AIMS | African Institute for Mathematical Sciences CAMEROON

Optimizing Multiplexing for High-Throughput Experiments

Omer A.M. Elhussien (omer.elhussien@aims-cameroon.org)
African Institute for Mathematical Sciences (AIMS)
Cameroon

Supervised by: Prof. Bernhard Renard
ROBERT KOCH Institute, Germany

19 May 2017

Submitted in Partial Fulfillment of a Structured Masters Degree at AIMS-Cameroon

Abstract

Reproducibility is often hard to achieve in high throughput experiments due to missing calibration which holds particularly for proteomic mass spectrometry data. This fact posses a great problem at the analysis stage. To solve the arising situation, we employ a variety of multiplexing kits for a specific labelling of every sample, mixed and analysed as a single experiment. We adopt the commonly used $4-8$ markers and fight to increase the number of samples that we can possibly mark with the given limited number of markers. Using the concept of orthogonal arrays, we create a connection with the problem in question. We adopt two main methods, the Bush's method from finite field and the Rao-Hamming method from error-correcting code to create the orthogonal arrays. With the former, we are able to mix 6 samples using 4 markers while the latter is capable of mixing 14 samples with 8 markers. Finally, we seek by simulation an optimal setup which is able to analyse a higher number of samples in a single trial. We successfully obtained a number of mixture setups capable of combining 12 samples using 6 markers and 14 samples using 8 markers. We determined the effect of different levels of noise from different probability distributions on the mixture setups we obtained.

Declaration

I, the undersigned, hereby declare that the work contained in this essay is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Omer A.M. Elhussien, 19 May 2017.

Contents

1. Introduction

Reproducibility is often hard to achieve in high throughput experiments due to missing calibration. This fact posses a great problems at the analysis stage. Such a scenario is rampant in cases where scrutiny of different experimental conditions are exercised as in the mass spectrometry experiments that demands the analysis of every sample's protein. Normally, taking measurements with improper calibration leave the experimenter with less hope on the resulting data. A good example is in measuring the change in protein level when a given drug is examined for some time interval. In an attempt to solve such problems that may arise, we mostly employ multiplexing kits (Tandem Mass Tag, Isobaric Tags for Relative and Absolute Quantification) for a specific labelling of every sample, which is mixed and analysed as a single experiment. By so doing, we are able to separately determine abundance for each protein, thanks to their unique markers.

In practice, $4-8$ markers have been adopted. However, we are interested in increasing the number of samples that can possibly be marked by this limited markers.

In this research, we try to study the possibility of repressing the problems stated above. The question in mind is whether we can mix the unique markers in some already determined quantities and apply the result as pseudo-marker for generating beyond the common range of $4-8$ samples. We are also interested in taking measurements of the demultiplex, resulting in the intensity of the markers which in turn yield the intensity of the fundamental samples through a simple linear regression model. Given that the measured intensity of markers and the mixture of markers for each sample is known for every single protein, our focus is in decoding the implicit intensity for each sample. Nevertheless, a mathematical difficulty arises as the system in question is not well defined. More still, we are faced with extra constraints like require positivity to hold for all intensities or the limited dynamic range of the system.

1.1 Background

Mass spectrometry(MS) is amongst the most crucial tools that have a significant contribution in the study of relative quantitative scrutiny of proteins so as to realize the role they play in biological systems. A common method for relative quantification via MS is stable isotope labelling of proteins in samples preceding analysis.

Labelling can be achieved by the application of combinatorial heavy isotopologues of C,H, N and O. It can be introduced in proteins not only through metabolic means but also through a process known as chemical derivatization. The latter includes isotope-coded affinity tag (ICAT), isobaric mass tags (such as TMT or iTRAQ), and dimethyl labelling [16].

Stable isotope derivatization methods introduce a small variation to indistinguishable peptides from two or more samples so that they can be separately identified in the MS1 spectrum. Two types of ions product; reporter ion peaks and peptide fragment ion peaks are generated during MS/MS event. The quantification is realized by correlating the proportional intensity of reporter ions to that of the peptide designated for MS/MS fragmentation.

1.1.1 Experimental protocol of isobaric tag. Isobaric labelling experiment involves extraction of protein mixtures from different samples and are digested according to laboratory specific shotgun proteomics protocols. Different samples are then tagged by separate isobaric tags reaction mixtures for various labelling. These sample mixtures are afterwards combined in equal ratio for multiplexing prior

to detachment by high-performing liquid chromatography and MS [1].

The MS/MS process involves a number of reporter ions each assuming a unique mass. Further, the relative amount of peptide in the mixture, each labelled with a given reagent is shown by the intensity of the reporter ion. Here we presume efficient label description for the peptides irrespective of protein sequence. Isobaric tags come in different forms with the most common tag being amine reactive. Generally, the distribution of the isobaric mass tags depends on the heavy isotopes that form their structure. They are however, indistinguishable in terms of total mass.

1.1.2 Different types of isobaric tag. A specific group of tags, Isobaric Tags for Relative and Absolute Quantification commonly referred to as iTRAQ is made up of reporter group, a mass balancer group and a peptide reactive group. A process known as differential isotopic enrichment is tasked with keeping constant both the general mass of the reporter and the balance components of the molecule. The reaction chain results in the formation of an amide bond. Normally the process starts by tag reacting with peptide, loosing the balance moiety and the reporter group retaining some charge.

Optionally another isobaric tag that is commonly used is the Tandem Mass Tag (TMT) reagent which is responsible for indicating for each sample that is mixed, the reaction between an amine reactive group and peptides and normalization group whose task lies in balancing mass peptides upon MS/MS.

Let us consider a third isobaric mass tag, Dileu, one that employs a different type of reporter and is viewed as a better substitute for the ones discussed earlier. The reporter in question is an isotope-encoded dimethylated leucine. It is preferred because of its cheaper cost and ability to analyse up to four samples at the same time, quality disintegrations resulting in relatively higher reporter ion intensities. This in turn is a sign of better identification and efficiency in quantification.

Another option for isobaric tagging is Deuterium isobaric Amine Reactive Tag (DiART). The latter reagents have three functional groups namely a reporter, a balancer, and an amine reactive for coupling of peptides. A maximum of six samples can be analysed at once, and without consideration to the peptide sequence high-intensity report ions are generated in comparison to iTRAQ. As a result, DiART quantifies more peptides with different abundance, and dependable results. Users have capability to alternate between the techniques as a result of DiART 's use of same means of labelling as that of TMT and iTRAQ [16].

1.1.3 Identification and Quantification process. To successfully quantify the amount of reporter ions in the peptide, particular labelling focusing on each residues is required which must always carry on to the end. The comparative magnitudes of the reporter ion are a prerequisite for deducing the quantitative information from the labelled peptides in the different class of samples.

From laboratory results, it was noticed that proteins and peptides identified by iTRAQ 8-plex appeared relatively smaller compared to those identified by TMT 6-plex. The iTRAQ 4-plex identification resulted in the largest sizes possible. However, the reverse is observed when comparing their quantify them, that is iTRAQ 8-plex produced the largest sizes followed by TMT 6-plex and finally the iTRAQ 4-plex. Search algorithm and scoring functions were thought to be the cause of the observed variation in the identification process when applying a given n-plex isobaric mass tag. Of the different mass tags discussed so far, 8-plex is more advantageous in that per experiment, it has the capability of admitting up to eight experimental conditions [16].

The superlative quantity of peptide is indicated by the proportion of the intensity of reporter ions but to maximize on detection and have a better approximation, one needs to uniquely evaluate labelling each time a new experiment is in place. Normally, readjusting reporter ion peaks is a way of attaining reliable quantitative measurements as the process eliminates the effect of neighbouring reporter ion.

By considering a lone precursor ion, a more precise proportion can be assessed in all experiments that employ isobaric labelling. This is because the usage of non- specific peptide on a given protein results in a reasonably large measurement error.

A new concept, peptide abundance proportions [16] results from joining data from a number of fractions. These are then averaged to give the proportions of each protein. By definition, fraction effect for any peptide is a representation of the relation of measured ratio and the proportion of the source data.

1.1.4 Advantages and Disadvantages of isobaric tags. Comparatively, quantitative labelling based on isobaric tag is capable of handling a number of samples at a time while maintaining their structure, resulting in high-throughput quantification. Moreover, this kind of tag is not susceptible to the randomness of data-dependent MS. It also exceeds metabolic labelling in taking precision and has the ability to replicate. Still an isobaric tag is more advantageous in that it assumes a large range when identifying high as well as low-abundance proteins. As such, records indicate that it improves the efficiency of MS/MS fragmentation leading to a higher signal intensity when dealing with some usual peptides.

Just like in the case of lone precursor ions, reporter ion signal intensity also plays a very important role in the quality achieved while using an isobaric tag. Thus, if the right bound of intensity is not applied, then there may result an overestimated or underestimated value of the quantification proportion. However, given the biaseness of isobaric tag towards greater proportion of abundant proteins, signal intensities mostly fall outside the required range.

Since a specific range of reporter ion intensity gives the desired results of measurement, one need to always balance the intensity. In case it is low, a simple solution lies in prolonging the time for MS/MS, giving rise to an increased intensity. In general, a larger intensity is recipe for selecting a class of peptide at the time when analysis of LC-MS/MS is run [16].

1.2 The Research problem

In this research project, we focus on two crucial problems in high throughput experiments specially in MS. Firstly, an isobaric tag has the capability to analyse up to 8 samples in each experiment, while the need for more comparisons exist. Consequently, we want to study the possibility of extending the number of samples by combining unique markers with common quantities then use them as pseudo-markers. Secondly, we want to trace the true amount of proteins in each sample according to the above mixture setup.

So, the research problem can be formulated as follows [3]. Consider a matrix $X\ (n \times k)$ and $Y\ (n \times 1)$, a vector $\beta\ (k \times 1)$ is sought to

$$\text{minimize} \quad \|Y - X\beta\|^2$$

subject to :

$$0 \leq \beta_i \leq 10^6 \quad ,\forall i = 1, 2, \ldots, k.$$

Where β_i is an element of β and is the concentration of proteins in the i^{th} sample. Further, it is known that the concentration of proteins β_i cannot exceed the given limit. However, the matrix X with n rows represent the number of samples, and k columns correspond to the number of markers that has been used during the experiment.

The matrix X has a special form. For instance, let us consider that, four samples have been combined

using three markers. Assuming the matrix X is given as follows

$$X = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

So, it can be interpreted as, sample 1 is encoded by markers 1 and 2 being present while the third marker being absent and so on.

