Graph Segmentation in Scientific Datasets

Rajat Sahay
Department of Computer Science
Vellore Institute of Technology
Vellore, India
rajat.sahay2018@vitstudent.ac.in

Savannah Thais
Research Computing
Princeton University
Princeton, NJ 08544
sthais@princeton.edu

Abstract

Deep learning tools are being used extensively in a range of scientific domains; in particular, there has been a steady increase in the number of geometric deep learning solutions proposed to a variety of problems involving structured or relational scientific data. In this work, we report on the performance of graph segmentation methods for two scientific datasets from different fields. Based on observations, we were able to characterize the individual impact each type of graph segmentation methods has on the dataset and how they can be used as a precursors to deep learning pipelines.

1 Introduction

Machine learning methods are increasingly employed in a range of scientific domains to complement or even replace traditional statistical or data-driven solutions (1). Many physical science datasets such as those from particle physics (2), physical chemistry (3), material science (4), and others, have inherent geometric or relational structures that can be well represented by graphs. This has accelerated the development and adaptation of graph-based deep learning solutions like Graph Neural Networks (GNNs) (5), Graph Convolutional Networks (GCNs) (6), and Graph Attention Transformers (GATs) (7), among others, in scientific research.

A common task in many of these datasets is clustering the raw data based on certain attributes in order to obtain relevant contextual information or identify underlying geometric structures. In this paper we study two such tasks: mapping the trajectories of charged particles at the Large Hadron Collider (LHC) (2) and identifying ligand sites in proteins (8). We explore graph segmentation as a precursor to a larger deep learning pipeline. By understanding more about the underlying geometric structure of the data and, when possible, separating the input data into distinct sub-graphs, we are able to simplify the tasks of downstream graph-based deep learning architectures (9) and parallelize inference in constrained computing environments (10).

Graph Segmentation Specifically, we assume $G = (V, E)$ to be an undirected graph with vertices $v_i \in V$ and edges $(v_i, v_j) \in E$ corresponding to pairs of neighbouring vertices. Vertex-based graph segmentation involves the construction of a segmentation S which is a partition of V into n components or regions, $C_1...C_n \in S$ such that each region corresponds to a connected component in a graph $G' = (V, E')$ where $E' \subseteq E$. In other words, any segmentation is induced by a subset of edges in E. A good segmentation would ensure that elements in the same component would be similar to each other and dissimilar from elements in different components.

2 Datasets

TrackML Track ML (2) is a dataset created by CERN to address new and developing software needs for real-time preprocessing and filtering of data produced by proton-proton collisions at the

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

LHC. The dataset consists 8850 simulated events (proton-protong collisions) interacting with a realistic detector (11). Each event can be defined as an unordered set of particle measurements (*hits*) within the detector; each *hit* is a 3D measurement in Cartesian coordinates (x, y, z) of the location where a particle interacted with the detector. All events are statistically independent and contain directional information, unique identifiers for the true hit position and unique particle identifier as well as representations of the final state of the particle.

The geometric learning task for the TrackML dataset is to identify groups of hits that correspond to the trajectories (tracks) of individual charged particles produced from the collision. By implementing graph segmentation methods we can begin to separate subgraphs that correspond to individual tracks or groups of tracks that can be used as inputs to a more precise track finding algorithm.

sc-PDB The sc-PDB database (8) is a comprehensive collection of over seventeen thousand ligandable binding sites available in the Protein Data Bank (PDB). The database provides an all-atom description of a protein, its ligands, binding sites and modes of binding. The steady development of sc-PDB over the last 15 years has offered a wealth of information to assist in a variety of fields like computer-aided drug discovery applications.

Binding sites in sc-PDB can be represented as 3D shapes of cavities generated using VolSite (12). A binding cavity is described by VolSite using a set of pharmacological properties laid out in a 3D grid, based on the properties of the neighbouring proteins. For the purposes of this paper, the data was split into 10 groups, based on the UniProt ID. This ensured that structures of a single protein were present in the same group. VolSite was also used to describe a cavity structure for each ligand (apo structures were aligned to their holo counterparts) and the ligands were used to select pockets (orthosteric ligand binding-sites).

3 Methodology

3.1 DBSCAN

Density-based clustering (13) provides a general and rigorous probabilistic framework in which the clustering task is well-defined and amenable to statistical analysis. The Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (14) is an extremely popular methodology for *flat* clustering. Based on a set of points, DBSCAN groups together points that are close to each other based on a distance measurement and a minimum number of points.

