Gene Expression Infection Detection, Analyzation & Prediction System

*Rohaan Advani(a), Kushal Gupta(b)
(a: student, College of Engineering Pune, b: student, Manipal Institute of Technology)

----------ABSTRACT----------

DNA Microarray *is a laboratory tool that utilizes florescence to quantify* ***Gene Expression*** *after hybridization and/or genetic manipulation. Strands of DNA with known nucleotides are used to check the level of expression of certain genes and their efficacy.*

The following system takes input data from a ***Dataset of DNA Microarray samples*** *and performs a multi-layered analysis using* ***Image Manipulation****,* ***Computer Vision****,* ***Exploratory Data Analysis****,* ***Machine Learning*** *and* ***Colour Space Manipulation*** *to produce the Ratio of Expression of certain Genes in a given sample, an analysis of Gene Expression in a given Sample Space / Dataset as well as predicts the Ratio of Gene Expression in a larger sample space. This has many industry applications in the field of scientific computation, medicine and agriculture.*

Date of Submission: 31 - 01 -2022 **Date of Acceptance: - -2022**

I. INTRODUCTION

Microarray is a laboratory tool that is used for analysis of gene expression. These are slides with thousands of microscopic DNA molecules, with known nucleotide sequence, established in a grid format. The known DNA sequence models are also known as the transcriptome or the set of messenger RNA (mRNA) transcripts expressed by a group of genes. Gene manipulation is an experimental science, where the efficacy and throughput of the output is unpredictable. Many Variables control the level of expression post genetic manipulation, most of which are independent and cannot be controlled.

Therefore, the need of such a quintessential tool which helps us weed out the expressionless strands of DNA from those which have a viable degree of expression and have industry applications. There are, however, many potential pitfalls in the use of microarrays that result in false leads and erroneous conclusions.

This paper attempts to discuss the steps / methods / algorithm used in order to build a DNA Microarray and subsequently the **Gene Expression Infection Detection, Analyzation & Prediction System**. Using this system, Gene expression in a given DNA Microarray can be recognized with Image Manipulation and Colour Space Manipulation algorithms.
Moreover, it discusses Exploratory Data Analysis of the Dataset of DNA Microarrays which are successfully compiled by the program and Building of a Machine Learning Model in order to

predict the Infection of Genes with reference to the given Sample Space/ Dataset. This has many industry applications in the field of scientific computation, medicine and agriculture.

II. IMPORTANT TERMINOLOGY

- DNA Microarray - A device used in Life science laboratories for easier visualization of gene expression post hybridization or more commonly to check mutation in DNA.

- Gene Expression - In our bodies information is read from DNA (genes) and they are then processed by other subcellular organelles to express the traits the genes were coding for. This entire process is called Gene Expression.

- Genetic Engineering - The study and practice of manipulation and crossing of genes.

- cDNA - Stands for, Complementary DNA. All DNA comprise Purines and Pyrimidines which are two types of nucleotide groups. DNA has a double strand helical structure. Each of those strands has specific nucleotides from both groups. But they show specific affinity. Adenine only binds with Thymine and Guanine only binds with Cytosine. Therefore one strand of DNA will comprise all the complementary nucleotides to the other strand therefore they are known as complementary DNA. A strand of DNA with all the complementary bases to the original strand.

- mRNA - Messenger RNA a single strand of nucleotides with a replacement of Thymine with Uracil. It is a complementary strand to one of the strands of DNA.

- RNA isolation - The process of extraction and purification of RNA from a cell sample.

- Reverse Transcription - A technique by which a complementary strand of DNA can be made from a given strand of RNA since the nucleotides are complementary.

- Hybridization - Practice of Recombining DNA strands in order to make new DNA strands with desired phenotypic qualities.

- Image Manipulation - In programming, an Image is recognized as a 2D Function F(x,y), where x and y are spatial coordinates of pixels having colour values for a certain display. Image manipulation is the processing of the image using multiple different methods until we reach our output which can be either a form of the image or a corresponding feature of that image used for further analysis and decision making.

- Computer Vision - Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information. In Python it is implemented using the cv2 Library.

- Canny Edge Detection - Canny edge detection is a multi-step algorithm that can detect edges and suppress image noise simultaneously.

- Gaussian Blur - In image processing, a Gaussian blur is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.

- Contours - Contours are the curves joining all the continuous points (along the boundary), having the same color or intensity in an image. Contours are a useful tool for shape analysis and object detection and recognition in Images.

- Threshold - Thresholding is a type of image segmentation, where we change the pixels of an image to make the image easier to analyze.

- ColourSpace - A ColorSpace is used to identify a specific organization of colors. Each color space is characterized by a color model that defines how a color value is represented in an image.

- RGB ColourSpace - An RGB color space is any additive color space based on the RGB color model. A particular color space that employs RGB primaries for part of its specification is defined by the three chromaticities of the red, green, and blue additive primaries and can produce any chromaticity that is the 2D triangle defined by those primary colors .These primary colors are specified in terms of their color space chromaticity in a color range of 0(dark) to 255(light), linking them to human-visible color.

- LAB ColourSpace - The CIELAB color space also referred to as L*a*b* is a color space defined by the International Commission on Illumination (abbreviated CIE) in 1976. It expresses color as three values: L* for perceptual lightness, and a* and b* for the four unique colors of human vision: red, green, blue, and yellow. CIELAB was intended as a perceptually uniform space, where a given numerical change corresponds to a similar perceived change in color. While the LAB space is not truly perceptually uniform, it nevertheless is useful in industry for detecting small differences in color.