1.3 Outline of the research project

In chapter 2, we introduce the concept of an orthogonal array and its properties. Then, we present Bush's method for building an optimal mixture setup with 8 samples and 4 markers. Further, we introduce several concepts related to the Error-Correcting codes. Finally, we end by discussing the Rao-Hamming method to build an optimal mixture setup with 14 samples and 8 markers.

In chapter 3, we discuss the demultiplex algorithm for our simulated setups. Then, we introduce the linear regression model which we build on this algorithm. Moreover, we present the mixture algorithm for simulated setups and we discuss the results we get from this algorithm. Finally, we end by determining the effect of different levels of noise on the generated mixture setups.

In chapter 4, we conclude the research project.

2. The Optimal mixture setup

The concept of orthogonality plays a significant role in statistical work. Its application ranges from Experimental design to estimation and analysis of variance as an orthogonal vector basically is equivalent to a random variable with no correlation.

The matrix with unique frame which has been displayed in the pervious part is referred to in [6] and [10] as orthogonal array and has a particular definition with numerous characteristics which makes it invaluable to be utilized in practical.

In this chapter, we describe the idea of an orthogonal array and its properties, we explain some hypotheses identified with this field. Then, we examine Bush's strategy for building an orthogonal array. Furthermore, we clarify the idea of Error-Correcting code and its connection to the orthogonal array. At last, we present the Rao-Hamming technique for developing a code.

2.1 Definitions and properties

In this area, we think of some expressions and characteristics about the orthogonal array concept. At that point we present Rao's inequality and the idea of tight array. At long last, we demonstrate fascinating change in the past inequality.

2.1.1 Definition. [10]

Given S as a set of s levels and A is $N \times k$ array. If every $N \times t$ sub-array $(0 \leq t \leq k)$ of A exhibits each $t-$tuple specifically λ times as a row, then A is called an orthogonal array with s levels, strength t, index λ, and is denoted by $OA(N, k, s, t)$.

From the above definition, we realise that N represents the number of samples or runs and k represents the number of factors which have been used in the experiment.

The above array has several properties which can be stated as follows [10],

(a) The index parameter fulfil the following equality

$$\lambda = \frac{N}{s^t} \quad .$$

(b) If A is an $OA(N, k, s, t)$ with index λ, then A is also an $OA(N, k, s, t^{'})$ such that $0 \leq t^{'} < t$ with index $\lambda^{'} = \frac{N}{s^{t^{'}}} = \frac{\lambda s^t}{s^{t^{'}}} = \lambda s^{t-t^{'}}$.

(c) If A_i , $i = 1, 2, \ldots, l,$ is an $OA(N_i, k, s, t_i)$ then $A = \begin{bmatrix} A_1 & A_2 & \ldots & A_l \end{bmatrix}^T$ is an $OA(N_1 + N_2 + \cdots + N_l, k, s, t)$, with $t \geq \min\{t_1, t_2, \ldots, t_l\}$. Moreover, if $l = s$, A_i is an $OA(N, k, s, t), \forall i$ and by appending i to each row of A_i, $i = 1, 2, \ldots, l$; then A is an $OA(sN, k+1, s, t)$.

(d) A replacement of the samples or factors does not affect on the orthogonal array parameters.

(e) A replacement of the levels of any column does not affect on the orthogonal array parameters.

(f) Let A be an $OA(N,k,s,t)$ then any sub-array $A^{'}(N \times k^{'})$ is an $OA(N,k^{'},s,\min\{k^{'},t\})$.

(g) Let A be an $OA(N,k,s,t)$ then by considering the rows which begin with a specific level and removing the first column, we get an $OA(\frac{N}{s},k-1,s,t-1)$.

(h) Let M be the set of all possible samples which can be constructed in a specific orthogonal array A, f_b is the number of b in A where $b \in M$ and f is the maximal f_b overall $b \in M$. The complement of A is the array which contains sample b with iteration $f - f_b \ , \forall \ b \in M$. Mathematically, an $OA(fs^k - N,k,s,t)$ is a complement of an $OA(N,k,s,t)$.

(i) Let A be an $OA(\lambda 2^t,k,2,t)$ then by replacing the two symbols, we get $\overline{A}$ which is known as the binary complement of A.

(j) Especially, let $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}^T$ be an $OA(N,k,s,t)$ with A_1 is an $OA(N_1,k,s,t_1)$ then A_2 is an $OA(N-N_1,k,s,t_2)$ where $t_2 \geq \min\{t,t_1\}$.

From the above properties and specifically (d) and (e), we can define the concepts of Isomorphic and Statistically equivalent according to the orthogonal array perspective.

2.1.2 Definition. [10]

Let A and B be orthogonal arrays. If A can be retrieved from B after a series of replacements of the samples, the factors and the levels of the factors, then A is Isomorphic to B. However, if A can be retrieved from B by replacing the samples, then A is Statistically equivalent to B.

There are two issues in the field of orthogonal array which have been drawing in scientists for quite a while and a considerable measure of work has been done around there [2], [18], [17], [9]. The main issue is that, given k, s and t, what is the minimum number of N for such an $OA(N,k,s,t)$ exists and is signified by $F(k,s,t)$. The second issue is that, given N, s and t, what is the greatest number of k for such an $OA(N,k,s,t)$ exists and is indicated by $f(N,s,t)$. Nevertheless, our exploration issue is somewhat extraordinary and can be expressed as, given $s=2$, $k=4$, $k=6$ and $k=8$, we look for the maximum number of N for which an $OA(N,k,s,t)$ occurs.

With a specific number of parameters s, k and t there is a limit which an orthogonal array exists with N samples and this relation is presented by an inequality called Rao's inequality and is stated as follows.

2.1.3 Theorem. [10]

Given s, t, k and $m \geq 0$ then the parameters of an $OA(N,k,s,t)$ fulfil the following inequalities

$$N \geq \sum_{j=0}^{m} \binom{k}{j} (s-1)^j \ , \quad \text{if} \ \ t = 2m, \tag{2.1.1}$$

$$N \geq \sum_{j=0}^{m} \binom{k}{j} (s-1)^j + \binom{k-1}{m} (s-1)^{m+1} \ , \quad \text{if} \ \ t = 2m+1. \tag{2.1.2}$$

Proof. Let A be an $OA(N,k,s,t)$ with elements $a_{ij} \in S$ and $S = \{0,1,\ldots,s-1\}$. Let V_1 and V_2 be the right hand side of 2.1.1 and 2.1.2 respectively. In this proof, we will use A to build a matrix H with $N \times V_1$ elements and rank V_1. So, $V_1 \leq N$ and that proves the theorem.

Let B be a matrix with $(s-1)\times s$ elements and it has pairwise orthogonal rows. Also, all the rows of B are orthogonal to 1_s which is $s\times 1$ vector with all elements equal to 1. B will be in the form,

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ s-1 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & \dots & s-1 \end{array} \\ \begin{pmatrix} b(1,0) & b(1,1) & \dots & b(1,s-1) \\ b(2,0) & b(2,1) & \dots & b(2,s-1) \\ \vdots & \dots & \vdots & \\ b(s-1,0) & b(s-1,1) & \dots & b(s-1,s-1) \end{pmatrix} \end{array}.$$

Then,

$$\sum_{j=0}^{s-1} b(i,j)\ b(i^{'},j)=0,\ i\neq i^{'},$$

and

$$\sum_{j=0}^{s-1} b(i,j)=0.$$

In the case when $t=2m$. We define $H(i_1,i_2,\dots,i_u)$ where $i_j\in\{1,2,\dots,s-1\}$ and $u\in\{1,2,\dots,m\}$ with $N\times\binom{k}{u}$ elements. Let $h^{(i_1,i_2,\dots,i_u)}_{j,(l_1,l_2,\dots,l_u)}$ be an element of H in row j and one of the possible columns $(l_1,l_2,\dots,l_u)$. Then, we define

$$h^{(i_1,i_2,\dots,i_u)}_{j,(l_1,l_2,\dots,l_u)}=b(i_1,a_{jl_1})b(i_2,a_{jl_2})\dots b(i_u,a_{jl_u}),$$

and H is given by

$$H=\left[1_N,H(1),\dots,H(s-1),H(1,1),H(1,2),\dots,H(\underbrace{s-1,\dots,s-1}_{m\text{ elements.}})\right].$$

So, the number of columns in H is $1+\binom{k}{1}(s-1)+\binom{k}{2}(s-1)^2+\dots+\binom{k}{m}(s-1)^m=V_1$.

In the case when $t=2m+1$. We define $\overline{H}(i_1,i_2,\dots,i_{m+1})$ where i_j is same as before with $N\times\binom{k-1}{m}$ elements. Let $h^{(i_1,i_2,\dots,i_{m+1})}_{j,(1,l_2,\dots,l_{m+1})}$ be an element of $\overline{H}$ in row j and one of the possible columns $(1,l_2,\dots,l_{m+1})$. Then,

$$h^{(i_1,i_2,\dots,i_{m+1})}_{j,(1,l_2,\dots,l_{m+1})}=b(i_1,a_{j1})b(i_2,a_{jl_2})\dots b(i_{m+1},a_{jl_{m+1}}).$$

By adding the columns of $(s-1)^{m+1}$ matrices in $\overline{H}$ to H we get $N\times V_2$. From the form of H and $\overline{H}$ elements, we notice that all the columns of H and $\overline{H}$ are non-zero.

Now, we want to confirm that all the columns are orthogonal. By using the fact that any u columns of the orthogonal array A with $u\leq t$ form an orthogonal array with $t=u$ and $\lambda=\frac{N}{s^u}$.

$$\begin{aligned}\sum_{j=1}^{N} h^{(i_1,i_2,\dots,i_u)}_{j,(l_1,l_2,\dots,l_u)} &= \sum_{j=1}^{N} b(i_1,a_{jl_1})b(i_2,a_{jl_2})\dots b(i_u,a_{jl_u}) \\ &= \frac{N}{s^u}\left[\sum_{j_1=0}^{s-1}\cdots\sum_{j_u}^{s-1} b(i_1,j_1)b(i_2,j_2)\dots b(i_u,j_u)\right]\end{aligned}$$

$$= \frac{N}{s^u}\left[\sum_{j_1=0}^{s-1}\cdots\sum_{j_{u-1}=0}^{s-1} b(i_1,j_1)\ldots b(i_{u-1},j_{u-1})\left(\sum_{j_u=0}^{s-1} b(i_u,j_u)\right)\right] = 0.$$

□

For instance, given $s=2$, $k=4$ and $t=2$ then from the above theorem, we get $m=1$ and $N\geq 5$.

