Development of computational methods for Custom protein arrays analysis. A case study on a 100-protein ("CT100") cancer/testis antigen array.

by

Jean-Michel SAFARI SERUFURI

Thesis presented for the degree of Master of Science
(Bioinformatics)
In the Department of Molecular and Cell Biology
University of Cape Town (UCT)

Supervisors :

Prof. Jonathan Blackburn
Assoc. Prof. Nicola Mulder
Dr. Judit Kumuthini

March 2010

Abstract

Custom antigen arrays offer a platform to assay the serological response of cancer patients to a set of selected cancer testis antigens in order to infer a diagnosis value or to assess the patient responses to particular treatments. However, the acquisition of the array data is subject to bias and noise. Therefore, array data processing and analysis is required to clear the data from bias, reduce the noise and to learn from the data.

This study aims to address the issues of normalization and sample qualitative clustering for custom protein arrays. For the issue of normalization, a method aiming to deal with the relatively small number of housekeeping spots, as well as to be robust against the eventuality of the flagging of some of them has been developed. This method was shown to be valuable for within - and between array normalization both through a simulation and through a significant improvement of the dataset from a reproducibility experiment, in addition to being easily implementable as a computer program. Furthermore a pipeline using these methods was implemented in python to carry out the spot filtering and normalization of the array data for within - and between array normalization. Qualitative clustering has been achieved using the factor analysis method in order to get quantitative information about the strength which links a particular profile to each cluster. The factor analysis was shown to be straight forward compared to the *K-Mean* using a Pearson correlation metric and also to be more easily implementable.

Acknowledgements

To Professor Jonathan Blackburn, Professor Nicola Mulder and Judit Kumuthini for their supervision, support and patience during this study;

To the staff and post-graduate students at Computational Biology group (CBIO) for fruitful discussions, and especially to Elizabeth Kelly and Cashifa Karriem for their availability and assistance;

To the Centre for Genetic and Proteomic Research (CPGR) for the access to their facilities, and especially to Dr Aubrey Shoko, Project Scientist at CPGR, for his assistance;

To my family and friends for their unconditional support;

I wish to express my gratitude.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the deciphering of the human genome, the amount of information concerning the genes and their involvement in biological processes has significantly increased. The science of genomics has enabled a revolution in both research and drug discovery. However, the cellular, subcellular and supracellular functions are indirectly governed by genes through their products which are proteins. Indeed, the genes code for proteins but that process is followed by random modifications called post-translational modifications which occur after the translation from gene to protein and are responsible for functional changes in proteins. In this context, Proteomics arises as a complementary field to genomics and aims to study the biological processes at the protein level [1, 2].

Proteomics uses different technologies to separate complex mixtures of proteins into their individual components and Proteomics analysis can be classified into three main categories: expression proteomics, bioinformatics analysis and functional proteomics; they all play different roles but feed into each other. The expression proteomics is the initial step used to identify candidate protein for further functional studies. After the identification of the candidate proteins, the bioinformatics analysis is supposed to make functional analysis more focused by enabling a selection among the candidate proteins. Bioinformatics analysis provides additional information such as protein structures (primary, secondary and tertiary), protein sequence alignments, annotations, etc. The functional proteomics aims to understand the role of target proteins in the cellular functions using approaches based on protein binding activities [1]. To this end, protein microarrays have been developed in analogy to DNA microarrays to assess the expression of large sets of proteins or to study

their functions.

1.1 DNA versus Protein microarray technology

DNA microarray technology has proven to be a valuable tool for investigations on biological processes, especially for the identification of differentially expressed genes in response to various stimuli or several disease states. However, the actual biological relevance of the identified genes may require further investigation. Indeed, the identified genes may only be indirectly related to the causative protein or proteins, or not at all. As a result, DNA microarray experiments suffer the handicap of their limitations to provide direct information on the way gene products interact between them in the regulation of cell life. Proteomics aims to provide biological information with a more direct level of comprehensiveness, mainly on protein expression and function [3].

