

Particle Trajectory Visualization and Identification with Timepix Detectors

Summer 2016

Ameir Shaa¹, Stephanie Baines² and Jessica Flores³

¹ Department of Physics and Applied Physics, Nanyang Technological University, Singapore

² School of Physics and Astronomy, The University of Manchester, United Kingdom

³ Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY, USA

The TimePix detector is a largely passive detector deployed at point 8 on the Large Hadron Collider (LHC) ring. It is able to detect different kinds of radiation in real time. Energy deposits in the detector is registered as signal describing the spacial trajectories of passing particles. The signal data is analysed and classified in order to understand the types of particle interactions occurring and identify anomalous occurrences. Using topological features if trajectories described by the data and one or more related features of said topology, the data was categorized using an algorithm to an efficiency of 85%.

Contents

1	Abstract	2
2	Introduction	2
3	The Timepix detector	3
4	Cluster Classification	4
4.1	Clusters and pixels	4
4.2	Cluster variables	4
4.2.1	Size, aspect ratios, counts, inner/edge pixels	4
4.2.2	Center of cluster	4
4.2.3	Radius and Density	5
4.2.4	Linearity	5
4.2.5	Horizontal/Vertical neighbouring pixel distance	5
4.2.6	Evaluation of the Goodness of fit	6
4.3	Cluster Types	6
4.4	Cluster Sorting Algorithm	7
5	Results	9
6	Conclusion	10

1 Abstract

The TimePix detector is a largely passive detector deployed at point 8 on the Large Hadron Collider (LHC) ring. It is able to detect different kinds of radiation in real time. Energy deposits in the detector is registered as signal describing the spacial trajectories of passing particles. The signal data is analysed and classified in order to understand the types of particle interactions occurring and identify anomalous occurrences. Using topological features if trajectories described by the data and one or more related features of said topology, the data was categorized using an algorithm to an efficiency of 85%.

2 Introduction

The Monopole and Exotics Detector at the LHC (MoEDAL) was initialized in 2010 after being approved by the European Centre for Particle Physics Research (CERN) research board. MoEDAL is the seventh international experiment at the Large Hadron Collider (LHC) located in front of the LHC beauty (LHCb) experiment. It is designed to search for manifestations of new physics signified
5 by highly ionizing particles (HIPs). It seeks to provide answers to the fundamental questions of particle physics including questions surrounding the existence of monopoles, dark matter and extra dimensions. It is a key player in the search for new fundamental symmetries of nature (SuperSymmetry for example), alternative mechanisms of mass generation and understanding how the big bang unravelled in the early universe. MoEDAL is dedicated to the detection of HIPs that could have
10 originated from scenarios other than what is known in the Standard Model. In essence, this detector aims to trap these long-lived HIPs and take photographic evidence of the existence of such HIPs. [1]

The MoEDAL detector, deployed at point 8 on the LHC ring together with the vertex locator (VELO) detector, comprises mainly of two sections: the plastic Nuclear Track Detector (NTD)
15 stacks of approximately 280 m² and the paramagnetic trapping volumes which captures both electrically and magnetically charged HIPs. Armed with these detection devices, new particles can be captured, studied and magnetic charge and be directly measured. The ability to capture these particles also raises the possibility of freeing such particles for further study.

The Timepix detector in the MoEDAL experiment at the LHC measures the radiation deposited
20 by passing particles. This section of the detector relies on the response of silicon based semiconductor detector chips to track particle events, and the passage of a charged particle is observed as a signal through a cluster of adjacent semiconductor chips. The visual interpretation of these signals are dark tracks on a blue canvas representing an array of semiconductor chips as shown in Fig. [1].
25 Each type of particle has a distinct trajectory in the detector depending on its mass and charge, and hence on its unique interaction with external electromagnetic fields.