There is a particular kind of orthogonal arrays achieve the equality of Rao's inequalities and they are called tight or complete orthogonal arrays. A considerable work has been done in this type of arrays with strength $t=4$ [14] and we are interested in the case when the number of levels is 2.

2.1.4 Theorem. [10]

A tight orthogonal array with parameters $t=4$, $s=2$, $k=5$ and $N=16$ exists and is unique.

A proof can be found in [14].

From the above theorem, we can conclude that, given five markers, we can combine up to 16 samples in one experiment.

The maximum number of factors for an orthogonal array with strength $t=2$ or $t=3$, has been obtained in [10] using 2.1.3 and is given by the following Corollary.

2.1.5 Corollary. [10]

Let A be an $OA(\lambda_1 s_1^2, k_1, s_1, 2)$ and B an $OA(\lambda_2 s_2^3, k_2, s_2, 3)$ then

$$k_1 \leq \frac{\lambda_1 s_1^2 - 1}{s_1 - 1}, \tag{2.1.3}$$

$$k_2 \leq \frac{\lambda s_2^2 - 1}{s_2 - 1} + 1 \tag{2.1.4}$$

and if $\lambda_i - 1 \not\equiv 0 \pmod{s_i - 1}$, $i = 1, 2$. then

$$k_1 \leq \lambda_1(s_1+1) + a_1 - \lfloor\theta_1\rfloor - 1, \tag{2.1.5}$$

$$k_2 \leq \lambda_2(s_2+1) + a_2 - \lfloor\theta_2\rfloor, \tag{2.1.6}$$

where, $\lambda_i - 1 = a_i(s_i-1) + b_i$, $0 \leq b_i \leq s_i - 2$ and $\theta_i = \frac{\sqrt{1+4s_i(s_i-1-b_i)}-(2s_i-2b_i-1)}{2}$ for $i = 1, 2$.

Using the above Corollary, by assuming $s_1 = s_2 = 2$, $\lambda_1 = \lambda_2 = 1$ then $k_1 \leq 3$ and $k_2 \leq 4$ respectively. Therefore, we can combine up to 8 samples using 4 markers and 4 samples using 3 markers.

A considerable improvement has been achieved in Theorem 2.1.3 by [5], [11] and an interesting results can be deduced from the following theorem.

2.1.6 Theorem. [5]

Let A be an orthogonal array with parameters s, t, k, $N = s^t$ and index unity, then

$$\begin{aligned} k &\leq t+1 \ , && \text{if } s \leq t, \\ k &\leq s+t-2 \ , && \text{if } s > t \geq 3 \text{ and } s \text{ is odd}, \\ k &\leq s+t-1 \ , && \text{otherwise.} \end{aligned}$$

A proof can be found in [5].

Let us assume $t = 3$ then by using the above theorem, we get $k \leq s + 2$. Therefore, we conclude there exists an $OA(s^3, s + 2, s, 3)$.

2.2 Bush's method for constructing an orthogonal array

Bush's construction is advantageous as it results in reduction of time and expenditure in statistical experiments. As such, we employ the technique to construct the orthogonal array of index unity that gives the least number of unique rows per every level per strength. It has been constructed using the concept of Galois field (GF) and is given as follows.

2.2.1 Theorem. [5]

Let $s \geq 2$ be a prime power and $s \geq t - 1 \geq 0$, then there exists an $OA(s^t, s + 1, s, t)$.

The method of constructing such an array is given by the following steps:

(a) Let $\{\alpha_0, \alpha_1, \ldots, \alpha_{s-1}\} \in \mathrm{GF}(s)$, $\psi_i(x)$ be the polynomial over $GF(s)$ for $i = 1, 2, \ldots, s^t$. with degree at most $t - 1$ to ensure the linearity, and a_i is the leading coefficient of ψ_i.

(b) Construct such a matrix with $s^t \times s$ entries,

	α_0	α_1	α_2	$\ldots$	α_{s-1}
$\psi_1(x)$	$\psi_1(\alpha_0)$	$\psi_1(\alpha_1)$	$\psi_1(\alpha_2)$	$\ldots$	$\psi_1(\alpha_{s-1})$
$\psi_2(x)$	$\psi_2(\alpha_0)$	$\psi_2(\alpha_1)$	$\psi_2(\alpha_2)$	$\ldots$	$\psi_2(\alpha_{s-1})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$
$\psi_{s^t}(x)$	$\psi_{s^t}(\alpha_0)$	$\psi_{s^t}(\alpha_1)$	$\psi_{s^t}(\alpha_2)$	$\ldots$	$\psi_{s^t}(\alpha_{s-1})$

(c) We add another column which consists of the leading coefficient of ψ_i, $\forall i$. And the matrix is given below,

$$\begin{bmatrix} \psi_1(\alpha_0) & \psi_1(\alpha_1) & \psi_1(\alpha_2) & \ldots & \psi_1(\alpha_{s-1}) & a_1 \\ \psi_2(\alpha_0) & \psi_2(\alpha_1) & \psi_2(\alpha_2) & \ldots & \psi_2(\alpha_{s-1}) & a_2 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ \psi_{s^t}(\alpha_0) & \psi_{s^t}(\alpha_1) & \psi_{s^t}(\alpha_2) & \ldots & \psi_{s^t}(\alpha_{s-1}) & a_{s^t} \end{bmatrix}.$$

Proof. We proof the validity of the above construction. We want to show that all the rows are distinct in any sub-array $s^t \times t$ using contradiction. First, we assume that all the t factors in the sub-array are among the first s factors. Let us assume

$$\psi_j(z_i) = \psi_{j'}(z_i) \text{ for } i = 1, \ldots, t \text{ and } j \neq j'$$

or, we can write

$$\psi(z_i) = 0 \ , \ i = 1, \ldots, t.$$

where $\psi = \psi_j - \psi_{j'}$. Since ψ has degree at most $t - 1$ and it has t zeros in this case that means ψ is identically zero and $j = j'$ which is a contradiction.

Also, we assume that the last column is one of the t-factors of the sub-array. Then, by assuming

$$\psi_j(z_i) = \psi_{j'}(z_j) \text{ for } i = 1, \ldots, t - 1 \text{ and } j \neq j'$$

and the leading coefficients of ψ_j and $\psi_{j'}$ are the same. Similarly, we can rewrite it as

$$\psi(z_i) = 0 \ , \ i = 1, \ldots, t-1.$$

where $\psi = \psi_j - \psi_{j'}$ and ψ now has degree $t-2$ that implies ψ is identically zero and $j = j'$ which is a contradiction. As a result, from the two cases, we conclude that all the rows of the sub-array are unique. □

An improvement of the above theorem has been done and is given by the following theorem.

2.2.2 Theorem. [10]

Let $s = 2^m$ with $m \geq 1$ and $t = 3$, then an $OA(s^3, s+2, s, t)$ exists.

The method of constructing such an array is given below:

(a) Let b_i be the coefficient of x in ψ_i for $i = 1, 2, \ldots, s^t$.

(b) By using Theorem 2.2.1 we construct an $OA(2^{3m}, s^{3m}+1, 2^m, 3)$.

(c) We add another column which contains b_i for each row i.

Proof. In the case when all the three factors of $2^{3m} \times 3$ sub-array are from the first $2^m + 1$ has been proved in the previous theorem. Assume the new factor is from $2^{3m} \times 3$ sub-array and specifically from the first 2^m factors. Further, by assuming the new factor corresponds to column z_1 then

$$\psi(z_1) = 0,$$

where ψ is polynomial of degree zero and it is clear that ψ is identically zero.

In the case when the two factors are from the first 2^m factors, specifically columns z_1 and z_2. We get $\psi(z_1) = \psi(z_2) = 0$, where ψ is of degree at most 2 with x coefficient equal to zero. By using the fact that $z_1^2 = z_2^2$ in GF(2^m) if and only if $z_1 = z_2$, we conclude that ψ is identically zero. Hence, all the rows in any $2^{3m} \times 3$ sub-array are unique. □

The array which has been constructed using the previous theorems has two specific properties which are simplicity and linearity.

2.2.3 Definition. [10]

A simple orthogonal array is an array with unique samples.

2.2.4 Definition. [10]

Given s a prime number, with $S = \{\alpha_0, \alpha_1, \ldots \alpha_{s-1}\} \in$ GF(s) and A is an $OA(N, k, s, t)$. A is linear, if it is simple and for any two rows R_1 and R_2 in A, then $\alpha_1 R_1 + \alpha_2 R_2$ is a row in A.

In comparison with orthogonal array, linear orthogonal arrays are more desirable in two ways,

(a) Let $v_1, v_2, \ldots, v_n$ be a basis for the vector space, $M = \begin{bmatrix} v_1 & v_2 & \ldots & v_n \end{bmatrix}^T$ and is known as a generator matrix with $(n \times k)$ entries. Then a matrix M can be used to get the rows of a linear orthogonal array by simply calculating all the linear combinations $\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$, with $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \in$ GF(s).

(b) The rows of a linear orthogonal array can be considered as the codewords of a linear error-correcting code. Consequently, it can be analysed using the concepts of coding theory.

In the previous section, we proved the existence of an $OA(s^3, s+2, s, 3)$ and using the above theorems we construct such a matrix in the following example.

2.2.5 Example. Given $S = \{0, 1\} \in \mathrm{GF}(2)$, construct an $OA(8, 4, 2, 3)$.

First, by using Theorem 2.2.2 and the properties of a linear orthogonal array. We are given $\{0, 1\} \in \mathrm{GF}(2)$ and such polynomials over $\mathrm{GF}(2)$ are $\{1, x+1, x^2+x+1\}$.

Second, we construct a generator matrix as follows,

	0	1		
1	1	1	0	0
$x+1$	1	0	0	1
x^2+x+1	1	1	1	1

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Finally, we find all the linear combinations for the rows of M and the $OA(8, 4, 2, 3)$ is given below,

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}^T.$$

2.3 Error-Correcting codes

Error-correcting codes are intended to correct mistakes in the transmission of information through noisy communication channels. A code is defined as a collection of codewords or vectors in S^k where S is the set of s symbols and S^k is the set of all vectors s^k with length k based on S.

