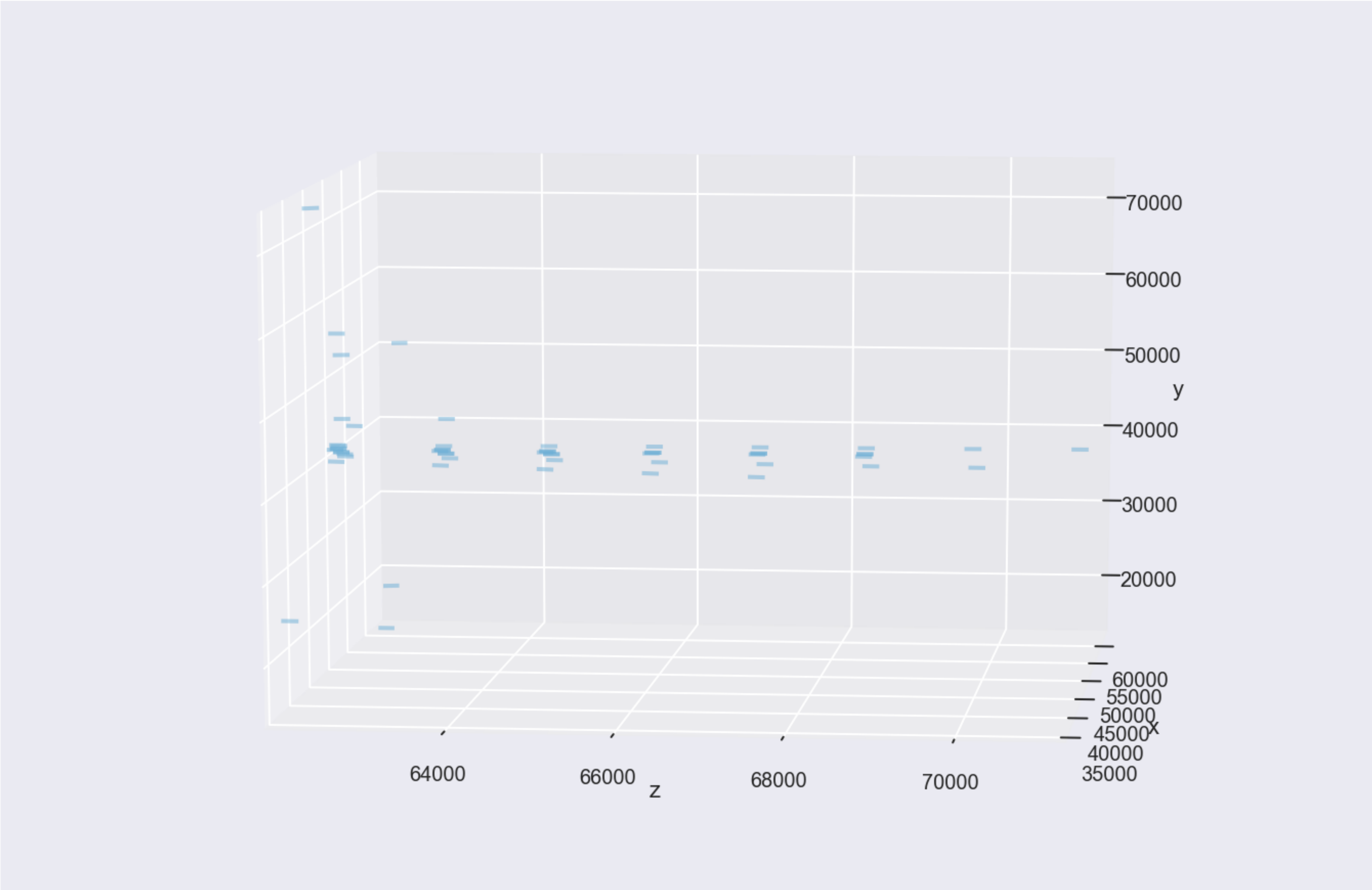
During optimization (e.g. min-cut/max-flow) changes E_{ij} so that in the end signal tracks are connected only to signal and background only to background. As a result it gives labels $t_i \in \{0, 1\}$ to every track that can be interpreted as background/signal.