(14) defines a simpler version of the DBSCAN algorithm where k is fixed. In (14), the parameters defined as h and k are known as Eps and MinPts which define the neighbourhood radius and the number of points required to seed a cluster respectively.

3.2 Spectral Clustering

Spectral Clustering (15) is a technique with roots in graph theory, where the primary approach is to identify communities in nodes of the graph based on the edges connecting them. It usually outperforms traditional clustering algorithms such as the k-means algorithm (16) and is also solved efficiently by most linear algebra software applications.

We assume that our data consists of n arbitrary 'points', $x_1 ... x_n$. We leverage a non-negative and symmetric similarity function to measure pairwise similarities s_{ij} and construct a similarity matrix S. S can be mathematically defined as: $S = (s_{ij})_{i,j=1...n}$. This gives rise to a (n, n)-sized matrix. For the purposes of our experiments, we use the method proposed by (17), where the entry (i, j) is the Euclidean distance between i and j. However, we also leverage the Eigengap Heuristic approach to determine the number of clusters the data should be partitioned into.

Eigengap Heuristic There usually exists a well-defined criteria to choose the number of clusters in a model based setting. This might be based on the log-likelihood of the data which can be treated in a Bayesian way (18). The Eigengap Heuristic is a procedure designed specifically for Spectral Clustering. Here, the goal is to choose k such that all the eigenvalues $\lambda_1, ... \lambda_k$ are small but λ_{k+1} is relatively larger. An explanation for this can be provided by spectral graph theory, where many geometric invariants of the graph can be expressed with the help of the first eigenvalues of the graph Laplacian. The sizes of the cuts are related closely to the size of the first eigenvalues.

3.3 Dynamic KNNs

k-Nearest Neighbours (kNNs) has been widely been used in areas like data mining due to its simplicity, effectiveness and robustness. It is an instance-based learning algorithm and a typical example of lazy learning; since it stores training instances to delay learning till classification. The classification accuracy of a kNN clustering approach is sensitive to the value of k. A simple method to increase this classification accuracy would be to learn the best k-value amongst different measurements of k. Dynamic kNN (DkNN) (19) works on the basis of this principle to choose an optimal value of k in the training datasets dynamically. DkNN uses an individual observation from the original sample for validation purposes and treats the remaining observations as training data. This is recursively repeated in a way such that each observation from the original sample is used for validation once.

3.4 Gaussian Mixture Models

Gaussian Mixture Models (20) are probabilistic models that use a soft-clustering approach to distribute data points into different clusters. The objective function of a GMM is to maximize the likelihood value of data X, defined as $p(X)$. By assuming a mixture of G Gaussians, $p(X)$ can be formulated as the marginalized property, summed up over all G clusters. Mathematically: $p(X_i) = \sum_{g=1}^{G} p(X_i|c_g)p(c_g)$.

Gaussian Mixture Models work on a two-step algorithm called Expectation-Maximization (EM). When given the number of clusters, the EM algorithm tries to figure out the parameters of these Gaussian distribution in two distinct steps: The Estimation Step guesses the parameters of the method based on available data while the Maximization step updates cluster parameters based on results from the Estimation step. This is repeated until convergence is reached. To calculate the number of clusters, we make use of Bayesian Gaussian Mixture Models, which allow us to infer an approximate posterior distribution over the parameters of a Gaussian Mixture Distribution.

4 Experiments and Results

TrackML The graph segmentation algorithms we discuss in this paper are responsible for the graph construction in a a full ML-based track reconstruction pipeline. Events present in TrackML are converted into hitgraphs and individual hits are assigned to graph nodes after conditional filtering. The algorithms cluster these hit nodes and decide whether an edge e_{ij} should be extended between hits x_i and x_j. We present 4 different quantitative measurements taken for TrackML, measured over 6 different values of minimum particle momentum, p_T^{min}. We measured Truth Efficiency; the fraction of true track segments represented as graph edges, Edge Efficiency; the ratio of true edges to the total edges in the graph, and the Number of Nodes and Edges for each method. A high Truth Efficiency is necessary to ensure all tracks can be reconstructed while a high Edge Efficiency means fewer false edges must be removed by a downstream edge-classifying network. The number of Nodes and Edges describe the size of the graphs and the potential for these methods to be used in resource constrained computing environments like those at the LHC.