Protein microarray technology immobilizes proteins onto a glass slides as capture probes for the detection of their biochemical activities with some other molecules contained in sample solutions. This technology enables several types of biological questions to be asked and answered. It also enables significant increases in the amount of proteomic information by permitting high-throughput analysis of proteins [3].

The high-throughput manipulation of proteins is more challenging than genes. Indeed, where a simple Polymerase Chain Reaction (PCR) is required for the amplification of DNA fragments; a more complex and often relatively unstable process is required for protein acquisition, and the biochemical activities being assessed are simpler in DNA microarrays where the single stranded DNA (ssDNA) displays equivalent biochemical properties which results in high affinity binding and specific binding partners; whereas proteins exhibit diverse biochemical features and therefore do not always have specific binding partners. The issues of protein acquisitions; their immobilization on the surface of the slide and the detection of their respective binding partners are outlined further below [3, 4].

Protein acquisition

The methods used in protein acquisition rely on the identification of the Open Reading Frames (ORFs) and the selection of the more appropriate plasmids for protein expression. An ORF is a protein-coding segment within the DNA sequence, and encodes all the amino-acids between initiation and termination codons. This ORF is amplified by PCR before being inserted into a plasmid. However, depending of the selected plasmid, variations in the initiation codons or even splicing signals may occur and result, for the same ORF, in different messenger RNA (mRNA) after transcription. To prevent from such biochemical alterations, the protein should be expressed in homologous systems or organisms where they will be subject to native modifications, and therefore preserve their potential to interact with their natural partners [5–7].

Heterologous expression is generally used even though it may lead to problems of expression. In upwards of 60% of the cases, soluble proteins expressed in *Escherichia coli* or *E.coli* conserve their solubility. An alternative to *E.coli* is the use of insects cells which results in similar modifications to mammalian cells [6].

Protein immobilization on the surface of the slide

The techniques of immobilization are important both for effective concentration and orientation of immobilized proteins on the surface, and also to preserve their folded conformations. There are two categories of protein immobilization methods, covalent and noncovalent. The biotin-streptavidin is one of the noncovalent immobilization method based on the high affinity of biotin and streptavidin interaction. This method links biotinylated macromolecules to a surface that was chemically coated with streptavidin (Fig. 1.1). By contrast, the covalent immobilization methods are based on a covalent coupling to a crosslinker attached to the surface [8, 9].

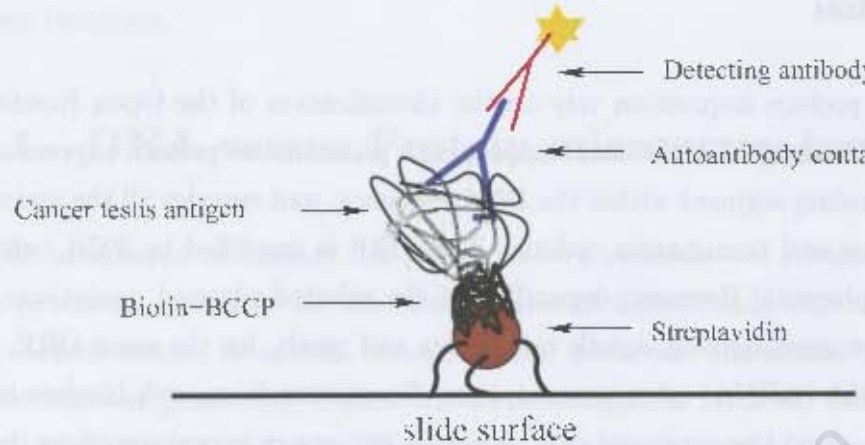


Figure 1.1: Biotin-streptavidin immobilization method

Detection of the respective binding partners

The main interest in microarray experiments is based on the possibility to quantify the interactions between the probes immobilized on the slide and the target contained in the sample solution. To this end, some molecules with specific interaction properties are labelled with either fluorescent, photochemical or radioisotope tags. The choice of one detection method against the others is the result of a trade off between the desire to reach a high signal-to-noise ratio and an affordable cost.