CRF processing

Background: ~5 * 10^3
Signal: 77

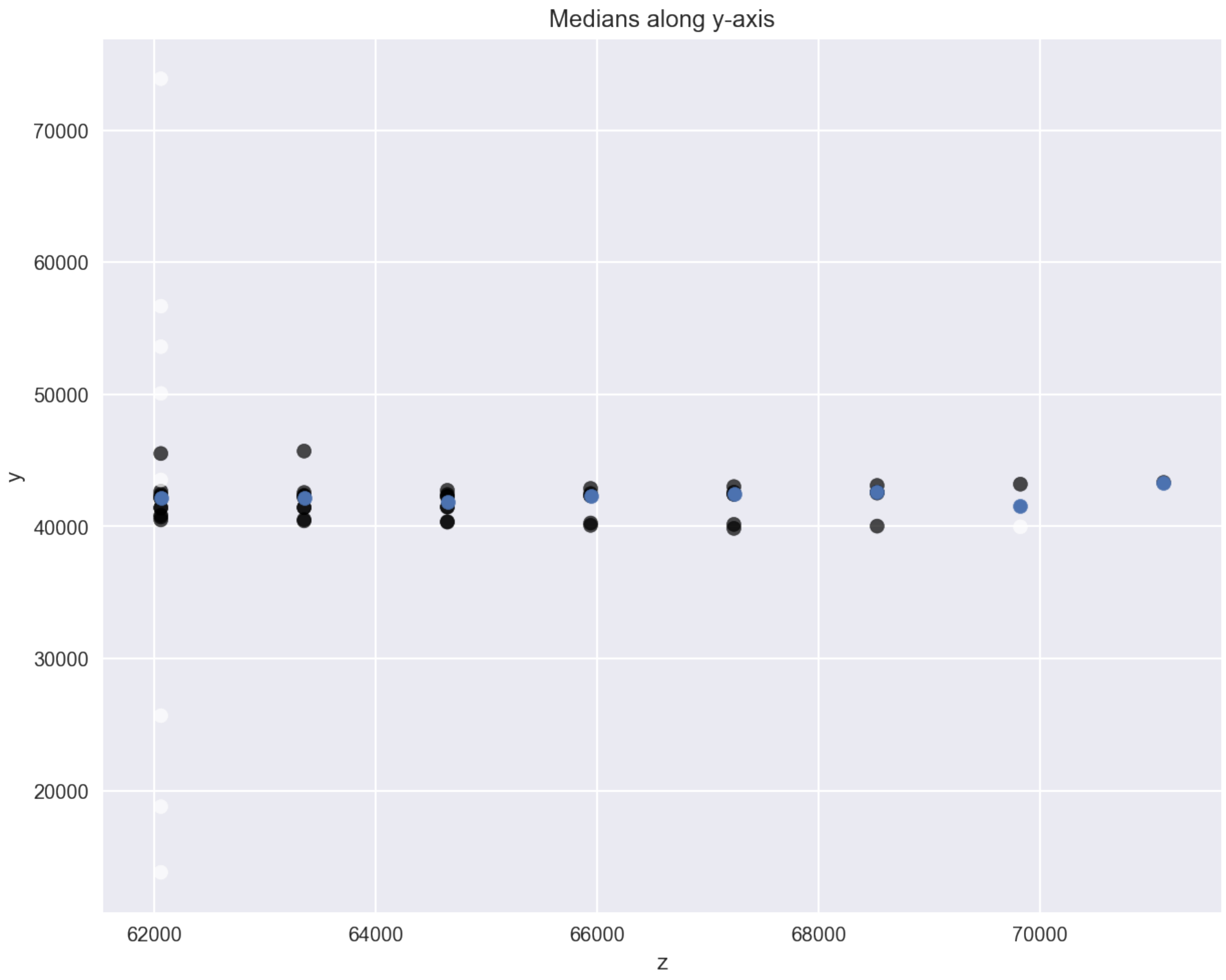


Background: <10
Signal: 47



Andrey Ustyuzhanin

Estimation of em-shower origin



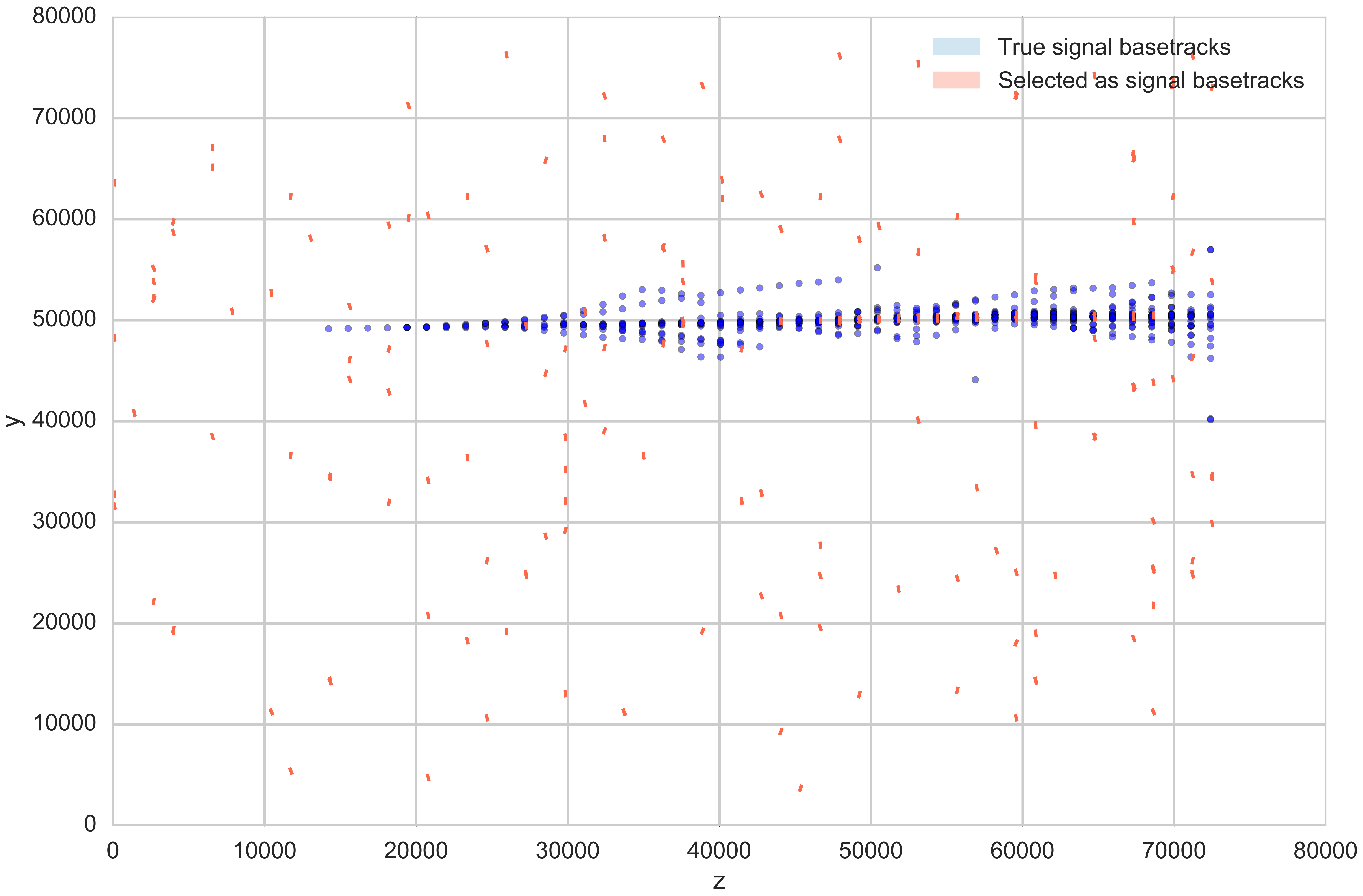
- X,Y coordinates of the shower origin are well predicted by medians on the tracks from the downstream plates
  - σ <7mm

- At the same time, prediction of Z coordinate looks more difficult

Algorithm 2

- Use all the plates;
- For each basetracks select nearest K(=10) tracks from the same plate and calculate features (\Theta, \Delta, chi2 difference) plus track's own Chi2 value.
- Pre-selection algorithm (X): Logistic Regression (100 events)
- Topology filtering (Y): median line reconstruction + estimation of ΔZ from upper-most plate to shower origin (60 events)
- Apply OPERA shower selection algorithm from reconstructed shower origin

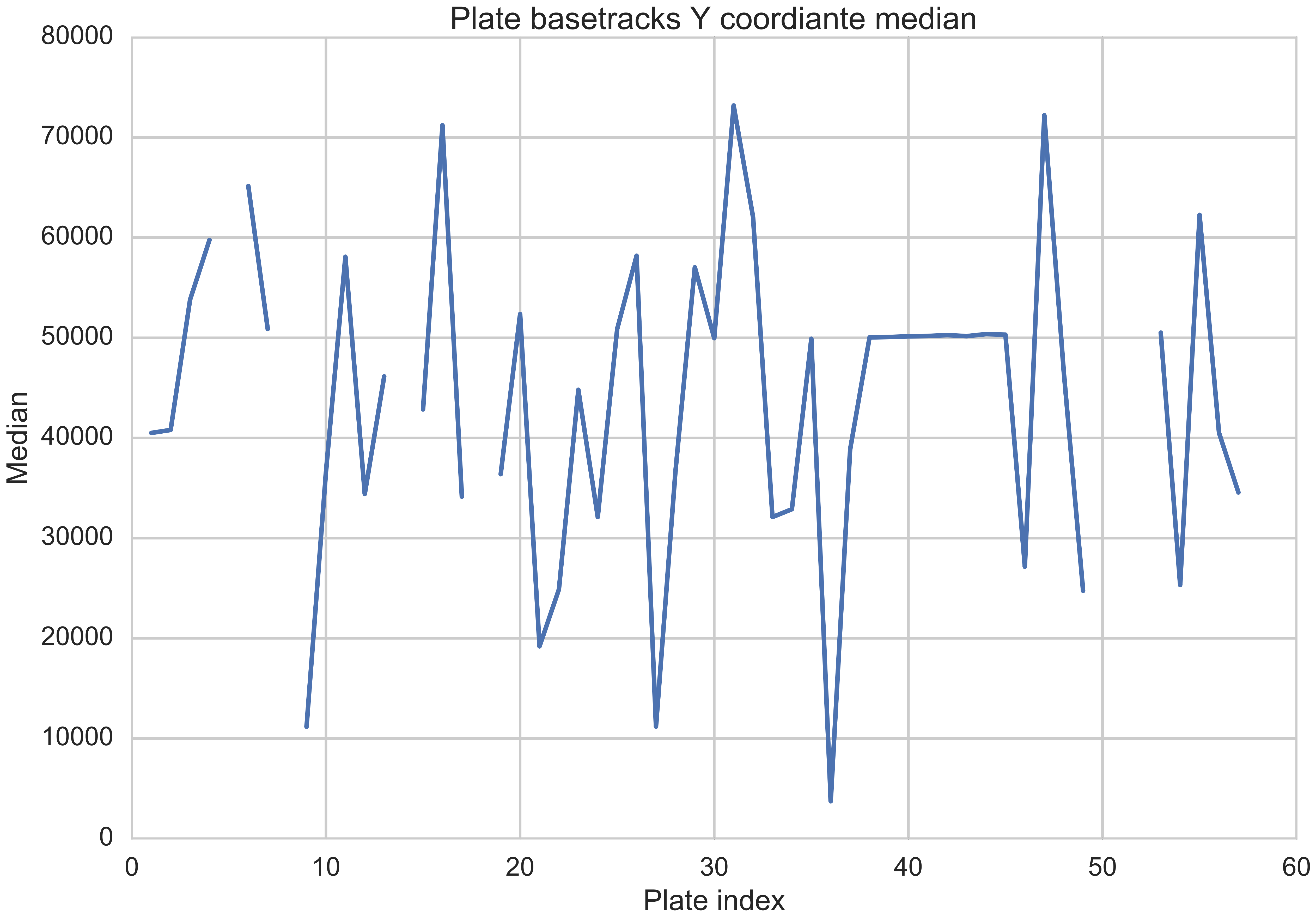
Typical (Y, Z) brick after pre-selection