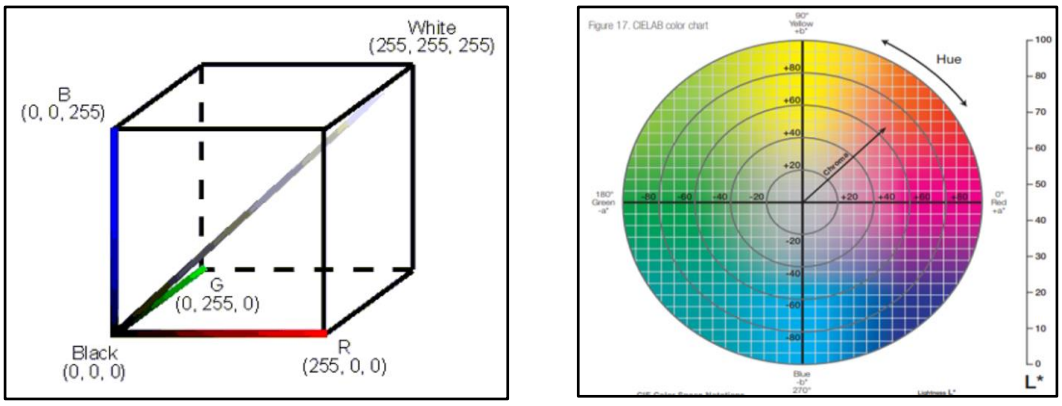


Figure 2.1: ColorSpace RGB Cube and LAB Graph

- Pandas - Python Library created to help developers work with “labeled” and “relational” data intuitively.

- MatPlotLib - Python Library which helps in Data Analyzation using numerical plotting.

- Seaborn - Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

- Sci-kit Learn - Industry standard Library for Data Science Projects in Python.

- Exploratory Data Analysis - Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypotheses and to check assumptions with the help of summary statistics and graphical representations.

- Univariate Analysis - “Uni” means one and “Variate” means variable hence univariate analysis means analysis of one variable or one feature. Univariate basically tells us how data in each feature is distributed and also tells us about central tendencies like mean, median, and mode.

- Bivariate Analysis - Bivariate Analysis is used to find the relationship between two variables. Analysis can be performed for a combination of categorical and continuous variables.

- Multivariate Analysis - Multivariate Analysis is used to find the interdependence of variables on each other in a given dataset.

- InterQuartile Range - In descriptive statistics, the interquartile range (IQR), also called the midspread, middle 50%, or H-spread, is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles, IQR = Q3 − Q1.
- Outliers - An outlier of a dataset is defined as a value that is more than 3 standard deviations from the mean. Removing outliers from a Pandas DataFrame removes any rows in the DataFrame which contain an outlier. Outlier calculations are performed separately for each column.

- Machine Learning - Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

- Linear Regression - Linear regression analysis is a Machine Learning Algorithm used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

III. PROPOSED METHOD

Part 1: Setting up the DNA microarray:

In theory one could assume that increasing the number of genes used to test the data set would lead to a smarter model and a more accurate one at that, in practice we observe a much slower rate of intake and the model takes longer to train. It is observed that the model tends to over fit the training data and prevent.

There is a basic protocol that is followed while setting up the microarray, we must follow the conventional steps for microarray setup as our program deals with post production analysis. The following would be the ideal steps to be followed.

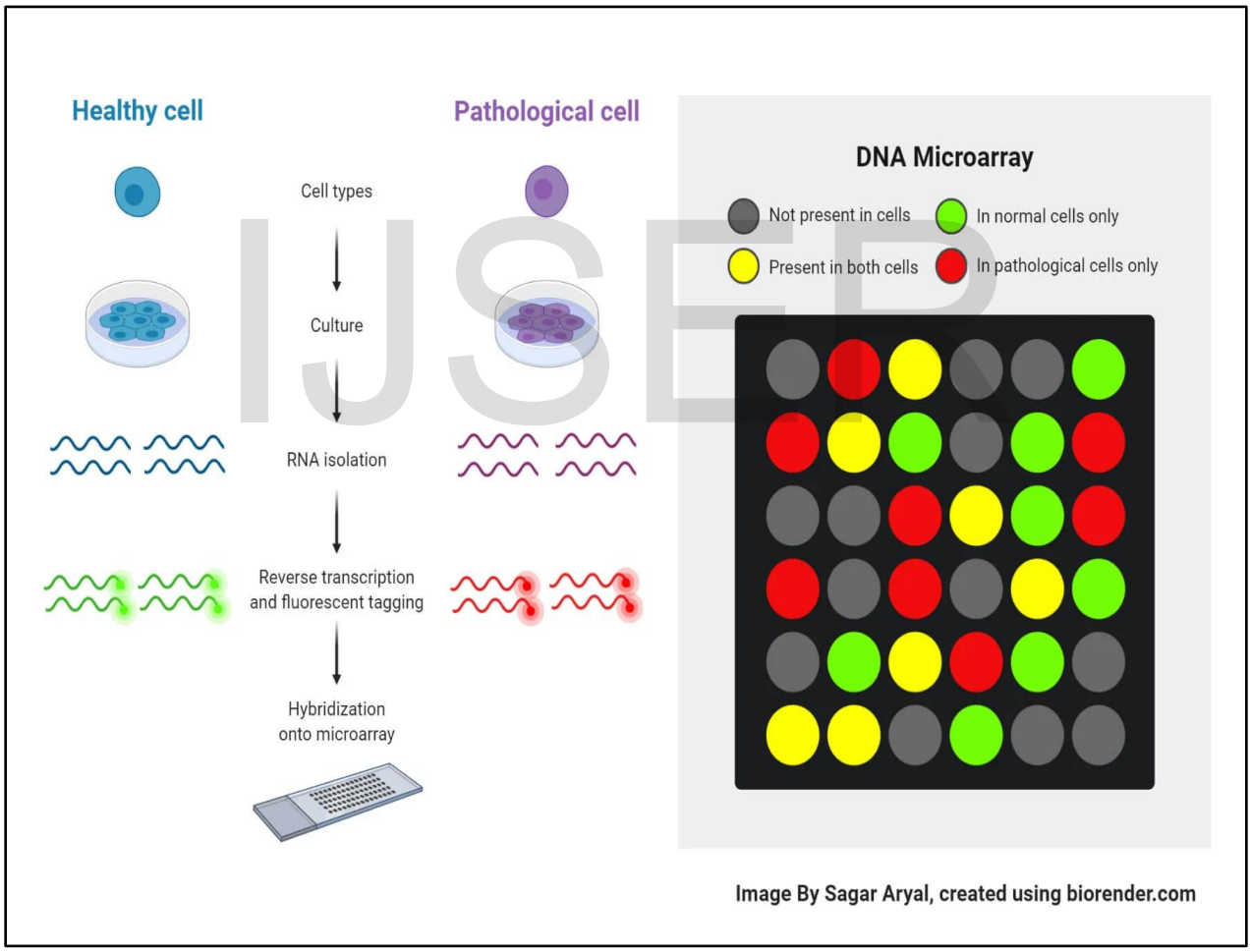


Fig 3.1: Steps to build the DNA Microarray

Step 1 - Isolation of gene of interest

We do this by isolating the pure mRNA samples from the organism in question, this is a comparative analysis therefore we will have one sample working as our control sample and one as an experiment sample.

