**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 named experiments.txt, and saves outputs to results.txt. The program also produces various plots based on the user’s specifications.

First, the user must set the data file name in the Clustering\_Analysis.organize\_data() function. This will allow the program to read in the specific data file the user wishes to analyze. Following the data file, the user must specify which combinations of bands they wish to analyze, and an arbitrary number of clusters they wish to use. These pieces of information will be saved in *experiments.txt*, and will be loaded into the program when it is run.

Following the inputs, 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.

*experiments.txt*

* List of all tests that are being run
* 4 band names are required to create two colours to be compared
  + Band names correspond to the wavelength of light stored in the hlsp\_wfc3ers\_hst\_wfc3\_m83\_cat\_all\_v1.txt file

*results.txt*

* Output file of the program
* Lists 4 band names from each trial
* Number of clusters from mean-shift clustering
* Silhouette score of each trial
* Total number of objects
* Number of objects in each cluster

*Clustering\_Analysis.py*

* Main file
* Global variables
  + Band\_names
    - Dictionary of all column numbers and corresponding wavelengths
    - Each name can be used in the list of tests in experiments.txt
  + cluster\_colours
    - list of colours to use when making plots
  + inputdata
    - The name of the data file that will be used for clustering
    - Use this global variable to make it easier to change the data set
      * Only the name of the file and the dictionary names will need to be changed to run clustering on different files
  + data
    - load the .txt file from inputdata
    - ‘data’ is used throughout code
    - Necessary in order to change the input file
  + max\_num\_clusters
    - Used to format the results.txt
      * File must have a constant number of columns regardless of how many clusters each trial has
    - Can be changed and have no effect on clustering output

*Clustering\_Analysis.userinput()*

* Allows user to set analysis specifications
* Prompts user for various inputs
  + Types of analysis
  + Plots to be generated
  + K-means inputs

*Clustering\_Analysis.do\_everything (input\_file, output\_file)*

* Function used to automate clustering process
* Used to call other functions and run k-means, mean-shift, make various plots, and produce a results summary
* Inputs
  + Input\_file
    - Experiment.txt
  + Output\_file
    - Results.txt
  + User inputs
    - Saved inputs are passed to do\_everything
* Variables
  + run
    - loads the experiments.txt
    - used through this function to run each trial
    - each object in the *run* array is the band name that will be run in each trial
  + results
    - used to open results.txt
  + for loop
    - used to cluster each trial listed in *run*
  + wave1-wave4
    - wave1, wave2, wave3, wave4 each take the name of the corresponding band in the current trial
      * all values taken from *run*
  + gooddata1, gooddata2, greatdata
    - Used to remove all data points with no value from the *wave* variables
    - Only data points that have a valid value in all *wave* variables will be used in clustering to reduce outliers
    - Parameters can be changed based on data file
  + colour1, colour2
    - colour1: difference in magnitude between wave1&wave2
    - colour2: difference in magnitude between wave3&wave4
  + numberofclusters
    - this variable is created from the result of the mean-shift clustering
    - passed to k-means function as an input
  + input\_str
    - used to format results.txt
  + score, num\_obj
    - used to call k-means clustering function
    - returns the silhouette score along with the number of objects in each cluster
  + output\_str
    - uses score, num\_obj to create the second part of results.txt
  + results.write
    - writes the result of the trial into results.txt
  + if results\_summary
    - if the user wants a summary of the results to be produced

*Clustering\_Analysis.organizedata()*

* Used to load in data file and remove any outliers

*Clustering\_Analysis.do\_meanshift (band1, band2, band3, band4, colour1, colour2, make\_plots)*

* Used to perform mean-shift clustering on each trial
* Output is the number of expected clusters in each colour combination
* Function is called in *Clustering\_Analysis.do\_everything* from the variable *numberofclusters* and is used as an input to the k-means function
* Variables
  + Input Checking
    - Used to make sure that each band in experiments.txt exists in the band dictionary and in the input file
    - If the bands are the same or do not exist, the user will know
  + X, X\_scaled
    - Used to preprocess data
    - See Data Mining textbook online resource for further information
  + bandwidth
    - bandwidth is a variable used to determine the number of clusters in a data set
    - can be set manually or estimate\_bandwidth() can be used
    - estimate\_bandwidth() is time consuming but is the best estimate, as a skewed bandwidth drastically changes the number of clusters
  + ms, ms.fit
    - Perform mean-shift clustering on data set
  + n\_clusters
    - find the number of clusters generated by performing mean-shift

*Clustering\_Analysis.make\_ms\_plots (colour1, colour2, n\_clusters, X, ms, band1,band2,band3, band4)*

* function is called when user makes mp=True in do\_everything function
* generates the mean-shift clustering plot
* Variables
  + Fig, ax
    - Generate axis to plot data
  + H, C1\_bins, C2\_bins
    - Generate histogram of cluster data
  + For loop :
    - Plot cluster data

*Clustering\_Analysis.do\_kmeans (band1,band2,band3,band4, colour1, colour2, greatdata, number\_clusters, make\_plots, output\_cluster\_id)*

* Function called from *Clustering\_Analysis.do\_everything()* after computing mean-shift
* Function computes number of objects in each cluster and prints values to results.txt
* Computes silhouette score which measures the accuracy of the clustering, score is printed in results.txt
* Variables
  + X, y
    - Retrieves the x/y coordinates of each object from the data file
    - Passed to xy\_plot function to plot the coordinates of each object in trial
  + Id
    - Retrieves the object ID from the data file
    - Used to compare clustering to other clustering experiments run on M83 data set
  + clusterdata
    - process data for clustering
  + scaler, clf
    - Perform k-means clustering
  + Cluster\_number
    - Extract the cluster number that corresponds to each object in the data set
  + If output\_cluster\_id
    - If statement will generate .txt file with a list of the object number and object id
      * File used to compare each clustering experiment with experiments performed by others
  + Score
    - Compute the silhouette score to measure clustering accuracy
  + Objects\_per\_cluster & for loop
    - Assign each object in the data set to its cluster
    - Used to plot xy position and number of objects in each cluster printed to results.txt
  + If make\_plots
    - If user wants graphs of clustering and xy position, make\_plots =True in *Clustering\_Analysis.do\_everything()*

*Clustering\_Analysis.colour\_kmeans\_plot(band1,band2,band3,band4, clf, scaler, colour1, colour2, number\_clusters)*

* Used to plot the results of k-means clustering
* Similar structure to the plot of mean-shift clustering
* Called from *Clustering\_Analysis.do\_kmeans* when user requires plots to be created

*Clustering\_Analysis.xy\_plot(x, y, number\_clusters, cluster\_number, band1,band2,band3,band4)*

* Produces colour coded x-y plots of each object in clustering
  + Colours correspond to cluster number
* Variables
  + For loop
    - Plot x-y positions of each object in each cluster with specific colour
* Each plot is saved to code folder

*Clustering\_Analysis.results\_summary(input\_file= ‘results.txt’)*