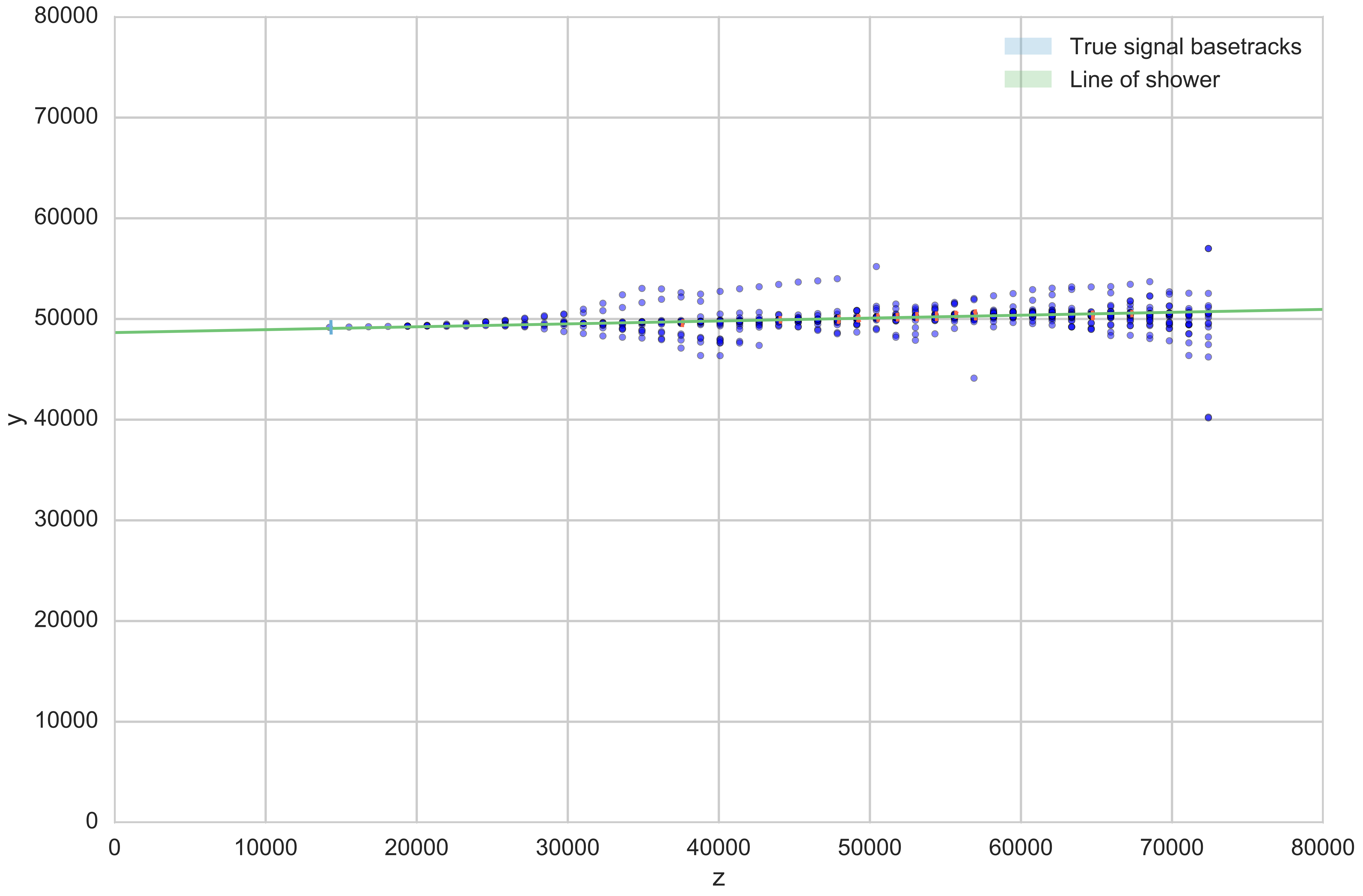
Clear selected tracks after iteration

Algorithm detects plato and throws away tracks which are away from it, since the plato accounts for central shower points.

After that, the line of shower is drawn on cleared data.



Median Line (shower center candidate)



Shower origin estimation

Use regression model (interested in ΔZ):

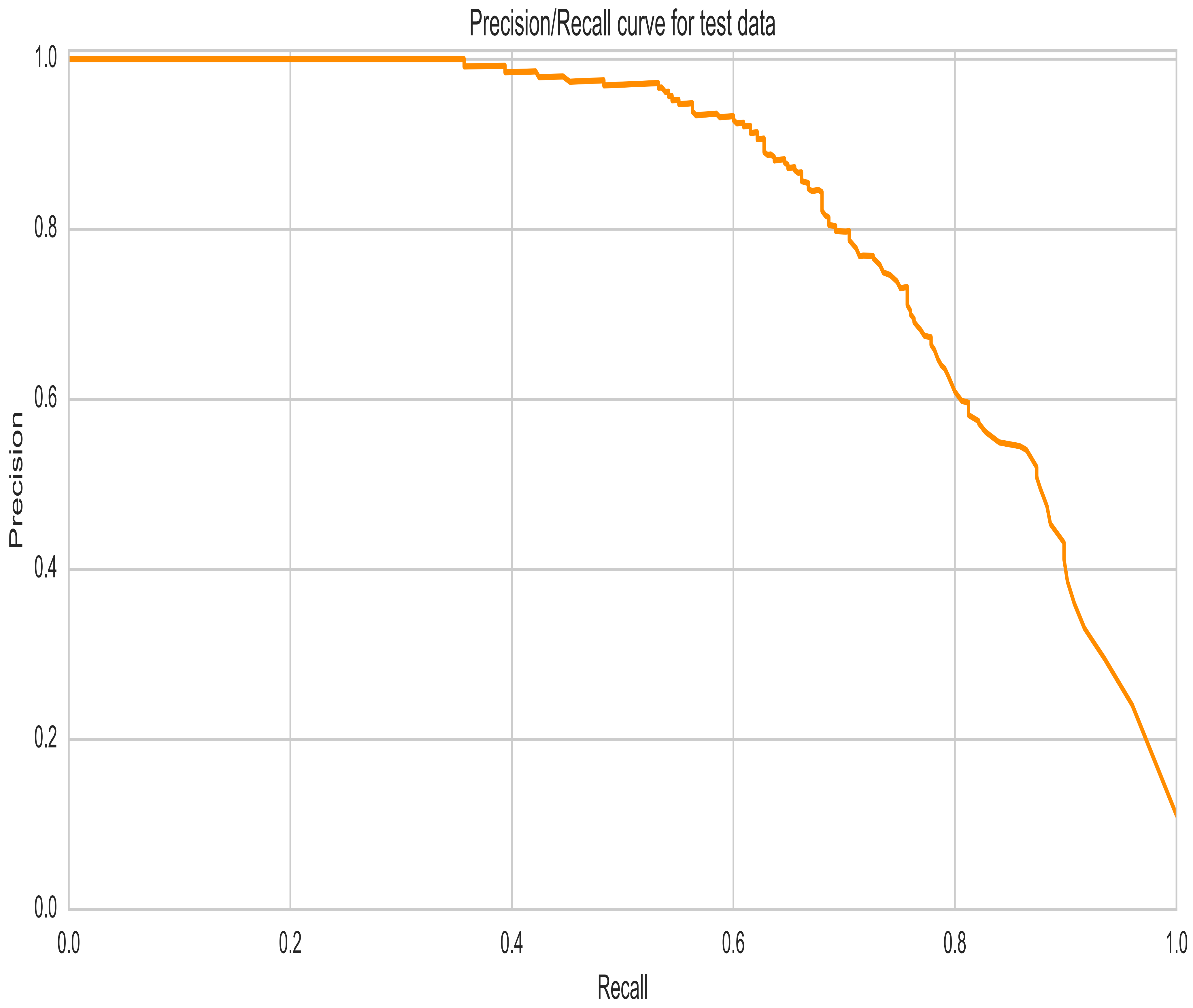
$$Plate_{origin} = Plate_{last\ detected} + \Delta Z$$

As we figured ΔZ we can use to identify the shower origin as intersection of the median line with plate ΔZ plates from the upper-most plate.

