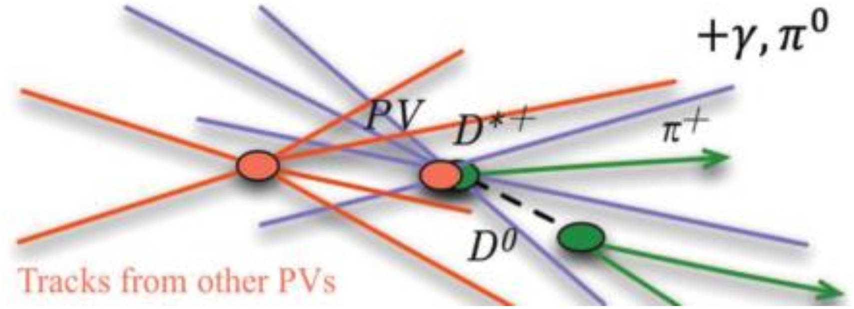
Matrix Factorization based algorithm for PV finding

Alan Aneeth Jegaraj

Mentors: Gowtham Atluri, Mike Sokoloff

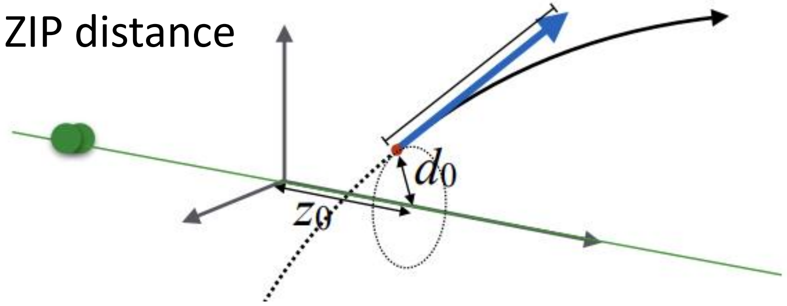
University of Cincinnati

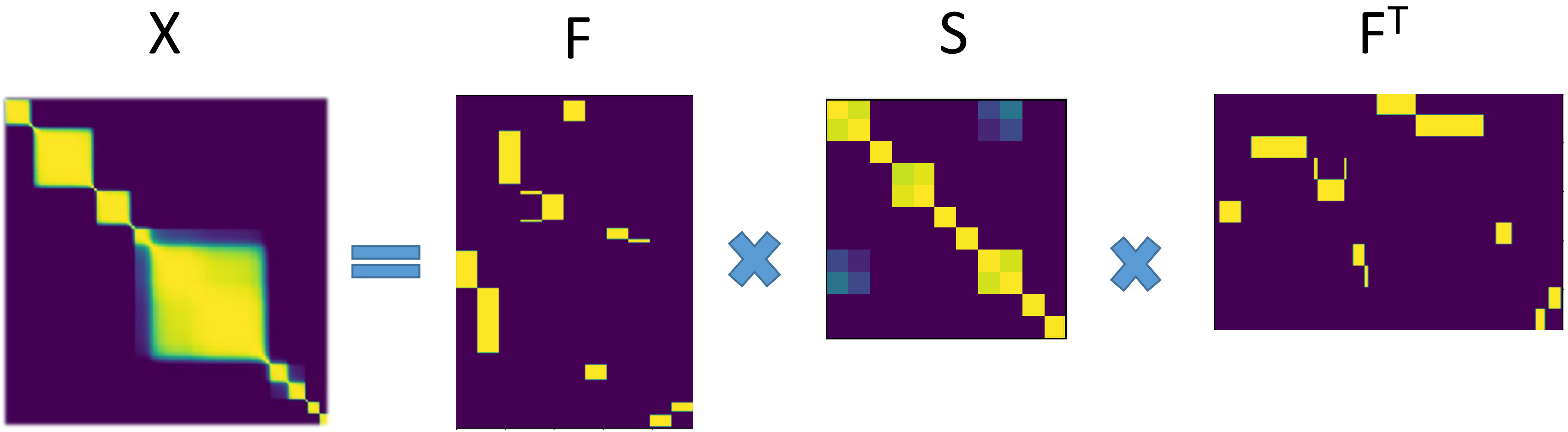
Primary Vertex(PV) Finding



- Accurate estimation of primary vertices is crucial for progress in LHCb.
- A general strategy for PV finding has two steps:
    1. Reconstruct tracks from hits
    2. Identify PVs using reconstructed tracks
- Clustering algorithms have been used for PV finding
    - Kucharczyk *et al*. 2014, Primary Vertex Reconstruction at LHCb
- Matrix factorization methods from machine learning provide a robust alternative to clustering methods
- Utility of matrix factorization for PV finding is evaluated in this work.