The frequency of non-zero elements in a specific vector $v = (v_1, \ldots, v_k) \in S^k$ is known as Hamming weight $w(v)$. However, the frequency of differences between two vectors u and v is referred to as Hamming distance and is given as, $\mathrm{dist}(u, v) = w(u - v)$ where $u, v \in S^k$. Moreover, the minimal distance of a code is defined to be $d = \min \mathrm{dist}(u, v)$ where u and v are unique codewords and such a minimum exists since $\mathrm{dist}(u, v) \subset \mathbb{Z}^+$. A code with N codewords, minimal distance d and each codeword has a length k based on s symbols is denoted by $(k, N, d)_s$ [12].

Similarly, as we have mentioned in the orthogonal array, we say that a code is simple if it has distinct codewords. Also, we say that the code is linear if it is simple and is a vector subspace of S^k with minimal distance $d = \min w(u)$, for $u \neq 0$. Given a simple code C with minimal distance d and $e = \lfloor \frac{d-1}{2} \rfloor$, then the code C can correct up to e errors per message and is called an $e-$error-correcting code.

In addition, parity check matrix and generator matrix are two parameters for representing a linear code, the basis of the code forms the rows of the generator matrix G $(n \times k)$, while the orthogonal space of

the code forms its rows in the case of the parity check matrix P $(k-n \times k)$. Let A be a matrix with elements from S, then G and P can be given as [10],

$$G = \begin{bmatrix} I_n & A \end{bmatrix} \ , \ P = \begin{bmatrix} -A^T & I_{k-n} \end{bmatrix}.$$

For a given linear code C there exists its dual $C^{\perp}$, this concept of Duality has been working as a backbone which connects the orthogonal arrays with coding theory. A dual code has several properties which have been stated in the following theorem.

2.3.1 Theorem. [10]

Given C as $(k, s^n, d)_s$ linear code, with P and G as parity check and generator matrices respectively. Then, there exists $C^{\perp}$ as $(k, s^{k-n}, d^{\perp})_s$ linear code with generator matrix P and parity check matrix G. Further, $C = (C^{\perp})^{\perp}$.

Proof. $\forall u \in S^k$, u is codeword of C if and if $P\ u^T = 0$. Also, we need to define the dual code of C as follows,

$$C^{\perp} = \{v |\ u\ v^T = 0\ \forall u \in C\},$$

$u.v^T = 0$ or equivalently, $v.u^T = 0$, that means $C^{\perp}$ is the set of all parity checks on C. So, if C has generator matrix G with $n \times k$ elements and parity check matrix P with $(k-n) \times k$ elements, then $C^{\perp}$ has generator matrix P and parity check matrix G. Hence, $C^{\perp}$ has length k, dimension $k-n$ and minimal distance $d^{\perp}$. So, $C^{\perp}$ is $(k, s^{k-n}, d^{\perp})_s$ linear code. $(C^{\perp})^{\perp}$ has generator matrix G and parity check matrix P, that means it is C. □

For a given $OA(N, k, s, t)$ there exists a $(k, N, d)_s$ code and vice versa. There are two theorems which connect the orthogonal arrays with codes and they are given below.

2.3.2 Theorem. [10]

Let C be a code and A be an orthogonal array related to C. We say C is linear if and only if A is linear.

A proof is trivial from definition 2.2.4 and the concept of a linear code.

2.3.3 Theorem. [10]

Let S be a set of s symbols over GF(s). C is a $(k, N, d)_s$ linear code with elements from S if and only if there exists a linear $OA(N, k, s, t)$ with $d^{\perp} = t + 1$.

Proof. We need to use the theorem that mentioned, Let A be an $OA(N, k, s, t)$ with elements from GF(s). Then, any t columns of A are linearly independent over GF(s).

In the case when C is a $(k, N, d)_s$ linear code and A is the array formed by its codewords. Any $d^{\perp} - 1$ columns of A must be linearly independent over GF(s). In the case when one of these columns is dependent, that means there is a codeword in the dual code with weight less than $d^{\perp}$ which is contradiction. So, A is an $OA(N, k, s, d^{\perp} - 1)$.

Also, in the case when A is an $OA(N, k, s, t)$ and C is a code related to A. Any t columns of A are linearly independent and so some $(t+1)$ columns of A are dependent that means there is a codeword of weight $t+1$ in the dual code. Therefore, $d^{\perp} = t + 1$. □

Isomorphic concept is not only applied to the orthogonal array but also is applied to the code and we say two codes are isomorphic if the following definition holds.

2.3.4 Definition. [10]

Let C and D be two codes. If C can be retrieved from D by replacing the codewords and the elements of each codeword, then C is isomorphic to D. However, if C and D are two linear codes and C can be retrieved from D by replacing the codewords, then C is isomorphic as linear code to D. Further, isomorphic as linear code is similar to statistically equivalent in the orthogonal array.

There are numerous ideas we have to introduce before we go further to the strategies for developing a code. Thus, we present the presence of a particular sort of codes and the ability of such a code to be developed, then we demonstrate the idea of cyclic code.

2.3.5 Theorem. [12]

Let C be a $(k, N, 2m+1)_2$ code and D be a $(k, N, 2m+2)_2$ code. C exists if and only if D exists.

A proof can be found in [12].

By adding 0 to each codeword with even Hamming weight and 1 otherwise, we get a $(k+1, N, 2m+2)_2$ code from a $(k, N, 2m+1)_2$ code and we say that the code $(k, N, 2m+1)_2$ has been extended. Further, the code $(k+1, N, 2m+2)_2$ is known as the extended code.

2.3.6 Definition. [12]

Let C be a linear code with such a codeword $(c_0, c_1, \ldots, c_{k-2}, c_{k-1}) \in \mathrm{GF}(s)$. If C has another codeword $(c_1, c_2, \ldots, c_{k-1}, c_0)$, then C is called a cyclic code and is denoted by $\langle c_0, c_1, \ldots, c_{k-2}, c_{k-1}\rangle$.

A cyclic code with size $N = s^n$ has a vector $\psi = (\beta_0, \beta_1, \ldots, \beta_{k-1})$ known as a generating vector. Therefore, the generator matrix of the code is given by $G = \begin{bmatrix}\psi & \psi_1 & \psi_2 & \ldots & \psi_{n-2} & \psi_{n-1}\end{bmatrix}^T$ where ψ_1 is a cyclic shift of ψ and ψ_i is cyclic shift of ψ_{i-1} for $i = 2, 3, \ldots, n-1$. Furthermore, the generator polynomial of the code is given by $\psi(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_{k-1} X^{k-1}$.

A generator polynomial of any cyclic code has several properties need to be satisfied and they are stated in the following theorem. Also, we connect a cyclic code with its dual.

2.3.7 Theorem. [10]

Let C be a cyclic $(k, s^n, d)_s$ code. If $\psi(X)$ is a polynomial with minimal degree $k-n$ and the leading coefficient is unity. Further, $\psi(X)$ divides $X^k - 1$ over $\mathrm{GF}(s)$ and $\phi(X)\psi(X)$ represents all the code where $\phi(X)$ are polynomials with degree at most $n-1$ and coefficients from $\mathrm{GF}(s)$. Then $\psi(X)$ is the generator polynomial of the code C and is unique.

Proof. Let ψ' be another monic polynomial with minimal degree $k-n$, then $\psi'(X) - \psi(X) \in C$ and has degree less than $k-n$ which contradict the fact that ψ' and ψ have minimal degree unless $\psi' = \psi$. So, $\psi(X)$ is unique.

Let $\varphi(X) \in C$ and write $\varphi(X) = \phi(X)\psi(X) + \Gamma(X)$ in $\mathrm{GF}(s)$, then $\Gamma(X) = \varphi(X) - \phi(X)\psi(X) \in C$ since the code is linear. Therefore, $\Gamma(X) = 0$.

Let us write $X^k - 1 = \Omega(X)\psi(X) + \Lambda(X)$ in $\mathrm{GF}(s)$, then $\Lambda(X) = -(\Omega(X)\psi(X)) \in C$ which is a contradiction unless $\Lambda(X) = 0$. So, $\psi(X)$ divides $X^k - 1$. □

2.3.8 Theorem. [10]

Let C be a cyclic code with codewords from GF(s) and each codeword has length k. If C has a generator polynomial $\psi(X)$ then its dual $C^{\perp}$ has a generator polynomial

$$\frac{X^k-1}{\overleftarrow{\psi}(X)} \quad , \quad \text{where} \quad \overleftarrow{\psi}(X) = X^{\deg(\psi)}\psi(\frac{1}{X}).$$

A proof can be found in [12].

2.4 The Rao-Hamming method for constructing a code

The $(k, s^{k-m}, 3)_s$ code where s is a prime power and $k = \frac{s^m-1}{s-1}$, $m \geq 2$ is referred to as Hamming code with single-error-correction. The parity check matrix $P(m \times k)$ of such a code has unique columns with elements from GF(s). Further, the orthogonal array related to the Hamming code is given by $OA(s^{k-m}, k, s, s^{m-1}-1)$ and is known as orthogonal array of Rao-Hamming type.

Moreover, this type of code has a special technique of construction for some symbols s. At the point when $s = 2$, the coordinate places of a Hamming code can be organized to form the code cycle, thus a more brief depiction will be given for these codes and for orthogonal arrays of Rao-Hamming sort. As generator polynomial for the Hamming code we can utilize any primitive irreducible polynomial of degree m over GF(2). This code can be expanded to give a dual code $(2^m, 2^{m+1}, 2^{m-1})_2$ with $OA(2^{m+1}, 2^m, 2, 3)$ [10].

2.4.1 Example. Given $s = 2$ and $m = 3$. Construct the Hamming-code with those parameters then extend the code to get an $OA(16, 8, 2, 3)$.

For $s = 2$ and $m = 3$ we have $k = 7$. Now, the primitive irreducible polynomial of degree $m = 3$ over GF(2) is given by $1 + x + x^3$.

So,

$$\psi(X) = 1 + X + X^3$$

is a generator polynomial of $(7, 16, 3)_2$ code. Let us check the assumptions of Theorem 2.3.7.

We have $N = 16 = 2^4 = s^n$. So, the degree of ψ is $k-n = 7-4 = 3 = m$. And the leading coefficient is 1. Also,

$$\frac{X^7-1}{1+X+X^3} = X^4 + X^2 + X + 1.$$

The last assumption will be used to construct the orthogonal array related to the code. But, we notice all the three assumptions have been satisfied.