At the same time we identify the decision threshold corresponding to Recall = 0.5, to select basetracks in the test events sample (1k events).

- MSE distance in XY plate to true shower origin is 0,67 mm.
- MSE distance Z to true shower origin is 4.5 mm.

Precision/recall curve after origin identification

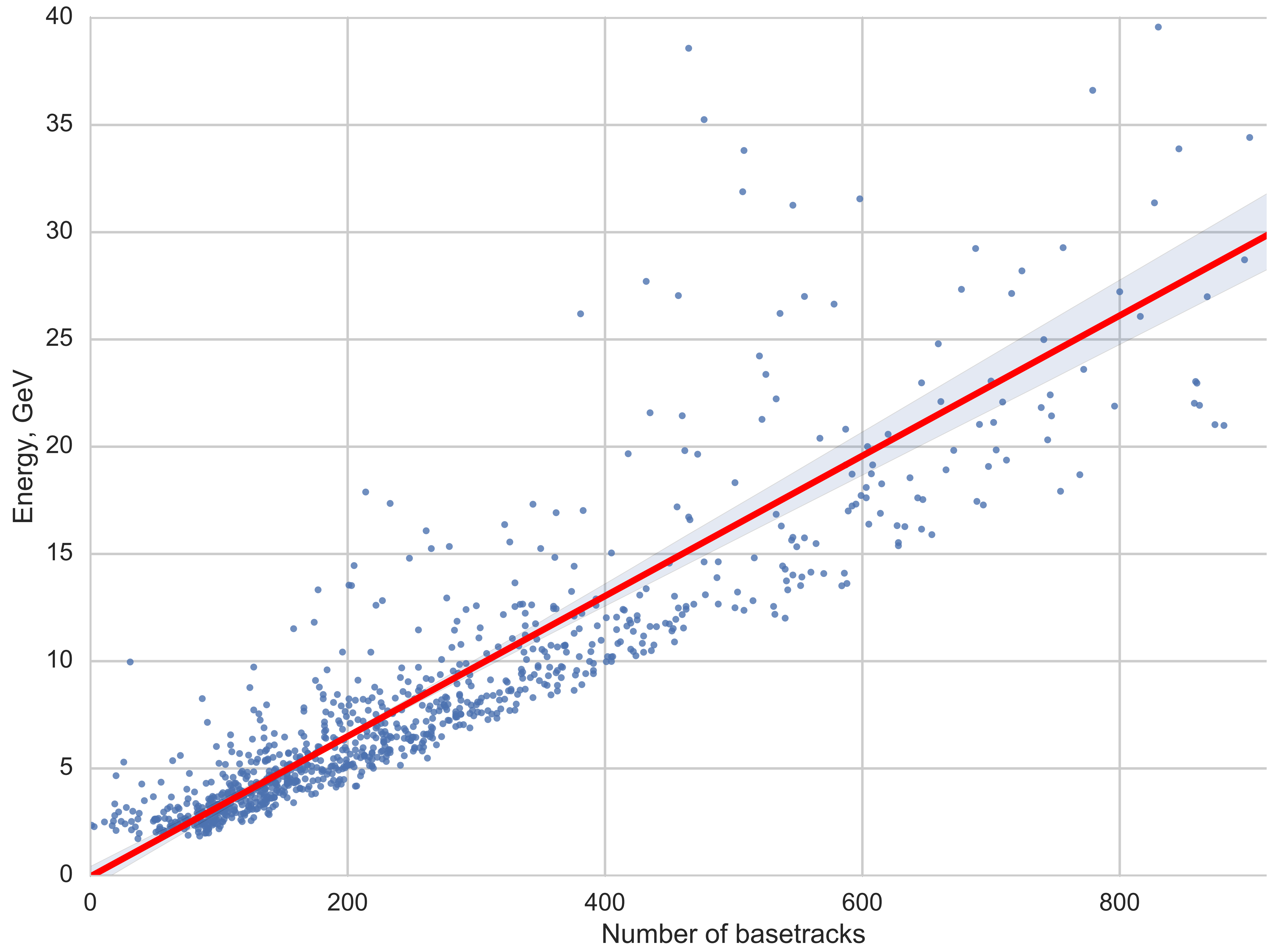


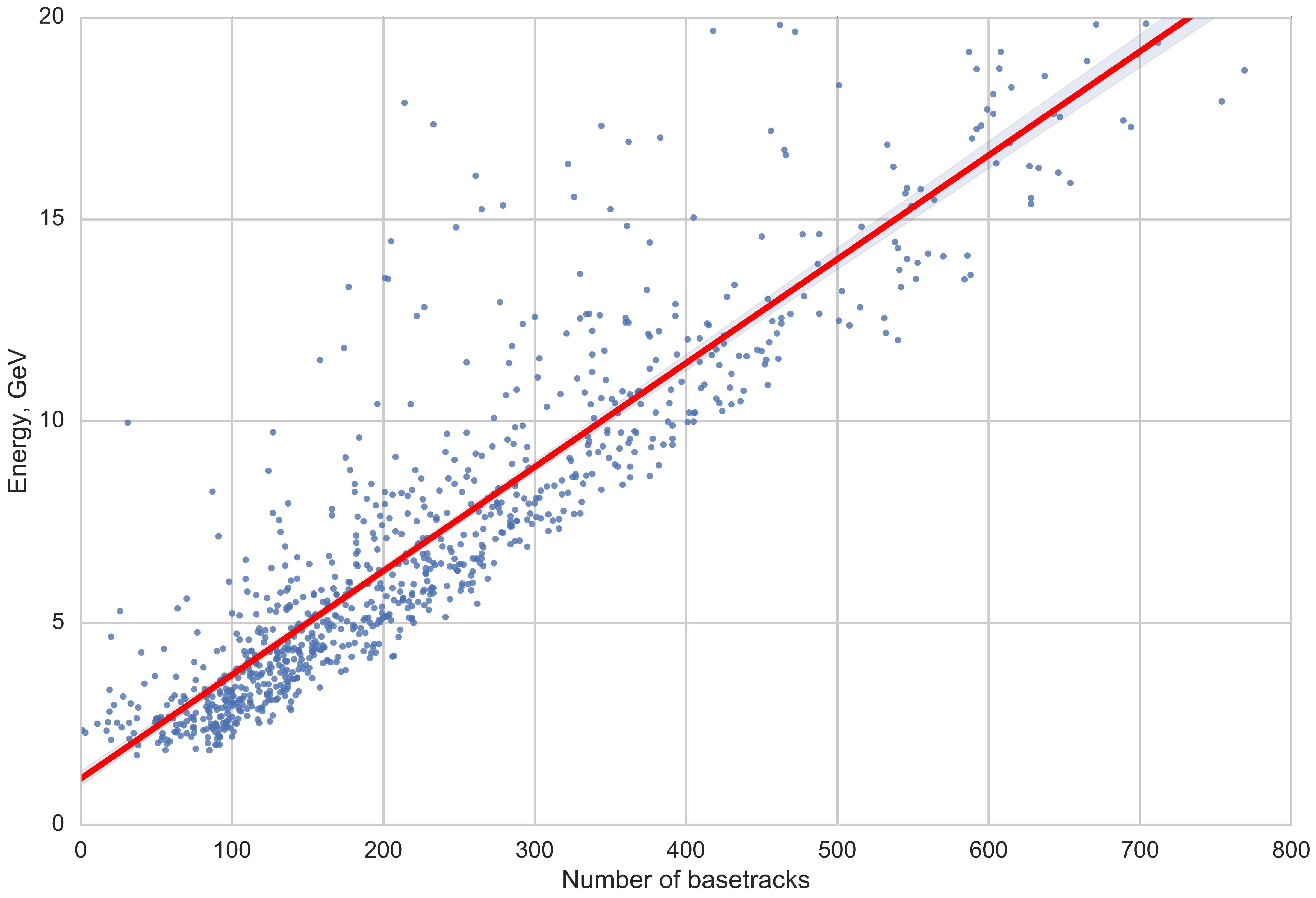
OPERA algorithm – identifies signal basetracks given initial point & direction

The precision/recall is built after applying this algorithm to initial brick from detected origin.

