

MICROORGANISM DETECTION USING RAMAN SPECTROSCOPY AND C-ICA

Final Year Project Report

Bachelor of Bioengineering

in the School of Chemical and Biomedical Engineering

Nanyang Technological University

Singapore

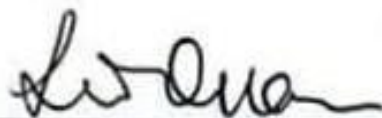
By

GOH HAN LONG, EUGENE

U1520852K

YEAR

2017/2018



Assoc. Prof. Liu Quan

Supervisor

School of Chemical and Biomedical Engineering
Nanyang Technological University

ABSTRACT

Microbial keratitis is an infection of the cornea that is caused by a variety of non-viral pathogens. It is the most potential complication of contact lens wear. Left untreated in time, it can cause serious damage to the eyes, to the point of rendering the patient blind.

In this study, five sets of Raman Spectroscopy data were provided and processed using machine learning techniques. Principal Components-Linear Discriminant Analysis (PC-LDA) was first performed to classify the data and to obtain the accuracy of classifying each set of data. Next, Constrained Independent Component Analysis (C-ICA) was performed on the same datasets, and the correlation coefficient of the extracted signal was compared against the original signal.

PC-LDA has been tested to be a proven technique in classifying the Raman spectra of the respective pure microorganism samples and the mixed microorganism samples on contact lens, but classification does not necessarily mean detection as there may be unknown contaminants in the sample.

Synthetic data of the microorganism, namely *P. Aeruginosa* and *C. Albicans*, were successfully extracted from a source signal and it was shown that the C-ICA was able to detect the microorganism of interest on the surface of contact lens, even in low dosages.

C-ICA has shown to be potential method of determining the presence of such pathogens, and the importance of which could lead to timely and appropriate treatment of microbial keratitis.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Assoc. Prof. Liu Quan for giving me the opportunity to work on this project, where I am truly fortunate to be able to work on a project that I am passionate about. His guidance and assistance that provided me the tools to work on this project is greatly appreciated.

I would also like to thank Ms. Bai Yanru for her constant supervision and important feedbacks. Her teachings about the applications of algorithms and data analysis in relation to the project will not be forgotten. Without her guidance and help, the completion of this project would not have been possible.

Table of Contents

List of Figures.....	v
Chapter 1: Introduction	1
1.1 Background	1
1.2 Objective	4
1.3 Scope	4
Chapter 2: Literature Review	5
2.1 Raman Spectroscopy and applications to microbial analysis.....	5
2.2 Pre-processing of Raman spectroscopy data	8
2.3 Data analysis using C-ICA	10
Chapter 3: Experimental.....	13
3.1 Data Preparation	13
3.2 PC-LDA	15
3.3 RICA	23
3.4 C-ICA	31
Chapter 4: Discussion and Conclusion	40
4.1 Comparison of RICA and C-ICA	40
4.2 Summary	44
References	
Appendix	

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1	Raman Spectra of various bacteria	2
2	PCA and LDA on bacterial classification	3
3	Set-up of Raman spectroscopy instrumentation	7
4	Comparison of reference signal and output signal by C-ICA	12
5a	Original signal obtained from Raman Spectroscopy	14
5b	Signal obtained after smoothing and subtracting baseline	14
6	Flow chart for PC-LDA	18
7a	LDA Scatter Plot of Pure CA, CA on CL, PA, PA on CL	19
7b	LDA Scatter Plot of CA, CA on CL	20
7c	LDA Scatter Plot of CA on CL, PA on CL	20
7d	LDA Scatter Plot of CA, PA	21
7e	LDA Scatter Plot of PA, PA on CL	21
8	LDA line vectors for CL, CA on CL, CA	22
9	Flow chart for Reconstruction ICA	27
10	RICA on CL, CA, Noise	28
11a	Correlation Coefficient between reference and unmixed signal (PA)	30
11b	Correlation Coefficient between reference and unmixed signal (CA)	30

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
12a.	Extraction of CA signal using raw CA signal as reference	34
12b.	Extraction of CA signal using threshold CA signal as reference	34
12c.	Extraction of PA using raw PA signal as reference	35
12d.	Extraction of PA using threshold PA signal as reference	35
13.	Extraction of CL and Noise signals	36
14a.	SNR against dilution of CA	37
14b.	SNR against dilution of PA	37
15a.	SNR against dilution of CA when the random noise is not set to default	38
15b.	SNR against dilution of PA when the random noise is not set to default	38
16a.	RICA on raw measurements of CA on CL	40
16b.	RICA on raw measurements of PA on CL	41
17a.	C-ICA on raw measurements of CA on CL	42
17b.	C-ICA on raw measurements of PA on CL	43
18.	Calculations for Dilution Factor	62

INTRODUCTION

1.1 Background

Microbial keratitis is an infection of the cornea and it is a severe complication contact lens wear, as prolonged wear and inadequate lens disinfection significantly increases the infection rate. Timely treatment is of paramount importance as it leads to permanent blindness.

P. Aeruginosa and *C. Albicans* are common pathological agents of this disease, and proper identification of the pathogen is necessary as the treatment method would differ. Traditional methods of identification include microscopic analysis of smears, gram staining of microbial culture and molecular diagnostic techniques^[1]. However, these methods require culture preparation prior to identification.

Raman spectroscopy is a non-invasive optical technique that uses light scattering to characterize the chemical composition and molecular structure of a sample. It produces a graph comparing the intensity of scattered light against the frequency of the laser, as shown in Figure 1. Technological advances made in the past decade have established it to be the gold standard in analyzing microorganisms^[2]. Although it is powerful, there are still drawbacks to this modality.

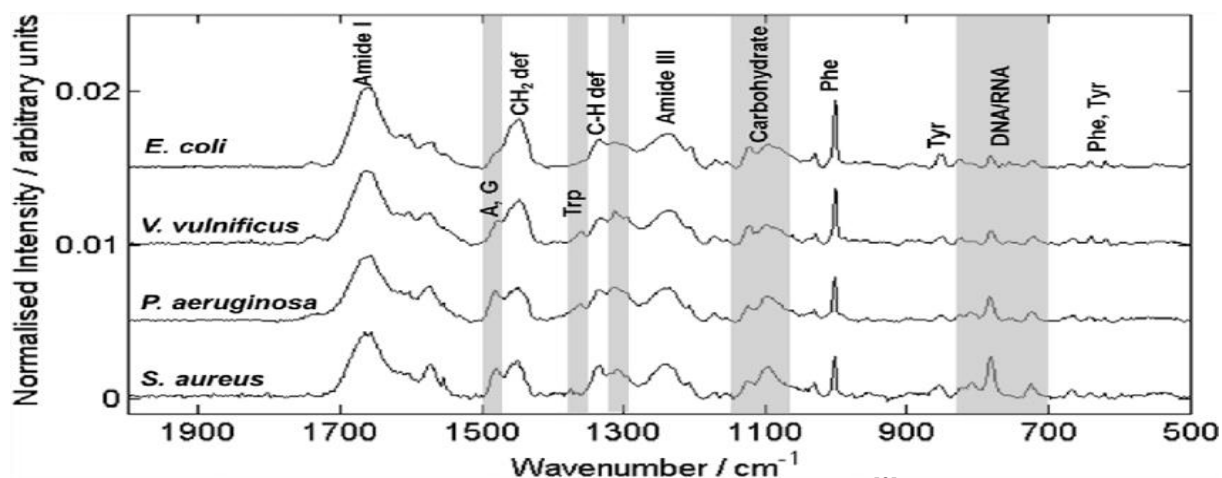


Figure 1. Raman spectra of various bacteria^[3]

Due to the small scale of the sample, the signal will be too weak for analysis. One method to enhance the signal is Surface-Enhanced Raman Spectroscopy (SERS), whereby a nanosized metallic platform containing gold or silver nanoparticles is used to significantly amplify the signal. However, this comes at the cost of additional preparation time and does not align to the objective of the project^[2].

Classification of the analyte is also subjective to the algorithm used. Principal Component Analysis with Linear Discriminant Analysis (PC-LDA), as shown in Figure 2, is one widely adopted method due to its advantages of being simple to implement with a fast computing speed. However, it fails to perform when the correlations between the samples are high and when the signal contains significant noise^[4].

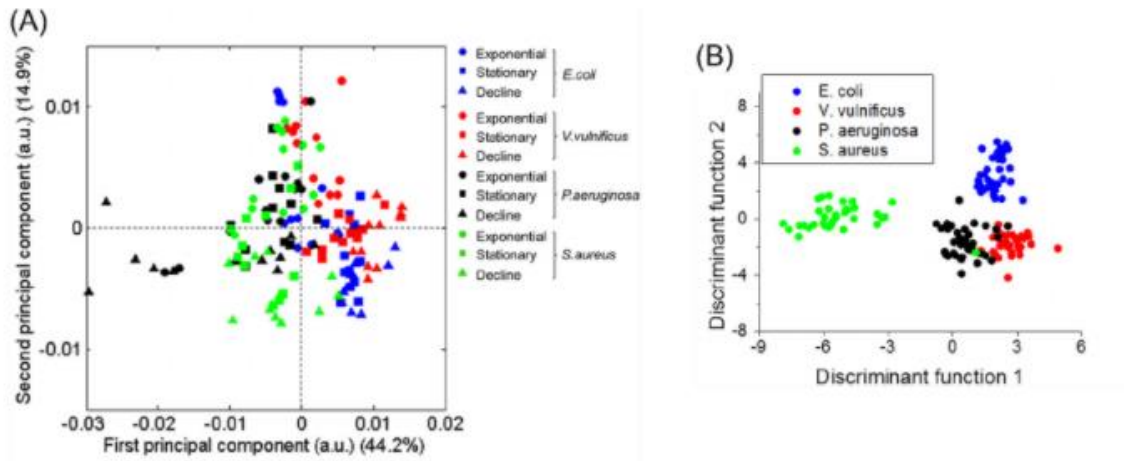


Figure 2. PC-LDA on bacteria classification^[3]

Independent Component Analysis (ICA) is an algorithm that separates linearly-correlated mixed sources by maximizing their non-Gaussianity^{[5][6]}. Unlike PC-LDA, it is less commonly used in spectrum analysis because it cannot classify components. Constrained Independent Component Analysis (C-ICA) is a modification of the algorithm that uses *a priori* information to extract the signal of interest. It has not been widely applied to microorganism analysis as it is computationally expensive.

Wang et al. has demonstrated that C-ICA is able to detect a certain chemical agent, and the successful application in this case could remove the need for any preparation^[7]. It can be assumed that approximate knowledge exists on the causative agent of this disease and its “fingerprint” is known from a stored database. This would not only save labor, resources and administrative costs, but also reduce the risk of permanent damage to the eyes.

1.2 Objective

There are two objectives of this project; the first part is to classify the Raman spectra of the pure bacteria samples (*P. Aeruginosa* and *C. Albicans*) and the respective bacteria samples on contact lens using PC-LDA. The second part is to use C-ICA on the same database to extract the signal of pure bacteria from the mixed signal. For the second part, Reconstruction ICA (RICA) will be performed prior to C-ICA to check if the signal is able to be decomposed. The results of RICA will then be compared against C-ICA to determine the effectiveness of the C-ICA algorithm.

1.3 Scope

Five sets of Raman spectroscopy data were provided for this experiment, namely 'Contact Lens' ('CL'), which is the spectra of purely contact lens, 'Pure PA', which is the spectra of pure *P. Aeruginosa*, 'Pure CA' which is the spectra of pure *C. Albicans*, 'PA on CL' which is the spectra of *P. Aeruginosa* on Contact Lens, and 'CA on CL' which is the spectra of *C. Albicans* on Contact Lens.

Each given sets of data will be processed by PC-LDA and C-ICA, and the accuracy of both methods will be validated using k-fold cross validation and correlation coefficient respectively. The preparation of the microorganism samples will not be discussed as synthetic data was used.

LITERATURE REVIEW

2.1 Raman Spectroscopy and applications to microbiological analysis

Raman spectroscopy makes use of the phenomenon of Raman Scattering. Co-discovered by C.V. Raman and K.S. Krishnan in the late 1920s, it can measure the chemical and molecular composition of a sample. The biological composition can be derived from the obtained information, as well as other parameters such as crystallinity, temperature and thickness.

It is based on a scattering technique – when monochromatic light is passed through a non-absorbing medium, most of the light particles (photons) will be transmitted without any change to its energy, frequency and wavelength. This phenomenon is known as Rayleigh scattering. However, some of the photons will be inelastically scattered, in an extremely low probability of occurrence of about 1 in 10^8 , where it will have a different frequency from the incident light and will be used to construct a Raman spectrum. When energy is lost, the Raman scattering is designated as “Stokes” while when energy is gained, it is designated as “anti-Stokes”^[8].

In the analysis of microorganisms, Raman spectroscopy offers high molecular specificity as well as a label-free and non-invasive method to characterize a biological sample based on their distinctive biological “fingerprint”. This is possible because biological samples contain reliable taxonomic markers such as DNA, proteins, lipids, carbohydrates and

other cellular biomolecules that each has a unique chemical bond, and the interaction between the incoming photons and respective chemical bonds give rise to a unique range of wavenumbers associated with their vibrational modes.^[9]

Raman spectroscopy can be performed using a Raman microscope or a handheld spectrometer. The basic instruments include a laser source, a filter for collecting Raman scattered light, a diffraction grating and a Charged Couple Device (CCD) detector. At the start of the experiment, a laser of a pre-selected excitation wavelength is focused onto a fiber-optic cable. Raman scattering is removed from the optical filter as it passes through a laser line filter. The beam is then reflected by a dichroic mirror and concurrently, the laser will be focused on the sample. Only Raman scattered light is transmitted through the mirror while Rayleigh scattered light is reflected.

The Raman scattered light is filtered once more through a notch filter to remove the remaining Rayleigh scattered light. The light is reflected by a mirror to exit the microscope to enter the spectrometer, where it will be separated according to its wavelength. A CCD detector will then measure the Raman shift and this input will be processed by a computer. A diagram of the set-up is shown in Figure 3.

Finally, the data is displayed in the form of a graph by a software, which gives us the Raman spectra of the sample. Although there are many different techniques that Raman spectroscopy can be implemented, each with their respective benefits and limitations for specific applications, this basic working principle remains the same^[10].

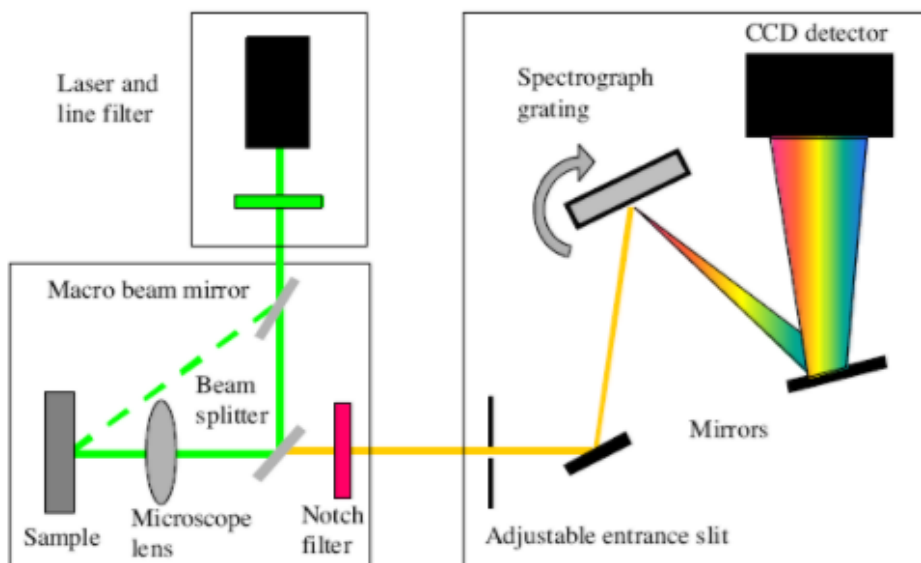


Figure 3. Set-up of Raman spectroscopy instrumentation^[11]

In the application of microorganism analysis or in situations where the quantity of the sample is small, SERS is often applied to amplify the signal. In this case, the microorganism of interest is analyzed on a platform of silver or gold colloids. It makes use of the concept of Surface Plasmon Resonance (SPR), where light from the laser is absorbed by the gold or silver colloids, causing their electrons to resonate. This excitation of the surface plasmons can enhance the signal up to seven orders of magnitude, which makes the pre-processing and identification step considerably easier^[12].

