**Program Functions**

Clustering\_Analysis is designed to allow users to perform three types of clustering analysis on a set of data. Mean-shift, K-means, and Minimum-Spanning-Tree clustering algorithms are available for use. The user can specify the analysis that will be run, the inputs to use, and the outputs that will be saved. Clustering\_Analysis uses inputs from a file. This file is set to the default name experiments.txt, and saves all outputs to a file set to results.txt. These file names may be changed using the inputs from the user. The program also produces various plots based on the user’s specifications.

First, the user must import Clustering\_Analysis into their python module and run Clustering\_Analysis.userinput(). The user will then be prompted with various questions about what type of analysis will be run. The user must leave a space between each type of analysis. The inputs will then be used to run the program, and the outputs will be saved.

The user will be prompted to enter the name of the data file they would like to analyze. This name is passed to the load\_data\_file function, where it is imported to Cluster\_Analysis as a table. The data file contains brightness measurements of various wavelengths of light. Each column of the file is assumed to be a separate wavelength. Following the data file, the user must specify which combinations of wavelengths they wish to analyze, and an arbitrary number of clusters they wish to use. The combination of wavelengths is used to determine what “colours” of stars will be used to group the objects. Since each object emits a certain magnitude of each wavelength, the difference in magnitude is used to define a colour range. These pieces of information will be saved in experiments.txt, and will be loaded into the program when it is run.

The user may select the types of analysis to perform, and the outputs that should be saved. Each analysis has a plot that is produced, as well as various results that are saved in the results.txt file.

After the user inputs are saved, they are passed to the main() function. This function acts as the central function of Cluster\_Analysis, and determines which functions must be run based on the input.

The main() function calls various analysis functions, and passes the data from the organize\_data function, which ensures that all the data points being passed to the analysis are valid. The user must set the correct criteria before running Cluster\_Analysis.

The analysis is run, and the results are saved to a results.txt file. This file records the bands that were used, the number of clusters in the data, the strength of the clustering, and the number of objects in each cluster. The user has the option to produce a results summary, which produces various graphs of the size and strength of each cluster.

**Clustering Methods**

Mean Shift

Mean shift clustering is a clustering technique that does not constrain the shape of the clusters, and does not require prior knowledge of the number of clusters present in the data (<http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TUZEL1/MeanShift.pdf>). The mean shift algorithm treats the data as a probability density function. Dense regions of the data set correspond to local maxima in the distribution. In order to perform a mean shift analysis, each data(point?) is subject to a gradient ascent procedure. The stationary points resulting from this procedure are the local maxima of the distribution, and represent the center of each cluster. Similar data points are assumed to be members of the same cluster (<http://www.cse.yorku.ca/~kosta/CompVis_Notes/mean_shift.pdf>).

K-Means

K Means clustering is a method to group data into a number of specified clusters. The algorithm assigns arbitrary cluster centers, and each data point is assigned to the closest center. The cluster center is then updated to the mean of the assigned data points. The process is repeated until there is no change in center assignment (https://web.cse.msu.edu/~cse802/notes/ConstrainedKmeans.pdf). K Means clustering is used in this program to determine the number of objects that belong to each cluster identified using the Mean Shift algorithm.

Minimum Spanning Tree

Minimum Spanning Tree clustering is an efficient form of clustering as it does not require user input. Euclidean MST clustering creates a spanning tree of *n* points in a space. The edge lengths in this tree are the Euclidean distances between pairs of points in a data set. EMST clustering is useful as it is able to identify clusters in data sets with irregular boundaries, instead of the typical spherical shape (<http://www4.ncsu.edu/~zjorgen/ictai06.pdf>).

**Application to Galaxy Survey Data Sets**

The clustering methods outlined above were applied to a survey of the M83 galaxy. The data set comprises brightness measurements at various wavelengths of light for individual objects within the galaxy (e.g. stars, star clusters), as well as object sky positions and identification numbers. The data are clustered in colour space, where “colour” is the astronomical term for the ratio of observed brightness at two different wavelengths; since multiple brightness measurements are available for each object, different combinations of colours can be tested to see which provide stronger clustering. The data are first clustered using the mean shift algorithm. The results of this algorithm provide the number of clusters that are apparent in the data set. These clusters group the data based on similar colours.

The K Means algorithm can be used to determine how many objects belong to each cluster. This determines how many objects in the data fall into each colour group. The K Means function then determines where each object resides in the galaxy, to determine what type of star clusters exist.

The identification numbers may be used to compare the results of this analysis to other analyses performed. If the identification numbers of various studies match, the results are strong.