Step 2 - Reverse transcription

The extracted mRNA sample is subjected to reverse transcription and we obtain a complementary strand of DNA from that, also known as cDNA. (Reverse transcription is the process by which rna sequences are transcribed to their complementary DNA sequence using promoters and complementary nucleotide sequences)

Step 3 - fluorescence

The cDNA is then tagged with floressence markers. Primary colours such as red and green are used as markers. These fluorescent markers are added to the cDNA to indicate healthy and infected genes upon expression.

Step 4 - Probe design

We construct DNA probes, single stranded DNA molecules with known nucleotide sequence. These probes have nucleotides of complementarity to the cDNA that have been previously acquired. (hybridisation is the name given to the process in which the cDNA will combine with the probes with specific gene sequences and express a particular color depending on the degree of expression of healthy vs infected gene). There exist approximately 6000 probes on the DNA chip which act as sites or hybridisation.

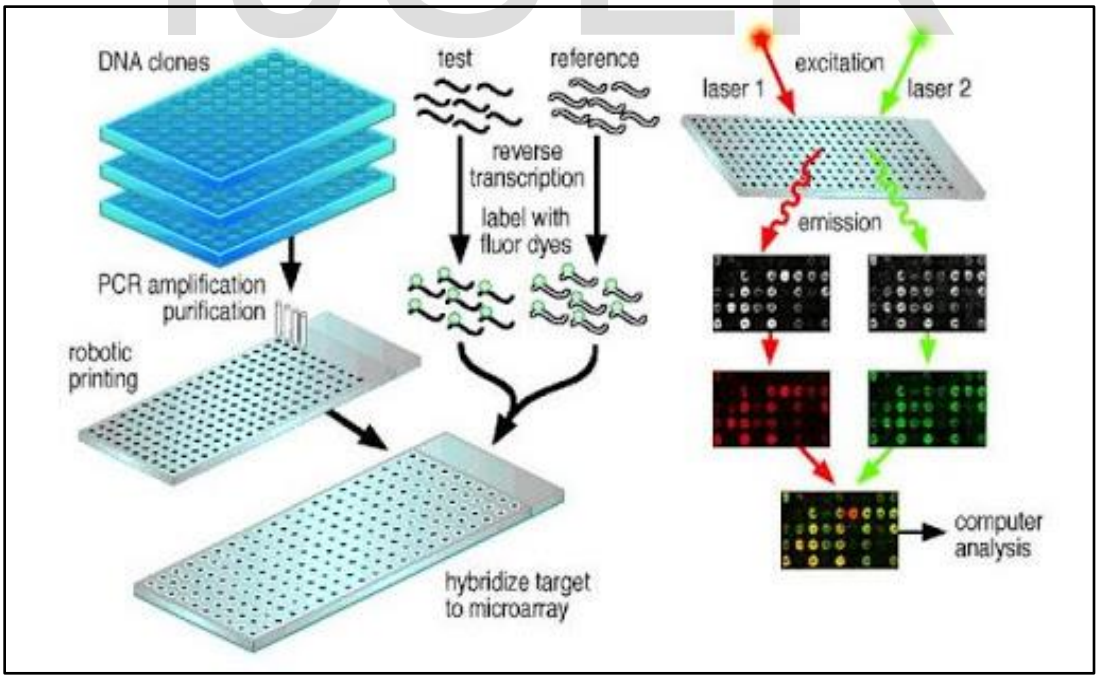


Fig 3.2: Probe Design

Step5 - Analyzer

The DNA chip or the slide with the probes are made with low-fluorescence material. Fluorescence is the measure of ability of a substance to absorb light. The chip is then placed in a Microarray Analyzer for further processing.

Part 2: Gene Expression Detection in DNA Microarray:

A Sample Space of Microarrays has been set up. This Sample Space is the Dataset which is going to be used in the program.