Fluorescent labelling is the most widely used detection method for microarray experiments, it is sensitive, stable, safe and effective; and can be archived for future imaging. It requires a laser scanner to be read. However, the labelling of molecules might affect their ability to interact with their partners. Chemiluminescence is another highly sensitive label-based detection method, however, it has the inconvenience of having low resolution due to its limited dynamic range. Radioactivity based methods are much less frequently used because of their requirement for long exposure which results in safety concerns. Both Chemiluminescence and radioactivity methods can only be performed once. Another unlabelled method for protein array detection is the Surface Plasmon Resonance (SPR). In this method probes are immobilized on a gold-coated surface and the binding activities are measured by the changes in the light reflection angles. The SPR is not very sensitive and requires more protein to be immobilized on the surface than fluorescence, chemiluminescence or radioactivity detection methods [1, 5, 10].

1.2 Microarray experiments

Even though microarray technologies have been around for many years, they are still subject to bias and variations. In addition to the variations introduced by probe acquisition, immobilization and the detection method; there is still a large number of environmental factors which might introduce variabilities in the experiments. Among them, there are the ambient conditions when the arrays were processed, the person conducting the experiments, the recombinant sample differences, the variations in sample collection, the non-uniformity in the hybridization, the distribution of artifacts or smears onto surface of the array or simply changes in the scanner settings in the case of fluorescent microarrays. Nevertheless, a good design of experiments can reduce the noise and be beneficial for the downstream data analysis [11–15].

In addition to designing lab protocols to reduce the noise within the microarray data, the experimental design should provide the means to assess the quality of the microarray data, as well as the means to correct or normalize them. To this end, some controls should also be immobilized onto the arrays. The most important type of controls are *housekeeping* and the *exogenous spiked-in* controls. The first are those which are assumed either singly or collectively not to change their signals within the different conditions of the experiments, while the second are those from species other than the one under study, generally selected to not hybridize on the arrays. Before selecting any controls, it is important to make sure that a stable source of the control exists [16].

1.3 Custom Antigen Arrays as Serological diagnosis tools

Different types of protein microarrays are currently used to study the biochemical activities of proteins. Among them, the analytical microarrays are typically used to profile complex mixtures of proteins and to estimate their level of expression, as well as their binding affinities and specificities. These types of experiments have interesting applications in molecular medicine where they provide a means to perform biomarker discovery for diagnosis, drug design and development as well as to get further understanding of pathogenesis and disease

biology [5].

For the diseases such as cancers which have a significant autoimmune response, antigen arrays are promising tools. By profiling the human serum which is the primary clinical sample used for disease diagnosis, antigen arrays have the potential to both reduce the intrusiveness of the cancer diagnosis techniques and improve personalized therapies in the clinical management of patients. However, despite the growing number of successful applications of antigen microarrays, their use is limited to specific investigations definable with a relatively small set of antibodies. Therefore, for reasons of disease specificities, custom or boutique antigen arrays were developed to focus on antigens of maximal interest and to contain very few irrelevant probes. The customization process requires one to define the content of the array well, according to the range of the desired specificity [11,13,17–19].

The study presented in this thesis is based on the analysis of a 100-protein Cancer-Testis antigen array ("CT100 array") for the *in vitro* diagnosis of cancer developed by the Blackburn group at the University of Cape Town in collaboration with the Centre for Proteomic and Genomic Research (CPGR).

In contrast to systemic autoimmune diseases where the presence of a particular autoantibody might have a diagnostic value, tumour associated antibodies, when detected individually, have little diagnostic value for three reasons. Firstly, the frequency with which anyone antibody specific for a particular cancer antigen is found within a cohort of patients often relatively low. Secondly, certain tumor-associated antigens are responsible for tumorigenesis in multiple cancer types, so the detection of the associated antibody can only indicate the presence of developing tumour without enabling discrimination between different cancer types. Thirdly, certain cancer-testis associated antibodies lack specificity because they might arise from events associated with Cancer or other diseases. Therefore, the characterization of antibody profiles against panels of cancer-testis antigens is potentially more informative than detection of antibodies against individual specific antigens [18, 20].