Previous works [2] [3] have sought to classify the particles and interactions that appear in the Timepix detector using the shapes of clusters of activated semiconductor chips. These clusters are
30 described in detail in this report. The classification of such clusters was performed by researchers and in Langton Star Centre as part of CERN@school, in an activity similar to the Juggernaut project of the LHCb collaboration, which uses the Zooniverse's Galaxy Zoo technology to identify radiation tracks in the Timepix detector. In the wake of this research, this report offers insight into a computerised method of cluster classification, developed for fast identification and processing
35 of large data sets into six main categories. This process is intended to precede the distribution of clusters to participants in a Zooniverse type project for meticulous identification for the sake of the MoEDAL experiments research, and improved student learning within the CERN@school framework.

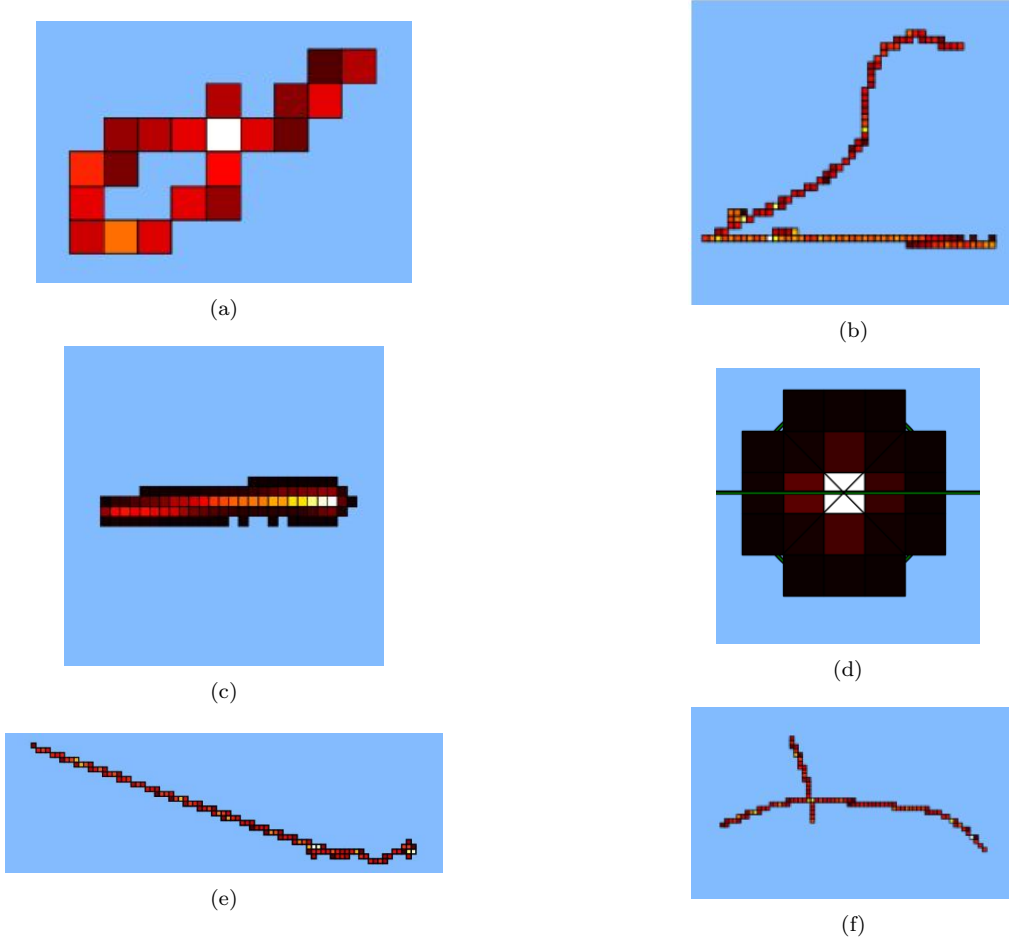


Figure 1: Examples of (a) loopers, (b) brancher, (c) slug, (d) boxy, (e) SW, (f) crossover clusters .