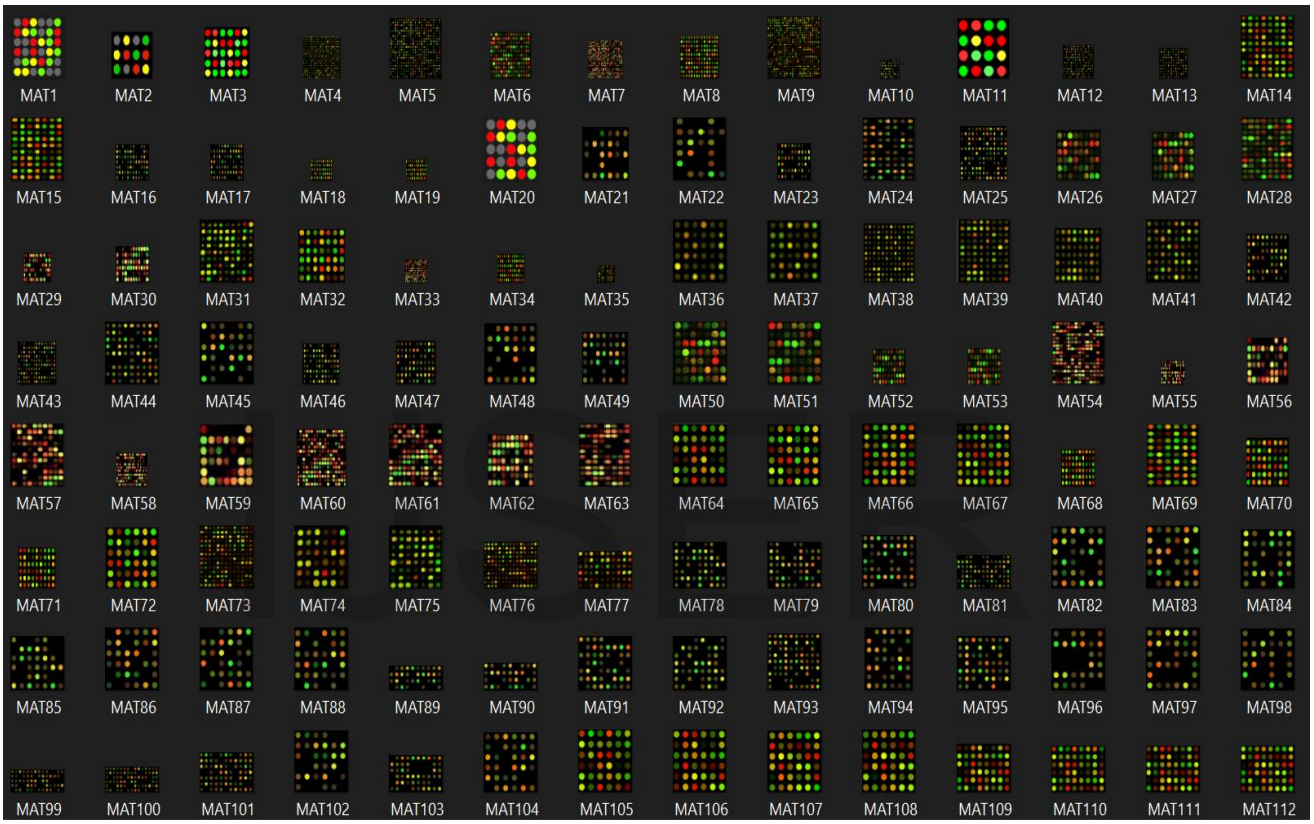


Fig 3.3: The Dataset for gene expression detection

In this program we are processing images of Microarrays in order to detect the following:

1. Total Number of Probes in the Given Sample.
2. Total Number of Probes in the Filtered Sample - Excludes Empty Probes.
3. Total Number of Infected Probes in the Filtered Sample.
4. Total Number of Healthy Probes in the Filtered Sample.

Step 1- Read the Microarray Image.

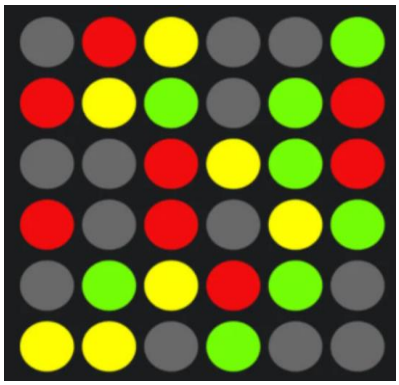


Fig 3.4: Microarray Image

Step 2- Use Canny Edge Detection to Detect Probes in the Original Sample.

Step 3- Use Gaussian Blur to reduce image noise.

Step 4- Detect the Number of Contours in the sample. Draw the Contours.

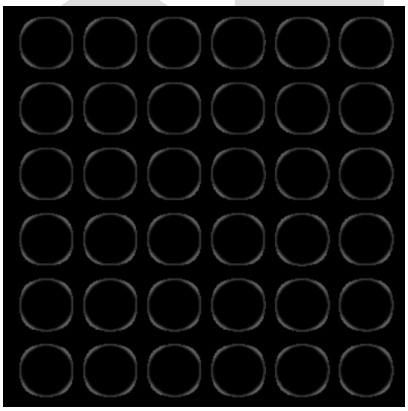


Fig 3.5: Detect Contours

Step 5- Apply Binary Threshold > 127 in order to filter out the Blank Probes from Sample.

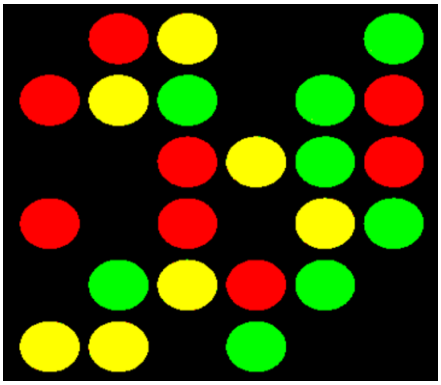


Fig 3.6: Remove Blank Probes

Step 6- Repeat Steps 2-4 for the Filtered Sample.

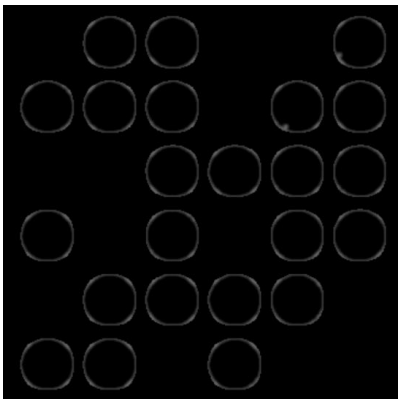


Fig 3.7: Detect Edges of non-blank probes

Step 7- The Current sample is in the BGR Colour Space Convert it to the LAB Colour Space in order to distinguish between the intensity of light in the image easily.

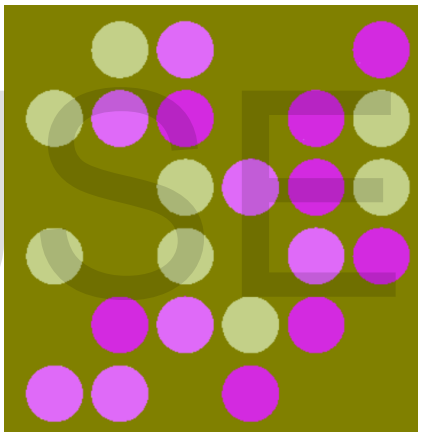


Fig 3.8: Convert to LAB ColorSpace

Step 8- Apply the inRange function to detect the Colours of Infected Probes from the Filtered Sample. The filtered LAB range used is [20, 150, 150] to [190, 255, 255].