From Table 1, we see that GMMs generally achieve high combined Truth Efficiency and Edge Efficiency across the full range of p_T^{min}. High Truth Efficiency indicates that hits belonging to the same track are likely to be clustered into the same component, while high Edge Efficiency indicates that distinct tracks are likely to be separated into different components. With these partially segmented graphs we can separate the raw input into distinct subgraphs. This can help accelerate graph-based deep learning solutions for particle tracking which are being run on distributed architectures (10).

sc-PDB Binding sites for proteins in the sc-PDB were generated using the SiteHopper create tool. The default *SiteHopper PatchScore* represents a summation of Tanimoto similarity coefficients (21) weighted 3:1 in favor of color similarity over shape similarity. SiteHopper made use of the fpocket [1] surface protein atoms as pseudo-ligands and was able to isolate binding site patches from the original protein structures. fpocket is a freely available binding site detection tool capable of operating in large molecular and genomics datasets. Each graph segmentation method, when applied to the sc-PDB dataset generates a map with potential binding sites clustered together. The clusters vary

[1] fpocket: https://bioserv.rpbs.univ-paris-diderot.fr/services/fpocket/

Method	p_T^{min}	Truth Efficiency	Edge Efficiency	Number of Nodes	Number of Edges
DBSCAN	0.5	0.972 ± 0.03	$0.046 \pm$ *0.00*	1.41×10^5	2.35×10^6
	0.6	0.974 ± 0.05	$0.051 \pm$ *0.00*	1.02×10^5	1.80×10^6
	0.75	0.979 ± 0.09	0.091 ± 0.01	8.68×10^4	1.17×10^6
	1.0	0.981 ± 0.10	0.14 ± 0.03	5.93×10^4	7.99×10^5
	1.5	0.983 ± 0.14	0.21 ± 0.05	2.19×10^4	9.27×10^4
	2.0	0.982 ± 0.15	0.25 ± 0.09	1.02×10^4	5.09×10^4
Spectral Clustering	0.5	0.968 ± 0.003	0.042 ± 0.01	1.59×10^5	2.382×10^5
	0.6	0.972 ± 0.004	0.108 ± 0.03	8.268×10^4	9.358×10^4
	0.75	0.979 ± 0.004	0.108 ± 0.03	7.97×10^4	9.27×10^4
	1.0	0.981 ± 0.005	0.180 ± 0.13	6.351×10^3	7.591×10^4
	1.5	0.981 ± 0.006	0.399 ± 0.17	3.742×10^4	4.260×10^4
	2.0	0.981 ± 0.008	0.719 ± 0.15	1.924×10^4	1.834×10^4
Dynamic kNN	0.5	0.972 ± 0.003	$0.028 \pm$ *0.00*	1.621×10^5	3.209×10^5
	0.6	0.974 ± 0.003	$0.074 \pm$ *0.00*	8.491×10^4	5.372×10^5
	0.75	0.977 ± 0.005	$0.081 \pm$ *0.00*	8.699×10^4	9.973×10^4
	1.0	0.979 ± 0.005	0.119 ± 0.01	7.372×10^4	6.138×10^4
	1.5	0.983 ± 0.008	0.253 ± 0.03	4.206×10^4	4.528×10^4
	2.0	0.984 ± 0.010	0.375 ± 0.04	2.519×10^4	1.972×10^4
GMM	0.5	0.966 ± 0.002	$0.115 \pm$ *0.00*	8.491×10^4	5.372×10^5
	0.6	0.974 ± 0.003	0.151 ± 0.01	7.916×10^4	4.684×10^5
	0.75	0.979 ± 0.006	0.207 ± 0.02	6.582×10^4	2.461×10^5
	1.0	0.982 ± 0.006	0.207 ± 0.03	5.218×10^4	9.207×10^4
	1.5	0.983 ± 0.008	0.361 ± 0.05	2.625×10^4	6.948×10^4
	2.0	0.982 ± 0.010	0.452 ± 0.05	9.350×10^3	9.958×10^3

Table 1: Quantitative Evaluation of Track Finding Efficiencies with different Graph Segmentation Techniques

in size, indicating that some of the cavities are more conservative than others. For all proteins, the largest cluster corresponds to the conserved, orthosteric ligand-binding site. The PatchScore can be considered a quantitative evaluation of these clusters. Each value in the Table 2 is associated with the maximum PatchScore exhibited by the members of the two binding site clusters.