1.4 CT100: a Cancer-Testis antigen array for the *in vitro* diagnosis of cancer

1.4.1 Description

The CT100 is a one colour Cancer-Testis (or CT) antigen array aiming to discover the interaction between 100 CT antigens printed as probes on an array and the autoantibodies contained in the patient serum in order to diagnose cancer, to determine the efficiency of anti-cancer treatments, or to monitor the rate of cancer progression in a patient.

The 100 CT antigens are a collection of functionally unrelated proteins expressed in a wide variety of human tumours and are recombinantly produced using either an insect cell vector or an *E.coli* vector, as fusions to a C-terminal biotinylation motif (BCCP). The CT antigens are immobilized onto the array using a biotin-streptavidin attachment and the detection method of the antigen-antibody interactions uses fluorescence (see Fig. 1.1).

Each array is made of 8 blocks and each one of them contains 49 probes (see Fig. 1.2). Among the probes are printed positive (or housekeeping) controls at two different concentrations (hIgG 10ng/ul and hIgG 50ng/ul), negative controls (insect cell and *E.coli* empty vectors), orientation signals (Cy5 BSA), exogenous controls (SIgG 200ng/ul) and the 100 CT antigens. On each block the probes are printed by different pins in triplicate except for the ICL empty vectors which are printed as a unique probe on each block. The probes are immobilized on a streptavidin coated surface on which biotinylated, BCCP-tagged CT antigens are deposited at specially defined locations.

1.4.2 Image processing

After the hybridization of the probes with a serum sample that potentially contains targeted autoantibodies, the array is first washed with a solution of detecting antibodies to enable the detection of the binding partners (see Fig. 1.1). The second washing has the purpose of removing the excess of detecting antibodies, dust, artifacts, etc. from the surface of the array, but also to reduce non specific hybridization. In spite of washing, the spot quantification still remains affected by intangible factors. As described in the Figure

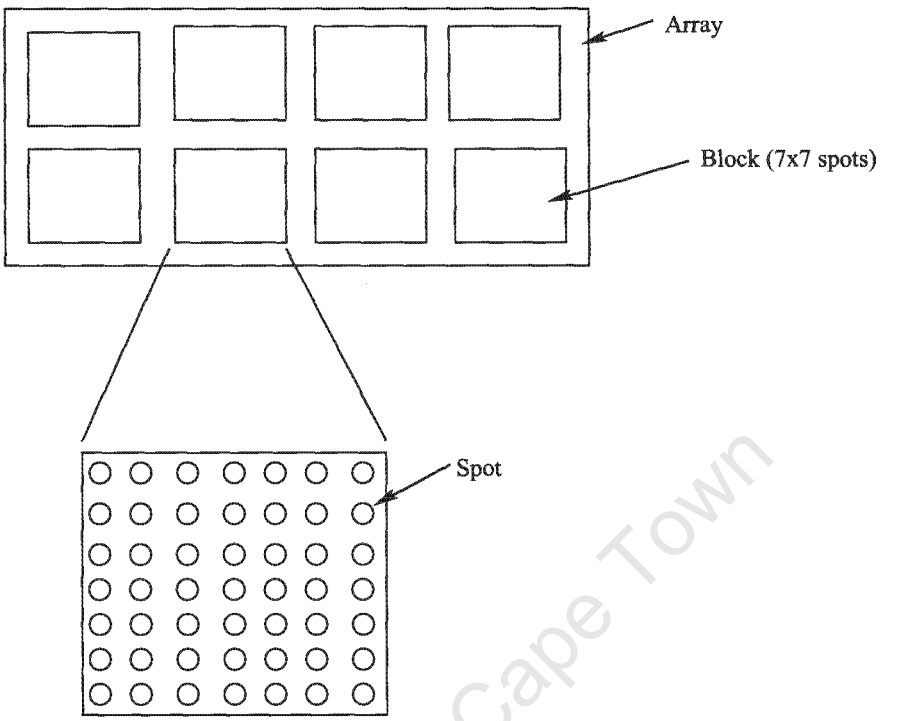


Figure 1.2: CT100 array layout

1.3, the same lab protocols can lead to different results in terms of noise in the background of the arrays. The measures of background intensity are supposed to represent the autofluorescence of the array surface for the different spot locations [16].