Mean PR AUC for 1000 test events is 0.85 ± 0.093

Energy resolution estimation





Electron Energy vs Number of basetracks selected.

Ntracks < 1000 (964 events), Energy resolution ~ 0.27 (Left)

E < 20 GeV (900 events), Energy resolution ~ 0.24 (Right)

Discussion

- The overall strategy works :

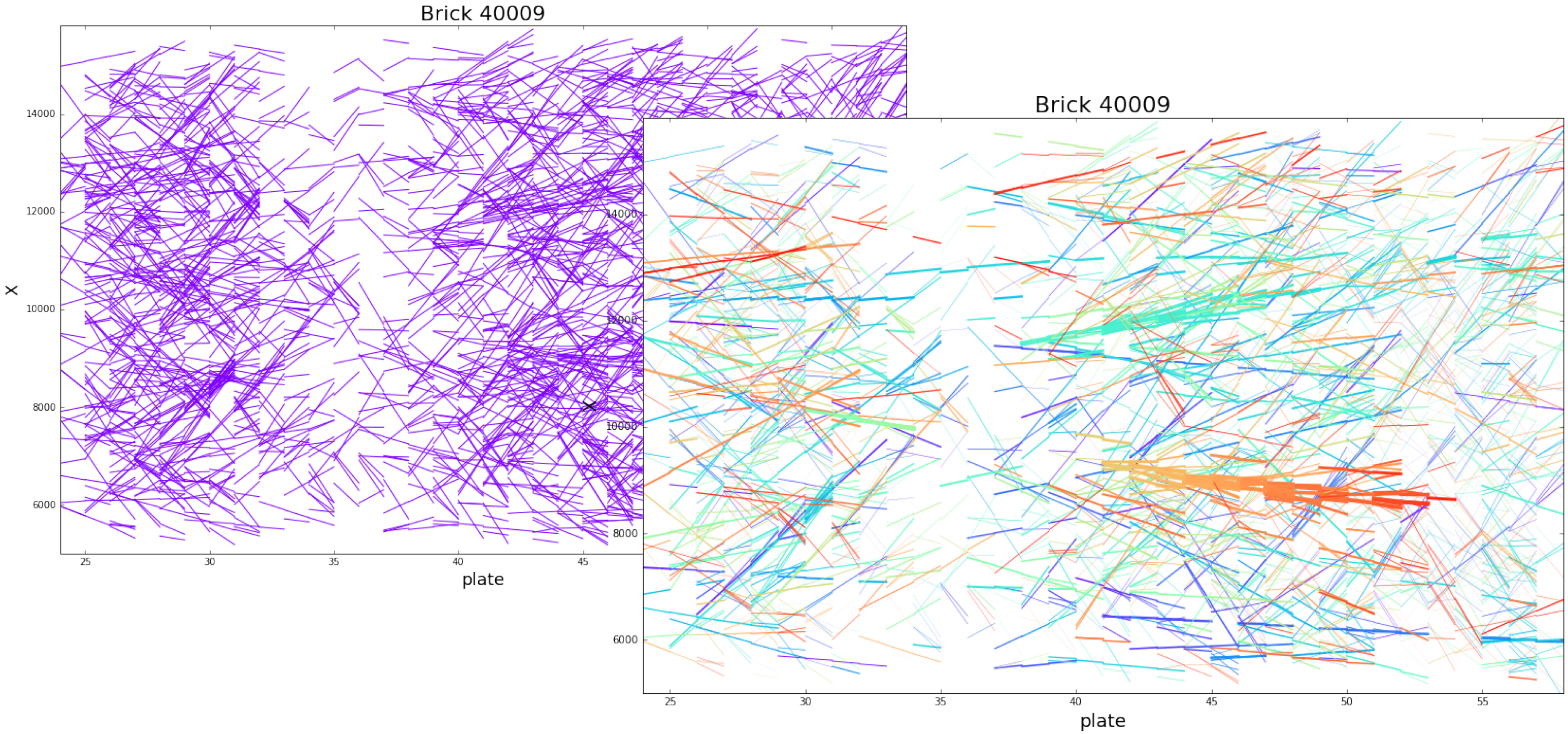
- **Basic pre-filter**
- **Use topology-aware filter (Conditional Random fields or Clustering)**
- **Post-filter to identify origin position**
- **Apply OPERA algorithm**

- Energy resolution is less ~ 25% (with no a priori information!). Can be improved with origin precision improvements

- Has to be tested on data (\Chi^2 might be rather dangerous feature)

- Has to be adapted to SHiP specific: use clustering

Clustering


Brick 40009
14000
12000
10000
8000
6000
X
25
30
35
40
45
plate
Brick 40009
14000
12000
10000
8000
6000
25
30
35
40
45
50
55
plate

Optimization of the Emulsion Detector for SHiP

Possible optimization parameters

- number of plates, lead thickness, emulsion thickness;
- passive material (tungsten, lead, ...);
- magnetic field (1.5Tesla, 1Tesla or off);
- exposure time (background density);
- brick X, Y dimensions.

Strategy:

- Generate signal sample, background sample (cosmic, beam), @Naples, Yandex;
- Specify Figure of Merit (energy resolution, angular resolution), @Naples;
- Identify key configurations for the parameters as starting point, @Naples;
- Generate detector geometry, update signal, background samples, @Yandex;
- Estimate detector efficiency (FOM, central part), @Yandex;
- Repeat previous 2 steps until convergence, @Both.

Conclusion

- The work is still in progress
- **Unify datasets, set of features and metrics**
- It is possible to identify e-m shower even in quite polluted brick without a priori information about shower origin:
  10^6 background tracks, 100+ signal tracks
- Have to test on real data
- Adjust for SHiP specific (200 em-showers per brick)
- Looking forward to optimization results

Backup

References

- Conditional Random Field http://jmlr.org/papers/volume15/mueller14a/mueller14a.pdf https://prateekvjoshi.com/2013/02/23/what-are-conditional-random-fields/
- Support Vector Machine, https://en.wikipedia.org/wiki/Support_vector_machine
- Density-based spatial clustering of applications with noise https://en.wikipedia.org/wiki/DBSCAN

Algorithm 3

- Uses all brick plates;

- Features: for each basetrack, build 100mrad cone towards the next plate. For every base track in that cone get its χ^2 and: \Delta$_{min}$, \Delta$_{max}$, \Theta$_{min}$, \Theta$_{max}$, χ^2_{min}, χ^2_{max}

- Pre-selection algorithm (X): XGBoost;