2.2 Pre-processing of Raman spectroscopy data

Data processing can be done in three steps: (i) data set pre-processing, (ii) feature extraction and (iii) classification or identification^[9]. Due to the large amount of datasets generated in Raman spectroscopy studies, pre-processing of the signal must be done to remove noise, so that accurate biochemical information of the sample can be obtained. Noise in Raman spectra can come from various sources, including the hardware itself, natural fluorescence of the sample, contaminants, and cosmic ray spikes. Because there is no fixed method to perform data pre-processing of Raman spectroscopy data, the amount of variability is extremely high. Cross-validation is done to judge the best method to use, which allows us to select the final predictive model.

For data pre-processing, the most obvious type of noise to be noticed is the cosmic ray spike, where a small portion of the graph will be significantly higher than the rest. Along with other minor noise in the signal, a smoothing filter, such as Moving Average or Savitzky-Golay, can be applied to remove such abnormalities. Next, baseline removal is performed on the smoothed spectra to identify the major peaks of the sample. Normalization is usually done after baseline removal to correct for sample and experimental variables. Resampling is also done in this step to ensure that all of the datasets from different samples have the same data points, where the data can be “up-sampled” by extrapolating additional data points, or “down-sampled” or to remove redundant data points.

Feature extraction is the next step to be done and it is related to dimensionality reduction. This step is important in the processing of Raman spectra to remove the redundant part of the signal while keeping those that contain important information about the chemical composition and molecular structure. Feature extraction is subjective to the algorithms we wish to apply on the data and it can range from simply defining an area of the graph to using complex computational methods to transform the data. Ultimately, the goal of this step is to select the algorithm that will lead to better interpretation of the data.

Classification of Raman spectroscopy data is done after the important features have been extracted from the data. It attempts to identify and categorize the data for storage and further analysis. In this step, a “training set” of data is provided containing prior data whose category membership is already known. New data, also known as the “test set”, will be compared against the “training set” and will be respectively assigned to their categories. Classification can either be supervised or unsupervised; supervised classification is guided by humans and unsupervised classification is calculated by software.

After performing the classification of the respective data, cross-validation is the final step done to assess the performance of the algorithm. Naturally, we want to select the algorithm that gives us the higher sensitivity (true positive rate). The result of this step is the ‘Accuracy’ of the algorithm, where it will be selected as a standard for other studies of the same purpose.

Identification is another method that can be performed on Raman spectroscopy data after feature extraction. Since it has been established that each sample has their distinctive “fingerprint” by their unique vibrational frequencies, it is possible to save a collection of these data into a database for future references. When a new sample needs to be identified, the Raman spectra of the new sample can be compared against existing signals (where the composition is known) through “Correlation Coefficient” to check the similarity of both samples. If the Correlation Coefficient of both samples are above a certain threshold, it can be assumed that the new sample is the same as the existing sample that it was compared against.

2.3 Data Analysis using C-ICA

C-ICA is an algorithm designed to extract a particular signal from a set of mixed signals from even a rough template of the reference signal, and it discards the other “uninteresting” signals. It is an extension of ICA that provides a framework to incorporate *a priori* knowledge of the Raman spectroscopic data of the microorganism that we want to analyze. Prior information can be implemented in the form of constraints into the contrast functions of ICA, which will be explained into detail in the next chapter.

Some applications of C-ICA includes being successfully applied to extracting signals of interest from the brain obtained from Magnetic Resonance Imaging or Electroencephalography, where the detection and identification of these signals could provide a greater specificity than just the classification of such signals.^[13] Because it is

not as sensitive to noise in the original signal, it holds much promise in detecting even a small dose of the microorganism, which suits our application as the standard amplification techniques, such as SERS, are not used on the sample beforehand.

Building on the first advantage of offering a higher accuracy, the other benefit that this method provides in the detection of the microorganism is that the constrained signal, or reference signal provided to the algorithm, need not be a perfect match to the sample we wish to analyze. As long as the reference signal contains enough important points, in this case, the peaks of the Raman spectra, it is possible to point the algorithm towards a particular Independent Component which we are interested in and to obtain a convergence, which indicates that the microorganism is present in the sample.

This is shown in a study conducted by Lu and Rajapakse, where they have deployed C-ICA to demonstrate the efficacy and accuracy of the algorithm on fMRI data. From their results shown in Figure 4, we know that we do not need to design an exact replica of the signal in order to extract it. The reference signals are simulated by a series of pulses having the same period as the desired sources.

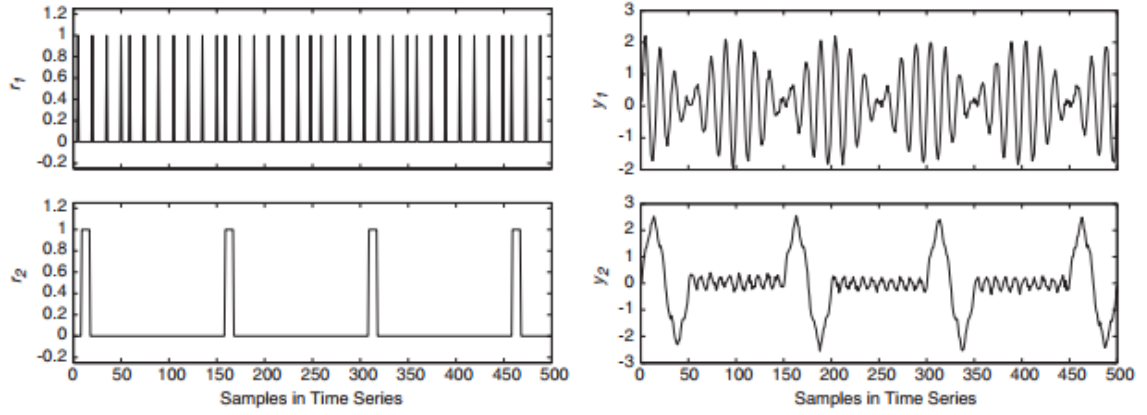


Figure 4. Comparison of reference signal (left) and output signal by C-ICA (right)^[14]

Relative to this project, this method is particularly attractive to use as it can be assumed that the contact lens sample directly taken from the patient will be analyzed wholesale, and the Raman spectra obtained will contain the bacteria we want to detect, as well as traces of other contaminants such as eye lashes, tears and dust.

EXPERIMENTAL

3.1 Data Preparation

From a given spreadsheet containing 1 measurement of 'CL', 8 measurements of 'Pure CA', 3 measurements of 'CA on CL', 10 measurements of 'Pure PA', and 3 measurements of 'PA on CL', it must be ensured that the data points of all the samples must be the same before subjecting it to any processing method.

The Raman spectra of 'Pure CA' and 'Pure PA' had 1574 data points whereas 'CL', 'CA on CL' and 'PA on CL' had only 1029 lines of data. A MATLAB code was written to standardize the 'Pure CA' and 'Pure PA' to 1029 lines of data via downsampling and removing some data points to fit the range of 300cm^{-1} to 2000cm^{-1} . The downsampling code is found in Appendix A.

Next, the table was transferred to OriginPro 9.1 and the signals were then smoothed using a 5-point Adjacent Averaging filter, which takes the average of the adjacent values in a 5-point window. This filter was used due to the ease of implementation that could provide a clear result as we only want to remove white noise from the signal^[15].

The baseline could then be subtracted by the Peaks Analyzer built-in application, where the peaks of the spectra are selected and a new graph will be plotted. Lastly, the data was normalized from [0 to 1] to ensure all of the data points are on the same scale. This

process was repeated for every dataset across the 5 tables, and the overall average was obtained. Figures 5a shows the original signal without any processing done, and Figure 5b is the signal after subtracting the baseline.

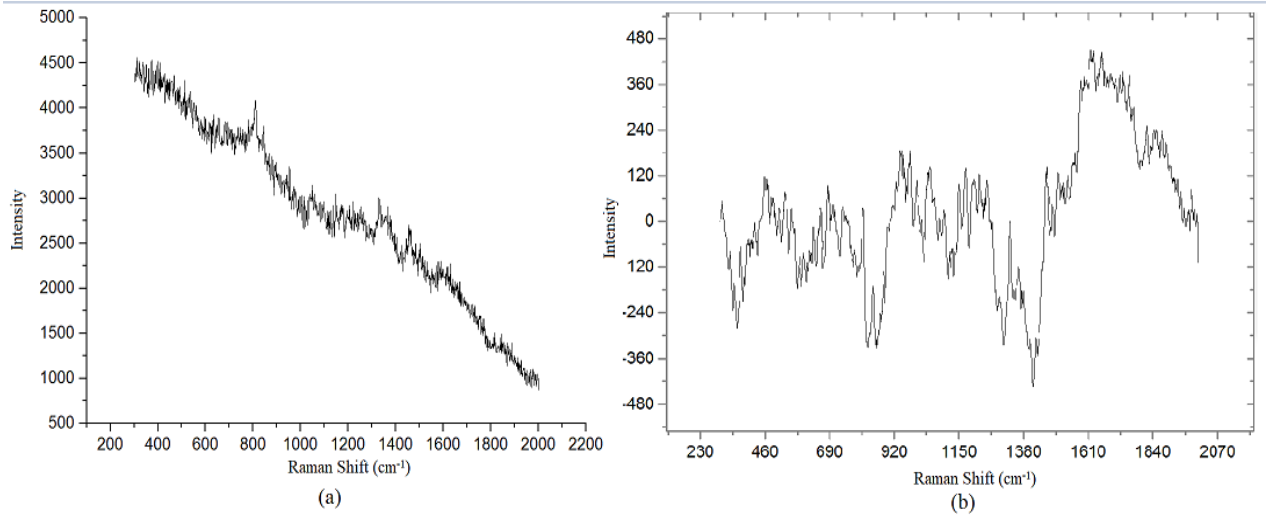


Figure 5. (a) Original signal obtained from Raman Spectroscopy
(b) Signal obtained after smoothing and subtracting baseline

The method used to subtract the baseline is based on Polynomial Fitting, which is one of the most commonly used methods for baseline removal. Selecting the right order of polynomial to subtract from the baseline is extremely important as a high polynomial fit would remove some important Raman bands, while a low polynomial fit would not be very efficient in subtracting the baseline. The fitting methods used in this experiment were either the 5th order polynomial or a cubic polynomial as a baseline estimate, depending on which is able to better preserve the Raman band spectra^[16].

3.2 PC-LDA

Principal Components-Linear Discriminant Analysis is a popular classification algorithm used as a finishing step for the analysis of Raman spectra due to its simplicity and ability to reduce the dimensions of large datasets into several Principal Components (PCs). It identifies PCs through orthogonal transformation, where the first PC would have the highest amount of variance (spread around the mean), the second PC would be orthogonal to the first PC, and so on. Mathematically, the PCs are eigenvectors of the covariance matrix of the original dataset, which correspond to the direction of the greatest variance in the data.

The mathematical expression of the transposition can be written as:

$$X(\lambda) = u_1 p_1(\lambda) + u_2 p_2(\lambda) + \dots + u_n p_n(\lambda) = \sum_{i=1}^n u_i p_i(\lambda) \quad (1)$$

where $X(\lambda)$ is the raw data from Raman spectroscopy, $p(\lambda)$ is the orthogonal vector with the most variation and u is the value of the new orthogonal space. The result of which is to represent $X(\lambda)$ as a single point in the new space^[17].

While performing PCA, normalization of the data first as attributes with large variances will end up dominating the first PC when they should not. Normalization puts each attribute on the same scale, so that each attribute has an opportunity to contribute in the calculation. After centering and standardizing the data, we get the covariance matrix of the dataset and perform Singular Vector Decomposition on it to get the eigenvectors and

eigenvalues. The results would then be ranked from highest to lowest, where eigenvectors with low eigenvalues bear the least information about the distribution of the data.

To find out exactly how many PCs need to be kept because of their significance, we can get the “explained variance” of the new data, which tells us how much information can be attributed to each of the principal components. The final step would be to plot the datasets on a new feature axis, where the different groups of PCs can be identified visually.

However, PCA by itself is not optimal for classification. PCA is an “unsupervised” algorithm that ignores the class labels, where the goal is to only find the direction with the most variance. In other words, even though the results can be visually obvious, PCA is not a classification method. Furthermore, using PCA alone is unwise from a classification point of view because the discriminant dimensions could be discarded, and unwanted information may be preserved. Fisher’s Linear Discriminant Analysis (LDA) is a “supervised” algorithm that maximizes the separability between groups to make the best decision. The general LDA approach is similar to PCA, but we are additionally interested in the axes that maximize the separability between classes.

For LDA, it is important to first obtain the between-class and within-class scatter matrix, defined by:

$$S_B = \sum_{i=1}^c N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \quad (2)$$

$$S_W = \sum_{i=1}^c (N_i - 1) \Sigma_i \quad (3)$$

where \mathbf{m} is the overall mean of the dataset, \mathbf{m}_i is the individual sample mean and N_i is the size of the respective classes. Next, we solve the generalized eigenvalue problem for $S_W^{-1} S_B$ to get the linear discriminants. We then select the linear discriminants for the new feature subspace and transform it onto the new subspace^[18].

$$J_{\text{Feature}} = \frac{\det(S_B)}{\det(S_W)} \quad (4)$$

In this study, PCA will be used to de-noise and reduce the dimension of the original dataset, and the PCs obtained will be used as the “training” set to build up the model, as well as the “test” dataset to validate the model. LDA will be used to classify the data. The holdout method is used to cross-validate the data to obtain the accuracy. MATLAB 2015b has implemented a ‘Classification Learner’ application that trains models to classify data, which will be used in this project.