The aim of the image processing is to estimate the amount of each specific anti-CT antigen autoantibody present in the serum by measuring the spot pixel intensities. The image analysis software associated with a scanner allows us to retrieve some statistics for the pixels measured in both the spots and their local backgrounds, for instance, the mean and median of the pixel distributions of the spots and their adjacent backgrounds. Among the scanner settings is one called *gain setting* which helps in the discrimination between a weak signal and the background. By increasing the gain setting, the sensitivity of the signal is improved but the selection of the optimal gain setting must be a trade off betwen the need to detect as many spots as possible and to avoid saturation of any of the spots.

The scanner software proceeds by matching a grid layout defined by the user to the actual image coming from the array and locating the signal spots in order to quantify them. The spot finding can be achieved manually, automatically or semi-automatically. In the

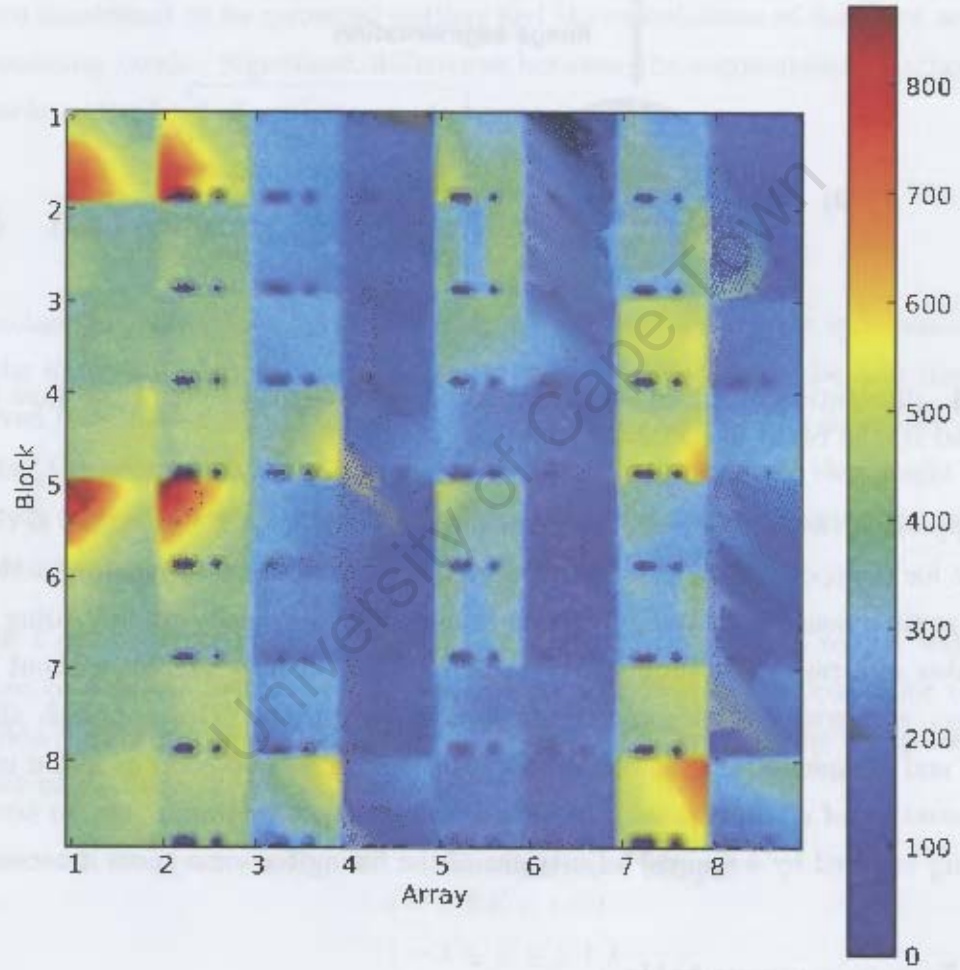


Figure 1.3: Illustration of the variabilities of the background intensities of 8 arrays with the application of the same experiment protocol. Here the 8 blocks of the arrays are represented in the columns.