3 The Timepix detector

The background radiation in the MoEDAL/VELO cavern is measured by the TimePix detector. The TimePix detector uses 300 μm wafer of n -type silicon as a sensitive element. The readout electronics are incorporated in the form of an application specific integrated circuit (ASIC) onto which the sensor is bump-bonded. There are three different modes on the readout electronics: Medipix mode for photon counting, TimePix mode recording the time of arrival of photons in the detector medium, and the time-over-threshold (ToT) mode. The detector 'visualizes' radiation using the ToT mode. Each silicon chip in the detector stores the length of time the charge measured by that sensitive area spends above the threshold value for detector inactivity. In return, each activated chip will return an analogue to digital count (ADC) associated with its x and y position where 1 count represents $0.1\mu\text{s}$. The count value is dependent on the energy deposited in the sensor per unit length, $\frac{dE}{dx}$, also known as Linear Energy Transfer or LET, the distance of the sensor interaction from the pixel readout pad and the detector settings. Each parcel of energy transferred activates a certain region of the sensor and the amount of energy transferred differs depending on the nature of the particle transferring this energy. For example an electron will activate a different set of pixels than an alpha particle. A group of activated sensor chips adjacent to each other are known as clusters, and there is a systematic approach to classifying the clusters to identify which particle

caused those clusters. The topological cluster variables for classifying these include size (number of pixels in the cluster), centre (mean of the pixels position), radius (distance from the centre to the pixel furthest away), density (number of pixels per unit area), linearity and inner pixels (the pixels that are completely surrounded by other pixels) [2].

In ToT mode, the output from the Timepix identifies a set of spacial coordinates on a 2D plane where energy is deposited, (x,y) , and a count value C attached to each set of coordinates representing the intensity of energy transferred to the detector at that point. These sets of values can then be developed into an images, or frames. Each frame has several regions of interest corresponding to the x , y , C coordinates in the data sets. These regions are isolated and analysed as individual clusters.

4 Cluster Classification

4.1 Clusters and pixels

Clusters define distributions of energy deposited in the TimePix detector, summarized as an array of pixels with spacial coordinates and count values. Throughout this report, the term pixel will refer to such a member of the cluster's array, and hence the presence of energy registered by the detector.

These cluster images can be represented by matrices A_{ik} in which the indices represent the spacial coordinates, and each (i, k) is associated with a count or pixel value.

4.2 Cluster variables

To differentiate between clusters, some topological cluster variables were extracted from existing CERN@School cluster analysis code [5]. These topological cluster variables [2] included size, counts (pixel value), number of inner and edge pixels, aspect ratios, centre of cluster, radius, density, linearity, horizontal/vertical neighbouring pixel distance, and chi-squared value.

4.2.1 Size, aspect ratios, counts, inner/edge pixels

The size n_j of a cluster refers to the total number of pixels it represents [2]. To determine the aspect ratio of a cluster, its width and height, defined as the maximum number of pixels that make up the cluster in the horizontal and vertical directions, are used. The aspect ratio is evaluated from Eq. 1.

$$aspect\ ratio = \frac{width}{height}. \quad (1)$$

The closer a cluster is to having a boxed shape, the more the aspect ratio tends to unity.

The inner pixel size Θ_j refers to the number of pixels that are contained within a cluster. In other words, it is the number of pixels in the cluster that are completely surrounded by other pixels [2]. Edge pixels are defined as the number of pixels that are contained on the outskirts of a cluster or not completely surrounded by other pixels.

4.2.2 Center of cluster

The centre of the cluster $(\bar{x}_j, \bar{y}_j)$ is the unweighted mean of the cluster's positions (x_k, y_k) [2]:

$$(\bar{x}_j, \bar{y}_j) = \frac{1}{n_j} (\sum x_k, \sum y_k). \quad (2)$$

95 It is unweighted because the pixel count values are ignored.

4.2.3 Radius and Density

The cluster radius r_j is evaluated using the unweighted cluster centre. It is the radius of the circle that is centered at $(\bar{x}_j, \bar{y}_j)$ that encloses all of the cluster's pixels [2]. Determining the radius of the cluster provides a rotationally invariant way of looking at the spatial extent of the cluster.

100 Using the radius given above, the cluster density ρ_j is defined as the number of pixels n_j per unit area assuming a circular cluster [2]. This is given in Eq. 3.

$$\rho_j = \frac{n_j}{\pi r_j^2} \quad (3)$$

From Eq. 3, circular clusters will have a density tending towards unity so that $n_j \propto r_j^2$. This variable may be useful for identifying heavy, ionizing particles.