One major disadvantage introduced by cavity detection is that it introduces noise due to non-conserved potential binding sites. Table 2 shows that Dynamic kNNs display higher values compared to other graph segmentation methods while clustering. This suggests that clustering methods like DkNNs can be considered an appropriate method for the generation of clustered binding sites as well as mitigating the noise introduced to the dataset by the fpocket cavity detection.

Method	Protein	Cationic Trypsin	BRD-4	CDK2	Estrogen Receptor	HIV-1 Protease	Prothrombin
DBSCAN	Cationic Trypsin	-	0.04	0.07	0.03	0.03	0.08
	BRD-4	0.03	-	0.08	0.11	0.05	0.13
	CDK2	0.09	0.13	-	0.06	0.11	0.18
	Estrogen Receptor	0.08	0.18	0.04	-	0.12	0.08
	HIV-1 Protease	0.17	0.01	0.19	0.18	-	0.31
	Prothrombin	0.16	0.05	0.24	0.11	0.09	-
Spectral Clustering	Cationic Trypsin	-	0.06	0.17	0.13	0.08	0.20
	BRD-4	0.15	-	0.18	0.09	0.12	0.06
	CDK2	0.14	0.15	-	0.07	0.14	0.05
	Estrogen Receptor	0.02	0.18	0.16	-	0.03	0.06
	HIV-1 Protease	0.15	0.02	0.08	0.07	-	0.16
	Prothrombin	0.11	0.11	0.13	0.22	0.05	-
Dynamic kNN	Cationic Trypsin	-	0.04	0.15	0.11	0.27	0.19
	BRD-4	0.19	-	0.35	0.14	0.08	0.13
	CDK2	0.16	0.19	-	0.15	0.19	0.22
	Estrogen Receptor	0.10	0.18	0.22	-	0.18	0.09
	HIV-1 Protease	0.15	0.15	0.21	0.17	-	0.24
	Prothrombin	0.19	0.28	0.31	0.23	0.17	-
GMM	Cationic Trypsin	-	0.04	0.10	0.08	0.16	0.09
	BRD-4	0.07	-	0.17	0.14	0.11	0.03
	CDK2	0.01	0.08	-	0.04	0.13	0.19
	Estrogen Receptor	0.11	0.18	0.09	-	0.11	0.07
	HIV-1 Protease	0.05	0.11	0.17	0.06	-	0.02
	Prothrombin	0.18	0.24	0.07	0.04	0.09	-

Table 2: A quantitative evaluation of clustering and mapping of potential binding sites between different proteins in the sc-PDB dataset.

In order to better understand the results displayed in Table 1 and Table 2, we also show the effectiveness of each graph clustering approach in comparison to the others. We define two values, $e_{TrackML}$ and e_{sc-PDB} to measure how well each clustering approach works on either dataset. $e_{TrackML}$ can be quantitatively defined as the ratio of segmented subgraphs to the total number of tracks in the event, which is equal to the number of particles produced. e_{sc-PDB} is the ratio of the total number of

Dataset	Method	$e_{TrackML} \uparrow$	$e_{sc-PDB} \uparrow$	$\chi_{TrackML} \uparrow$	$\chi_{sc-PDB} \uparrow$
DBSCAN	TrackML	0.579	-	0.7424	-
	sc-PDB	-	0.481	-	0.2863
Spectral Clustering	TrackML	0.602	-	0.5968	-
	sc-PDB	-	0.517	-	0.4262
Dynamic kNN	TrackML	0.513	-	0.5079	-
	sc-PDB	-	0.594	-	0.5038
GMM	TrackML	0.735	-	0.8194	-
	sc-PDB	-	0.408	-	0.3920

Table 3: Effectiveness of each graph segmentation method on both datasets

clustered binding sites to the total isolated binding sites identified by fpocket. From Table 3 we notice that Spectral Clustering and GMM perform well on TrackML while DkNN is the most effective and clustering potential binding sites in sc-PDB.

To get a deeper understanding of two e efficiency variables, we also look at how useful each segmentation measure is for downstream tasks. This is quantitatively measured by defining two variables, $\chi_{TrackML}$ which is the ratio of the number of clusters containing more than 50% particles from the same track and the total number of tracks. We also define χ_{sc-PDB} as the number of observed clusters which actually correspond to a distinct ligand binding site and the total number of observed clusters. We see that GMMs perform best on the TrackML dataset and is able to separate over 80% of tracks into distinct subgraphs. Dynamic kNN performs best on the sc-PDB dataset, however only 50% of segmented subgraphs correspond to true individual ligand binding sites.