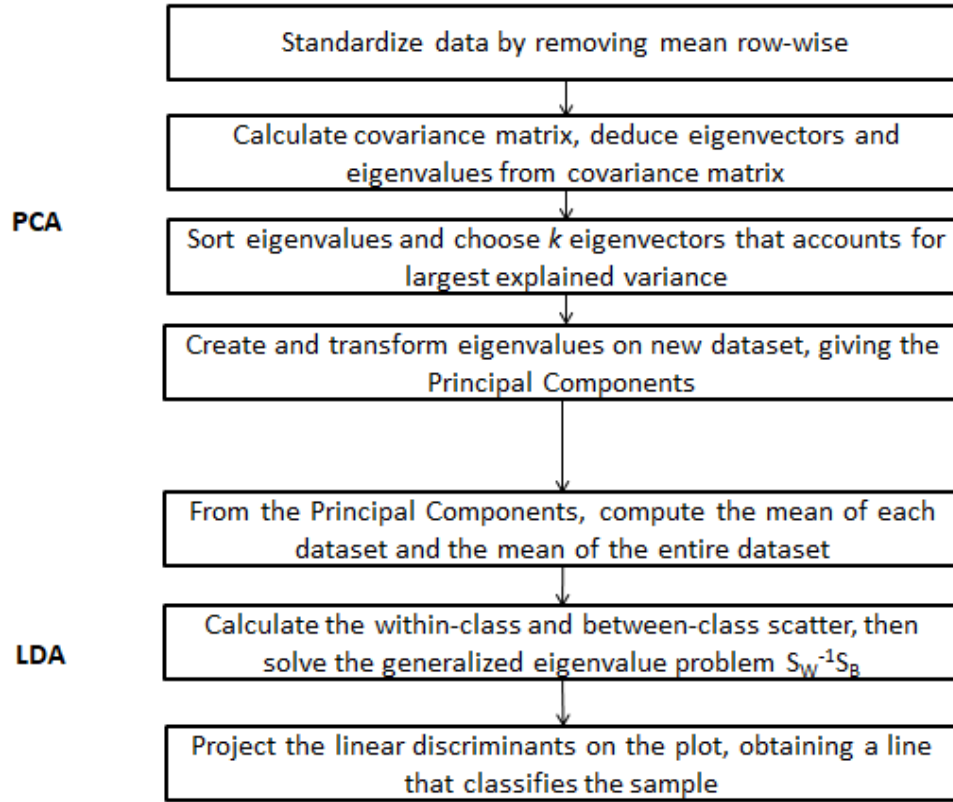


Figure 6. Flow chart for PC-LDA^[19]

Once the data has been prepared on a spreadsheet, the data will be transferred into MATLAB for deploying the respective algorithms. For PC-LDA, PCA is first performed to obtain the Principal Components (PCs) of each spectrum, and it is evident from the results that there is good separability in the different categories. Linear Discriminant Analysis is a built-in model in the “Classification Learner” application in MATLAB 2015b, and in this project, it is used to obtain the Accuracy of the algorithm. The results of PCA are simply loaded onto the software by selecting the variables and the response, which the application will use as a “training set”.

It will automatically generate the accuracy and display the results on a scatter plot, showing the correctly classified PCs, as well as the misclassified PCs. 5-fold Cross Validation was used for this process. This algorithm was deployed 5 times; each time for a different set. Figures 7a to 7e shows the scatter plot results of the respective set.

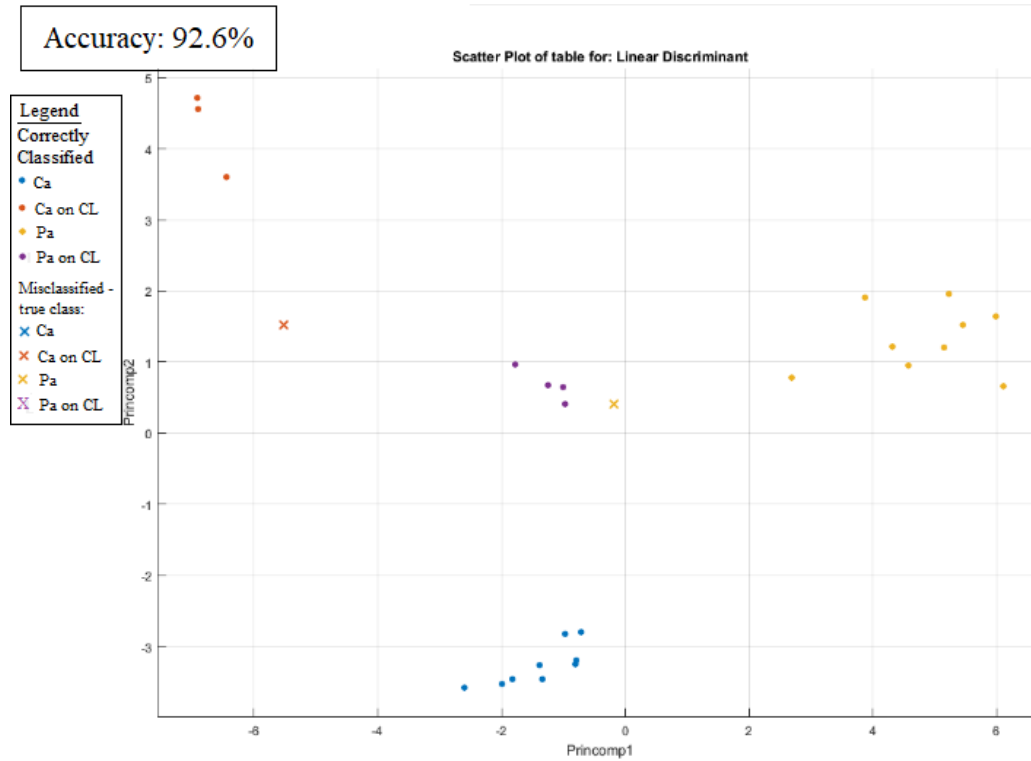


Figure 7a. LDA Scatter plot of Pure CA, CA on CL, PA, PA on CL

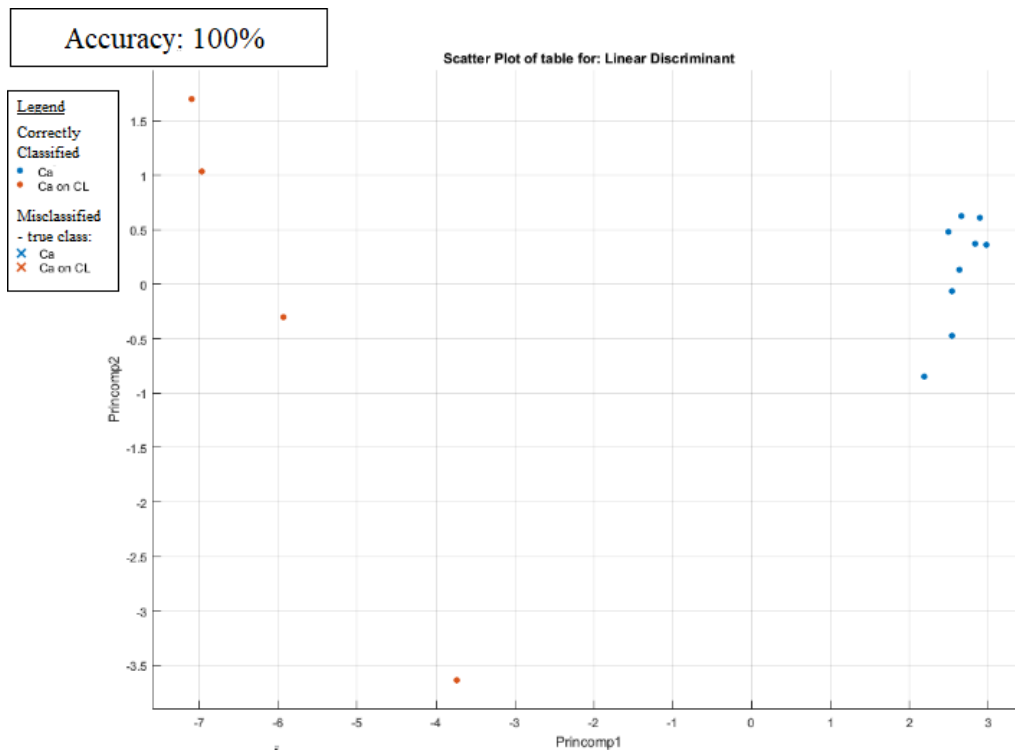


Figure 7b. LDA Scatter plot of CA, CA on CL

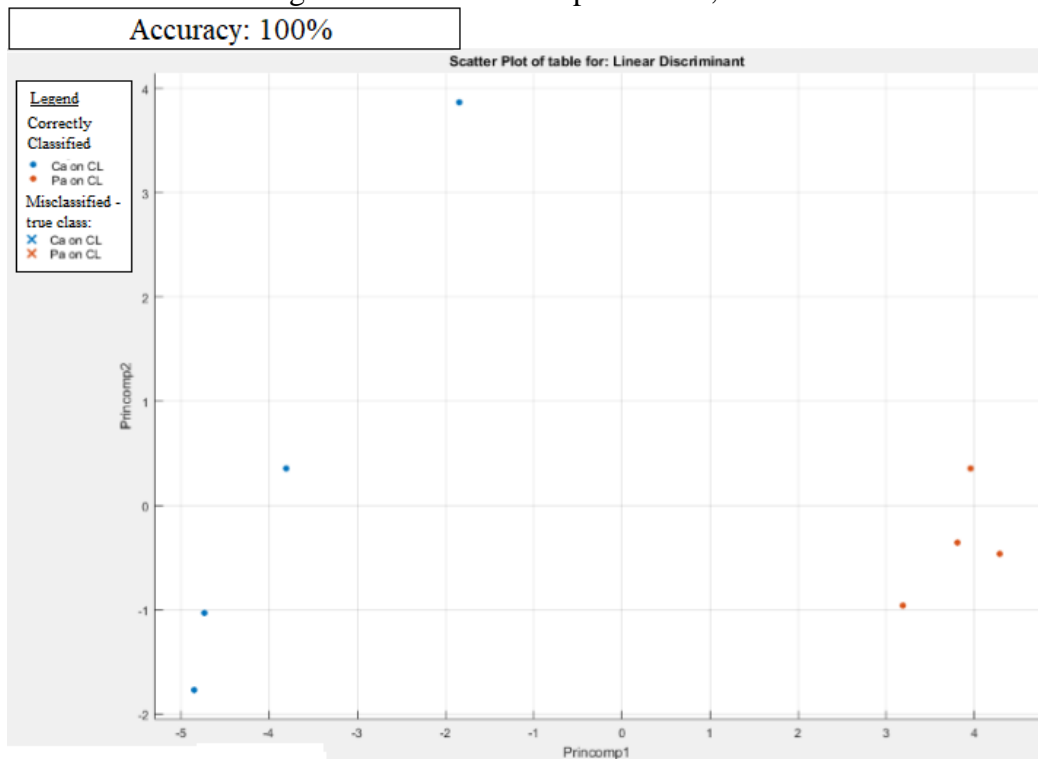


Figure 7c. LDA Scatter plot of CA on CL, PA on CL

Accuracy: 94.7%

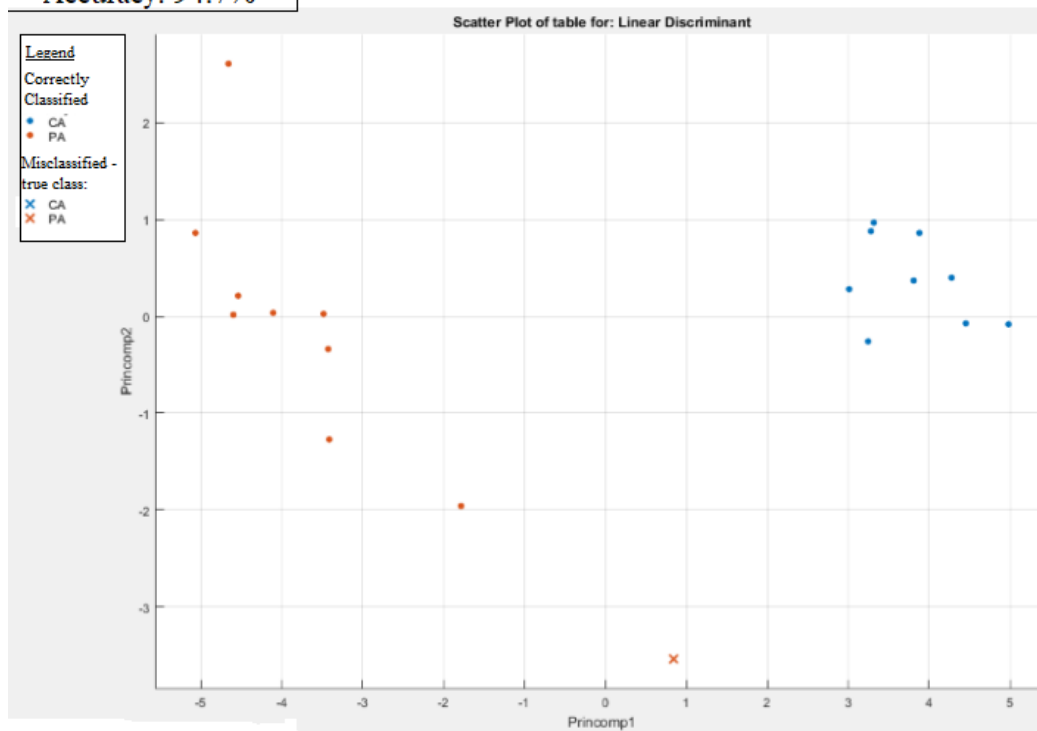


Figure 7d. LDA Scatter plot of CA, PA

Accuracy: 92.9%

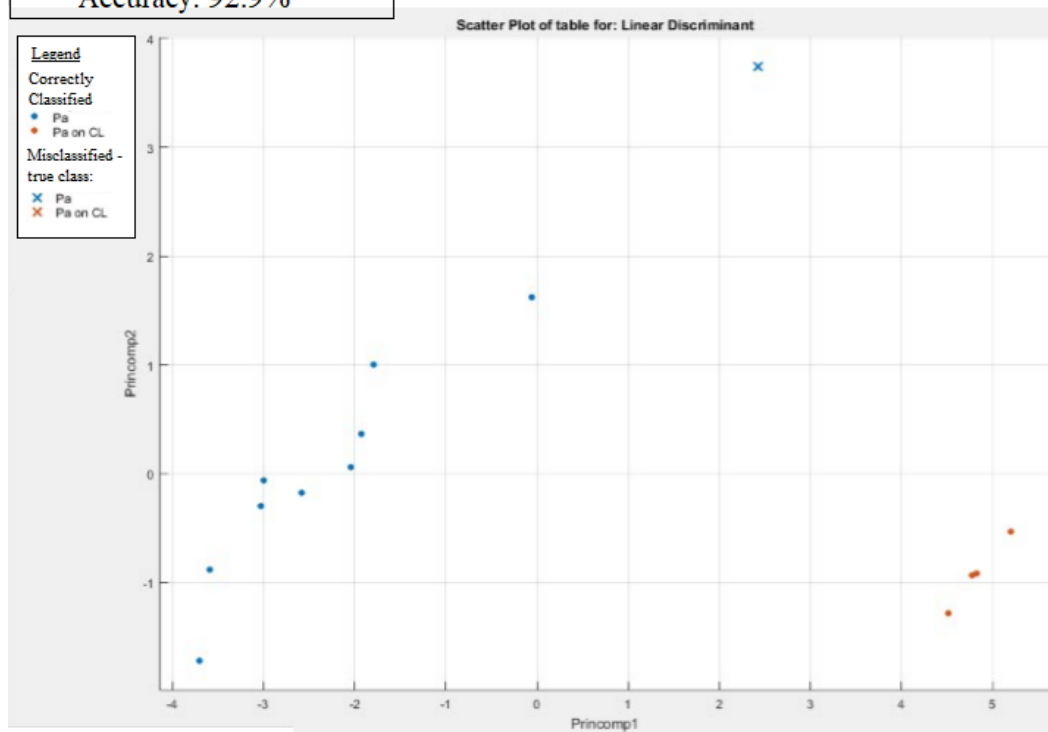


Figure 7e. LDA Scatter plot of PA, PA on CL

A LDA code was written and performed on PCA data, generating line vectors that tells us which group the PC of each spectrum would belong to, shown in Figure 8.