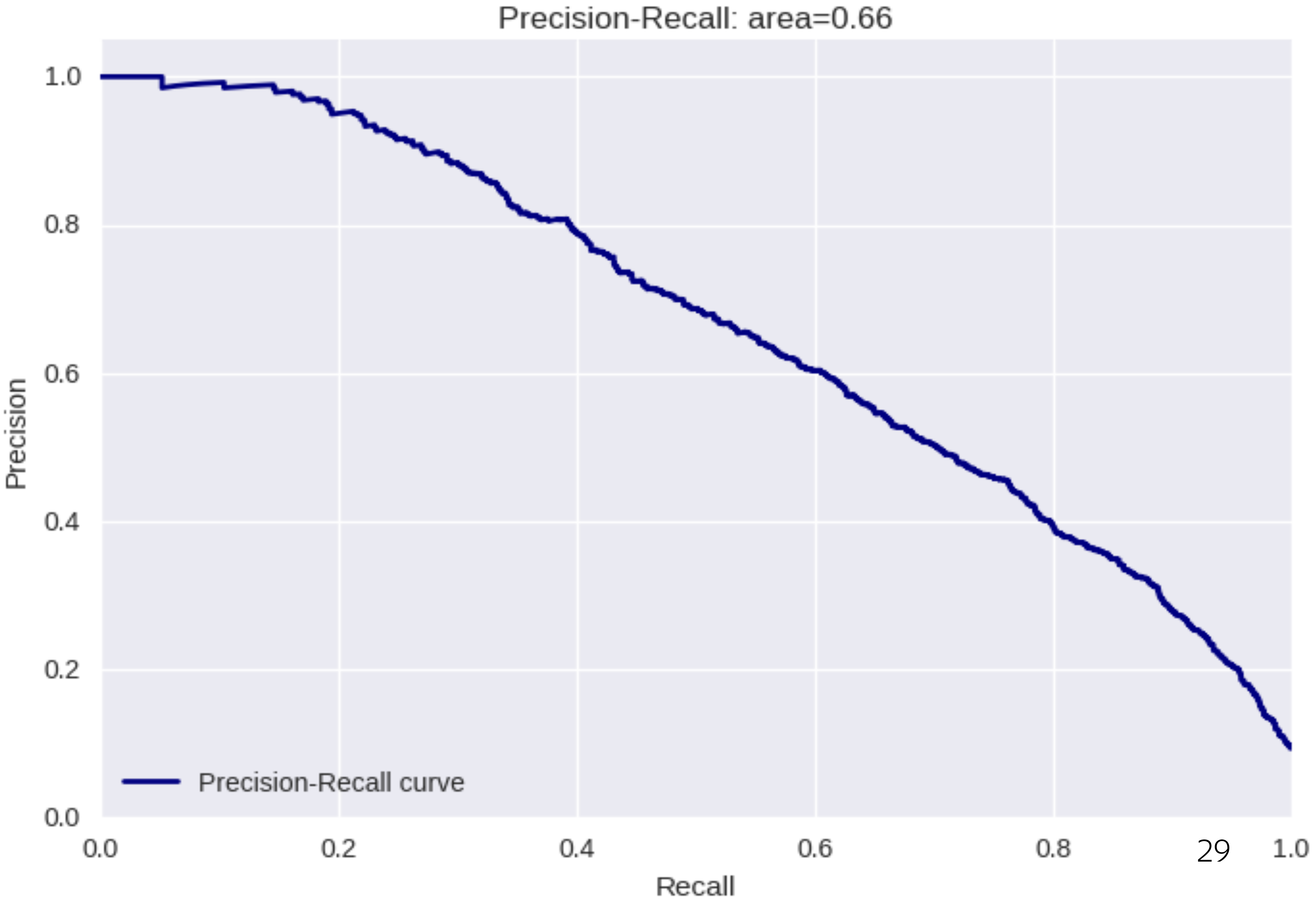
Pre-selection results (after XGBoost)

Recall: signal efficiency
Precision: signal purity

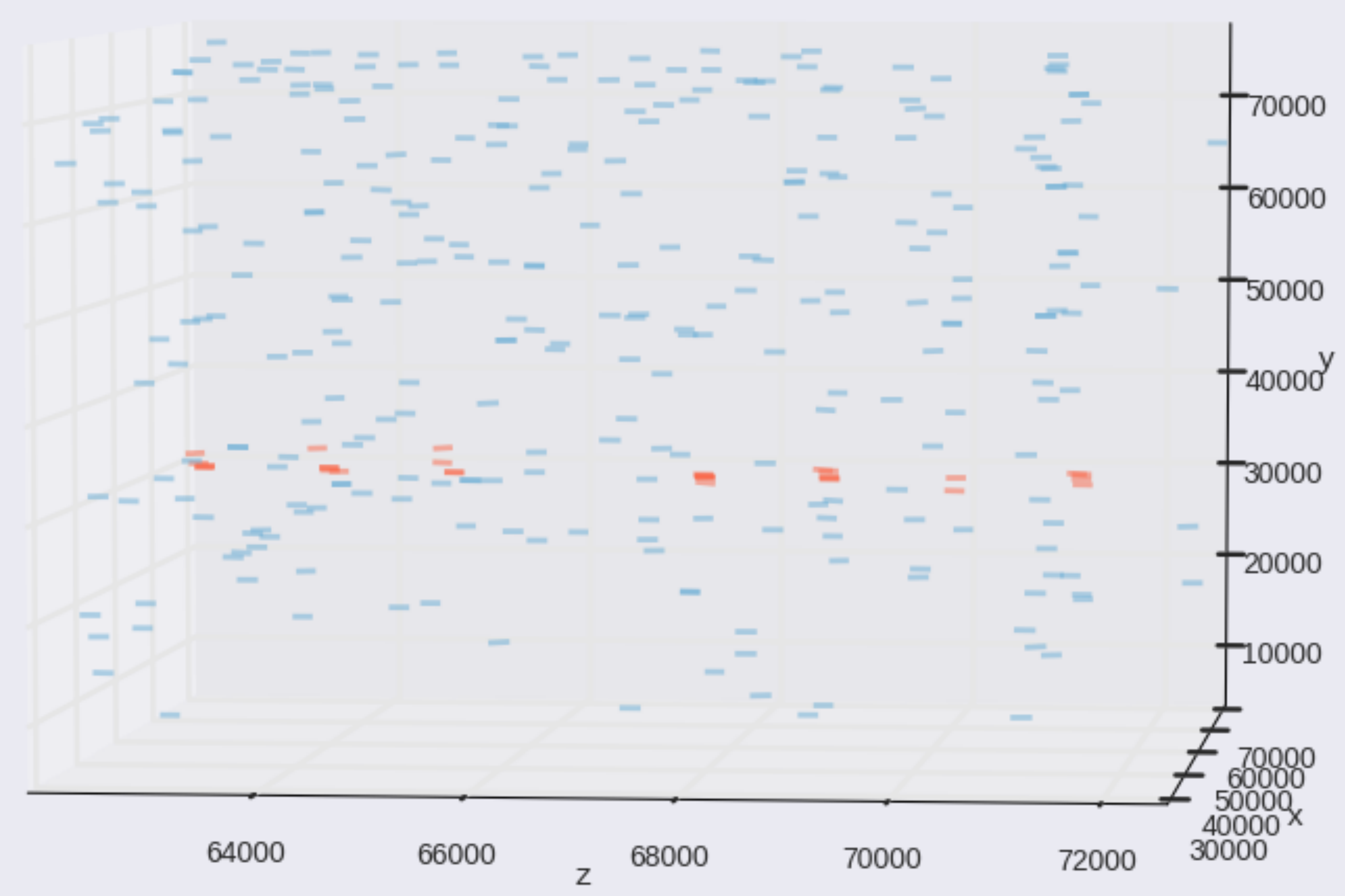
signal purity in the cone from shower origin.