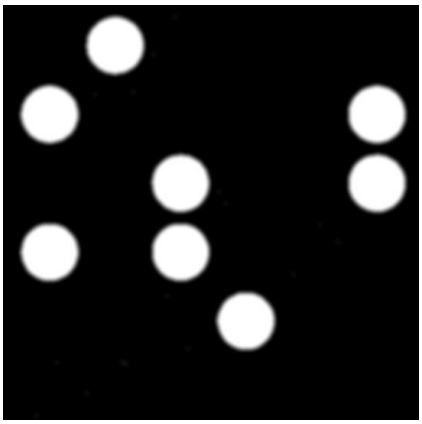


Fig 3.9: Infected probes detected

Step 9- Repeat Steps 2-4 for the Infected Probe Sample.

Step 10- Apply the inRange function to detect the Colours of Healthy Probes from the Filtered Sample. The filtered LAB range used is [100, -50, 60] to [255, 140, 215].

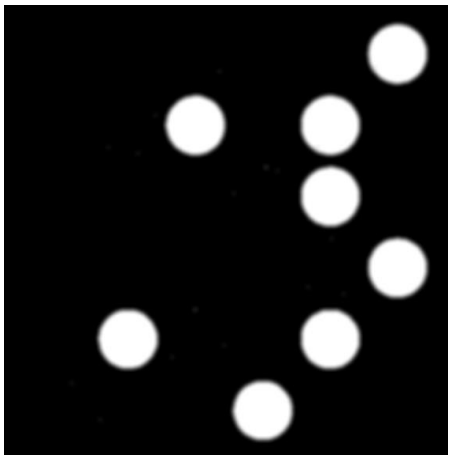


Fig 3.10: Healthy Probes Detected

Step 11- Repeat Steps 2-4 for the Healthy Probe Sample.

Step 12-Display all Processed Images and the Number of Contours Detected for all cases.

```
Number of Probes found in Original Sample = 36
Number of Probes found in Filtered Sample = 23
Number of Infected DNA Probes found = 8
Number of Healthy DNA Probes found = 8
```

Step 13- Repeat Steps 1-12 for the entire Dataset and build a csv file for further Exploratory Data Analysis.

Part 3: Exploratory Data Analysis on Dataset of Information collected by above program:

The CSV Dataset Built has information of a Total of 150 Images.
Exploratory Data Analysis Includes the following steps:

1. Variable Declarations and Calculations
2. Univariate Analysis
3. Bivariate Analysis
4. Multivariate Analysis
5. Treating Null Values
6. Treating Outliers
7. Dummy Variable Creation for Descriptive Data.

Step 1- Variable Declarations and Calculation:

The name of the CSV dataset imported for further use is eda.
Following are the variables used for the exploratory data analysis:

1. OGP - Number of Probes in Original Sample
2. FP - Number of Probes in Filtered Sample
3. IP - Number of Probes in Infected Sample
4. HP - Number of Probes in Healthy Sample

The above variables have been found in **Part 2** of the method and save as a csv dataset available in the given Github Repository for use.

Description of table variables are as follows:

```
eda.describe()
```

	TEST IMAGE NO:	OGP	FP	IP	HP
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	54.433333	40.066667	15.146667	15.433333
std	43.445368	58.501009	50.266493	28.390231	24.117558
min	1.000000	9.000000	9.000000	0.000000	1.000000
25%	38.250000	28.250000	19.000000	6.000000	6.000000
50%	75.500000	36.000000	27.000000	8.000000	10.000000
75%	112.750000	59.000000	37.000000	13.000000	15.000000
max	150.000000	424.000000	384.000000	238.000000	195.000000

Table 3.1: Dataset Description

The Calculated Variables and their respective formulas using Pandas in Python are as follows:

- MP - Number of Mixed Probes in the Sample - Certain Probes of a given DNA microarray are neither 100% Healthy nor 100% Infected. Such probes fall under the category of Mixed Probes as shown in **Part 1** of the proposed method.

Formula 3.1:

```
eda['MP']=eda['FP']-eda['IP']-eda['HP']
```

- RI - Ratio of Infection - Different Samples have different numbers of Infected and filtered probes. Thus in order to standardize the amount of Infection in the Probes amongst all the Images in the DNA Microarray Dataset this variable has been created.

Formula 3.2:

```
eda['RI']=eda['IP']/eda['FP']
```

- RH - Ratio of Healthy Probes - Different Samples have different numbers of Healthy and filtered probes. Thus in order to standardize the amount of Healthy Probes amongst all the Images in the DNA Microarray Dataset this variable has been created.

Formula 3.3:

```
eda['RH']=eda['HP']/eda['FP']
```

- RM - Ratio of Mixed Probes - Different Samples have different numbers of Mixed and filtered probes. Thus in order to standardize the amount of Mixed Probes amongst all the Images in the DNA Microarray Dataset this variable has been created.