From the generator polynomial, the generator vector is given by

$$\psi = \langle 1\ 1\ 0\ 1\ 0\ 0\ 0 \rangle.$$

Then, we extend the code to get the $(8, 16, 4)_2$ code and since the Hamming weight of the generator vector is odd, we add 1 to it. The generator vector of the extended code will be in the following form,

$$\psi = \langle 1\ 1\ 0\ 1\ 0\ 0\ 0 \rangle 1.$$

Further, we use the idea after Definition 2.3.6 to get the generator matrix as follows,

$$G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Finally, we use the same idea as in Example 2.2.5, by finding all the linear combinations of the rows of G and the $OA(16, 8, 2, 3)$ is given below,

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T.$$

2.5 Summary

In this chapter, we can develop an $OA(8, 4, 2, 3)$ and an $OA(16, 8, 2, 3)$ at the same time, the situation when $k = 6$ has not been considered in the strategies we talked about. Nonetheless, we guarantee there exist an $OA(28, 8, 2, t)$ and an $OA(15, 6, 2, t)$ for some strengths t. We demonstrate this amid the simulation part.

3. Demultiplex process and simulation

In this chapter, we present the demultiplex algorithm for simulated data and we introduce the idea of linear regression model which we built on this algorithm. Then, we introduce the concepts we used to build the simulation algorithm. Further, we present the algorithm and we discuss the results. Finally, we discuss the effect of different levels of noise on the results we get from the simulation.

3.1 Linear regression model

First, we discuss the case when we have one independent variable X. Then, the linear regression model can be stated as follows:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \quad , \quad i = 1, 2, \ldots, n.$$

where:

Y_i is the value of the dependent variable in observation i.

β_0 and β_1 are parameters.

X_i is the value of the independent variable in observation i.

ϵ_i is the random error term and it follows $\mathcal{N}(0, \sigma^2)$. Also, ϵ_i and ϵ_j are independent for all i, j when $i \neq j$.

$Y_i \sim \mathcal{N}(\beta_0 + \beta_1 X_i, \sigma^2)$.

Furthermore, we use the method of least squares to estimate the parameters β_0 and β_1 as follows:

$$Q = \sum_{i=1}^{n} \left[Y_i - \beta_0 - \beta_1 X_i \right]^2 .$$

We need to find estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ for β_0 and β_1 respectively, which minimize Q. So,

$$\frac{\partial Q}{\partial \beta_0} = 0 = -2 \sum_{i=1}^{n} \left[Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i \right]$$

and

$$\frac{\partial Q}{\partial \beta_1} = 0 = -2 \sum_{i=1}^{n} X_i \left[Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i \right] .$$

Those two equations are known as normal equations and they are given as follows,

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} X_i = \sum_{i=1}^{n} Y_i$$

$$\hat{\beta}_0 \sum_{i=1}^{n} X_i + \hat{\beta}_1 \sum_{i=1}^{n} X_i^2 = \sum_{i=1}^{n} X_i Y_i .$$

Since we are seeking for a minimum, we have to check the second derivative. But, it has not been shown. Also, all the ideas which have been presented in this section can be found in [13] or [7].

Then, the above normal equations can be written in a matrix form as follows:

$$X^{'}X\hat{\beta} = X^{'}Y,$$

where:

$$X^{'}X = \begin{bmatrix} n & \sum_{i=1}^{n} X_i \\ \sum_{i=1}^{n} X_i & \sum_{i=1}^{n} X_i^2 \end{bmatrix} \quad , \quad X^{'}Y = \begin{bmatrix} \sum_{i=1}^{n} Y_i \\ \sum_{i=1}^{n} X_i Y_i \end{bmatrix} \quad \text{and} \quad \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix}.$$

Finally,

$$\hat{\beta} = \left[X^{'}X\right]^{-1} X^{'}Y.$$

Now, the above model can be generalized to have $k-1$ independent variables. Thus, the generalized model can be written as follows:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{k-1} X_{i,k-1} + \epsilon_i \quad , \quad i = 1, 2, \ldots, n.$$

3.1.1 Under-determined System of Equations. In our case, for a given number of markers k and a given number of samples n with $k < n$. Then, we want to estimate the concentration of each sample using those markers without considering a specific probability distribution for the dependent variable. As a result, we have under-determined system of equations and can be solved using the pseudo-inverse of the matrix X in the $\hat{\beta}$ formula as follows [4]:

$$\hat{\beta} = X^{'}\left[XX^{'}\right]^{-1} Y,$$

where X is $k \times n$ matrix and it is our mixture setup.

3.2 Demultiplex algorithm

For a given mixture setup, we assume that each sample has a specific concentration in this setup. Then, we add noise to each marker which changes the real measurements. Consequently, we try to decipher the true concentration for each sample by estimating the coefficients of a linear regression model. This sequence of ideas is organized in the following algorithm:

1. Given a mixture setup in a matrix A;
2. Determine the number of rows (n_rows) and the number of columns (n_col) in A;
3. Assume the concentration of each sample is given by,
$$conc = [1, 2, \ldots, n\_rows]^T;$$
4. Determine the real measurement of each marker which is given by
$$real = A^T \times conc;$$
5. Add noise to the real measurements which follows the normal distribution with specific mean and standard deviation under the variable $(measured)$;
6. Estimate the coefficients of the linear regression model using A and the vector $measured$. The estimated values are given in the vector $recovered$.

Let us apply the above algorithm on the following example using R [15].

3.2.1 Example. Given six samples and four markers on the following mixture setup.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \text{and} \quad conc = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}^T.$$

Then, the real measurement of each marker is given by $real = \begin{bmatrix} 6 & 10 & 12 & 14 \end{bmatrix}^T$. Further, by adding noise which follows $\mathcal{N}(0, 0.01)$, we get $measured = \begin{bmatrix} 6.061 & 9.850 & 11.947 & 13.926 \end{bmatrix}^T$.

Finally, the recovered concentration of each sample is given by

$$recovered = \left(\left[A^T A \right]^{-1} \times A^T \right)^T \times measured = \begin{bmatrix} 0.991 & 2.040 & 3.029 & 3.934 & 4.924 & 5.972 \end{bmatrix}^T.$$

3.3 Mixture algorithm

In this section, we present the algorithm we used to build our simulated mixture setup with a specific number of samples and markers. There are many conditions we use in this algorithm and one of them is the condition number of the matrix. The estimated condition number can be defined as follows:

3.3.1 Definition. [8]

The condition number $k(A)$ for a given square matrix A is defined by

$$k(A) = \|A\| \cdot \left\|A^{-1}\right\|.$$

When $k(A)$ is large then we say that A is ill-conditioned and is well-conditioned when $k(A)$ is small.

In our algorithm we estimate the reciprocal condition number of the mixture matrix. The reciprocal value has a range between 0 for ill-conditioned matrix and 1 for well-conditioned matrix.

We need to add few lines in the Demultiplex algorithm which has been defined in the previous section. So, we state the modification of the previous one. Then, we present the mixture algorithm.

1. Define a function $mixt\_de$ with parameters A for the mixture setup and tol for the accepted tolerance.

7. If the maximum of the vector abs$(recovered - conc)$ is less than tol, *do*

    - Return True.

1. Define a function $mixtu\_sim$ with parameters n_row for the number of rows, n_col for the number of columns, tol and r_co_val for the tolerance and the reciprocal value we accept, respectively;
2. Define a variable t_times which is an integer and is given by, $t\_times = \frac{n\_row \times n\_col}{2}$;
3. Define the counter variable k with value 1;
4. Create an empty list li;
5. While the list li is empty, *do*
   - Define a vector with elements $(0, 1)$ and repeating them t_times.
   - Generate a matrix l by randomly permuting the elements of the above vector.
   - If the matrix l has unique rows, *do*
     - If the matrix $(l^T \times l)$ is not singular, *do*
       - If the reciprocal condition number of $(l^T \times l)$ is greater than r_co_val, *do*
         - If the returned value from the function $mixt\_de$ is true, by passing the matrix l and tol, *do*
         - Save the matrix l in the list li;
6. Return li.

3.4 Simulation results

Using the mixture algorithm, we are able to get different types of mixture setups by varying the number of rows and the number of columns in the algorithm. This algorithm has been written in R language and the code can be found in appendix A.1. All the results of those mixture setups have been obtained by using tolerance $(tol = 0.2)$, reciprocal condition $(r\_co\_val = 0.00001)$ and noise follows the normal distribution $\mathcal{N}(0, 0.01)$. Different types of difficulties have faced our simulation such as time and the speed of the processors. When writing the code, there were two options between making the code have quadratic running time and saving the generated matrices to prevent repetition by comparing the saved matrices with the new one. As a result, it will require a large memory. The other option, which has been used by making the running time linear and generating the matrix every time without comparing with the previous ones or even saving them.

For instance, using 4 markers and 8 samples, we are able to get several mixture setups and we selected three of them because we are interested in varying the levels of noise in each setup. Consequently, we want to determine which setup can be stable with different levels of noise.

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}^T, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T$$

$$A_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}^T .$$

Now, we want to vary the level of noise in the above matrices. By adding noise which follows the normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 0.2$, and running the $mixt\_de1$ code 1000 times, the matrix A_1 was able to achieve 70% correctly estimating the concentration of each sample from the total number of runs. By fixing the mean and varying σ to be 0.3, we get 37% correct estimation of the concentrations in all the samples. As the level of noise increases, the correct percent continues decreasing and achieving 20% correct estimation when $\sigma = 0.4$. On the other hand, by fixing σ to be 0.2 and varying the value of μ, we are able to get 70% correct estimation when $\mu = 0.1$. Also, with $\mu = 0.2$ the percent decreases to be 60% correct estimation from the total number of runs which was 1000 for all the results. Those results have been obtained by slightly modifying the $mixt\_de$ code and the modified code can be found in appendix A.1 under $mixt\_de1$.