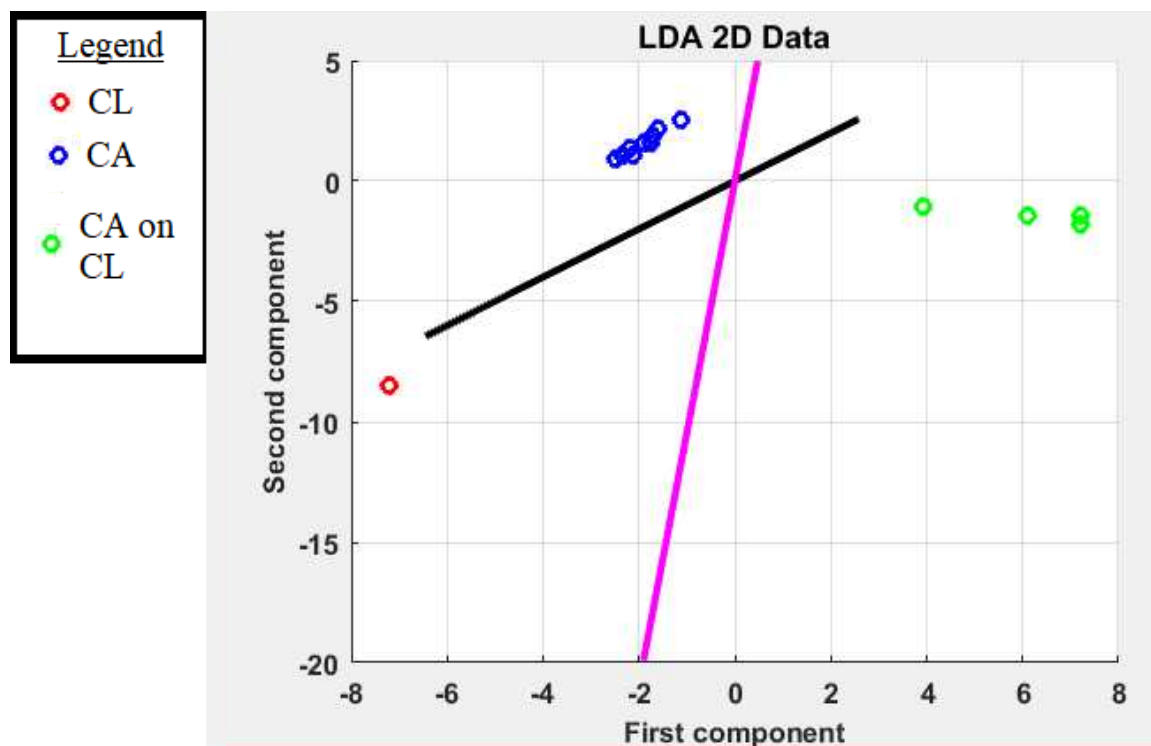


Figure 8. LDA Line vectors for CL, CA on CL, CA

The code for PC-LDA is listed in Appendix A.

3.3 ICA

Independent Component Analysis is another Blind Source Separation (BSS) technique that is related to PCA, but produces very different results. In contrast to PCA, the objective of ICA is to extract a data transformation that maximizes the statistical independence of basis vectors, and as such, ICA treats all components to be equally important and does not discard any data. ICA vectors also are not orthogonal, whereas PCA vectors are.

ICA recovers a set of independent signals from a set of mixed signals, and assumes the mixture is a linear combination of each independent signal. It is a prerequisite of ICA to have an equal number of independent signals and measured signals.

$$X_i = a_1 S_1 + a_2 S_2 + \dots + a_n S_n \quad (5)$$

Expressing all n measured signals:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \text{ or in matrix form, } \begin{bmatrix} x1(t) \\ \vdots \\ xN(t) \end{bmatrix} = \begin{bmatrix} a11 & \dots & a1m \\ \vdots & \ddots & \vdots \\ an1 & \dots & anm \end{bmatrix} \begin{bmatrix} s1(t) \\ \vdots \\ sM(t) \end{bmatrix} \quad (6)$$

Where \mathbf{X} represents the mixed signals; the results directly obtained from Raman spectroscopy containing known and unknown composition of the sample, \mathbf{S} representing the original signals where it only contains the known, pure composition of individual sample and \mathbf{A} is the $n \times m$ mixing matrix that allows us to find \mathbf{X} from \mathbf{S} . Given \mathbf{X} , the raw results, the aim of ICA is to find \mathbf{S} and \mathbf{A} . It should be noted that the recovered “original”

signals can be positive or negative, and does not have an order to which they are recovered.

Since we do not know the values of \mathbf{A} and \mathbf{S} , it is impossible to recover the exact original signal, but only produce an approximation of it. This brings about another equation (7), which is similar to (6) except that we are interested in finding the source signal from the obtained, mixed signal.

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \quad (7)$$

Where \mathbf{Y} is the estimated source signal and \mathbf{W} is the $\mathbf{n} \times \mathbf{n}$ mixing de-matrix. To obtain the scaled and permuted estimated source signal, we can express the de-mixing matrix \mathbf{W}

$$\mathbf{W} = \mathbf{D}\mathbf{P}\mathbf{A}^{-1} \quad (8)$$

Where \mathbf{D} is a non-singular diagonal matrix and \mathbf{P} is the permutation matrix. The mixing matrix \mathbf{A} must also be invertible. ^[20]

Before applying ICA, there are some mathematical rules and definitions that must be understood. Statistical independence means knowing independent random variables tells us nothing about the next, and their covariance is zero. For example, given two variables \mathbf{A} and \mathbf{B} ,

$$\text{Cov}(\mathbf{A}, \mathbf{B}) = \text{mean}(\mathbf{A} * \mathbf{B}) - \text{mean}(\mathbf{A}) * \text{mean}(\mathbf{B}) \quad (9)$$

In the case of finding statistical independent variables,

$$\text{mean}(\mathbf{A} * \mathbf{B}) = \text{mean}(\mathbf{A}) * \text{mean}(\mathbf{B}) \quad (10)$$

Although the covariance of two statistically independent variables must always be zero, it must be noted that it does not apply for the reverse situation. Only in the case of Gaussian variables when zero covariance means independence, and these features of Gaussian variables are used for finding \mathbf{W} in Equation (8).

Another rule that is applied in ICA is the Central Limit Theorem, which establishes that the sum of independent variables (in this situation, the original signals, \mathbf{S}), in most situations, their normalized sum tends towards a normal distribution even when the original variables are not normally distributed. This implies that if the measured signal has minimal Gaussian properties, it is likely that the signal is independent.

There are a few different approaches to perform ICA and estimate the independence of a signal, but the fundamental goal of these methods is to find the \mathbf{W} that maximizes the non-Gaussianity of $\mathbf{Y}=\mathbf{W}\mathbf{X}$. All ICA methods are optimization-based process, and the three methods to be understood are Kurtosis, Negentropy and Negentropy approximation. Kurtosis is the easiest approach to determining the independence based on the skewness of the curve, but it is very sensitive outliers and thus, not robust enough for our application.

Negentropy is used as a measure of distance to normality, which uses the concept of entropy to measure the non-gaussianity of a signal. The entropy of a random variable can be also interpreted as the degree of information it gives. A discrete signal would then have an entropy equal to the sum of products of probability of each event and the

logarithm of those probabilities. Negentropy can then be derived from the differential entropy of a signal, \mathbf{Y} , minus the differential entropy of a Gaussian signal with the same covariance of \mathbf{Y} . It is a better method to estimate Independence compared to Kurtosis, but it is difficult to calculate.

A computationally less expensive method has been developed to approximate the negentropy of a signal instead of directly calculating it, given by the equation:

$$J(\mathbf{x}) \approx \sum_{i=1}^p k_i [\text{mean}(G_i(\mathbf{x})) - \text{mean}(G_i(\mathbf{v}))]^2 \quad (11)$$

Where $G(\mathbf{x})$ is a non-quadratic function, \mathbf{v} is a Gaussian variable with unit variance and zero mean and k_i is a constant value. Two most commonly used functions for G that gives a good approximation to negentropy are:

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u \quad (12)$$

$$G_2(u) = -e^{-\frac{u^2}{2}} \quad (13)$$

These are known as “contrast functions”^{[21][22]}.

Relative to this project, the software MATLAB 2017b has implemented a Reconstruction ICA (RICA) Algorithm that is able to perform this function of feature extraction. The algorithm is based on minimizing an objective function and generally works the same as ICA, except with a reconstruction cost. It finds the mixing matrix using the limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) quasi-Newton optimization

algorithm, which will not be discussed in this report. It attempts to obtain a near orthonormal weight matrix that minimizes the sum of elements $g(XW)$, where g is a contrast function such as in Equation (12), where $a_1=2$. After the algorithm obtains an answer, it will be transformed to map the input data into new output features.

The steps of applying RICA to the observations are as follows:

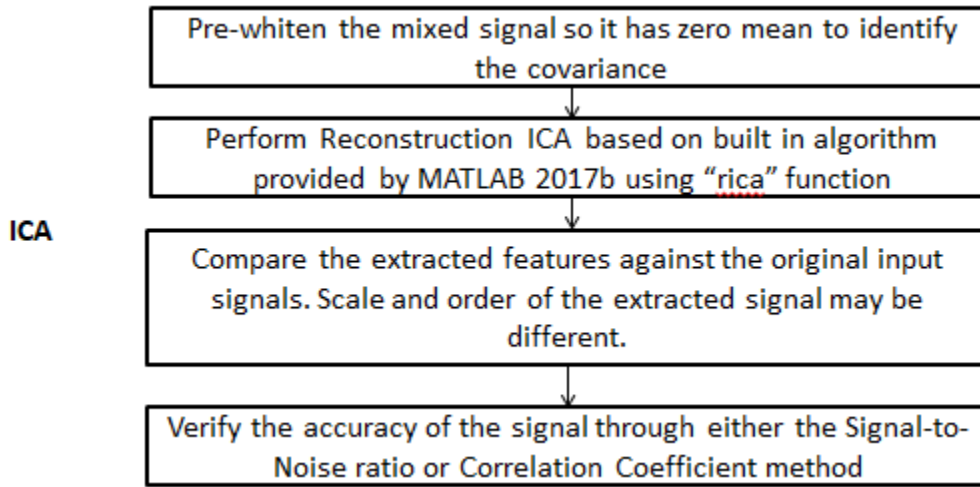


Figure 9. Flow chart for Reconstruction ICA^[23]

Because the orthonormality constrained is removed in RICA, a “reconstruction cost” is used in the algorithm to prevent degeneracy. Besides being easier to implement, a key benefit of using RICA over ICA is removing the constraint on the optimizer, leading to it able to learn overcomplete features, having a faster convergence rate and being less sensitive to whitening^[24].

Using the “rica” function in MATLAB 2017b, the data for the Raman spectra of Contact Lens (‘CL’) as well as pure bacteria *C. Albicans* (‘CA’) and *P. Aeruginosa* (‘PA’) were loaded onto the software. Figure 10 shows the result of the code, which works by first

mixing the three signals, 'CL', 'CA' and white noise are added in a random order, thereby generating Mixed Signals 1, 2 and 3. In this case, the randomness is set to a default for reproducibility of results. The mixed signals are then whitened to remove any correlations in the data before being processed by ICA, and it will be compared against the original signal.

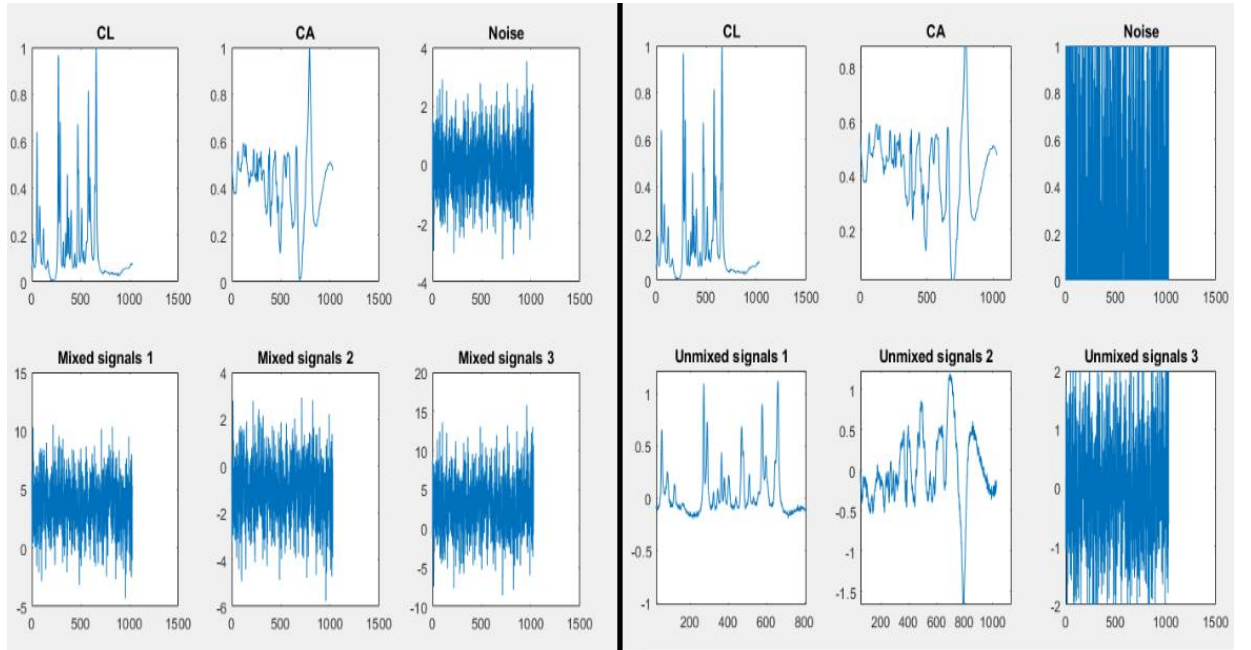


Figure 10. RICA on Pure CL, Pure CA and Noise

The downsampled Raman shift was used as the basis for the pre-whitening step, and the code for pre-whitening was taken from a tutorial on how to extract Mixed Signals on the MATLAB Mathworks website^[25]. After the data has been pre-whitened, RICA is performed, and it should be noted that the number of outputs will be equal to the number of inputs. After the algorithm has unmixed the signals, the features of the output signals should be proportional to the input signals. As seen in Figure 10 (above), the Unmixed

Signal 2 appears to be flipped. However, the signal can be corrected by simply multiplying the scale by -1.

After obtaining the extracted signal from ICA, it is required to verify the accuracy of the recovered signal compared to the actual signal. This step is akin to cross-validation in PC-LDA. This can be done by either measuring the Mean Square Error (MSE) first and then the Signal-to-Noise ratio (SNR), listed in equation (14) and (15), or through the Coefficient Correlation method in equation (16).

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^N (y_i - x_i)^2 \quad (14)$$

$$\text{SNR} = 10 \log_{10}[\sigma^2 / \text{MSE}] \quad (15)$$

$$R(s,y) = \text{cov}(s,y) / (\sqrt{\text{cov}(s,s)\text{cov}(y,y)}) \quad (16)$$

Where s is the signal that was extracted by the algorithm and y is the actual signal that we should expect to receive.

Figure 11 shows the results of using the Correlation Coefficient method on the actual PA signal against the unmixed PA signal, it was found that the matrix of Correlation Coefficient between the signals is 99.75%, which is very high because the best possible result of this method is '1'. This process was repeated for CA, and the Correlation Coefficient was found to be 98.88%.