5 Conclusion and Future Work

We take a look at how different types of graph segmentation approaches work on scientific datasets and how they could be used as a precursor for deep learning pipelines with graph-based data. We conduct comprehensive evaluations over two scientific datasets used in separate fields and show how graph segmentation would be able to point towards factors that would inevitably help speed-up or improve the accuracy of the overall pipeline it is fitted into. From these initial studies we demonstrate that relatively graph segmentation approaches can effectively identify and separate unique graph components. However, the most effective algorithm for graph segmentation depends on the specific dataset of interest. In future work we would like to examine other datasets in the particle physics and proteins and expand to other domains in physical sciences to understand if these conclusions are dataset specific or generalizable across datasets in a particular subfield.

6 Impact Statement/Ethical Considerations

This work may be used for a variety of purposes in different fields. Based on the datasets included in this paper, this work has the potential to be used for improve the efficiency of particle tracking at the LHC. It may also be used as a starting point for obtaining more information about newly discovered protein structures. The results of these tasks would yield many benefits to science and society. This work could also reasonably be used to improve processing of other structured scientific dataset and enable a number of additional learning tasks. We encourage any reseachers applying these methods to new dataset carefully design a quantitative measurement of accurate segmentation for the individual dataset.

We are also aware that this work exists in the broader context of computer vision and geometric machine learning. This can be exploited for many negative purposes such as surveillance and targeting. We are committed to continuing to learn more about the broader use of machine learning and support our communities in advocating against harmful uses.

Acknowledgments and Disclosure of Funding

This work is supported by IRIS-HEP through the U.S. National Science Foundation (NSF) under Cooperative Agreement OAC-1836650.

References

[1] A. Tsaris, D. Anderson, J. Bendavid, P. Calafiura, G. Cerati, J. Esseiva, S. Farrell, L. Gray, K. Kapoor, J. Kowalkowski, et al., in *Journal of Physics: Conference Series*, vol. 1085 (IOP Publishing, 2018), vol. 1085, p. 042023

[2] S. Amrouche, L. Basara, P. Calafiura, V. Estrade, S. Farrell, D.R. Ferreira, L. Finnie, N. Finnie, C. Germain, V.V. Gligorov, et al., The Springer Series on Challenges in Machine Learning p. 231–264 (2019). DOI 10.1007/978-3-030-29135-8_9. URL `http://dx.doi.org/10.1007/978-3-030-29135-8_9`

[3] P. Mishra, P.N. Pandey, Bioinformation **6**(10), 372 (2011)

[4] V. Srivastava, B. Biswas, in *International Conference on Advanced Informatics for Computing Research* (Springer, 2018), pp. 193–203

[5] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2021). DOI 10.1109/tnnls.2020.2978386. URL `http://dx.doi.org/10.1109/TNNLS.2020.2978386`

[6] T.N. Kipf, M. Welling. Semi-supervised classification with graph convolutional networks (2017)

[7] W.U. Ahmad, N. Peng, K.W. Chang. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction (2021)

[8] E. Kellenberger, P. Muller, C. Schalon, G. Bret, N. Foata, D. Rognan, Journal of chemical information and modeling **46**(2), 717 (2006)

[9] G. DeZoort, S. Thais, I. Ojalvo, P. Elmer, V. Razavimaleki, J. Duarte, M. Atkinson, M. Neubauer, arXiv preprint arXiv:2103.16701 (2021)

[10] A. Heintz, V. Razavimaleki, J. Duarte, G. DeZoort, I. Ojalvo, S. Thais, M. Atkinson, M. Neubauer, L. Gray, S. Jindariani, et al., arXiv preprint arXiv:2012.01563 (2020)

[11] C. Gumpert, A. Salzburger, M. Kiehn, J. Hrdinka, N. Calace, A. Collaboration, et al., in *Journal of Physics: Conference Series*, vol. 898 (IOP Publishing, 2017), vol. 898, p. 042011

[12] J. Desaphy, K. Azdimousa, E. Kellenberger, D. Rognan. Comparison and druggability prediction of protein–ligand binding sites from pharmacophore-annotated cavity shapes (2012)

[13] J.A. Hartigan, Journal of the American Statistical Association **76**(374), 388 (1981)

[14] M. Ester, H.P. Kriegel, J. Sander, X. Xu, et al., in *kdd*, vol. 96 (1996), vol. 96, pp. 226–231

[15] J. Liu, J. Han, in *Data Clustering* (Chapman and Hall/CRC, 2018), pp. 177–200