For A_2 and A_3, the results have been presented in the following graphs:

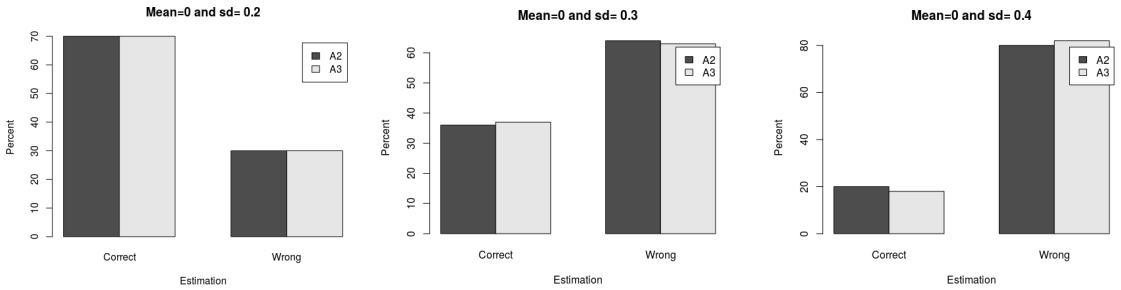

Figure 3.1: Matrices A_2 and A_3 for different values of σ

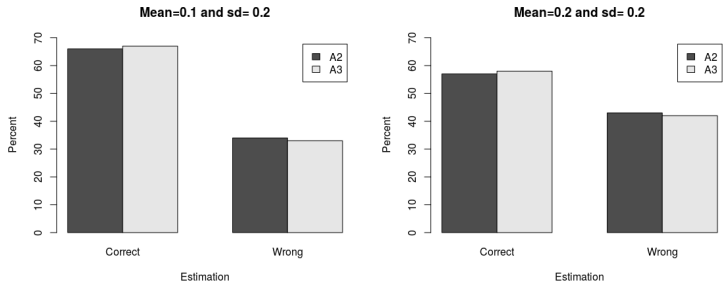

Figure 3.2: Matrices A_2 and A_3 for different values of μ

From the above graph we notice that there is no difference between the two mixture setups and they are almost giving the same result in all the cases.

For the remaining mixture setups, we present 2 matrices for each case and the remaining setups have been attached in appendix A.2. Also, the graphical results for different levels of noise are presented in appendix A.3.

In the case when 6 markers have been used, we are able to mix 10 samples in three different mixture setups and two of them are given as follows:

$$B_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}^T, \quad B_2 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}^T.$$

As the number of elements increases, difficulties start to arise and more running time is required to get such a mixture setup. The running time was between 2 hours up to 10 days to have a result. Just as before, using 6 markers we are able to analyse up to 11 samples and to decipher the true concentration of each sample. Those mixture setups are given below

$$C_1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}^T, \quad C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}^T.$$

In the case when we have 12 samples, we can mark them using 6 markers by using the following mixture setups and getting the correct results as required.

$$D_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T, \quad D_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T.$$

Finally, using 8 markers we can analyse up to 14 samples and getting the right concentration of each sample. One of those mixture setups is given below

$$E_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}^T.$$

By varying the levels of noise, an interesting results have been obtained using the error-correcting code matrix which has been constructed in Example 2.4.1 compared to the other results. Also, we are going

to denote such a matrix by E_4 through the remaining work.

$$E_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}^T.$$

Those results are given in the following graphs:

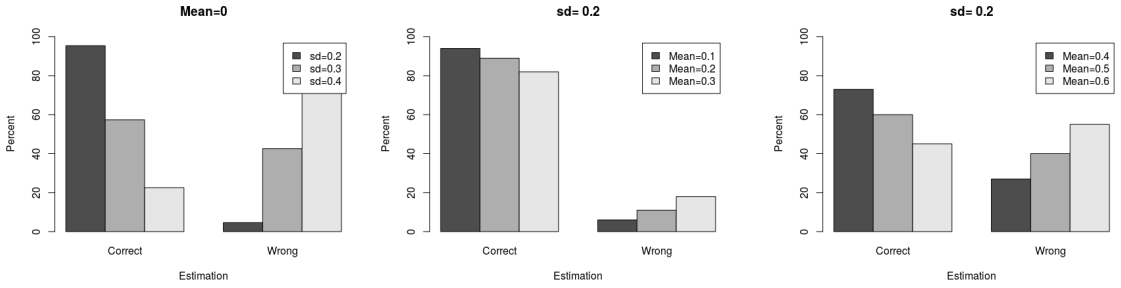

Figure 3.3: Matrix E_4 with different values of μ and σ

From the above figure, we notice there is slight effect from the different values of the mean when $\sigma = 0.2$. However, there is a considerable effect from σ on the mixture setup.

Now, we want to add noise from another probability distribution which is uniform continuous distribution with parameters a and b. We want to check our mixture setups for different values of b and by fixing a to be equal to zero. From the above results, we decided to consider one mixture setup from each group. The results of this experiment are given in the following graphs:

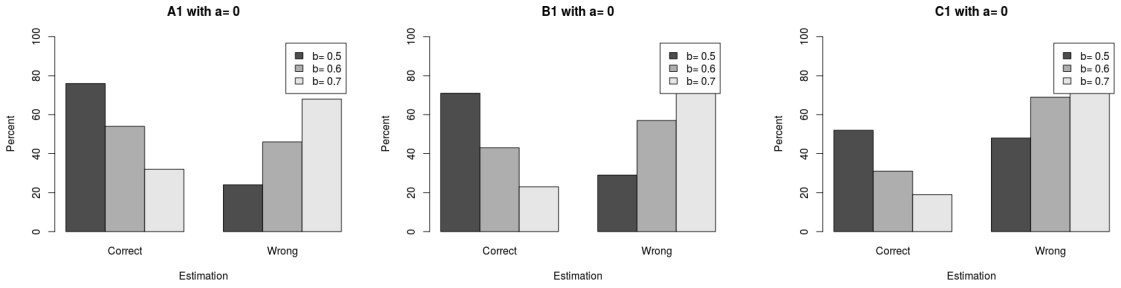

Figure 3.4: Matrices A_1, B_1 and C_1 for different values of b

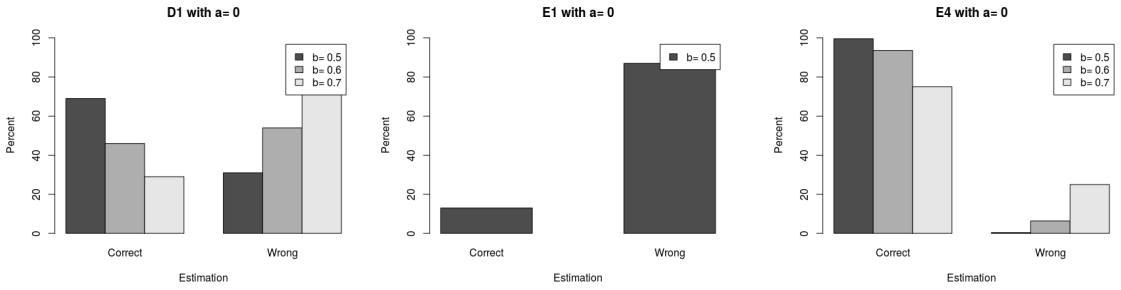

Figure 3.5: Matrices D_1, E_1 and E_4 for different values of b

From the above results, we notice that E_4 is stable with different levels of noise compared to the other mixture setups.

Also, we want to add noise which follows the gamma distribution with shape parameter α and scale parameter β. In this experiment, we fix $\alpha = 2$ and try to vary the values of β. Then, we determine the effect of those variations on our mixture setups. The results are given in the following graphs:

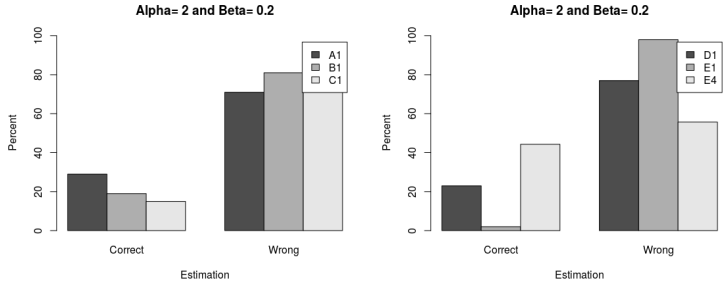

Figure 3.6: Matrices A_1, B_1, C_1, D_1, E_1, and E_4 when $\alpha = 2$ and $\beta = 0.2$

From the above results, we notice that all the mixture setups were not able to estimate the correct concentration, even the setup E_4 was able to estimate 44% correct estimation from the total number of runs. Then, it is followed by A_1 which achieved 29% correct estimation from the total runs. We conclude from that, by adding noise from gamma distribution, that we cannot trust one of the above setups.

Further, by considering the special case of gamma distribution when the shape parameter $\alpha = 1$ which gives the exponential distribution with parameter β. In this trial, we want to test the different values of β on our mixture setups. The results as before are given below

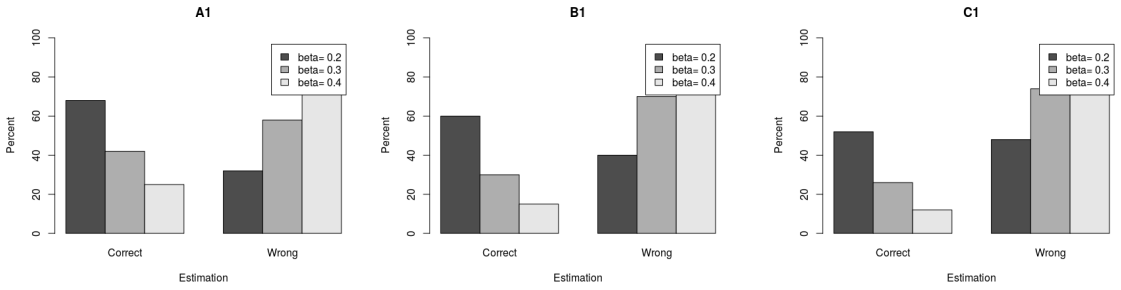

Figure 3.7: Matrices A_1, B_1 and C_1 for different values of β

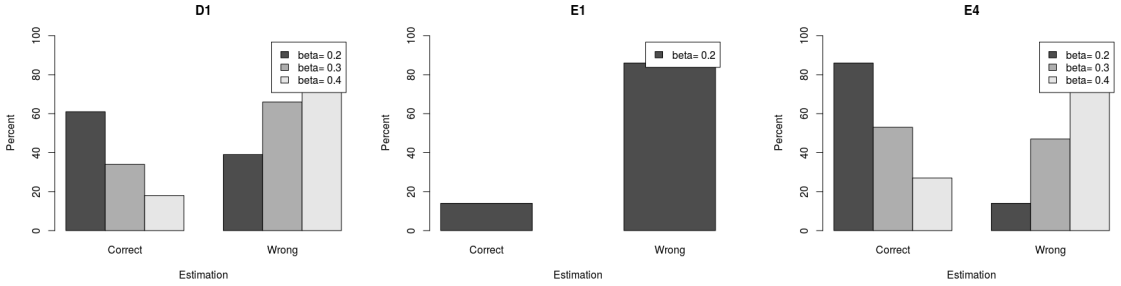

Figure 3.8: Matrices D_1, E_1 and E_4 for different values of β

From the above graphs, we notice that when $\beta = 0.2$, E_4 is able to estimate correctly up to 86% from the total number of runs. It is followed by A_1 which is able to estimate 68% correct concentration from the total number of trials. As the value of β increases the percent of the correct estimation decreases and it reaches up to 12% in C_1 when $\beta = 0.4$.