105 4.2.4 Linearity

For long and straight clusters, the spacial coordinates of the cluster distribution can be considered as a linear correlation between the x and y coordinates such that the line of best fit for the pixels can be determined. The sum of residuals R_k is given in Eq. 4.

$$\sum_k R_k^2 = \sum_k \frac{|y_k - m_j x_k - c_j|}{1 + m_j^2}, \quad (4)$$

where m_j is the gradient and C_j is the intercept of the line of best fit for cluster j [2]. The linearity L_j for cluster j is then average residual for the cluster,

$$L_j = \frac{\sum_k R_k}{n_j}. \quad (5)$$

4.2.5 Horizontal/Vertical neighbouring pixel distance

The horizontal and vertical neighbouring pixel distance were also important variables for differentiating clusters from one another. Some clusters did not show a singular "one-to-one" correspondence between x and y spacial coordinates. In these cases, multiple pixels in the y -direction corresponded to a single x coordinate. The distance between these y coordinates was determined. 115 If these y coordinates were separated by a region of inactive pixels, then it was conceivable that the cluster was formed of two distinct branches at these points.

$$\begin{aligned} \text{Pixel 1 : } (x, y_1), \text{ Pixel 2 : } (x, y_2) \\ \text{Distance : } \Delta y = |y_2 - y_1| \end{aligned} \quad (6)$$

The same was done for cluster images under a 90° rotation.

$$\begin{aligned} \text{Pixel 1 : } (x_1, y), \text{ Pixel 2 : } (x_2, y) \\ \text{Distance : } \Delta x = |x_2 - x_1| \end{aligned} \quad (7)$$

Both Δx and Δy were useful for classifying branching and looping clusters (Fig. 1; see section 4.3 for more information on cluster types).

4.2.6 Evaluation of the Goodness of fit

The equation for the straight lines of clusters was determined as the line of best fit evaluated from the spacial coordinated of pixels in the (x, y) plane, referred to here as pixel data. In the interest of isolating highly linear clusters, the goodness of fit of the pixel data was calculated using the standard equation for the chi-squared reduced, $\chi^2 - red$. This was given by

$$\chi^2 = \sum_{i=1}^N \frac{[Y_o^i - Y_e^i]^2}{Y_e^i} \quad [4] \quad (8)$$

$$\chi^2 - red = \frac{\chi^2}{N - dof}. \quad (9)$$

The expected value of the y coordinates, Y_e^i , is subtracted from the observed values Y_o^i and the $\chi^2 - red$ is the therefore the weighted square sum of the these divided by the number of degrees of freedom. If the data fit this trend well, $\chi^2 - red < 1$ and this condition was therefore used to isolate truly straight trends in the pixel data.

Furthermore, the goodness of fit was expected to hold for sections of the pixel data if the trend was consistently linear at each point. The $\chi^2 - red < 1$ was also evaluated iteratively through the pixel data using progressively larger fractions of this data. A rapid increase in $\chi^2 - red < 1$ was expected if the pixel data diverged at one end of the spacial spectrum indicating the presence of a straight line that became wiggly, defining a straight wiggly. This information was used to separated straight lines from straight wiggly lines.

4.3 Cluster Types

There are 6 main types of clusters: straight-wiggles, branchers, crossovers, slugs and loopers (fig. 1) [3].

1. **Looper** (fig. (1a)) : Tracks that loop over themselves are called loopers. These are characterized by a curved line with an opening in the shape of an annulus. This signal may tell of the decay of a heavier particle into a beta particle and its anti-particle. Since beta particles are electrically charged, they follow a curved trajectory and meet again to annihilate each other.
2. **Brancher** (fig. (1b)) : A brancher can best be identified by one or more lines branching off from the line of best fit. It may be caused by the decay of a heavier neutral particle into two neutral particles. The initial trajectory of the heavier particle would be more or less parallel to the line of best fit. In order to conserve momentum, the particles would have to travel in different directions altogether, inducing the branching off of the signal track.
3. **Slug** (fig. (1c)) : A slug refer to a large, dense region of pixels confined to a thick continuous sensor region that resembles the shape of a slug. This signal may be caused by a heavy ion fragment or nucleus for example.
4. **Boxy** (fig. (1d)) : A boxy is characterized by a dense region of pixels with a high energy in the centre. However, unlike the slug, the pixels of a typical boxy are arranged in the shape of a square. This could be whitness of the interaction of an alpha particle with the detector medium, leaving large deposits of charge in the sensor element, giving rise to the square-like clusters that we observe.
5. **Straight wiggly (SW)** (fig. (1e)) : Straight Wiggly lines are characterised by their "zig-zag" or wiggly deviation from the line of best fit towards the end of their trajectories, and are similar trajectories expected of fast moving particles. Initially, the particle has a high enough kinetic energy such that it is able to move through the lattice of the detector undeviated.