[16] K. Alsabti, S. Ranka, V. Singh, (1997)

[17] A.Y. Ng, M.I. Jordan, Y. Weiss, in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic* (MIT Press, Cambridge, MA, USA, 2001), NIPS'01, p. 849–856

[18] C. Fraley, A.E. Raftery, Journal of the American Statistical Association **97**(458), 611 (2002). URL `http://www.jstor.org/stable/3085676`

[19] L. Jiang, Z. Cai, D. Wang, S. Jiang, in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, vol. 1 (2007), vol. 1, pp. 679–683. DOI 10.1109/FSKD.2007.552

[20] X. Chen, A.Y. Zhang. Optimal clustering in anisotropic gaussian mixture models (2021)

[21] J. Batista, P.C. Hawkins, R. Tolbert, M.T. Geballe, Journal of Cheminformatics **6**(1), 1 (2014)

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6: Impact Statement and Ethical Considerations
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Code will be released soon
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

For the sake of brevity, the following tables and figures were not included in the main paper. However, we feel that they may be vital to some readers in order to get a deeper understanding and a better overall view of the topics discussed in the paper.

We take a look at the hitgraph generated by Spectral Clustering for an event in the $\eta - \phi$ space.

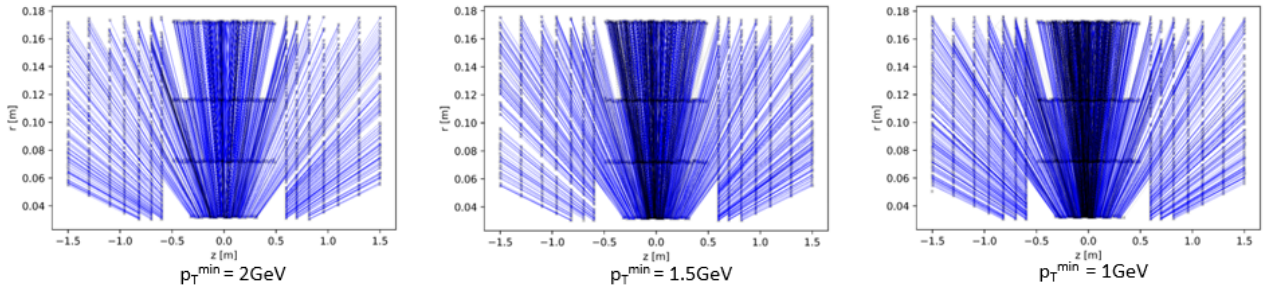

Figure 1: Progressively increasing number of false edges with a decrease in p_T^{min} in the $\eta - \phi$ space

Table 4 lists out the UniProt ID, among other features, of the proteins that were selected for our experiments. Here, *n* defines the number of predicted binding sites according to fpocket.

Protein	N	*UniProt ID*	n
Cationic Trypsin	315	P00918	968
Bromodomain-Containing Protein 4 (BRD4)	93	O60885	192
Cyclin-Dependent Kinase 2 (CDK2)	148	P24941	490
Estrogen Receptor (ER)	52	P03372	241
Human Immunodeficiency Virus-1 (HIV-1) Protease	335	N/A*	481
Prothrombin	142	P00734	336

Table 4: A list of proteins studied with their *UniProtID*
*HIV-1 Protease entries were defined by a 90% sequence similarity search using the consensus B protease sequence retrieved from HIVdb.

We also list out the hyperparameters that were used for the graph segmentation algorithms that we discussed in the paper. For DBSCAN, we refer to the implementation done by (9) and accordingly alter the value of ϵ and MinPts according to the value of p_T^{min}. We also look at *n* for Gaussian Mixture Models. It defines the number of initializationsto perform, and the result with the highest lower value bond on the likelihood is used. All experiments were run on on an NVIDIA GTX1080Ti.

Method	Dataset	p_T^{min}	ϵ	MinPts	*n*
DBSCAN	TrackML	0.5	0.22	3	-
		0.6	0.18	3	-
		0.75	0.1	3	-
		1.0	0.08	3	-
		1.5	0.06	3	-
		2.0	0.05	3	-
DBSCAN	sc-PDB	-	7	10	-
GMM	TrackML	0.5	-	-	3
		0.6	-	-	3
		0.75	-	-	3
		1.0	-	-	3
		1.5	-	-	3
		2.0	-	-	3
GMM	sc-PDB	-	-	-	9

Table 5: Hyperparameters