Finally, we add noise from the beta distribution with shape parameters α and β. We vary the values of both parameters. The results are given in appendix A.3.

From the results, we notice that when $\alpha = 1$, the lowest percent has been achieved in E_1 with $\beta = 3$ which was 7%. As β increases, the percent increases and it reaches up to 84% in A_1 when $\beta = 5$. Moreover, by fixing $\beta = 1$ and varying the value of α, we realise that the percent of correct estimation increases as α increases.

3.5 Summary

In this chapter, we were able to mix up to 14 samples using 8 markers and analysing up to 12 samples using 6 markers. We were not able to get an $OA(15, 6, 2, t)$ and $OA(28, 8, 2, t)$ because of the time factor and the random construction of the setups. We added different levels of noise and considerable results have been obtained and analysed. The matrix E_4 was always stable with different levels of noise except when the noise has been constructed from gamma distribution. We did not consider any discrete distribution because the level of the noise would be very high and all the mixture setups would fail to check the correct estimation.

4. Conclusion

In this research project, we studied the problem of missing calibration in high throughput experiments. We introduced the concept of the orthogonal array and its connection with the experimental design field. Then, we were able to use the Bush's method to mix 6 samples using 4 markers and to recover the correct concentration of each sample. Also, we introduced several concepts related to the error-correcting codes. Then, we used the Rao-Hamming method to construct an optimal mixture setup which was able to analyse 14 samples using 8 markers.

Since we were seeking for an optimal setup which is able to analyse a higher number of samples in one trial, we simulated our samples concentrations, then we tried to decipher the true concentration of each sample using simulated mixture setup. As a result, we were able to get several mixture setups. Due to the time factor, we were not able to mix more than 12 samples using 6 markers. Also, we were not able to exceed analysing 14 samples using 8 markers.

Finally, we tried to determine the effect of different levels of noise from different probability distributions on our mixture setups which we got. As a result, the matrix which has been constructed using the Rao-Hamming code was stable with different levels of noise. Also, the other mixture setups were able to estimate the correct concentration by increasing the range of our acceptable tolerance as we did in appendix A.4.

Future Work

We were not able to get an $OA(28, 8, 2, t)$ and $OA(15, 6, 2, t)$ as we guaranteed in chapter 2 because of the time factor. According to the results we got from the mixture setup which has been constructed using the error-correcting code idea, we claim there is a strong relation between the two fields and interesting results may be obtained. Also, we wish to apply this mixture setup with real life samples.

Appendix A. Appendices

A.1 The code of the simulation

The codes have been omitted and can be found on the CD with the project soft copy.

A.2 Mixture setups

$$B_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}^T, \quad C_3 = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T.$$

$$D_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T.$$

$$E_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}^T,$$

$$E_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T.$$

A.3 Figures

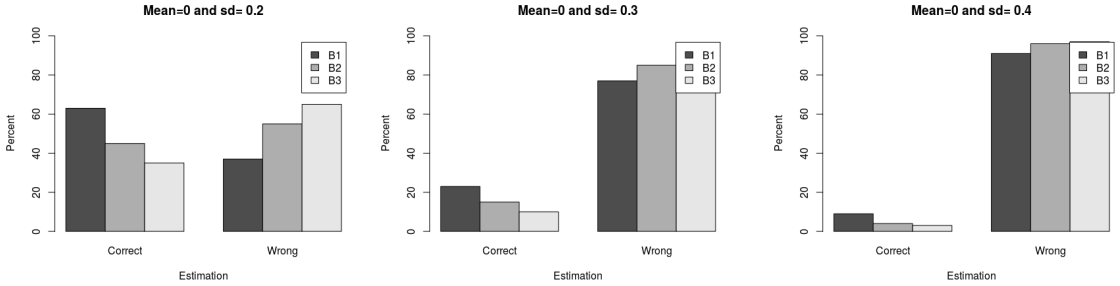

Figure A.1: Matrices B_1, B_2 and B_3 for different values of σ

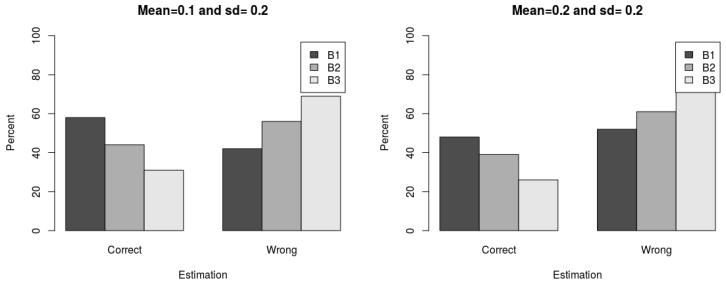

Figure A.2: Matrices B_1, B_2 and B_3 for different values of μ

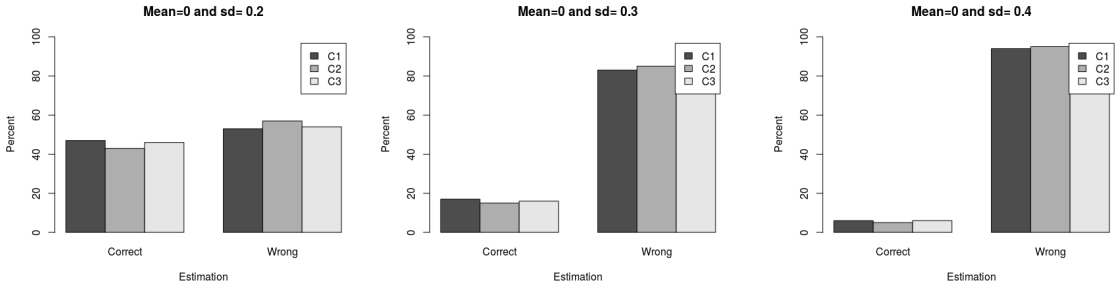

Figure A.3: Matrices C_1, C_2 and C_3 for different values of σ

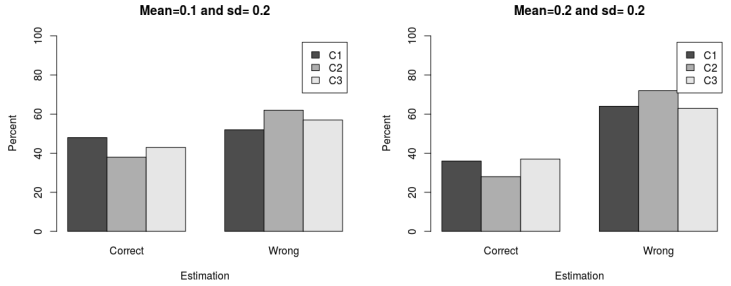

Figure A.4: Matrices C_1, C_2 and C_3 for different values of μ

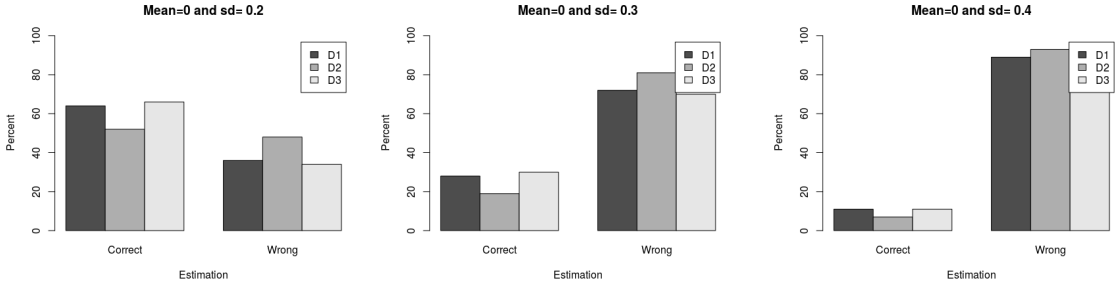

Figure A.5: Matrices D_1, D_2 and D_3 for different values of σ

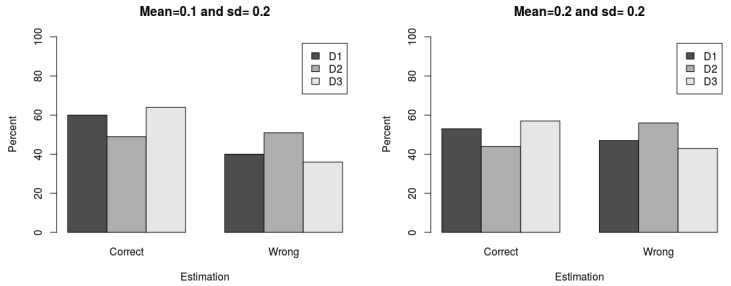

Figure A.6: Matrices D_1, D_2 and D_3 for different values of μ

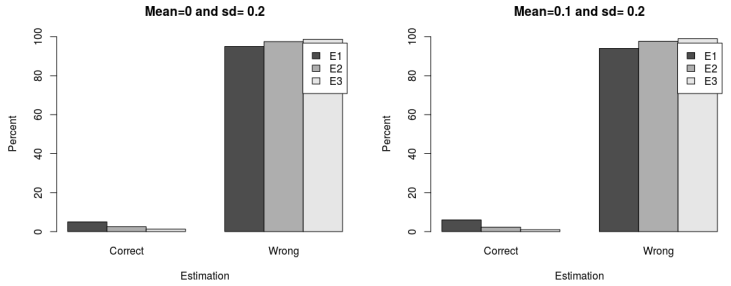

Figure A.7: Matrices E_1, E_2 and E_3 for different values of μ and σ

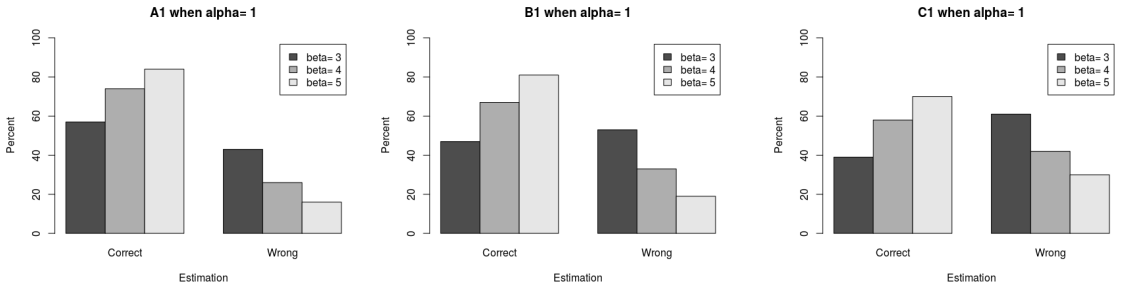

Figure A.8: Matrices A_1, B_1 and C_1 when $\alpha = 1$ for different values of β