Formula 3.4:

```
eda['RM']=eda['MP']/eda['FP']
```

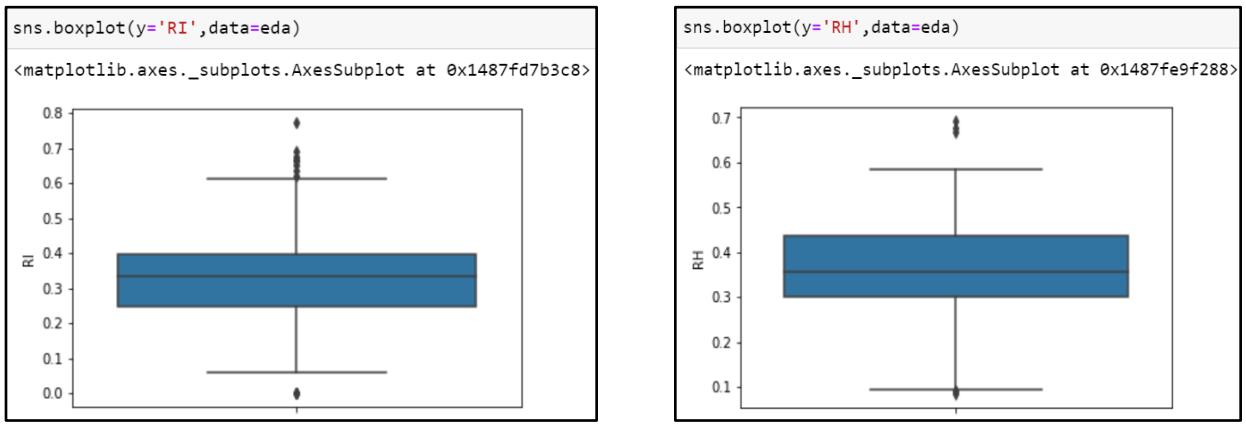
Since RI, RH and RM variables have standardized the above data there is no requirement of the other variables for further Analysis and thus they can be dropped.

Step 2- Univariate Analysis:

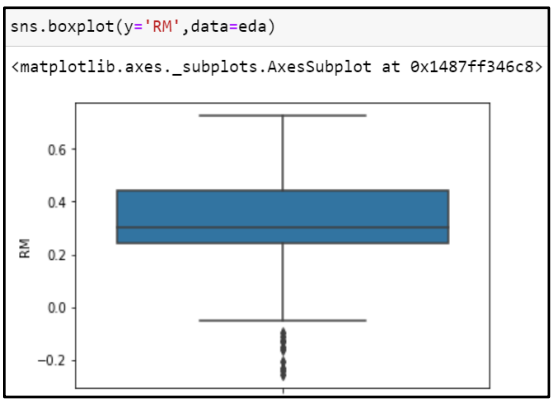
Univariate analysis of variables RI, RH and RM has been done using Boxplots and Distplots.

Boxplots are a measure of how well distributed the data in a data set is. It divides the data set into three quartiles. The graphs represent the minimum, maximum, median, first quartile and third quartile in the data set. Which gives us insights into the measures of central tendencies of the given data which is an important element of statistical data analysis.

The Following are Boxplots of variables RI, RH and RM and the calculated results:



Graph 3.1a: Boxplots of Infected probes and Healthy probes



Graph 3.1b: Boxplots of Mixed Probes

	RI	RH	RM
count	150.000000	150.000000	150.000000
mean	0.332412	0.359298	0.308290
std	0.143937	0.114378	0.192097
min	0.000000	0.083333	-0.258065
25%	0.250000	0.300000	0.240606
50%	0.333333	0.357143	0.303689
75%	0.397826	0.436821	0.442708
max	0.774194	0.692308	0.727273

Table 3.2: Dataset Description

The points outside the Box plot Distributions in the above graphs the Outliers which will cause an error while building the machine learning model and thus these Outliers will be treated in further steps of this part.

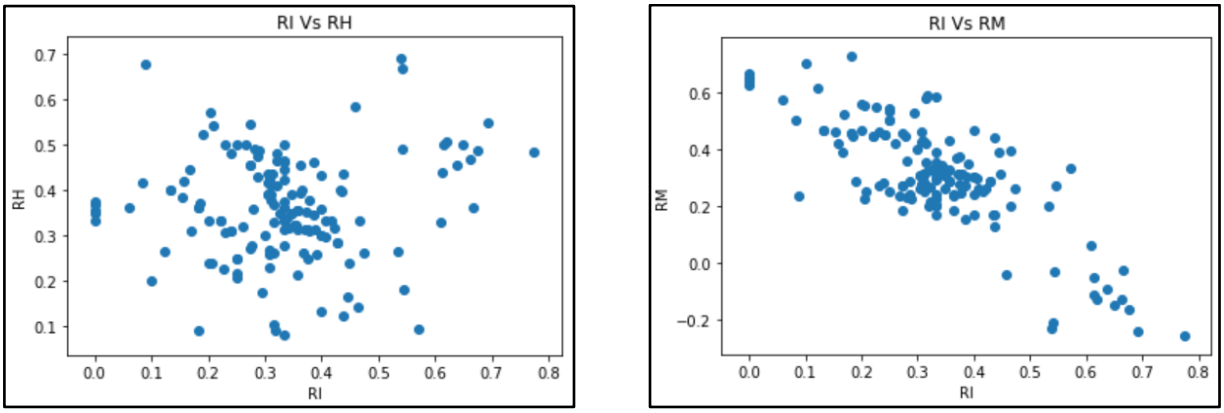
The table shows the measures of central tendencies which we intended to calculate in this analysis. 25%, 50% and 75% correspond to the first quartile, median and third quartile respectively.

The Distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution.

Step 3- Bivariate Analysis:

Bivariate Analysis of variables RI, RH and RM has been done in groups of two using scatter plots which plot X, Y coordinate plots of one variable against another and thus show us the tendency of maximum of the data lies towards the X value of the Y value.

Following are the Scatter Plots between the variables (RI, RH) and (RI, RM):

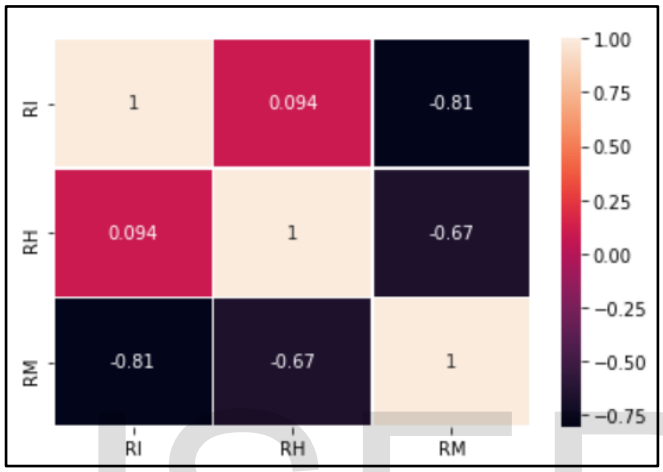


Graph 3.2: Bivariate Analysis of Infection against Healthy and Mixed Probes

Step 4- Multivariate Analysis:

Multivariate Analysis of variables RI, RH and RM is done using a Heatmap. A heatmap contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. It is used to compare all variables of a dataset during Exploratory Data Analysis.