Matrix (Tri)-Factorization





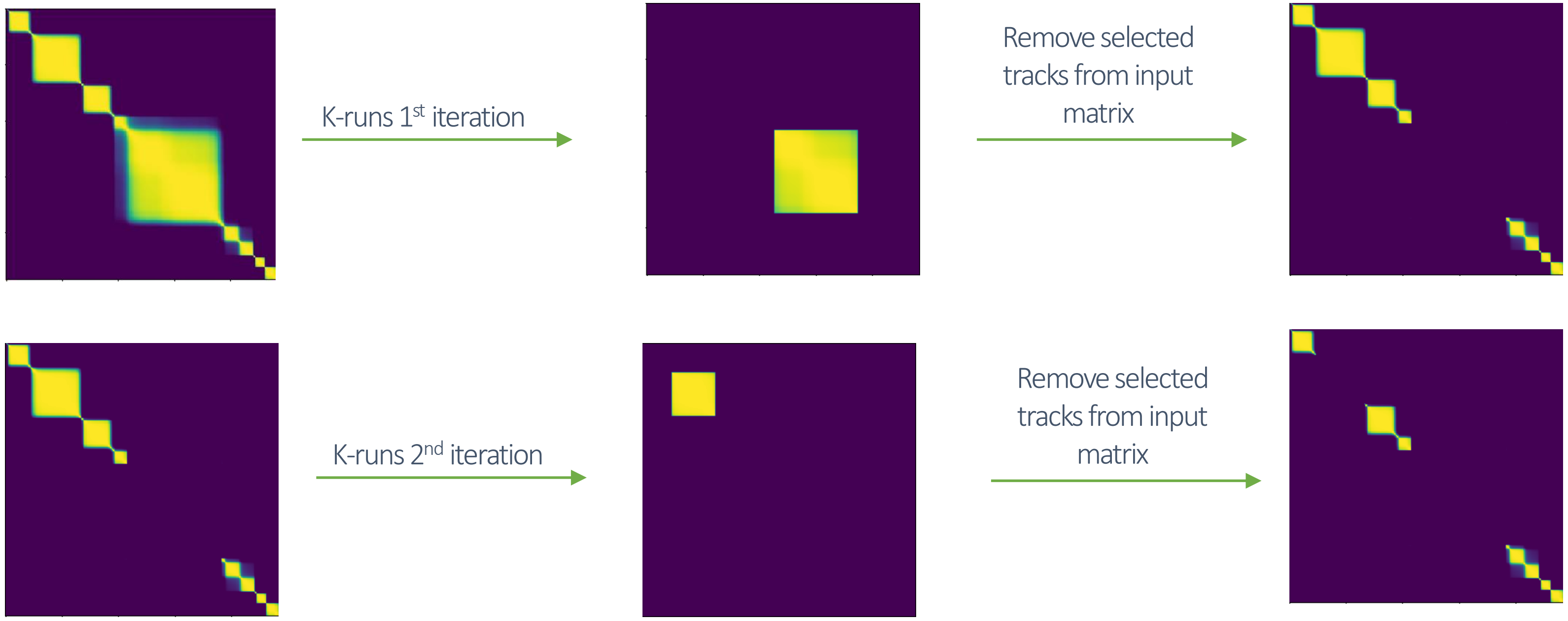
FNMTF, Wang *et al.* 2011

Why MF? Track affinity matrix shows distinct block structures

K-runs FNMTF implementation

Choose a value for K, which controls how many PVs to search for

Loop for K-iterations

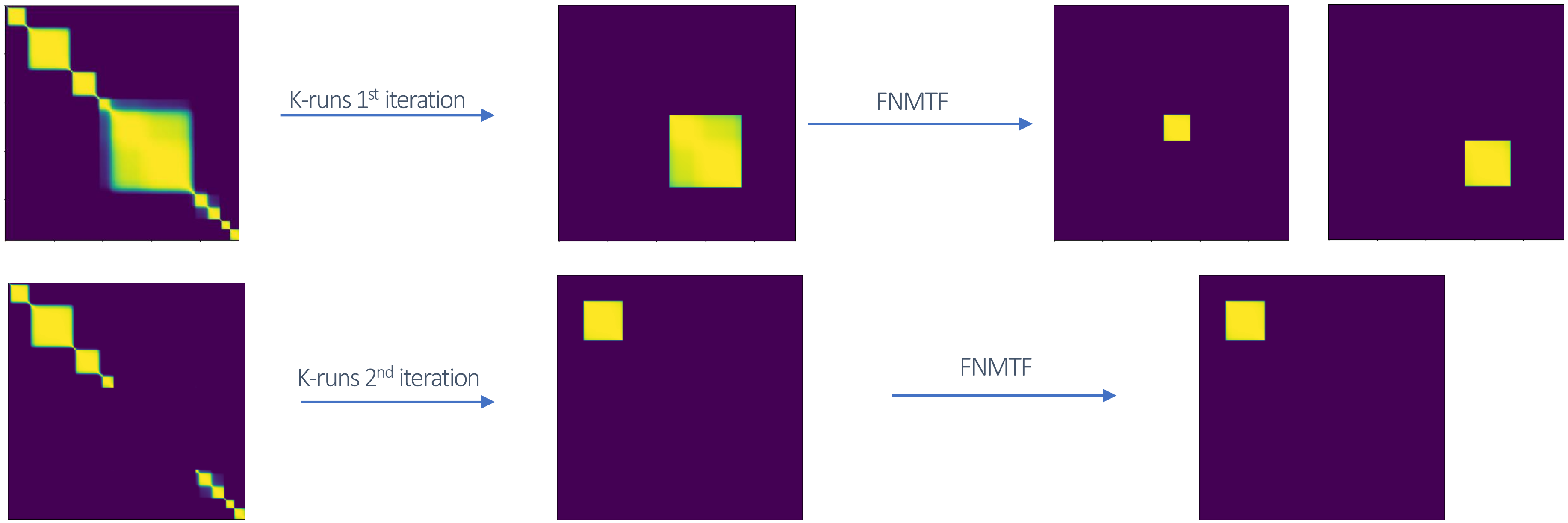


- This approach performs better than FNMTF when it comes to detecting PVs with low tracks and are isolated.
- It always prioritizes larger PVs over smaller PVs
- FNMTF is better able to distinguish between closer PVs with high track multiplicity

K-runs + Split

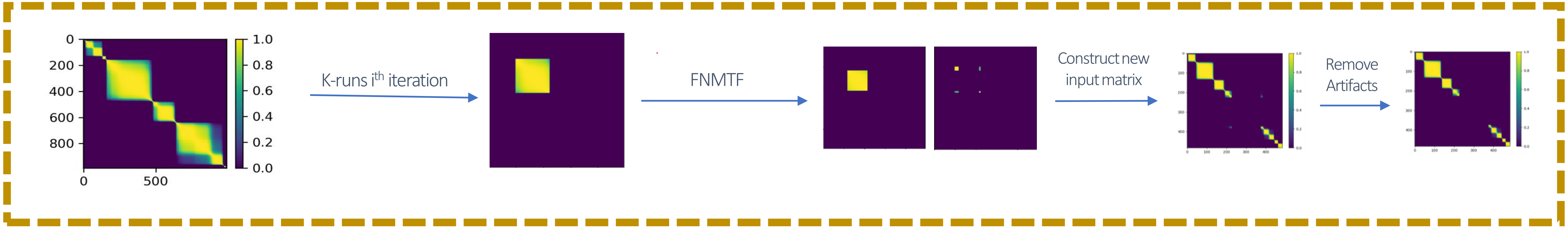
Choose a value for K, which controls how many PVs to search for

Loop for remaining K-iterations



Final Pipeline

Keep looping until stop condition is met



Combines the beneficial features of both K-runs and FNMTF

- K-runs' ability to better identify PVs with low track multiplicity and are isolated
- FNMTF's ability to better detect PVs that are merged and have a high track multiplicity

Experiment setup

- Data
  - 500 events randomly generated using pythia8
  - Total number of PVs in these 500 events = 3906
- Algorithms: K-Runs + Split, K-Runs FNMTF, FNMTF, k-means, HAC
- Metrics
  - GT PV must be within 500 microns of a reconstructed PV
  - A reconstructed PV can only find one GT PV
    - Reconstructed PVs and GT PVs matched using Hungarian algorithm