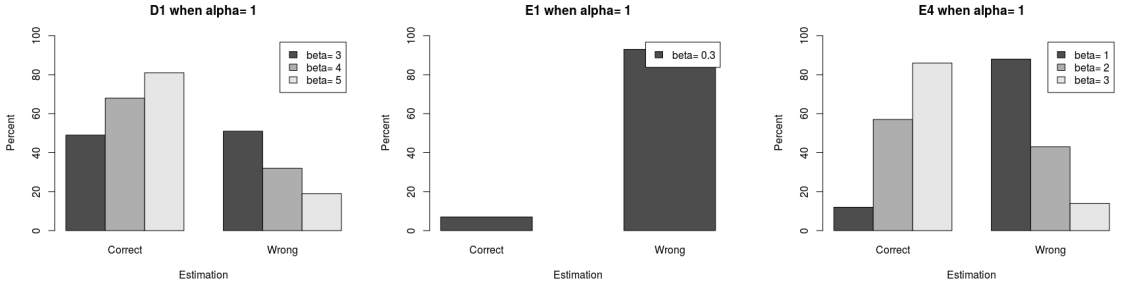

Figure A.9: Matrices D_1, E_1 and E_4 when $\alpha = 1$ for different values of β

A.4 Extension

We want to construct three simple setups with markers in each sample which are given as follows:

$$A_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } E_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then, we want to vary the value of the acceptable tolerance and to compare with the other mixture setups which have been constructed during the simulation part. The results are given below

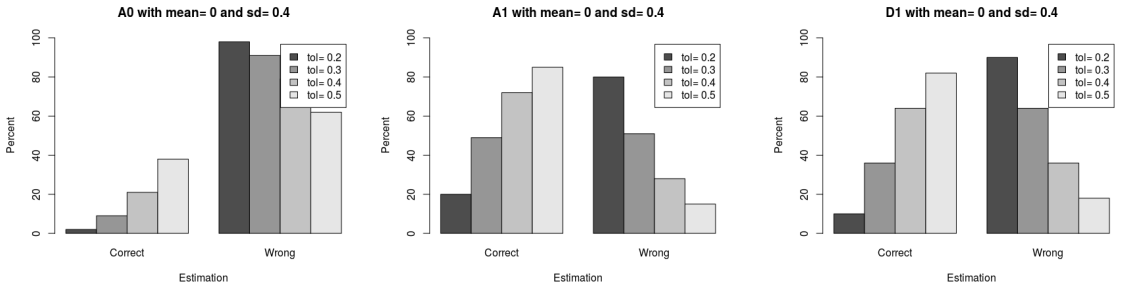

Figure A.10: Matrices A_0, A_1 and D_1 when $\mu = 0$ and $\sigma = 0.4$

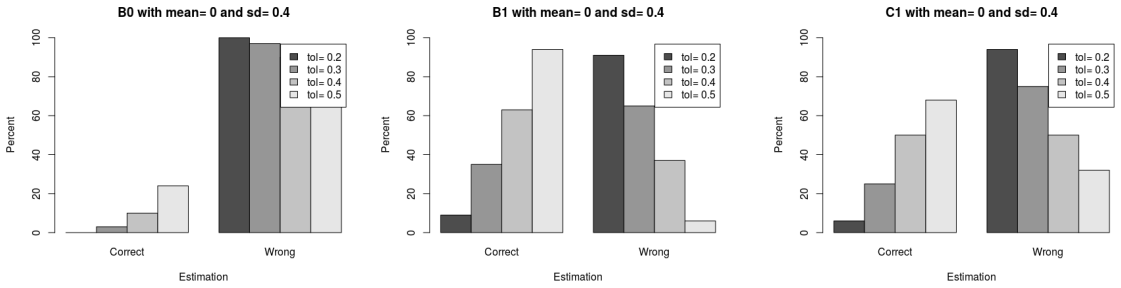

Figure A.11: Matrices B_0, B_1 and C_1 when $\mu = 0$ and $\sigma = 0.4$

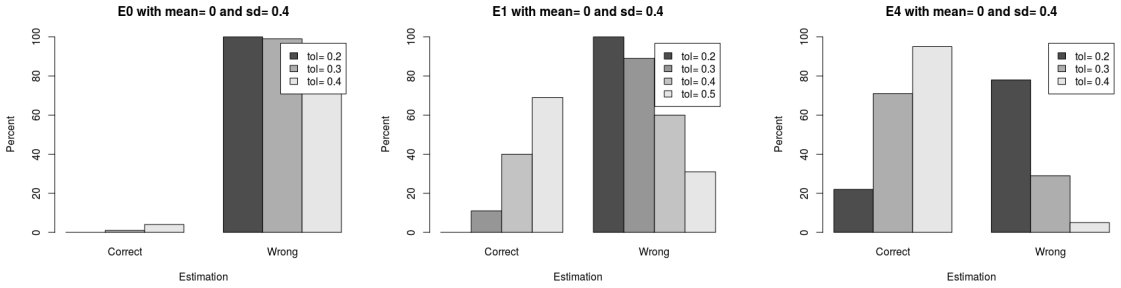

Figure A.12: Matrices E_0, E_1 and E_4 when $\mu = 0$ and $\sigma = 0.4$

Acknowledgements

It is my greatest pleasure to be able to finish this paper though the journey was not that smooth. The path was full of obstacles but above all, it is a reality. It could have been just but a dream without the collective efforts of different people whose care I really appreciate. I would like to give my sincere gratitude to my own supervisor, Professor. Bernhard Y. Renard and his well dedicated team, Dr. Muth, Martina and Kristina who stood with me through thick and thin. They supported me with academic insights and clarification and the motivation to move on.

To the AIMS family, hats off for your unwavering support and the chance granted to me to be part of the team to be equipped with such a world class education, all for the good of my country and my beloved continent at large. I also extends my gratitude to my tutor, Sedric for working with me tirelessly.

My greatest gratitude goes to the AIMS- Cameroon president Prof. Mama Foupouagnigni and the academic director Prof. Marco Garuti for their assistance and the encouragements to move on. The assistance from my fellows AIMS colleagues Ogola, Richard and Calvine was also awesome and heartily appreciated for they were the starting point when help was needed. The assistance and experience from each and every student at AIMS was really awesome. I also salute my language lectures Mr. Chuansi Doko and Mr. Talla Naoussi Arnold for guiding me through good language writing and grammar correction to make this piece attain this quality.

I thank my one and only beloved wife for her support, encouragements and inspirations during my stay at AIMS though we were that far. I also appreciate her support through prayers for God's guidance both in times of good and bad.

I thank my family from the depth of my heart for such a caring life they have for me, how they sacrificed for me to be able to reach this far. My parents are a blessing to me and through their encouragements and motivation throughout the academic year at AIMS, I have managed to reach the end. Their love cannot be underestimated, I really appreciate their care.

Above all, I thank the Lord God Almighty for it is only through His grace that I am who I am today. His mercies and grace always push me forward in life against all odds of drawbacks.

References

[1] Aggarwal, S. and Yadav, A. K. (2016). Dissecting the itraq data analysis. *Statistical Analysis in Proteomics*, pages 277–291.

[2] Bose, R. C. and Bush, K. A. (1952). Orthogonal arrays of strength two and three. *The Annals of Mathematical Statistics*, pages 508–524.

[3] Bro, R. and De Jong, S. (1997). A fast non-negativity-constrained least squares algorithm. *Journal of chemometrics*, 11(5):393–401.

[4] Bruckstein, A. M., Donoho, D. L., and Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81.

[5] Bush, K. A. et al. (1952). Orthogonal arrays of index unity. *The Annals of Mathematical Statistics*, 23(3):426–434.

[6] Cox, D. R. and Reid, N. (2000). *The theory of the design of experiments*. CRC Press.

[7] Draper, N. R., Smith, H., and Pownell, E. (1966). *Applied regression analysis*, volume 3. Wiley New York.

[8] Golub, G. H. and Van Loan, C. F. (2012). *Matrix computations*, volume 3. JHU Press.

[9] Gulati, B. R. (1971). Orthogonal arrays of strength five. *Trabajos de estadistica y de investigacion operativa*, 22(3):51–77.

[10] Hedayat, A. S., Sloane, N. J. A., and Stufken, J. (2012). *Orthogonal arrays: theory and applications*. Springer Science & Business Media.

[11] Kounias, S. and Petros, C. (1975). Orthogonal arrays of strength three and four with index unity. *Sankhyā: The Indian Journal of Statistics, Series B*, pages 228–240.

[12] MacWilliams, F. J. and Sloane, N. J. A. (1977). *The theory of error-correcting codes*. Elsevier.

[13] Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. (1996). *Applied linear statistical models*, volume 4. Irwin Chicago.

[14] Noda, R. (1979). On orthogonal arrays of strength 4 achieving rao's bound. *Journal of the London Mathematical Society*, 2(3):385–390.

[15] R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[16] Rauniyar, N. and Yates III, J. R. (2014). Isobaric labeling-based relative quantification in shotgun proteomics. *Journal of proteome research*, 13(12):5293–5309.

[17] Seiden, E. et al. (1954). On the problem of construction of orthogonal arrays. *The Annals of Mathematical Statistics*, 25(1):151–156.

[18] Seiden, E. et al. (1955). On the maximum number of constraints of an orthogonal array. *The Annals of Mathematical Statistics*, 26(1):132–135.