Area under curve (AUC):

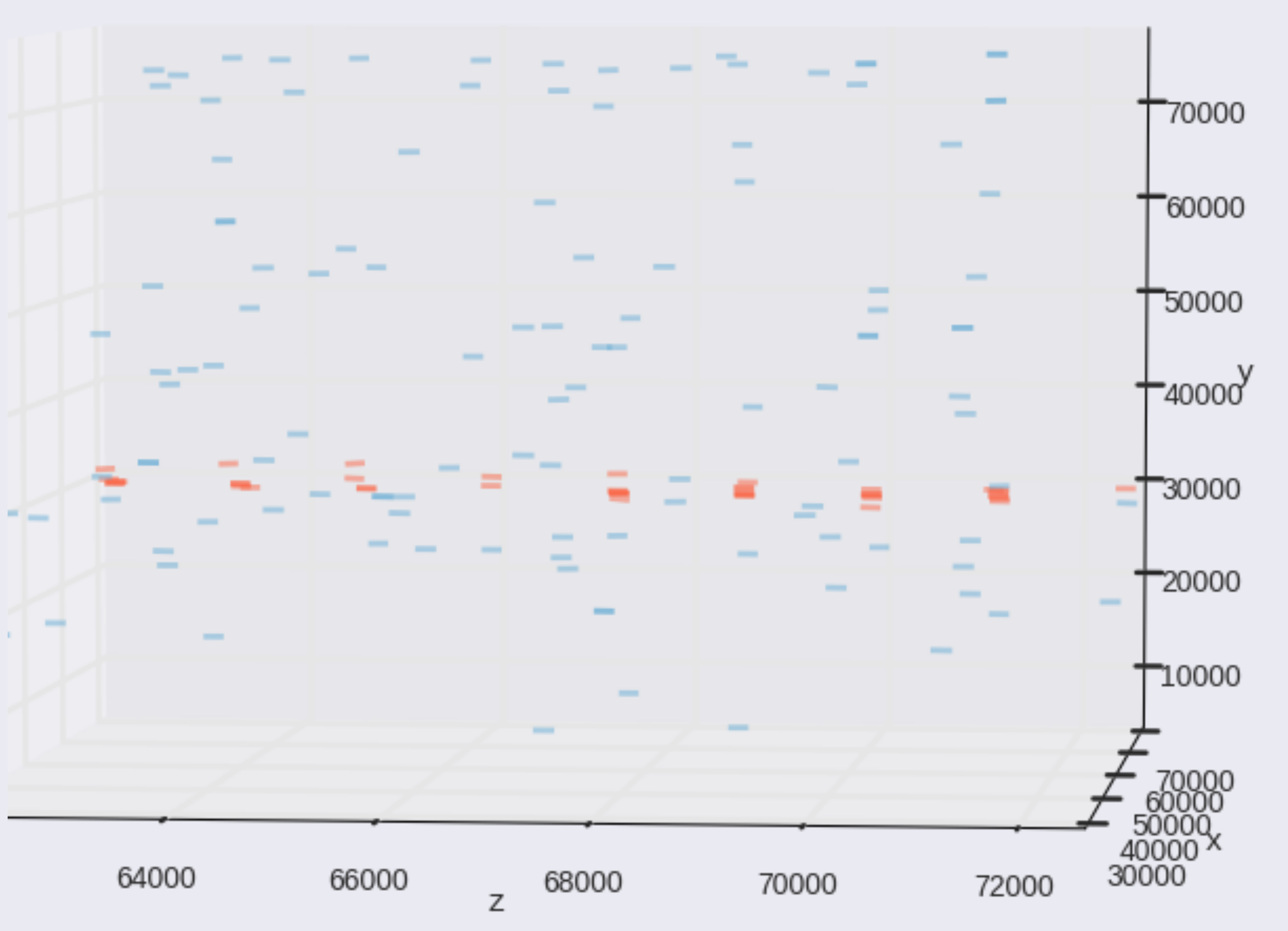
PR-AUC = 0.66



Before (left) and after (right) CRF



Xgboost threshold = 0.9, no CRF

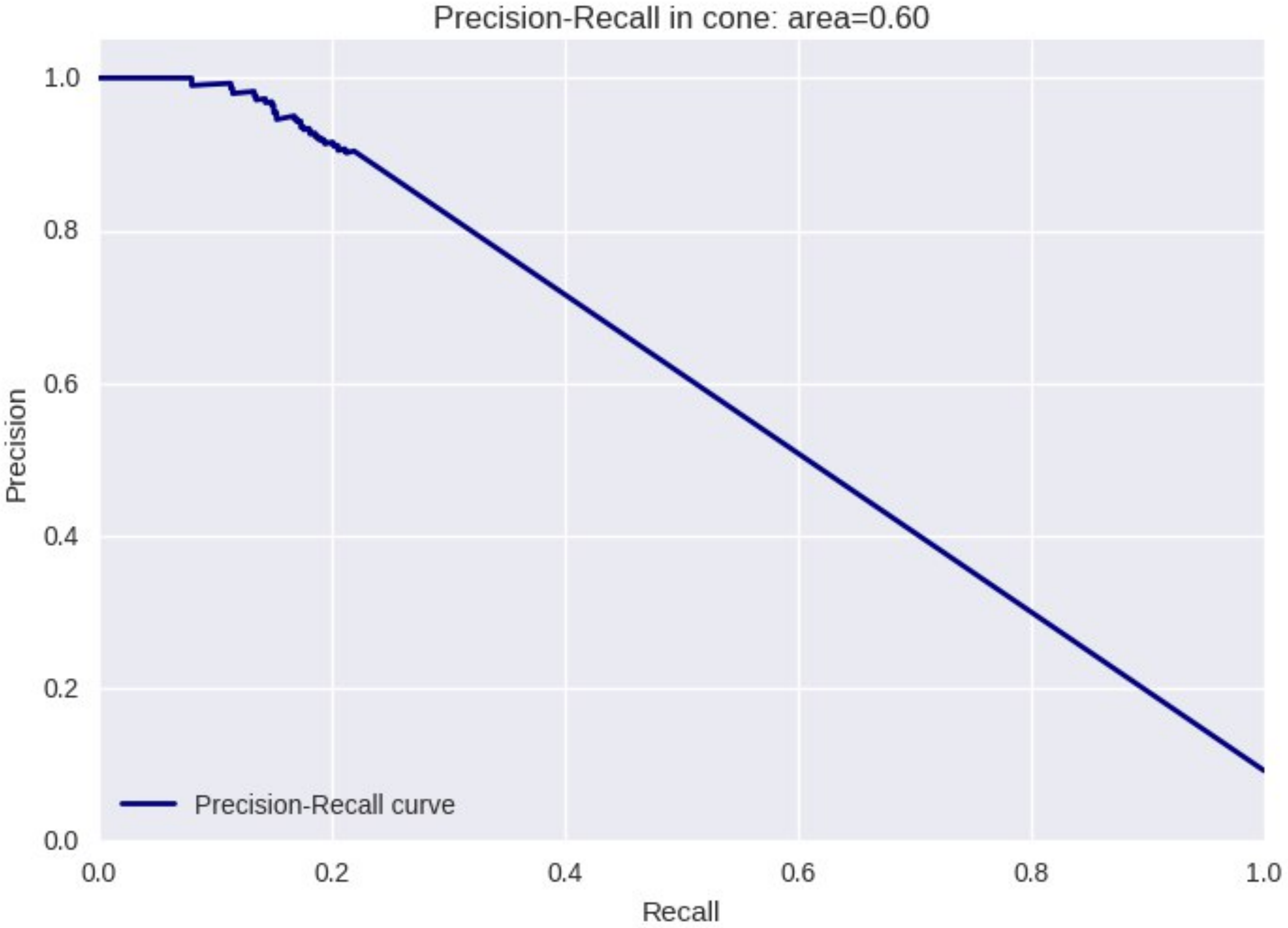


Xgboost threshold = 0.8 + CRF

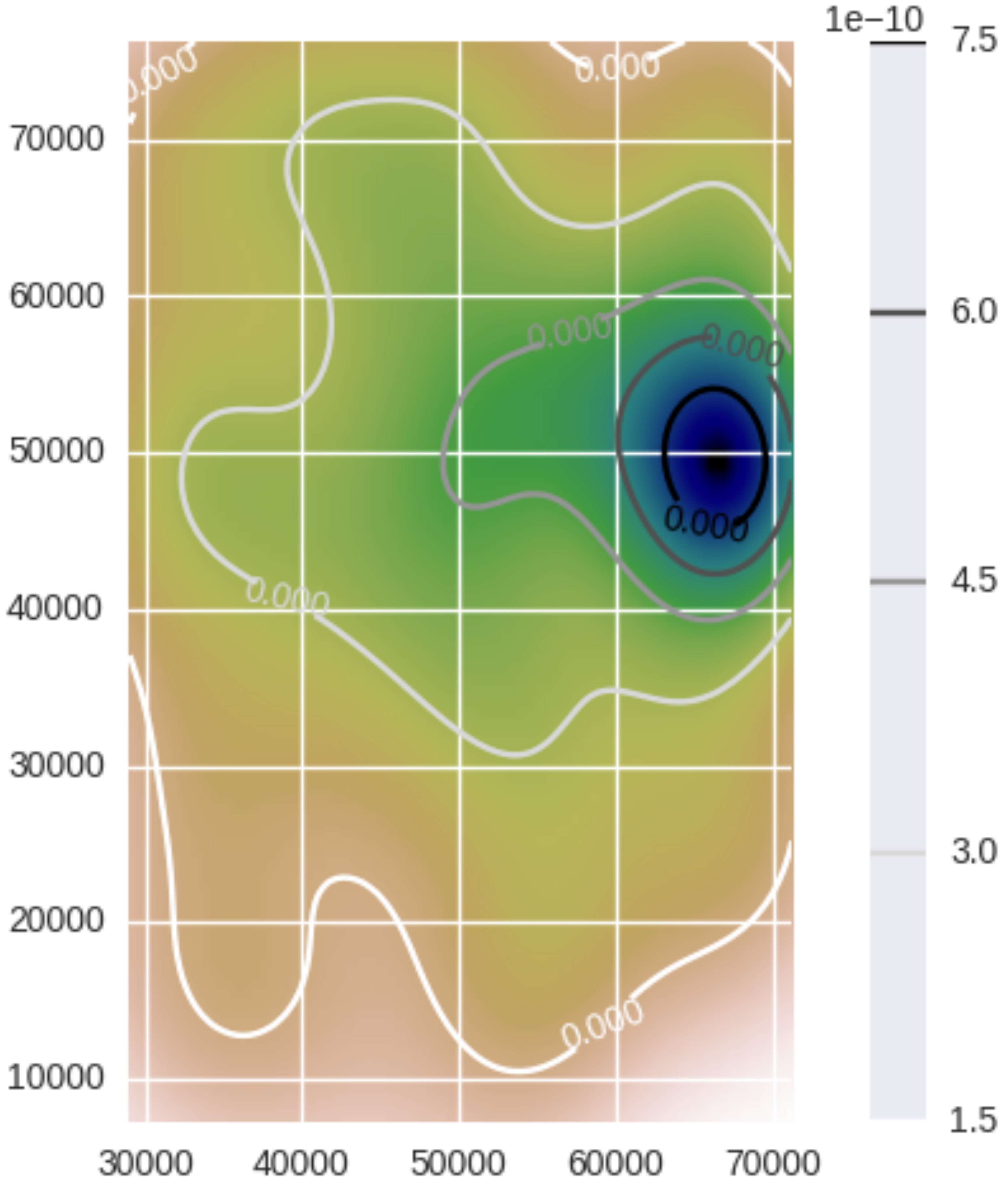
Precision/Recall after CRF

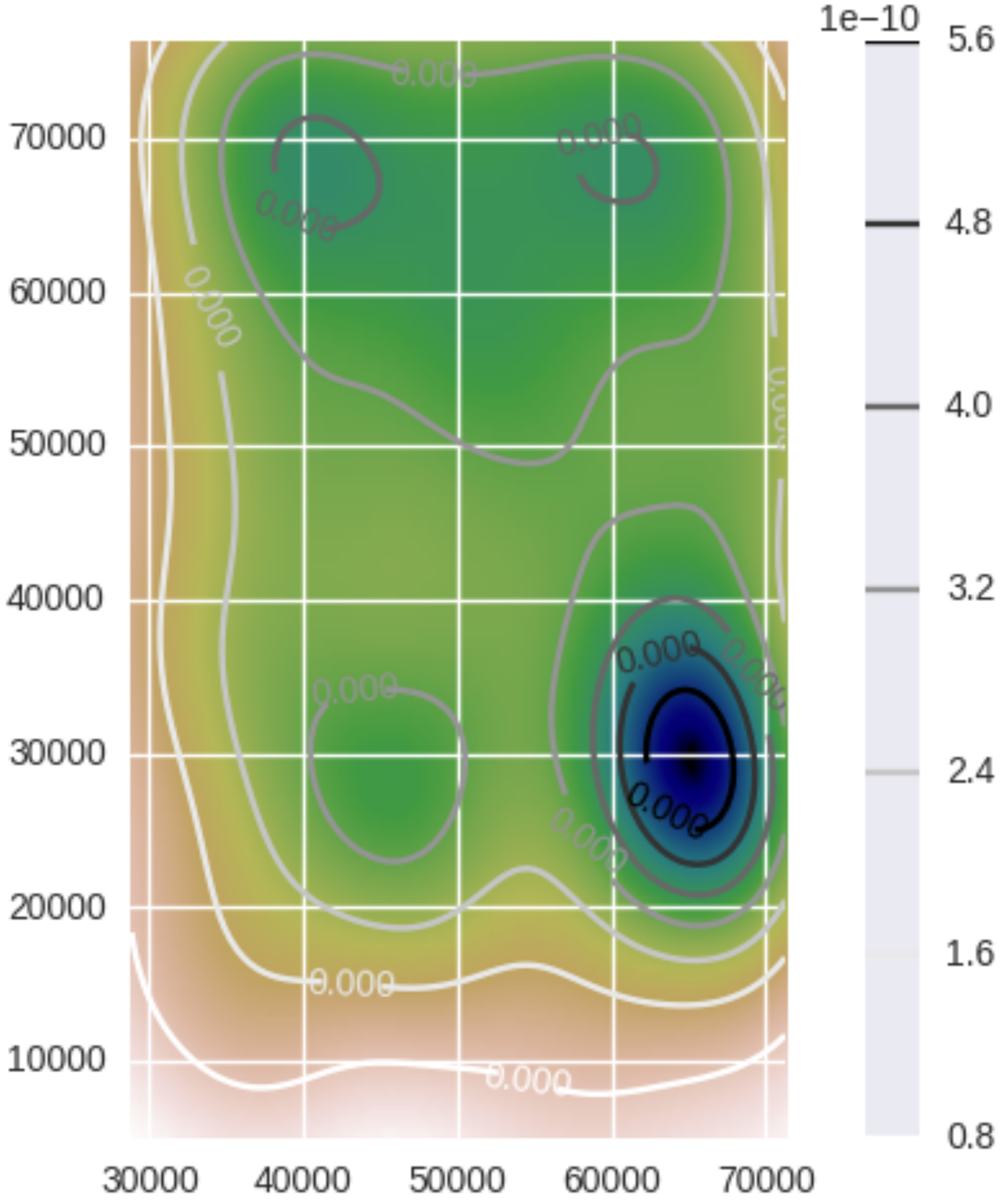
Average PR-AUC = 0.60
(over 5 events)

at Recall = 0.5
Precision = 0.612



Possible extension, 2D Gaussian post-filtering





Algorithm 2. Details

1. Create 57 layers of brick by randomly selecting last 8 layers given. Use 1 / 20 of background basetracks for each signal event. Put shower with more than 200 basetracks inside the brick.

2. For each basetracks select nearest K(=10) tracks from the same plate and calculate features (cosine distance, Euclidian distance, chi2 difference) plus track’s own Chi2 value. Train Linear classifier (LogisticRegression) on the created set. The classifier is trained on 100 events.

3. Clean the resulted data from rest of the background, by calculating median of OY in each plate and reject the basetracks which lies away from median value. Draw a line of shower using PCA on selected basetracks.

4. Iteratively go backward and specify line direction on each iteration and try to guess shower starting point from regression as well as threshold for recall fixed at 0.5. The train sample is 60 events.

5. Apply Hosseni paper algorithm from predicted starting point to classify events again and calculate energy resolution. The test sample is 1000 events