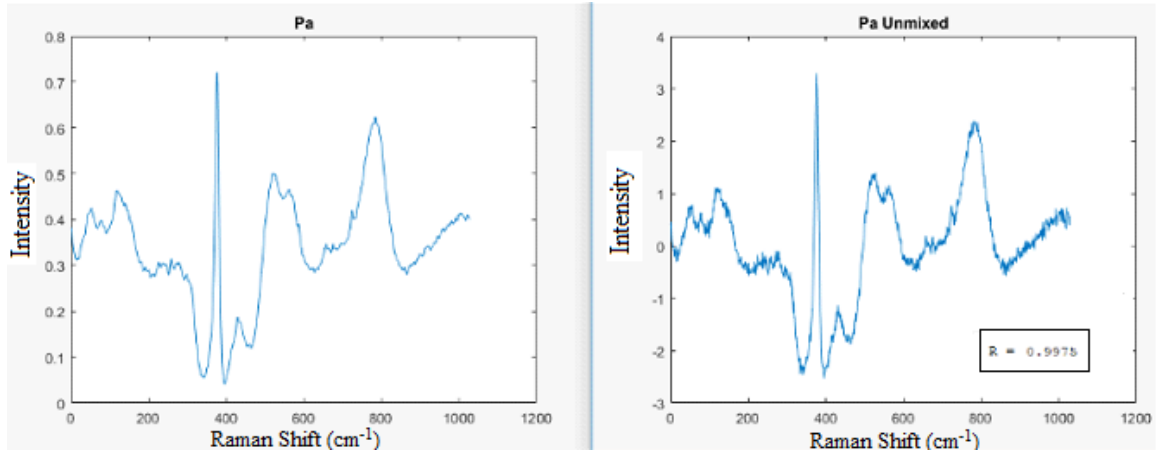


Figure 11a. Correlation Coefficient between reference and unmixed output signal (PA)

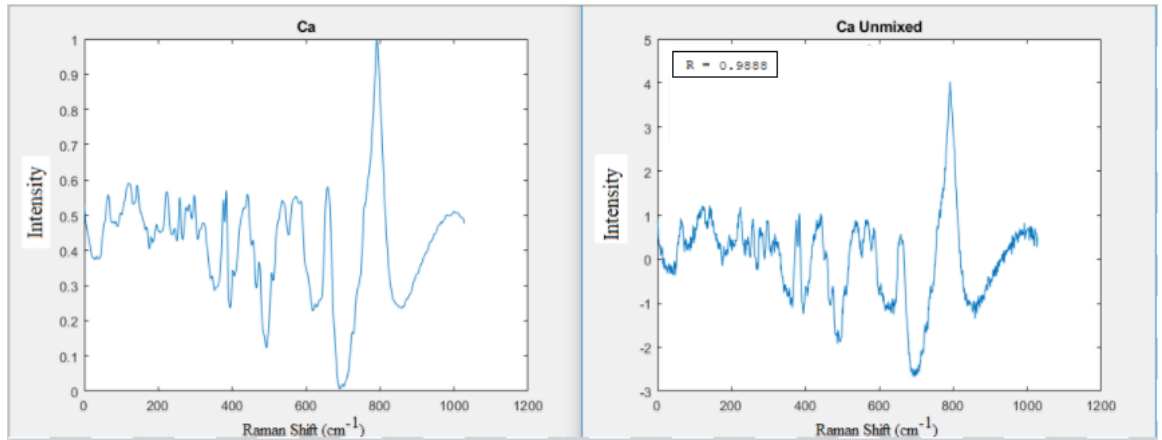


Figure 11b. Correlation Coefficient between reference and unmixed output signal (CA)

The importance of performing ICA in this project is to check if the decomposed mixed signal can recover the spectra of the microorganism. The code for RICA and Pre-whitening are listed in Appendix B and C respectively.

3.4 C-ICA

According to the article “ICA with Reference” by Lu and Rajapakse, the goals of C-ICA are to estimate the output as a subset of the Independent Components that are mixed in the input data and to extract the Independent Components that are closest to the corresponding signals. It becomes a constrained optimization problem that adds new variables to the existing ICA algorithm, where a Lagrange multiplier is used in conjunction with the contrast function, giving a constrained contrast function^[26]. The equation of the new contrast function to be maximized is listed in Equation 17, and it is subjected to $\mathbf{g}(\mathbf{w})$ (Equation 18), the feasible constraint to the contrast function as well as $\mathbf{h}(\mathbf{w})$ (Equation 19), the equality constraint used to bound contrast function $\mathbf{J}(\mathbf{y})$ and weight vector \mathbf{w} .

$$\text{Maximize } \mathbf{J}(\mathbf{y}) \approx \rho(\mathbf{E}\{\mathbf{G}(\mathbf{y})\} - \mathbf{E}\{\mathbf{G}(\mathbf{v})\})^2 \quad (17)$$

$$\text{Subject to } \mathbf{g}(\mathbf{w}) = \varepsilon(\mathbf{y}, \mathbf{r}) - \xi \leq 0 \quad (18)$$

$$\mathbf{h}(\mathbf{w}) = \mathbf{E}\{\mathbf{y}^2\} - 1 = 0 \quad (19)$$

Where \mathbf{E} is the column matrix that corresponds to unit-norm eigenvector, ρ is a positive constant, \mathbf{v} is a Gaussian variable with zero mean and unit variance, $\mathbf{G}(\mathbf{y})$ and $\mathbf{G}(\mathbf{v})$ are any non-quadratic function. $\varepsilon(\mathbf{y}, \mathbf{r})$ represents the closeness between the output \mathbf{y} and the reference signal \mathbf{r} . ξ is the threshold that distinguishes the desired signal from other

source signals. The desired source signal extracted is always closest to the reference signal \mathbf{r} .

A Newton-like learning algorithm was adopted to find the maximum Lagrangian multiplier, and the updated weights are given by (Equation 20).

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \boldsymbol{\Pi} \frac{\mathbf{R}_{\mathbf{xx}}^{-1} \mathbf{L} \mathbf{w}_k}{\delta(\mathbf{w}_k)} \quad (20)$$

$$\mathbf{L} \mathbf{w}_k = \rho \mathbb{E} \{ \mathbf{x} G'_y(y) \} - 0.5 \mu \mathbb{E} \{ \mathbf{x} g'_y(\mathbf{w}_k) \} \quad (21)$$

$$\delta(\mathbf{w}_k) = \rho \mathbb{E} \{ \mathbf{x} G''_y(y) \} - 0.5 \mu \mathbb{E} \{ \mathbf{x} g''_y(\mathbf{w}_k) \} \quad (22)$$

Where \mathbf{k} represents the iteration count, $\boldsymbol{\Pi}$ represents the learning rate and $\mathbf{R}_{\mathbf{xx}}$ represents the covariance matrix of the input. $G'_y(y)$ and $G''_y(y)$ are first and second derivatives of $G(y)$ with respect to Y and $g'_y(\mathbf{w}_k)$ and $g''_y(\mathbf{w}_k)$ are first and second derivatives of $g(y)$ with respect to $g(\mathbf{w}_k)$. Furthermore,

$$\mu_{k+1} = \max \{ 0, \mu_k + \gamma g(\mathbf{w}_k) \} \quad (23)$$

$$\lambda_{k+1} = \lambda_k + \gamma h(\mathbf{w}_k) \quad (24)$$

Where μ and λ are optimum multipliers updated iteratively and γ is the scalar penalty parameter^[27].

On MATLAB, the C-ICA algorithm used in this project was adapted from Zhi-Lin Zhang's C-ICA code for extracting weak temporally correlated signals^[28]. The paper focuses on using second-order statistics based approach to find suitable reference signals

for weak temporally correlated sources. Synthetic and real-world data were used to verify their approach, which were able to successfully extract the desired signals.

In the C-ICA code itself, the number of samples, sampling period, frequency of the reference signals had to be pre-set. The source signals were then standardized and whitened, as per the steps of ICA. The parameters, μ , λ , γ and $\mathbf{\Pi}$, were all set to '1', and the maximum iteration count is set to 200. Three sets of reference signals were provided, the first being the original reference signal, the second changing only the width of the signal and the third changing only the phase of the signal. The threshold, ξ , was set to 1.75^[29].

In the adapted C-ICA code, only one reference signal was used, which makes the changing of the width and the phase of the signal obsolete. Three source signals were mixed together, the first being the Raman Spectra of Contact Lens, the second being the bacteria (*P. Aeruginosa* or *C. Albicans*) and the third being random Gaussian noise. The reference signal used can either be the raw signal from the respective bacteria, or can be a thresholded signal from the respective bacteria, which will be in the form of a square wave. The threshold was lowered to 0.5 for it to accurately converge to the correct signal. The rest of the parameters, including the maximum iteration count, remained the same. The correlation coefficient of the output, ' $\mathbf{y1}$ ', was compared against the original source signal. Figures 12a-d shows the successful extraction of the desired bacterial signal using the original and thresholded signal as the reference.

Figure 13 shows that the algorithm is able to extract ‘CL’ when the reference signal given is ‘CL’, and extracts ‘Noise’ when the reference given is ‘Noise’. This indicates that the extraction of a desired signal is highly dependent on the input reference signal, so it can be modified to detect other materials in the raw Raman spectra as well, as long as the Raman spectra of the target is known. Furthermore, the threshold of the algorithm may need to be adjusted if the target signal is too similar to other individual source signals.

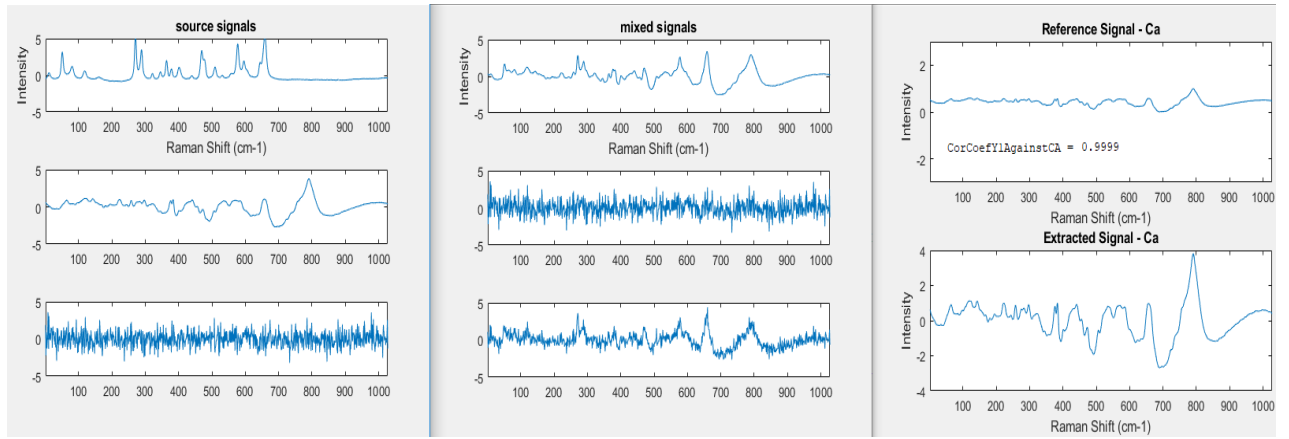


Figure 12a. Extraction of CA signal using raw CA signal as reference

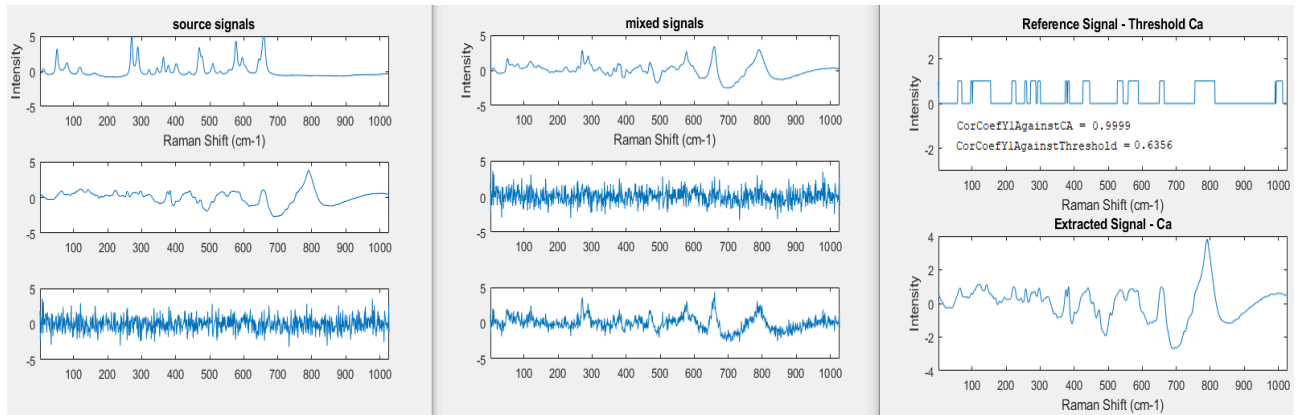


Figure 12b. Extraction of CA signal using threshold CA signal as reference

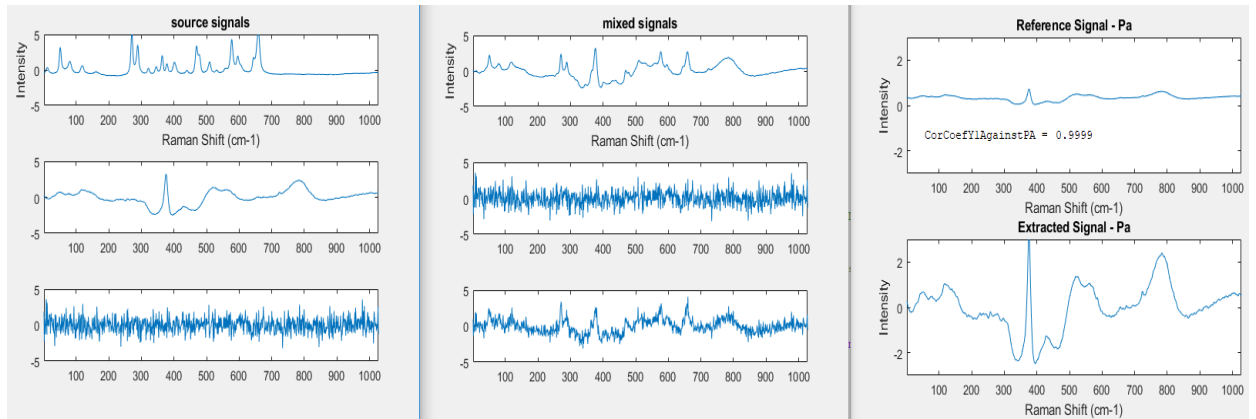


Figure 12c. Extraction of PA using raw PA signal as reference

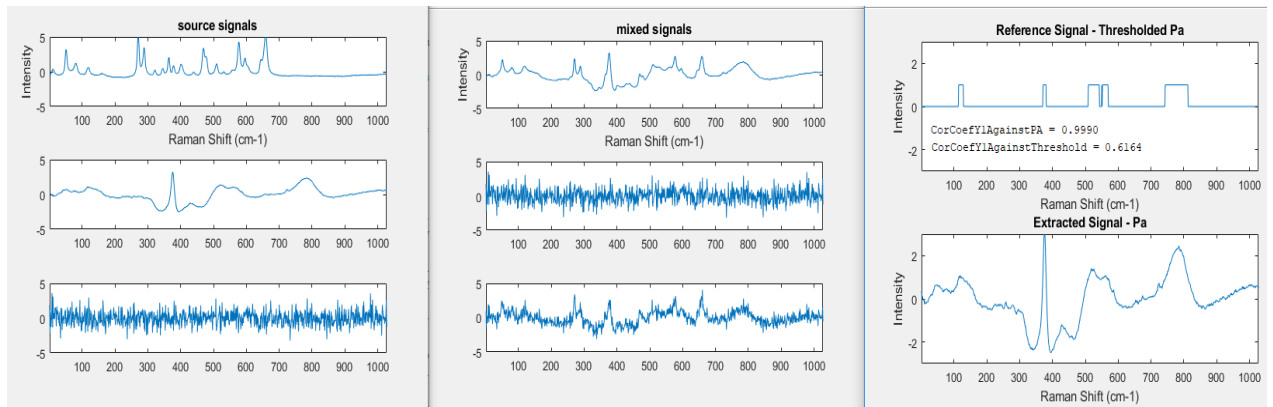


Figure 12d. Extraction of PA using threshold PA signal as reference

Correlation Coefficient of Raw CA against Extracted CA: 0.99
Correlation Coefficient of Extracted CA using Threshold CA against Raw CA: 0.99

