# User documentation

## Common Setup prerequisites

In order to run all of the four main applications user should make sure that his operating system supports *.NET Framework* version *4.7.2*. This is required by the user interfaces used in the applications. Another important requirement is to have the C# library ClusterCalculator downloaded. This library is a part of the solution Cluster Processor [8]. The library provides the utilities like calculations of the cluster attributes which are further used in the applications.

## Cluster Viewer

***Setup and input***

The Cluster Viewer application requires Classifier for Clusters to be downloaded. Regarding the third party software, for three-dimensional plotting we need Chart Director library package [11].

To view clusters we run the viewer and either type the path to our .ini file or click Browse button, select the desired .ini file and click Load Clusters button. If the message “File was loaded successfully”appears, it means that our collection of clusters is now ready for viewing. If we do not see this message it means that a problem occurred during loading of the file. Some of the common causes are:

* .ini file does not exist or is inaccessible
* .cl and .px files referenced by the .ini file do not exist or are inaccessible
* .cl, .px or .ini file is not in the correct format

For more information about input file format see the section [input format section]

The image displays the energy of each pixel logarithmically mapped to the color spectrum starting from white (for pixels with energy less than 2eV) through yellow (for pixels with energy greater than 2eV and less than 15eV) to orange and red (pixels with energy up to 500eV). If the same pixel is hit twice in the cluster, the pixel with more energy is displayed.

***Browsing***

After the cluster collection is loaded we can navigate through the collection using Previousand Nextbuttons. To find the n-th cluster in the collection quickly we can select the index of the cluster in Control Panel box and click Findbutton.

***Histograms***

Apart from collection browsing, user can also view histograms. There are two histograms available in the viewer – Collection Histogram and Pixel Histogram, where the first one displays the histogram of the whole collection with respect to the pixel count attribute. The Pixel Histogram represents the histogram of the pixels of the currently viewed cluster with respect to the energy. Both histograms are calculated and displayed after clicking Show Collection Histogram (or Show Pixel Histogram).

***3D visualization***

To get the better idea of how the trajectory of the cluster looked like, we can visualize the cluster in 3D based on the time of arrival of each pixel. In order view the three-dimensional image of the cluster user can click View 3D button. After the image is displayed user can rotate the image around the x-axis and z-axis by clicking Up, Down, Left and Right buttons.

***Cluster Attributes***

For the calculation of cluster properties we use button Show Attributes located in the Cluster Details section. By clicking Show Attributes we can view various attributes of the cluster in the JSON file format. These attributes range from the straightforward eg. Total Energy and Maximum Energy to the more sohphisticated ones like Branch Count and Relative Halo Size. For the further information about the cluster attributes please see the section [insert section].

***Skeletonization***

Skeletonization, often reffered to as a thinning proces “transforms an input binary image into a skeleton by reducing the original image which contains different thicknesses to a thin representation (a set of curves and lines).“ [9] Skeletonization can be used as a tool for removing halo effect of the clusters while also preserving the shape of the original cluster. During the skeletonization process, as the image is thinning, pixel energy is split among its neighbours, which means the total energy of the cluster and its skeletonized version is the same. To view the skeleton of the original cluster, we click Skeletonize. [add image of skeletonization]

***Branch analysis***

After we are able to find the skeleton of the cluster we can try to detect particle trajectories contained in the cluster. To do so, we can click the Show Branches button. The center point of the cluster is represented by the white dot, while the separate branches are denoted by the distinct colors – blue, red and green. Each branch can have its sub-branches - the starting point of the sub-branch is contained in its parent branch. The sub-branches are highlighted by the lighter shade of its parent branch color.

***Classification***

Using all the features we are able to calculate about the cluster we can classify the cluster into various categories. This is done via machine learning using neural networks. The default classes of particles that are implemented in the classifier are Pb, Fe and He while also including classes for elementary particles like muons, electrons, pions and protons. In many cases the classification process is difficult, so we cannot expect the class prediction to have 100% accuracy. To find out more about classification process and its reliability see the section [insert section]. [add sample image of the particles]

## Filter

The user interface of the Cluster Filter application is very similar to the Cluster Viewer. As the first step we select the input .ini file. This can be done either by typing out the path to the file or clicking the Browse button. Then, we choose the name for the output .ini file. After that we can choose the properties to filter by and set the lower and upper bounds for the property. If no bound (upper or lower) is specified, filter automatically sets the bound to the maximum (or minimum) possible value. To start the processing we click the Process button. During the filtering process a new .cl file is created that contains only those clusters, which fit the filtering criteria. However, no .px file is created in the process, because the output reuses the original pixel file to speed up the calculation and prevent unnecessary copying. After the filtering is done, filter displays the message “Filtering successfully completed”. If any other message is shown, it means there was an error. The message should provide more information about the problem – in most of the cases it is either the incorrect data format or file inaccessibility.

### Console Filter

[to do?]

## Description Generator

The Cluster Description Generator is a tool for the attribute calculation for the whole collection of clusters at once. This can be useful when creating training data for the machine learning algorithms. We start by clicking the Browse and Add… button where we select one or more .ini files and add them to the Selected Input collection. To add more input files we can simply repeat the proces until all desired files are shown in the Selected Input box. To remove the last element in the collection we click the Remove Last button. For each single file we select, we can edit its Class Name collumn by triple clicking – there can be multiple files containing the same class. This way we set the name of the particle that is present in the input file, which can then be used for the supervised machine learning algorithms. Then we can choose the output file name in the Select output text field and also tick the attributes that will be calculated. To include the particle name as an attribute, remember to tick the Class attribute.

After that we can choose either equal distribution of each class in the output, or the distribution proportional to the particular class size. For each file with given class (further referenced to as a class partition) we can choose whether we want to process those in a serial order (given by the order on input) or in a parallel order. The user can also select the ending condition of the process. When this condition is satisfied, the program finishes the calculation – for the large number of clusters the process may take several minutes to complete.

By default, no cluster on the input will appear in the output more than once. When generating the data for imbalanced classes (there are huge differences in class size) this could lead to machine learning problems with classification of the less frequent classes. To compensate for that, there is an option to choose Allign class. By choosing the allign class, any other class partition that was processed will not be discarded but will be processed again until the specified Allign class is fully processed.

## Classifier [possibly not part of user documentation?]

The Cluster Classifier is a console application which provides an interface for the classification process of the selected clusters. The prerequisite for the classifier is to have Accord.net NuGet packages installed. [10] When running the application, specify the path to the input file in the JSON format as the first command line argument. The result of the classification are the pairs of the class name and the number of clusters that were classified to this class (eg. Electron:123) which are printed to the console. Another option is to use the classifier as a library to classify a single cluster (similarily to the Cluster Viewer) .