Following is the Heatmap of variables RI, RH and RM.



Graph 3.3: Multivariate Analysis of **Infection: Health: Mixed**

Step 5- Treating Null Values:

In **Part 2** of the proposed method, we were successful in finding all values we require for data analysis and thus there are no Null values that are required to be treated for Analysis of the current data.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   RI      150 non-null    float64
 1   RH      150 non-null    float64
 2   RM      150 non-null    float64
dtypes: float64(3)
memory usage: 3.6 KB
```

Figure 3.11: Null Value Detection in Dataset

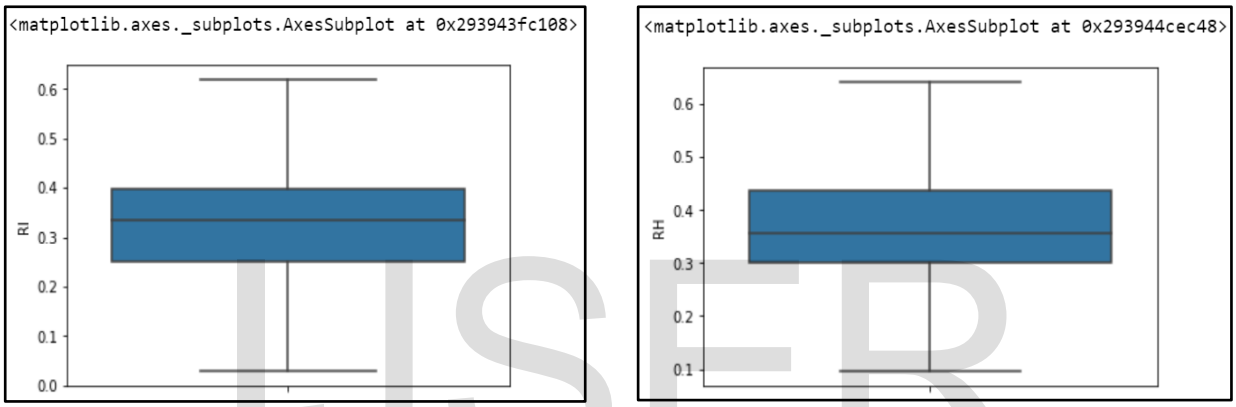
Step 6- Treating Outliers:

Outliers give faulty values for data while building Machine Learning models, thus for the next part it is vital to treat these values. Mathematically outliers are the values that lie outside the range (25% Quantile - 1.5(IQR)) and (75% Quantile + 1.5(IQR)) where IQR refers to Inter Quartile Range and the above range corresponds to values greater than 3 Standard deviations from the mean. The following are the Formulas and Iterative codes which were used to treat the outlier values for variable RI (Similar code executed for RH and RM):

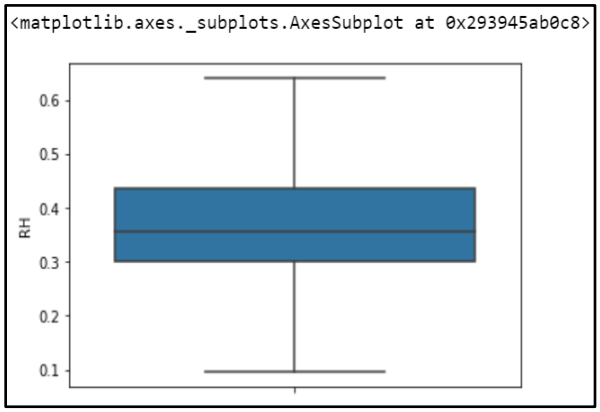
```
IQR_RI = eda['RI'].quantile(0.75)-eda['RI'].quantile(0.25)
Upper_OutlierLimit_RI = eda['RI'].quantile(0.75) + 1.5*IQR_RI
Lower_OutlierLimit_RI = eda['RI'].quantile(0.25) - 1.5*IQR_RI
OutlierValues_RI = eda[(eda['RI']>=Upper_OutlierLimit_RI) | (eda['RI']<=Lower_OutlierLimit_RI)]
```

```
for i in range(0,150):
    if eda['RI'][i] >= Upper_OutlierLimit_RI:
        eda['RI'][i] = Upper_OutlierLimit_RI
    elif eda['RI'][i] <= Lower_OutlierLimit_RI:
        eda['RI'][i] = Lower_OutlierLimit_RI
    else:
        continue
```

After the outliers have been treated the Boxplots for the variables RI, RH and RM are as follow:



Graph 3.4a: Boxplots without outlier values of Infected probes and Healthy probes



Graph 3.4b: Boxplots without outlier values of Mixed probes

Step 7- Dummy Variable Creation for Descriptive Data:

This is the last step of Exploratory Data Analysis. None of our 3 variables is Descriptive Data thus this step is not required for our dataset.
Dummy variable creation converts Descriptive Data to Quantitative Data so that it can be processed by Machine Learning Models.
Our dataset is now ready to be used for creation of a Machine Learning Model.

Part 4: Building a Regression Model to Predict for Amount of Infection(RI) for test values of Amount of Healthy(RH) and Mixed(RM) Probes:

The Exploratory Data Analysis on our Dataset has now been completed and our data is ready to be used for building a Machine Learning Regression Model for performing prediction of Amount of Infection (RI) for test values of Amount of Healthy (RH) and Mixed (RM) Probes.

The following steps will be followed for the same:

1. Splitting Data into Train and Test sets for performing Machine Learning.
2. Training the Regression Model with the Train set and analyzing its features.
3. Calculating Prediction set for the given Independent Test data.
4. Comparing Test set of dependent data to Prediction set to determine model accuracy.

Step 1- Splitting of Data in Tran and Test sets for performing Machine Learning:

The dataset has been divided into train and test data. The train data will be used to build the model and the test data will be used to analyze the model accuracy.

In this paper we intend to predict the Amount of Infection (RI) thus we will take RI and the Dependent Variable(y) and RH, RM will be the Independent Variable(x).

75% of the dataset has been taken as Train data and 25% as Test data.

Step 2-Training the Regression Model with the Train set and analyzing its features:

Since the dependent variable RI is a Continuous Variable, we will be making a Regression Model to perform prediction.

Use the Train Independent and Dependent Data sets to Build the Regression model.

Following are the Coefficient and Intercept of the built Regression Model for the Train data.