However, as it loses kinetic energy through Coulomb interactions, there is a higher probability of the particle being deflected causing the scattered trajectory.

6. **Crossover** (fig. (1f)) : Multiple linear tracks intercepting one another and the line of best fit are identified as crossovers. This is potentially caused by the scattering of two neutral particles as they collide head on, or the recording of two individual tracks crossing the same area of the detector at different times.

4.4 Cluster Sorting Algorithm

The JSON files, `pixels.json` and `klusters.json` with the relevant pixel data of the clusters were processed by the cluster sorting script implemented within `Python`. The relevant information was then stored in a text file and was submitted to the following analysis.

Initially, all clusters with a pixel count less than 19 were rejected as background deposits, `size>19`, uninteresting in the search for particle track in the TimePix detector. This followed from the fact that for all interesting clusters identified by eye, the smallest had a size of 20 and the rest was degraded as background.

Clusters were sorted into categories depending on their topological features. The sorting used existing and new quantities which quantitatively defined their topological distributions. Figure 2 shows the selection cuts we made, which are also explained below.

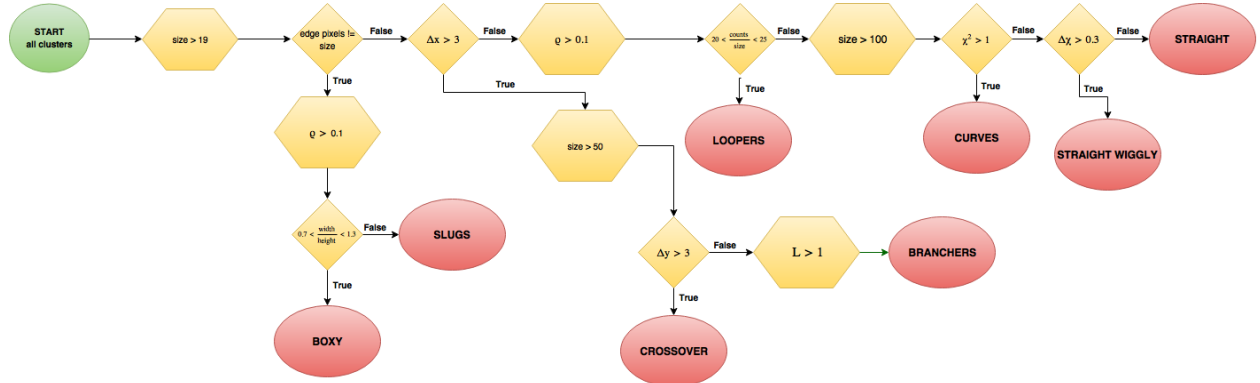


Figure 2: Cuts used to classify clusters.

Clusters were sorted into two large categories depending on whether these contained bulk pixels or not. When considering compact and linear distributions of pixels, the compact clusters were expected to contain at least one pixel entirely surrounded by other pixels. Hence, the shorter more compact boxys and slugs were isolated from the longer and thinner, or looping clusters realizing that the former pair would contain such bulk pixels, but the latter group not. A condition on the density of the clusters was also imposed to exclude background clusters. A cluster was considered slug or boxy like if the density satisfied `density>0.1`, where density is defined in Eq. 3.