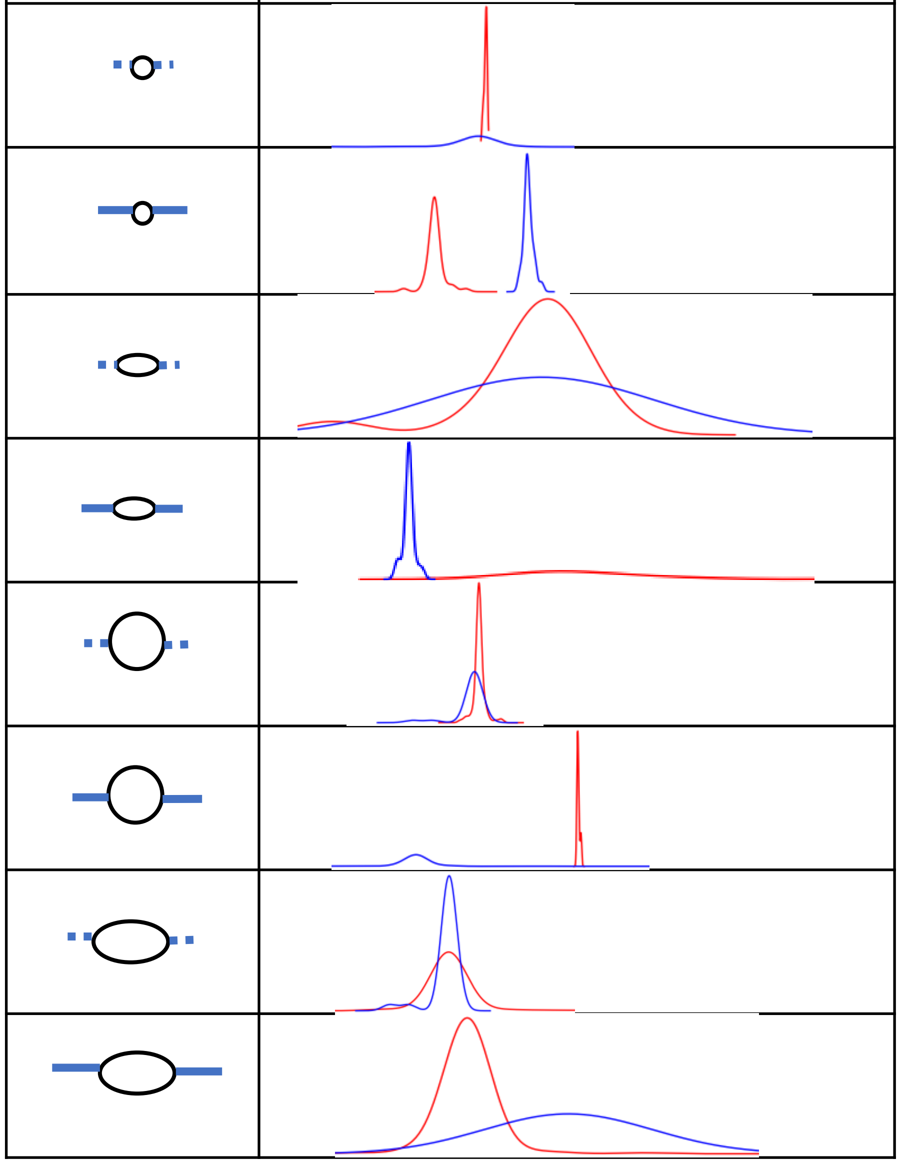
GitHub link: https://github.com/STDSLab/IRIS-HEP-PV-Finding

PV classification

- 3 PV characteristics are identified.
  - Less than 25th percentile is low
  - Greater than 75th percentile is high

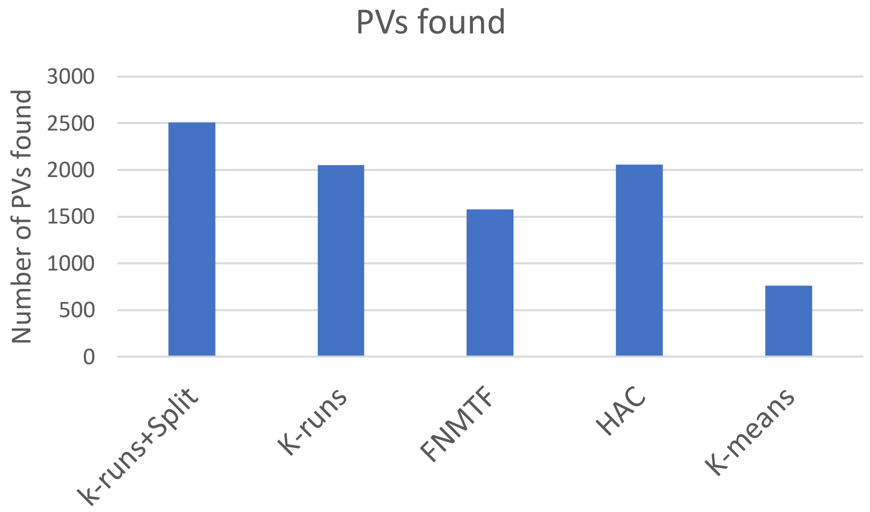
PV characteristic	Low	High
Num tracks	20	76
Variance	20.8	380.8
Distance to closest PV	4.3	22.6