```
lr.coef_
array([0.07212714])

lr.intercept_
0.3085886967189197
```

Fig 4.1: Coefficient and Intercept of the Linear Regression Model Built

Step 3- Calculating Prediction set for the given Independent Test data:

Now that the model has been built based on the Test Independent data, we should be able to calculate a predicted Test dependent data.

Following is the data predicted using the Regression model built:

```
pred_Y_lr=lr.predict(test_X)
pred_Y_lr
```

```
array([0.3406452 , 0.33463461, 0.33308471, 0.33910403, 0.33678385,
       0.32800754, 0.33918809, 0.33434839, 0.3259987 , 0.33594727,
       0.32756952, 0.33700242, 0.33950033, 0.34404102, 0.33681236,
       0.33121682, 0.32589921, 0.33263108, 0.33481675, 0.3300319 ,
       0.32756952, 0.32862401, 0.32772447, 0.33809525, 0.34793077,
       0.31618103, 0.33864167, 0.34187815, 0.31542412, 0.33455447,
       0.32662048, 0.3338332 , 0.33563637, 0.34465227, 0.33022684,
       0.34231048, 0.33580649, 0.33263108])
```

Fig 4.2: Predicted Values of infection by the Linear Regression Model

Step 4- Comparing Test set of dependent data to Prediction set to determine model accuracy:

Following is the Test dependent data that was initially split from the dataset:

```
test_Y
```

```
array([0.33333333, 0.61956522, 0.33962264, 0.30769231, 0.30909091,
       0.30769231, 0.33333333, 0.35714286, 0.44827586, 0.31034483,
       0.36842105, 0.36363636, 0.28571429, 0.54237288, 0.30434783,
       0.33333333, 0.2       , 0.22222222, 0.06060606, 0.40540541,
       0.47368421, 0.27777778, 0.12244898, 0.31168831, 0.27272727,
       0.31578947, 0.08333333, 0.38461538, 0.33333333, 0.4       ,
       0.25      , 0.02826087, 0.33333333, 0.33333333, 0.4       ,
       0.61956522, 0.37735849, 0.02826087])
```

Fig 4.3: Actual Values of infection by the Linear Regression Model

Calculate the Mean Absolute Error and the Root Mean Squared Error in order to determine Model Accuracy for Prediction of the Amount of Infection (RI) for test values of Amount of Healthy (RH) and Mixed (RM) Probes in the given Sample Space.

The mean_absolute_error function computes mean absolute error, a risk metric corresponding to the expected value of the absolute error loss or l1-norm loss.

If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean absolute error (MAE) estimated over n_{samples} is defined as

$$\text{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|.$$

For the Predicted Dependent Data the Calculated Mean Absolute Error is: 9.1 %

```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(test_Y, pred_Y_lr)
```

```
0.09129498562130552
```

Fig 4.4: Mean absolute error = 0.091

The mean_squared_error function computes mean square error, a risk metric corresponding to the expected value of the squared (quadratic) error or loss.

If $\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over n_{samples} is defined as

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

For the Predicted Dependent Data the Calculated Mean Squared Error is: 1.7%

```
from sklearn.metrics import mean_squared_error
mean_squared_error(test_Y, pred_Y_lr)
```

```
0.017729235631678563
```

Fig 4.5: Mean square error = 0.017

Thus, the machine learning model was successfully built and the Prediction for calculation of Amount of Infection in DNA Microarray Sample Space was successfully done with minimal error.

IV. REFERENCES

[1]. https://microbenotes.com/dna-microarray/

[2]. https://www.nature.com/scitable/definition/microarray-202/#:~:text=A%20microarray%20is%20a%20laboratory,known%20DNA%20sequence%20or%20gene.

[3]. https://www.aimspress.com/fileOther/PDF/Bioengineering/bioeng-04-00179.pdf

[4]. https://www.ibm.com/in-en/topics/computer-vision#:~:text=Computer%20vision%20is%20a%20field,recommendations%20based%20on%20that%20information.

[5].https://en.wikipedia.org/wiki/Gaussian_blur#:~:text=In%20image%20processing%2C%20a%20Gaussian,image%20noise%20and%20reduce%20detail.

[6]. https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html

[7]. https://artsandculture.google.com/entity/rgb-color-space/m0151f3

[8]. https://handwiki.org/wiki/Color_spaces_with_RGB_primaries

[9]. https://en.wikipedia.org/wiki/CIELAB_color_space

[10]. https://chromatone.center/theory/color/models/perceptual.html

[11]. https://www.dataquest.io/blog/15-python-libraries-for-data-science/

[12]. Data Analysis and Visualization Using Python, Author: Dr. Ossama Embarak, Issue: 2018.

[13]. https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

[14]. https://towardsdatascience.com/build-the-story-around-data-using-exploratory-data-analysis-and-pandas-c85bf3beff87

[15]. https://dbpedia.org/page/Interquartile_range

[16]. https://www.tutorialspoint.com/python_data_science/python_heat_maps.htm

[17]. Statistics for Data Scientists, Source Name: SAS for R Users, Issue: 2019, Page Number: 159-182