From there, the slug clusters were separated from the boxy clusters by realizing that the aspect ratio of slugs was significantly different to 1, and that of boxys faithfully approached unity. Hence, slugs maintained a width to height ratio greater than 1.3 or less than 0.7.

The next task involved sorting the second group of clusters with lower densities into sets of straight or curved simple line, loopers which doubled over on themselves and the branching and crossing type known as branchers and crossovers. If the topological distribution of pixels was

graphed in the x-y plane, the branchers and crossovers were not described by single-valued continuous function. These could therefore be separated from the straight and curved clusters by requiring the presence of multiple pixels with the same horizontal x-coordinates in the plane having vertical y-coordinates spread by more than 3 pixels. This condition was defined by `delta_x(xytuple)` within the sorting algorithm's functions script `functions.py`.

At this point, it is noted that a more stringent condition on the size of clusters was imposed, `size>50`, in order to cancel background tracks. The crossovers and branchers were then separated using the following set of conditions which distinguished one set cluster class from the other. The crossovers were extracted by requiring that the condition defined by `delta_x(xytuple)>0` as described above, but applied to the pixel frame rotated by 90 degrees. This was defined as `delta_y(xytuple)>0` in the Python script. Indeed, it was expected that the crossover pixel distributions would remain non-singular functions under rotation of the pixel coordinate frame. The branchers on the other hand could be defined such that the function was singular in the rotated frame. The reader is reminded that the condition for singularity was defined such that the function was single valued to within a margin of 3 pixels. The details on this cut are given in section 4.2.5.

The linearity of the pixel distribution was used to further refine the isolation of the brancher clusters avoiding the addition of surviving linear clusters. This quantity defined how well the pixel distribution fit a straight line as described by Eq. 5. Hence, branchers were required to have a poor linearity, `linearity>1`.

The remaining clusters not satisfying the condition `delta_x(xytuple)>0` were defined topologically by either straight or curved singular functions, or functions which looped over themselves referred to as loopers. Loopers clearly showed a higher density than straight or curved clusters. Hence a second density condition was imposed selecting a looper for `density>0.1`. A constraint on the average intensity of energy deposited per pixel was imposed to refine the selection of loopers over background clusters.

Finally, the straight wiggly, straight and curves were separated from the final set using the goodness of fit of all the pixels in each cluster to a linear trend, represented by the value of $\chi^2 - red$ given in Eq. 8. A good fit was defined for $\chi^2 - red < 1$, isolating the straight singular clusters from the curved singular clusters. Straight lines were expected to have a negligible $\chi^2 - red < 1$.

The $\chi^2 - red < 1$ was evaluated by calling the Python function `chi_squared_fun(x_values[i], y_values[i], m[i], c[i])` defined within `functions.py`.

The straight clusters were then sorted into categories of straight wiggly and straight. This was done using the fact that the goodness of fit of a function should not change depending on the range of a subset of data considered if the fit is consistently good along the curve. That is to say, if $R' = \{x'_1, x'_2\}$ is a subset in the range of pixel data $R = \{x_1, x_2\}$, then the goodness of fit of the data depends on the subset R' if the line diverges from a straight line at any point. This defined a straight wiggly line.

$\chi^2 - red < 1$ was evaluated iteratively for increasing subsets of the pixel data in each cluster. The cluster were tagged by `tag==1` if the $\chi^2 - red < 1$ was seen to increase significantly between two iterations as defined by the Python functions `chi_squared_set(x_values_o, y_values_o, M, C)` and `tag(chi_squared_set)` defined within `functions.py`.

5 Results

In this section, the best estimate of an efficiency for the algorithm here presented is discussed. These results are shown in **Graph 1**. As previously mentioned, interesting clusters were initially identified by eye. This preliminary classification was also compared to the results obtained from the algorithm described above.