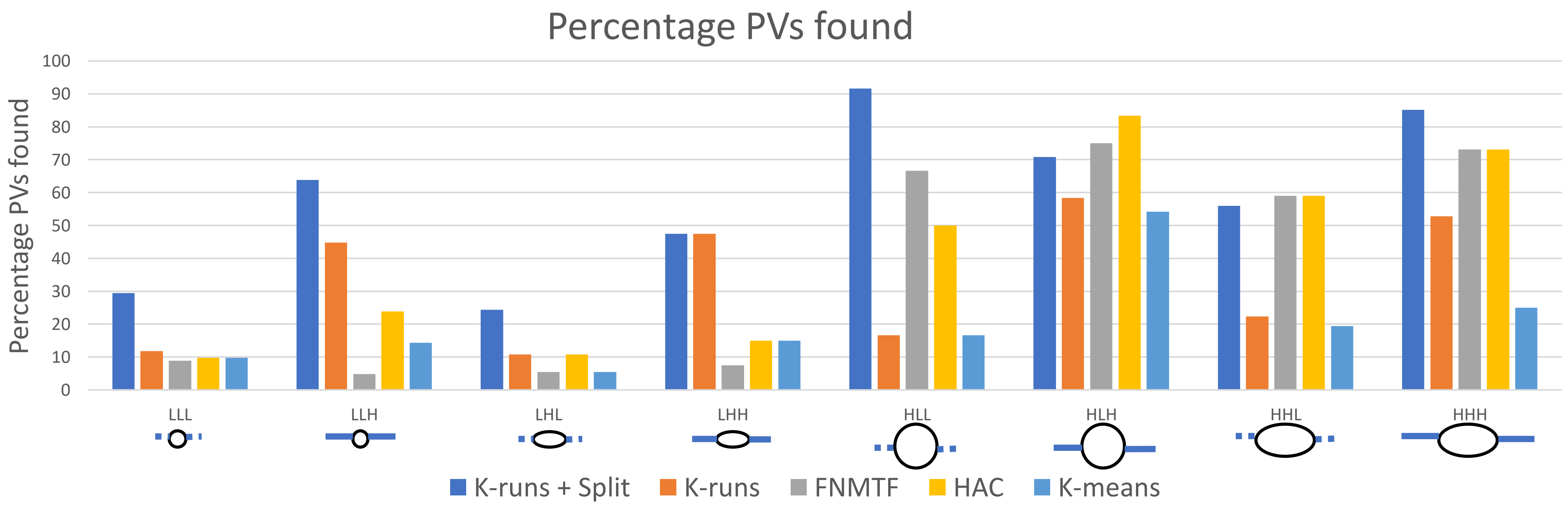
Legend	Track density distribution of PV corresponding to each scenario (Red=PV of interest)



Performance results

Algorithm	Total PVs	Total clusters	PVs found	Percentage PVs found	False positive rate
K-runs + Split	3906	4468	2510	64.26	0.56
K-runs FNMTF	3906	3906	2054	52.59	0.53
FNMTF	3906	3906	1580	40.45	0.40
HAC	3906	3906	2058	52.69	0.53
K-means	3906	3906	763	19.53	0.20





Future research directions and improvements

- Number of PVs discovered in K-runs + Split > # GT PVs
- An ML approach to detect noisy tracks
- A graph neural network to detect when 2 PVs are merged by FNMTF
- Comparing computational efficiency of FNMTF with other PV finding approaches

Conclusion

- Studied the utility of FNMTF for PV finding
- Created a better FNMTF-based pipeline for PV finding
- Experimental evaluation shows superiority of our pipeline to other clustering algorithms

References

[1] Kucharczyk, M., Morawski, P., & Witek, M. (2014). Primary Vertex Reconstruction at LHCb - CERN. from https://cds.cern.ch/record/1756296/files/LHCb-PUB-2014-044.pdf

[2] Wang. H. *et al*, 2011, Fast Nonnegative Matrix Tri-Factorization for Large-Scale Data Co-Clustering, from https://www.ijcai.org/Proceedings/11/Papers/261.pdf

Thank you / Questions?