Correlation Coefficient of Raw PA against Extracted PA: 0.99
Correlation Coefficient of Extracted PA using Threshold PA against Raw PA: 0.99

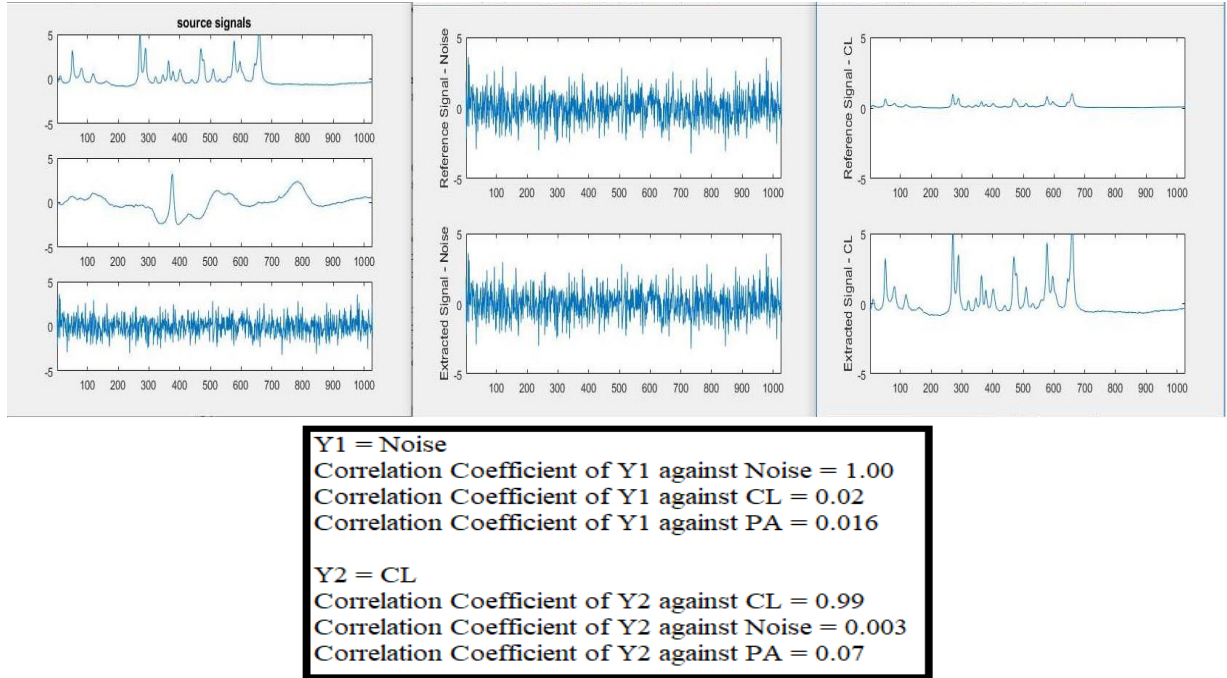


Figure 13. Extraction of CL and Noise signals

By being able to converge to the desired signal using the reference signal that was set, the results has confirmed the high sensitivity of the algorithm. This flexibility of selecting the reference also allows us to directly obtain the results that we want, while discarding the other “uninteresting” sources.

The adapted code further tests the detection limit of the algorithm, diluting the concentration of the bacteria up to a maximum of **70,000,000** for the *C. Albicans* signal and **80,000,000** for the *P. Aeruginosa* signal. A plot of the dilution against the SNR is provided in Figure 14a and 14b. Only the main C-ICA code for CA is provided in Appendix D, without modifications being made to the code to dilute the concentration of the target bacteria and to plot the SNR against dilution graph.

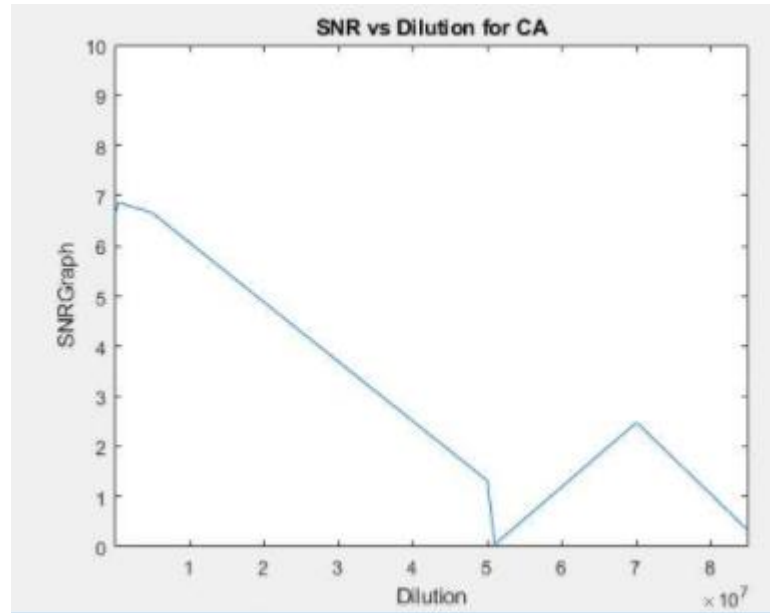


Figure 14a. SNR against dilution of CA

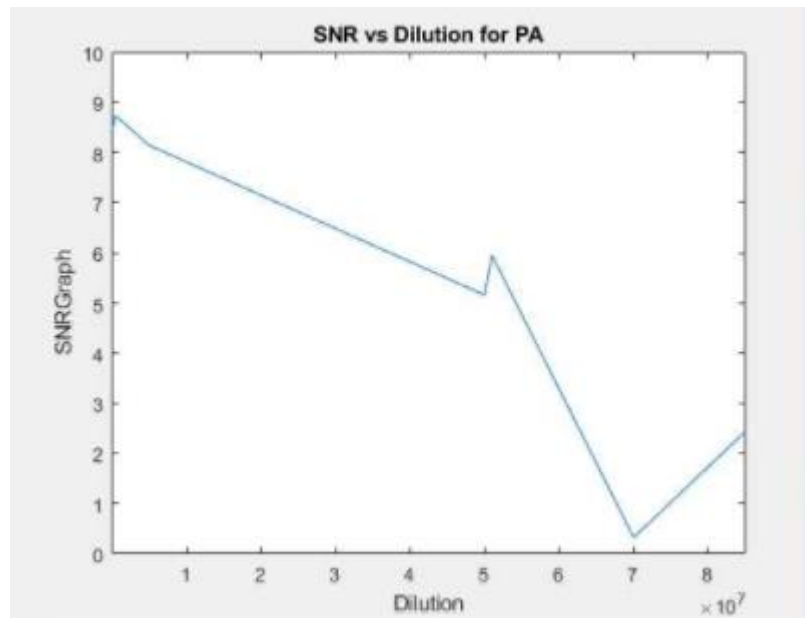


Figure 14b. SNR against dilution of PA

One discrepancy that the graphs have shown is that there is an abnormal slight increase in the SNR as the dilution is increased. This is due to the random noise signal being set to a default value for the sake of reproducibility of results. When the randomization is not set

to a default value, the graph displays the expected trend, as shown in Figure 15a and Figure 15b. It is also shown that the detection limit will not be as high as compared to when the randomization is set to default, with the SNR reaching the minimum value of about 50,000,000 for the *C. Albicans* signal and 70,000,000 for the *P. Aeruginosa* signal.

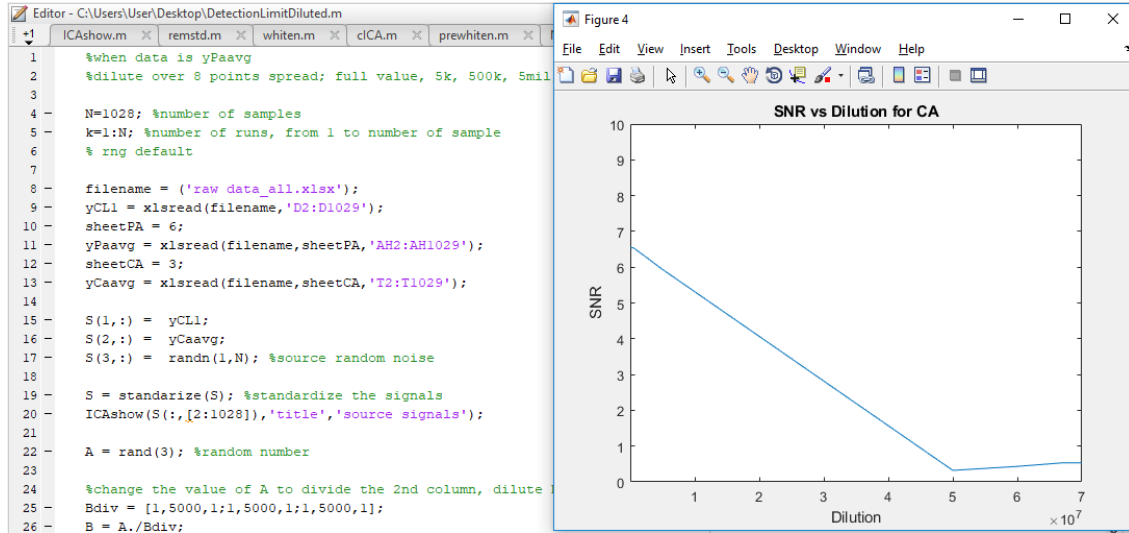


Figure 15a. SNR against dilution of CA when the random noise is not set to default

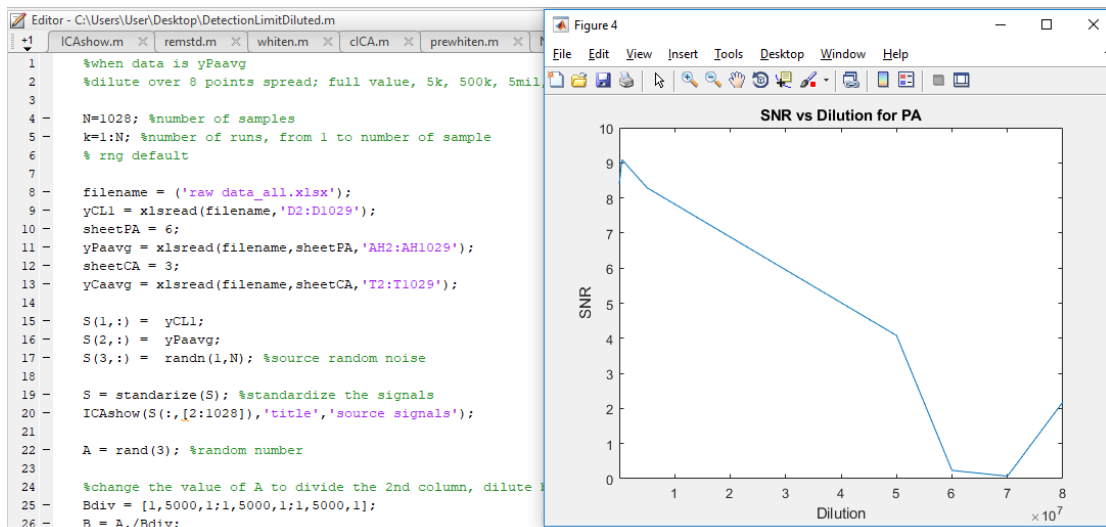


Figure 15b. SNR against dilution of PA when the random noise is not set to default

Identifying the detection limit where the SNR drops below a certain threshold is only the first step. It would be more accurate to report the contribution of the target analyte relative to the contribution of the non-target concentration and noise to the overall spectrum. As such, there is a need to find the original target concentration and compare it with the new target concentration, in terms of percentage. To do so, the percentage of the original concentration of the bacteria (Matrix A) was calculated, which was found to be 40.67%. Next, the diluted target concentration for CA and PA was calculated, which we have found to be 0.00000137% and 0.000000979%. Dividing the original percentage to the diluted percentage, we obtain the dilution factor of about **30,000,000** and **41,000,000** respectively. The workings are provided in Appendix E.

DISCUSSION AND CONCLUSION

4.1 Comparison of RICA and C-ICA

Having successfully performed RICA and C-ICA on the data, the last part of this report aims to compare and to demonstrate that Constrained ICA is the superior method despite being more computationally expensive.

The RICA step was repeated and this time, the raw measurements (e.g. [yCAonCL1, yCAonCL2 and yCaonCL3]) directly taken from Raman Spectroscopy were used as the input matrix instead of a mixture of the individual source signals of CL, the bacteria, and random noise. In this scenario, no reference signal is given for the algorithm. It shows that the algorithm is unable to decompose the mixed signals into the individual signals, but instead recovers a signal that does not share a strong correlation coefficient with any of the input signals.

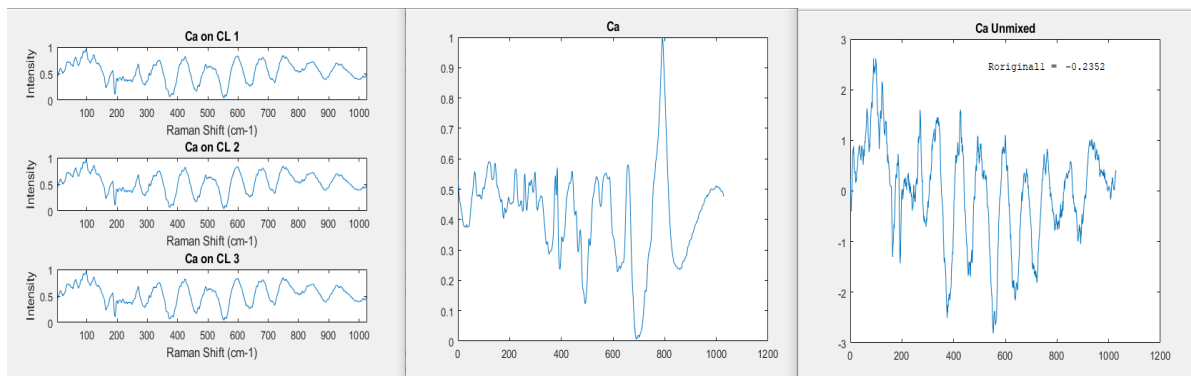


Figure 16a. RICA on raw measurements of CA on CL

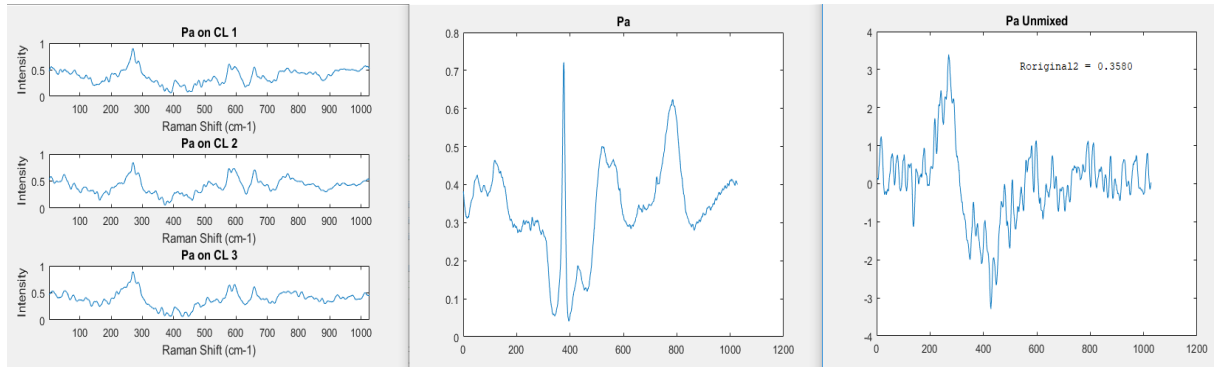


Figure 16b. RICA on raw measurements of PA on CL

**Correlation Coefficient of CA Unmixed against CA :
0.235**

**Correlation Coefficient of PA Unmixed against PA :
0.358**

By theory, for ICA-related applications to work, the mixtures must be a linear combination of the independent sources. The number of available mixtures must also be the same as the number of independent components^[30]. Since the results show that RICA is unable to recover any of the individual sources, it can be assumed that the input mixtures are not in a linear combination with the independent source signals, which may be due to the generated random noise.