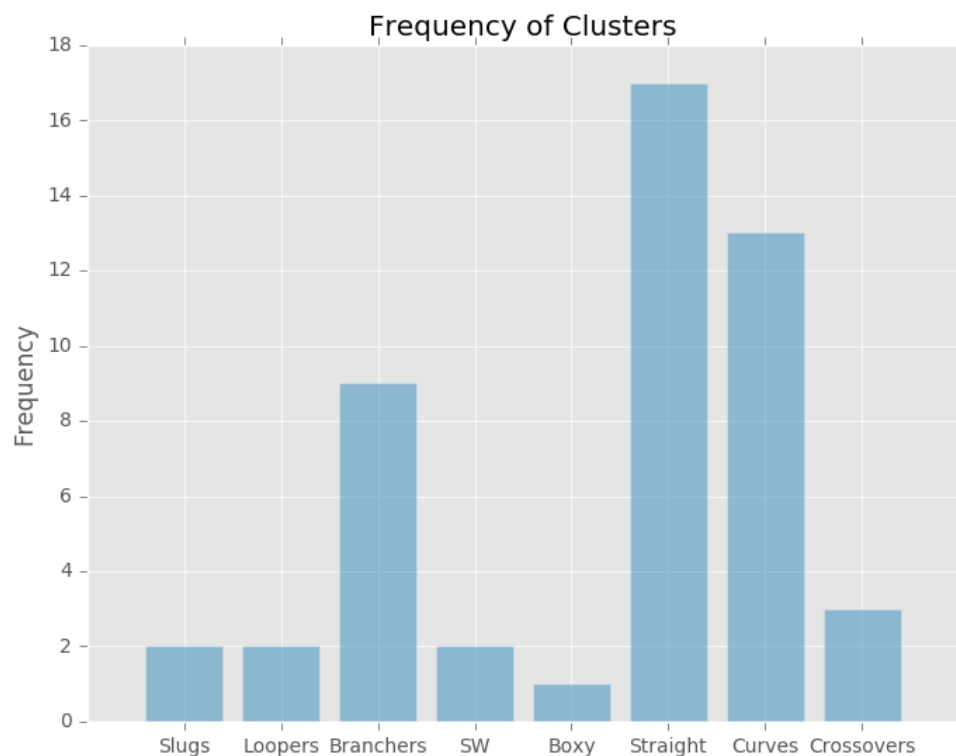


Figure 3: Frequency of clusters identified by our algorithm.

Table 1 below shows a comparison between the clusters identified by eye and the clusters correctly identified by the algorithm.

Type	Number Identified (by eye)	Identified Correctly (by algorithm)
Brancher	7	6
Crossover	4	2
Loopers	2	2
Slugs	2	2
Straight-Wiggly	1	1

Figure 4: Results comparison between manual and algorithm results for cluster classification.

The efficiency of the algorithm was evaluated at $Efficiency = 0.871$ where

$$Efficiency = \frac{\sum_{n=1}^N \frac{IdentifiedCorrectly_n}{NumberIdentified_n}}{N}$$

This efficiency was evaluated using a restricted data set and must be tested against a much wider sample space to bear absolute significance. This was however a vast improvement on the previous efficiency of 0.35 recorded in the first report [6].

6 Conclusion

So far, the sorting algorithm for cluster identification achieved 87% efficiency in the cluster classification. In order to clearly see if the cuts imposed are universally applicable, the algorithm must be subjected to a significantly larger data set. For this reason, the next steps of this project will be to perform the analysis of a larger data set.

References

- [1] James L Pinfold, *The MoEDAL Experiment at the LHC*, Physics Department, University of Alberta, EDP Sciences, DOI: 10.1051 20147100111
- [2] T. Whyntie, H. Bithray, J. Cook, A. Coupe, D. Eddy, R.L. Fickling, J. McKenna, B. Parker, A. Paul & N. Shearer (2015) *CERNschool: demonstrating physics with the Timepix detector*. Contemporary Physics, 56:4, 451-467, DOI: 10.1080/00107514.2015.1045193
- [3] S. Bibb, C. Cooke, A. Evans, E. Ireland, F. Pomeroy, T. Whyntie *MoEDAL Update Presentation* LangtonStar Centre, Queen Mary University of London, January 2015
- [4] Allen Rubin, *Statistics for Evidence-Based Practice and Evaluation* Third edition, Brooks/Cole Cengage Learning, 2013, University of Texas
- [5] CERN@school GitHub, Queen Mary University, London
- [6] J. Flores & A. Shaa (2016) *Report 1*.