The same procedure was repeated with C-ICA, again using the raw measurements as the input matrix. Despite using the Raman spectra of the pure bacteria as the reference signal, the output given by C-ICA is inconclusive. For the CA signal, the algorithm is able to produce an entirely different output from any of the input signals. However, the signal produced is extremely noisy, and a Savitzky-Golay filter had to be applied to recover a

clean signal. The output signal was also manipulated via flipping the signal around its axis and shifting of the wavelength to obtain the best coefficient correlation against the original 'Raw CA' signal. Signal manipulation is not required for the PA signal, as the output shows a correlation coefficient of 95.5% between the extracted signal and the input raw measured signal. The results of CICA on the raw measurements are displayed in Figure 17a and Figure 17b.

The same results could not be obtained for the PA signal, as it would always recover one of the signals from the input matrix. This may be due to the high correlation coefficient of the 'Raw PA' signal and the 'PA on CL' signal, compared to the lower correlation coefficient between 'Raw CA' signal and 'CA on CL' signal, which are 0.4 and 0.03 respectively.

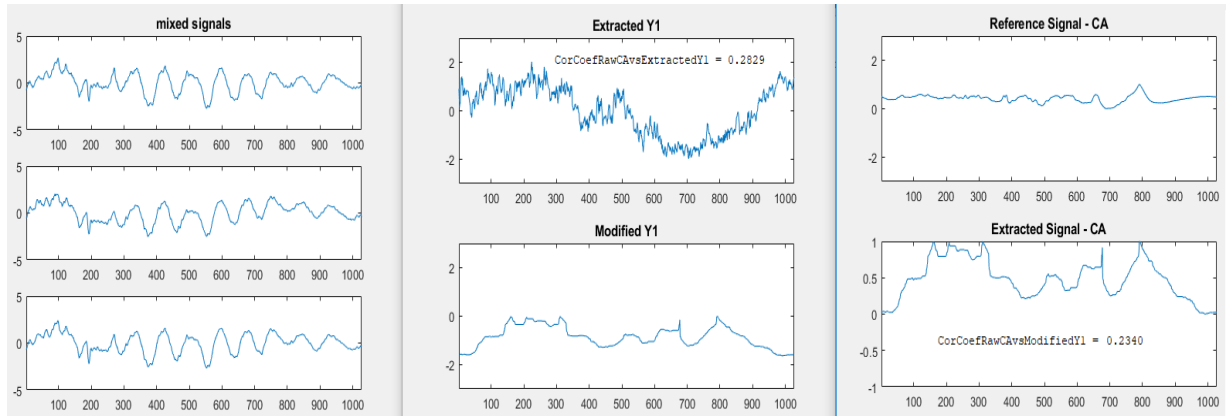


Figure 17a. C-ICA on raw measurements of CA on CL

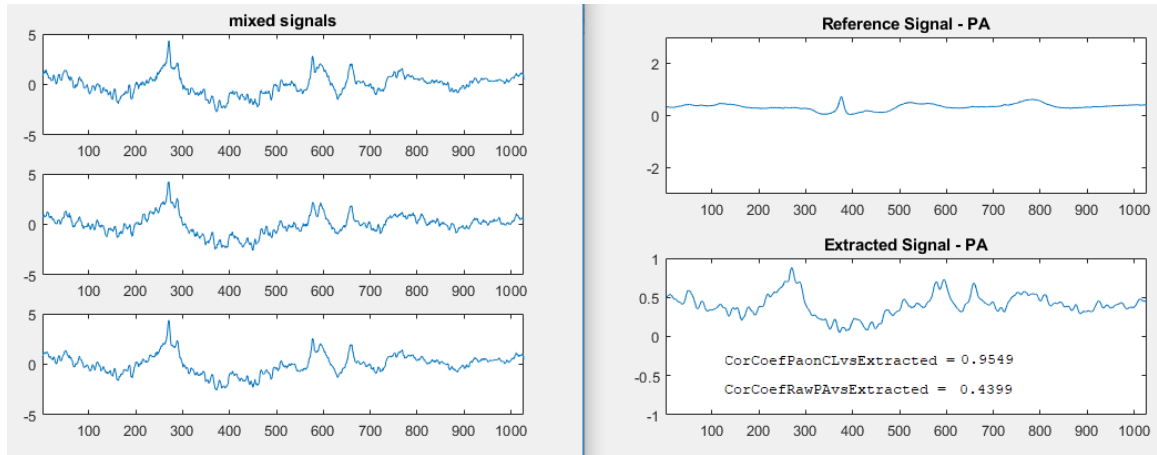


Figure 17b. C-ICA on raw measurements of PA on CL

**Correlation Coefficient of extracted CA against CA :
0.234**

**Correlation Coefficient of extracted PA against PA :
0.422**

4.2 Summary

In conclusion, the objective of this project is to classify the Raman spectra of the pure microorganism samples and the respective microorganism samples on contact lens using PC-LDA and to use C-ICA on the same dataset to extract the signal of pure microorganism from a mixed signal. This is important because it validates the using Raman spectroscopy for the purpose of microorganism identification, thus enabling a more accurate diagnosis at a much shorter time compared to traditional methods of microorganism identification.

Before the application of any algorithms, data pre-processing was performed on the dataset, and the first step would be to ensure that the datasets from different samples have the same data points via “up-sampling” or “down-sampling”. Signal smoothing via moving average filter or Savitsky-Golay filter is then performed to remove noise such as cosmic ray spikes and natural fluorescence. Finally, the important features of the signal are extracted via baseline removal to identify the major peaks of the sample.

PC-LDA is a popular algorithm to classify datasets generated by Raman spectroscopy due to its ease of implementation and dimensionality reduction. We first obtain the “explained variance” of the data to find out exactly how many PCs need to be kept because of their significance, and then execute the algorithm accordingly. K-fold cross validation was performed to validate the results of PC-LDA, providing us with the

accuracy whenever a new dataset is compared against an existing dataset to check if the model has predicted correctly.

RICA was used in substitution to ICA as it is an already built-in function in MATLAB 2017. It has proven to be able to mix the signals along with random generated noise, and then decompose the mixed signals back into the individual source signals. However, the drawbacks of this algorithm are that the decomposed signals are not in any order, and the recovered signal may not be an exact replica of the individual source signal as it may be inverted or flipped.

Lastly, C-ICA has been proven to be a technique that can be applied to Raman spectroscopic data to detect the presence of a microorganism, provided that there is *a priori* knowledge of the target that we wish to detect and there is a linear relationship between the individual spectra of components and the combined Raman spectra. This experiment has also proved that we do not need an exact match of the Raman Spectroscopic signature of the microorganism, for as long as the major “peaks” of the spectra signature are present. The threshold value of C-ICA can also be set to modify the freedom of convergence to ensure the specificity of the target. C-ICA has proven to have a high detection limit as it is able to detect the target that is diluted up to at least **30,000,000** times the original concentration.

REFERENCES

- [1] Noorpur Gupta and Radhika Tandon, “Investigative modalities in infectious keratitis”, Indian Journal of Ophthalmology, Vol. 56(3), p.209-213, 2008.
Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2636118/>
- [2] Christian N. Kotanen, Luis Martinez, Rene Alvarez, John W., “Surface enhanced Raman scattering spectroscopy for detection and identification of microbial pathogens isolated from human serum”, Sensing and Bio-Sensing Research, Vol. 8, Pp.20-26, 2016. Available:
<http://www.sciencedirect.com/science/article/pii/S2214180416300204>
- [3] Mya Myintzu Hliang, Michelle Dunn, Paul Stoddard, Sally L Mcarthur, “ Raman Spectroscopic Identification of Single Bacterial Cells at Different Stages of their Lifecycle”, Vibrational Spectroscopy, Vol. 86, Pp.81-89, 2016. Available:
https://www.researchgate.net/publication/304185492_Raman_Spectroscopic_Identification_of_Single_Bacterial_Cells_at_Different_Stage_of_their_Lifecycle
- [4] Chengxu Hu, Juexin Wang, Chao Zheng, Shuping Xu, Haipeng Zhang, Yanchun Liang, Lirong Bi, Zhimin Fan, Bing Han, Weiqing Xu, “ Raman spectra exploring breast tissues: Comparison of Principal Component Analysis and Support Vector Machine- Recursive feature elimination”, Medical Physics, Vol. 40, 2013.
Available: <https://www.ncbi.nlm.nih.gov/pubmed/23718612>

- [5] M. Boiret, D.N. Rutledge, N. Gorretta, Y.M. Ginot, J.M. Roger, “Application of independent component analysis on Raman images of a pharmaceutical drug product: pure spectra determination and spatial distribution of constituents”, Journal of Pharmaceutical and Biomedical Analysis, Elsevier, Vol. 90, Pp.78-84, 2014.
Available: <https://hal-agroparistech.archives-ouvertes.fr/hal-00948530/document>
- [6] Aapo Hyvarinen and Erkki Oja, “Independent Component Analysis: Algorithms and Applications”, Neural Networks, Vol. 13, Pp.411-430, 2000.
Available: http://cis.legacy.ics.tkk.fi/aapo/papers/IJCNN99_tutorialweb/
- [7] Wei Wang and Tulay Adali, “Constrained ICA and its Application to Raman Spectroscopy”, Antennas and Propagation Society International Symposium, 2005 IEEE, 2005. Available:
<http://ieeexplore.ieee.org.ezlibproxy1.ntu.edu.sg/document/1552752/>
- [8] Renishaw, “Raman Spectroscopy in more detail.”, Date Unknown.
Available:
<http://www.renishaw.com/en/raman-spectroscopy-in-more-detail--25806>

- [9] Butlet HJ, Ashton L, Bird B, Cinque G, Curtis K, Dorney J, Esmonde-White K, Fullwood NJ, Gardner B, Martin-Hirsch PL, Walsh MJ, McAinsh MR, Stone N, Martin FL, “Using Raman spectroscopy to characterize biological materials”, Nature Protocols, Vol. 11, Pp.664-687, 2016. Available: <https://www.ncbi.nlm.nih.gov/pubmed/26963630>
- [10] Cynthia Hanson and Elizabeth Vargis, “Microscopy and Raman Imaging: Open-system Raman microscopy”, LaserFocusWorld, Vol. 51, 2015.
Available: <http://www.laserfocusworld.com/articles/print/volume-51/issue-05/biooptics-world/biooptics-features/microscopy-and-raman-imaging-open-system-raman-microscopy.html>
- [11] Carolyn Rulli, “The Raman Spectrophotometer”, School of Arts & Sciences, University of Pennsylvania. Available: <https://www.sas.upenn.edu/~crulli/TheRamanSpectrophotometer.html>
- [12] Stephan Stöckel, Johanna Kirchhoff, Ute Neugebauer, Petra Rösch, Jürgen Popp, “The application of Raman spectroscopy for the detection and identification of microorganisms”, Journal of Raman Spectroscopy, Vol. 47, Pp. 89-109, 2016.
Available: <http://onlinelibrary.wiley.com/doi/10.1002/jrs.4844/abstract>

- [13] S.H. Ahn, W.H. Lee, M.H. In., T. –S. Kim, S.Y. Lee, “Extraction and Localization of Alpha Activity of the Brain in EEG and fMRI Using Constrained ICA”, Engineering in Medicine and Biology Society, 29th Annual International Conference of the IEEE, 2007. Available: <https://www.ncbi.nlm.nih.gov/pubmed/18003255>
- [14] Wei Lu, Jagath C. Rajapakse, “ICA with Reference”, Neurocomputing, Vol 69, Pp. 2244-2257, 2006.
Available: www.sciencedirect.com/science/article/pii/S0925231205003176
- [15] Prof. Tom O’Haver, “A Pragmatic Introduction to Signal Processing”, University of Maryland at College Park, Department of Chemistry and Biochemistry, 2017.
Available: <https://terpconnect.umd.edu/~toh/spectrum/Smoothing.html>
- [16] Rekha Gautam, Sandeep Vanga, Freek Ariese, Siva Umapathy, “Review of multidimensional data processing approaches for Raman and infrared spectroscopy”, EPJ Techniques and Instrumentation, Vol 2, Pp. Unknown, 2015.
Available: <https://epjtechniquesandinstrumentation.springeropen.com/articles/10.1140/epjti/s40485-015-0018-6>
- [17] Xiaozhou Li, Tianyue Yang, Siqi Li, Deli Wang, Youtao Song, Su Zhang, “Raman spectroscopy combined with principal component analysis and k nearest neighbor analysis for non-invasive detection of colon cancer”, Laser Physics,

Vol. 26, Pp. Unknown, 2016. Available:

<http://iopscience.iop.org/article/10.1088/1054-660X/26/3/035702/meta#citations>

- [18] Sebastian Raschka, “Linear Discriminant Analysis – Bit by Bit”, 2014.

Available: http://sebastianraschka.com/Articles/2014_python_lda.html

- [19] Olga Veksler, “CS434a/541a: Pattern Recognition”, University of Western Ontario, Computer Science department, 2004. Available:

http://www.csd.uwo.ca/~olga/Courses/CS434a_541a/Lecture8.pdf

- [20] Wei Kong, Charles R. Vanderburg, Hiromi Gunshin, Jack T. Rogers, Xudong Huang, “A review of independent component analysis application to microarray gene expression data”, Biotechniques, Vol 45, Pp. 501-520, 2008.

Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3005719/>

- [21] Alan Oursland, Judah De Paula, Nasim Mahmood, “Case Studies of Independent Component Analysis”, Independent Component Analysis (ICA) Tutorial, Date Unknown. Available: <http://www.oursland.net/tutorials/ica/>

- [22] Aapo Hyvarinen, Juha Karhunen, Erkki Oja, “Independent Component Analysis”, 2001. Available: https://www.cs.helsinki.fi/u/ahyvarin/papers/bookfinal_ICA.pdf

- [23] Mathworks, “Feature Extraction”, 2017. Available:
<https://www.mathworks.com/help/stats/feature-extraction.html#bvmxyf6-1>
- [24] Quoc V. Le, Alexandre Karpenko, Jiquan Ngiam, Andrew Y. Ng, “ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning”, NIPS, 2011.
Available: <http://ai.stanford.edu/~quocle/LeKarpenkoNgiamNg.pdf>
- [25] Mathworks, “Extract Mixed Signals”, 2017. Available:
<https://www.mathworks.com/help/stats/extract-mixed-signals.html>
- [26] Wei Lu, Jagath C. Rajapakse, “ICA with Reference”, Neurocomputing, Vol. 69, Pp 2244 – 2257, 2006. Available:
<http://www.sciencedirect.com/science/article/pii/S0925231205003176>
- [27] Balakrishna Iavu, Venkata Ashokkumar Potnuru, “Speech Enhancement using Constrained-ICA with Bessel Features”, Blekinge Institute of Technology, School of Engineering, Department of Electrical Engineering, 2011. Available:
<http://www.diva-portal.org/smash/get/diva2:829678/FULLTEXT01.pdf>
- [28] Zhi-Lin Zhang, “Morphologically constrained ICA for extracting weak temporally correlated signals”, Neurocomputing, Vol. 71, Pp 1669 – 1679, 2008.
Available: <http://dsp.ucsd.edu/~zhilin/papers/McICA.pdf>

- [29] Zhi-Lin Zhang, “Software of Zhilin Zhang, (9) cICA code for the constrained ICA algorithm”, 2008. Available: <http://dsp.ucsd.edu/~zhilin/Software.html>
- [30] Djuwari Djuwari, Dinesh Kant Kumar, Marimuthu Palaniswami, “Limitations of ICA for Artefact Removal”, Engineering in Medicine and Biology 27th Annual Conference, 2005. Available: <https://researchbank.rmit.edu.au/eserv/rmit:1577/n2006005092.pdf>

APPENDIX

Appendix A – Code for PC-LDA for all given sets of data

```
%row 1 is CL, 2-10 is CA, 11-14 is CA on CL, 15-24 is PA, 25-28 is PA on CL
y1=yCL1(2:1029)
y2=yCa1(2:1029)
y3=yCa2(2:1029)
y4=yCa3(2:1029)
y5=yCa4(2:1029)
y6=yCa5(2:1029)
y7=yCa6(2:1029)
y8=yCa7(2:1029)
y9=yCa8(2:1029)
y10=yCaavg(2:1029)
y11=ycaonCL1(2:1029)
y12=ycaonCL2(2:1029)
y13=ycaonCL3(2:1029)
y14=ycaonCLavg(2:1029)
y15=yPa1(2:1029)
y16=yPa2(2:1029)
y17=yPa3(2:1029)
y18=yPa4(2:1029)
y19=yPa5(2:1029)
y20=yPa6(2:1029)
y21=yPa7(2:1029)
y22=yPa8(2:1029)
y23=yPa9(2:1029)
y24=yPaavg(2:1029)
y25=yPaonCL1(2:1029)
y26=yPaonCL2(2:1029)
y27=yPaonCL3(2:1029)
y28=yPaonCLavg(2:1029)
data=[y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19,y2
0,y21,y22,y23,y24,y25,y26,y27,y28];
[coeff,score,latent,tsquared,explained] = pca(data);
%remove the mean row-wise
data=data-repmat(mean(data,2),1,size(data,2));
%calculate eigenvectors W and eigenvalue of covariance matrix
[W, EvalueMatrix]=sig(cov(data'));
Evalues=diag(EvalueMatrix);

% order by largest eigenvalue
Evalues = Evalues (end:-1:1);
W=W(:,end:-1:1);
W=W';
```

```

%generate PCA scores and transpose and save

pc=W*data;
pctranspose=pc';
pctranspose=pctranspose(:,1:2);
csvwrite('principalcomponents5.csv',pctranspose);

% % plot PCA of all PCs in 2D unlabelled
% figure;
% plot(pc(1,:),pc(2,:),'.') %plotting principal components of first row all
% columns, 2nd row all columns
% title('PCA 2D')

%plot PCA labelled
figure;
plot(pc(1,1:1),pc(2,1:1),'^') %plotting principal component of 1st row 1st
column against 2nd row 1st column, pure CL
hold on
plot(pc(1,2:10),pc(2,2:10),'o') %plotting principal component of 1st row
1st column against 2nd row 1st column, pure CA
hold on
plot(pc(1,11:14),pc(2,11:14),'.') %plotting principal component of 1st row
2nd-5th column against 2nd row 2nd-5th column, Ca on CL
hold on
plot(pc(1,15:24),pc(2,15:24),'*') %plotting principal component of 1st row
6th-14th column against 2nd row 6th-14th column, pure PA
hold on
plot(pc(1,25:28),pc(2,25:28),'x'), %pa on CL

legend('CL','Pure CA','Ca on CL','Pure PA','PA on CL');
title('PCA 2D labelled')

explained

%obtain LDA matrix from 1st 2 principal components
LDAMatrix2=pc(1:2,:);

```

```

% % start LDA here
X1=LDAMatrix2(:,1) %all rows, 1st col, CL
X2=LDAMatrix2(:,2:10) %all rows, 2nd to 5th col, CA
X3=LDAMatrix2(:,11:14) %all rows, 6th to 14th col, CA on CL
X4=LDAMatrix2(:,15:24) %all rows, 6th to 14th col, PA
X5=LDAMatrix2(:,25:28) %all rows, 6th to 14th col, PA on CL
% %find mean and overall mean
Mu1=mean(X1)';
Mu2=mean(X2)';
Mu3=mean(X3)';
Mu4=mean(X4)';
Mu5=mean(X5)';
Mu=(Mu1+Mu2+Mu3+Mu4+Mu5)/5;
%covariance and within class
S1=cov(X1)';
S2=cov(X2)';
S3=cov(X3)';
S4=cov(X4)';
S5=cov(X5)';
Sx=S1+S2+S3+S4+S5;
%number of samples per class
N1 = size(X1,2);
N2 = size(X2,2);
N3 = size(X3,2);
N4 = size(X4,2);
N5 = size(X5,2);
%between class
SB1 = N1.*(Mu1-Mu)*(Mu1-Mu)';
SB2 = N2.*(Mu2-Mu)*(Mu2-Mu)';
SB3 = N3.*(Mu3-Mu)*(Mu3-Mu)';
SB4 = N4.*(Mu4-Mu)*(Mu4-Mu)';
SB5 = N5.*(Mu5-Mu)*(Mu5-Mu)';
SB = SB1+SB2+SB3+SB4+SB5;
% %computing LDA projection
invSw=inv(Sx);
invSw_by_SB=invSw*SB;
%get projection vector
%[V,D] = EIG(X) produces diagonal matrix D of eigenvalues and a full matrix
%V whose columns are corresponding eigenvectors
[V,D]=eig(invSw_by_SB);
%projection vectors
P1=V(:,1);
P2=V(:,2);

```

```

% %visualization on scatter plot, uncomment to see 2D data
hfig=figure;
axes1=axes('Parent',hfig,'FontWeight','bold','FontSize',12);
hold('all')
%create x and y labels
xlabel('X_1 - first feature','FontWeight','bold','FontSize',12,...
'FontName','Garamond');
ylabel('X_2 - second feature','FontWeight','bold','FontSize',12,...
'FontName','Garamond');
%first class
scatter(X1(1,:),X1(2:),'^','LineWidth',2,'Parent',axes1);
hold on
%second class
scatter(X2(1,:),X2(2:),'o','LineWidth',2,'Parent',axes1);
hold on
%third class
scatter(X3(1,:),X3(2:),'p','LineWidth',2,'Parent',axes1);
hold on
%fourth class
scatter(X4(1,:),X4(2:),'*','LineWidth',2,'Parent',axes1);
hold on
%fifth class
scatter(X5(1,:),X5(2:),'x','LineWidth',2,'Parent',axes1);
hold on
%drawing projection vectors
%first line
t=-5:5
line_x1=t.* P1(1);
line_y1=t.* P1(1);
%second line
line_x2=t.* P2(1);
line_y2=t.* P2(2);
plot(line_x1,line_y1,'k-','LineWidth',3);
hold on
plot(line_x2,line_y2,'m-','LineWidth',3);
hold on
legend('CL'.'Pure CA','Ca on CL'.'Pure PA','PA on CL');
hold on
title('PC-LDA')
dim = [1.15 .5 .3 .3];
str = 'Accuracy = 92.6%';
annotation('textbox',dim,'String',str,'FitBoxToText','on');

%to check accuracy
pcatable = readtable('principalcomponentsall.csv')
Type = pcatable(:,3)
Type = table2cell(Type)
measure = pcatable(1:27,1:2)
measure = table2array(measure)
indices = crossvalind('Kfold',Type,10);
cp = classperf(Type);
for i = 1:10
test = (indices == i); train = ~test;
class = classify(measure(test,:),measure(train,:),Type(train,:));
classperf(cp,class,test)
end
cp.CorrectRate

```

Appendix B – Code for Reconstruction ICA for unmixing Pure CL, Pure PA and random noise

```
% steps from https://www.mathworks.com/help/stats/extract-mixed-
signals.html
xlength=xCl(2:1029)
s1=yCl1(2:1029);
s2=yPaavq(2:1029);
s3=randn(1,1028);
s3=s3';

s=[s1,s2,s3]
rng default % For reproducibility
mixdata = s*randn(3) + randn(1,3);
figure
for i = 1:3
    subplot(2,3,i)
    plot(s(:,i))
    title(['Signal ',num2str(i)])
    subplot(2,3,i+3)
    plot(mixdata(:,i))
    title(['Mixed signals ',num2str(i)])
end
mixdata = prewhiten(mixdata);

q = 3; %where q is number of signals
Mdl = rica(mixdata,q,'NonGaussianityIndicator',ones(3,1));

unmixed = transform(Mdl,mixdata);

unmixed1 = unmixed(:,1); %separate unmixed signal into 3
unmixed1 = unmixed1*-1;
unmixed2 = unmixed(:,2);
unmixed3 = unmixed(:,3);

figure
for i = 1:3
    subplot(2,3,i)
    plot(s(:,i))
    title(['Signal ',num2str(i)])
    subplot(2,3,i+3)
    plot(unmixed(:,i))
    title(['Unmixed signals ',num2str(i)])
end

unmixed = unmixed(:, [2,1,3]);
for i = 1:3
    unmixed(:,i) = unmixed(:,i)/norm(unmixed(:,i))*norm(s(:,i));
end

figure
for i = 1:3
    subplot(2,3,i)
    plot(s(:,i))
    ylim([0,1])
    title(['Signal ',num2str(i)])
    subplot(2,3,i+3)
    plot(unmixed(:,i))
    ylim([-2,2])
    title(['Unmixed signals ',num2str(i)])
end

% calculate the Coefficient Covariance of actual signal vs unmixed signal
figure;
plot(s2)
title('Pa')
figure;
plot (unmixed1)
title ('Pa Unmixed')

R = corrcoef(s2,unmixed1)
```

Appendix C – Pre-whitening code

```
%provided by MATLAB website

function Z = prewhiten(X)
% X = N-by-P matrix for N observations and P predictors
% Z = N-by-P prewhitened matrix

% 1. Size of X.
[N,P] = size(X);
assert(N >= P);

% 2. SVD of covariance of X. We could also use svd(X) to proceed
but N
% can be large and so we sacrifice some accuracy for speed.
[U,Sig] = svd(cov(X));
Sig      = diag(Sig);
Sig      = Sig(:)';

% 3. Figure out which values of Sig are non-zero.
tol = eps(class(X));
idx = (Sig > max(Sig)*tol);
assert(~all(idx == 0));

% 4. Get the non-zero elements of Sig and corresponding columns
of U.
Sig = Sig(idx);
U   = U(:,idx);

% 5. Compute prewhitened data.
mu = mean(X,1);
Z = bsxfun(@minus,X,mu);
Z = bsxfun(@times,Z*U,1./sqrt(Sig));
end
```


Appendix D – C-ICA code

```
N=1028; %number of samples
k=1:N; %number of runs, from 1 to number of sample
rng default;

filename = ('raw data_all.xlsx');
yCL1 = xlsread(filename,'D2:D1029');
sheetPA = 6;
yPaavg = xlsread(filename,sheetPA,'AH2:AH1029');
sheetCA = 3;
yCaavg = xlsread(filename,sheetCA,'T2:T1029');

S(1,:) = yCL1;
S(2,:) = yCaavg;
S(3,:) = randn(1,N); %source random noise

S = standarize(S); %standardize the signals

NoiseSignal = S(3,:);

ICAshow(S(:,[2:1028]),'title','source signals');

A = rand(3); %random number
X = A*S; %mix the signal
ICAshow(X(:,[2:1028]),'title','mixed signals');
[X,V] = whiten(X); %perform whitening on signal

w = rand(size(X,1),1);
w = w/norm(w); %set w to random number

mu0 = 1;
lambda0 = 1;
gamma = 1;
learningRate = 1;
OverValue=0.000001; maxIter = 200; %set all parameters to 1, set maximum
iteration to 200

%Begin the cICA part here, if working on PA, run this part

yCa = yCaavg;
yCatranspose = yCa'; %transpose it so matrix dimension agree

% genRectangleRef = (length, period, firstPulsePos, pulseWidth)
% set ref1 to either yCatranspose, yPatranspose for raw signal. THE
REFERENCE SIGNAL DO NOT CHANGE. ONLY THE INPUT SIGNAL CHANGES.

ref1 = yCatranspose; threshold = 0.5;
[y1, w1] = cICA(X, ref1, threshold, w, learningRate, mu0, lambda0, gamma,
maxIter, OverValue); %perform cICA on the signals

% plot the results for Ca
figure;
subplot(2,1,1);plot(yCa(2:1028)); axis([-inf,inf,-5,5]); ylabel('Reference
Signal - Ca');
subplot(2,1,2);plot(y1(2:1028)); axis([-inf,inf,-5,5]); ylabel('Extracted
Signal - Ca');

CorCoefY1AgainstCA = corrcoef(yCaavg,y1)
```

Appendix E – Dilution Factor of CA and PA

Matrix A (Undiluted)	CL	Bacteria	Noise	Total Conc.	% of Target to Overall Conc.	Dilution Factor
Row 1	0.535664191	0.98914491	0.018177534			
Row 2	0.08707722	0.066946258	0.683838614			
Row 3	0.802091441	0.939398362	0.78373648			
Total	1.424832851	1.99548953	1.485752627	4.906075009	40.67384878	
Matrix F (Diluted by 50,000,000)						
Row 1	0.535664191	1.98E-08	0.018177534			
Row 2	0.08707722	1.34E-09	0.683838614			
Row 3	0.802091441	1.88E-08	0.78373648			
Total	1.424832851	3.99E-08	1.485752627	2.910585518	1.37E-06	29663076.02
Matrix G (Diluted by 70,000,000)						
Row 1	0.535664191	1.41E-08	0.018177534			
Row 2	0.08707722	9.56E-10	0.683838614			
Row 3	0.802091441	1.34E-08	0.78373648			
Total	1.424832851	2.85E-08	1.485752627	2.910585507	9.79184E-07	41538496.41
Notes						
For Original concentration	$Y = C1 * X1 + C2 * X2 + C3 * X3$					
	Target % = $C2 / (C1 + C2 + C3)$					
For Diluted Concentration	$Y = C1 * X1 + (C2 * a * X2) + C3 * X3$, where a is the dilution coefficient (50,000,000 or 70,000,000)					
	Where $(C2 * a * X2) = C^A$					
	Diluted Target % = $C^A / (C1 + C^A + C3)$					
To get the dilution factor	Target % / Diluted Target %					

Figure 18. Calculations for Dilution Factor