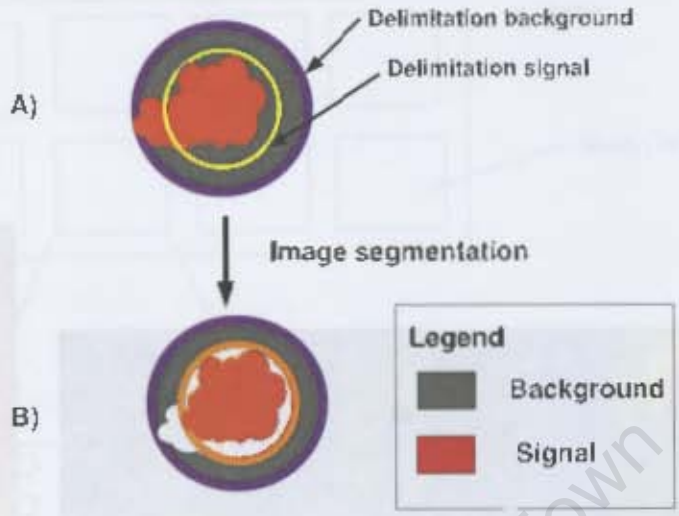


Figure 1.4: Illustration of image segmentation. A) shows the result of the spot finding process and B) the result of segmentation of the image.

manual approach, the user adjusts the grid over the array and fits the extent of the spot to account for the spot size variations and uneven spacing between spots. In practice, this approach is time consuming and subject to human error especially when dealing with a large number of arrays. The automatic approach aims to identify and fit, without human intervention, the extent of each spot using specific algorithms. This approach allows to save time and compensate for human errors. However, noise and contamination can lead to false detection of certain spots. The semi-automatic spot finding is just an automatic spot finding followed by a manual adjustment of the fitting for some spots if necessary.

1.4.3 Image segmentation

Image segmentation is the part of the image processing which deals with the decision on which pixel in the extent of the spot area belong to the spot signal, to the background or to different sources such as dust or artifacts. As mentioned above, the scanner reports some statistics based on pixel distributions. The segmentation of the image (see Fig. 1.4) defines rules to filter out bad pixels.

Among those rules there are the pixel filtering and the trimmed pixel methods. The pixel

filtering consists of setting up a threshold to filter out low intensity pixels and performs the statistics calculations with the remaining pixels. The trimmed pixel method assumes that most of the pixels in the extent of the spot area belong to the actual signal of the spot, and most of those in the adjacent area belong to the background. Therefore, for each spot or background intensities the pixels falling out of defined quantiles are trimmed because they are considered to be potential outliers and the calculations of statistics are based on the remaining pixels. Significant differences between the segmentation methods become noticeable as the level of artifacts on the arrays increases.

1.4.4 Background correction and subtraction

The background intensity is estimated from the adjacent area of the spots and subtracted from the foreground intensity of the spot to get the true value of the spot intensity that is derived from the specific interaction of antigens-antibodies. The issue here is to avoid including the artifacts in the estimation of the background because they might arbitrarily induce an overestimated background intensity and, as a result, diminish the true value of the spot intensity.

Module 1 of the Protein Chip Analysis Tools (ProCAT) [19] is a robust way to tackle the issue of artifacts in the background signals. The ProCAT approach for background correction basically replaces the background of a specific spot by the background median intensity of a surrounding 3 x 3 spot window.

$$\hat{B}_{i,j} = \underset{\substack{i-1 \le i' \le i+1 \\ j-1 \le j' \le j+1}}{\text{median}} \{B_{i',j'}\}$$

where i, j, i' and j' design the row and column coordinates of spots.

The background correction smooths the local background (see Fig. 1.5) by reducing the effect of artifacts and noise in the background. This enables calculation of the true value of the spot intensities with more accuracy by subtracting the value of the foreground by the value of the corresponding corrected background.

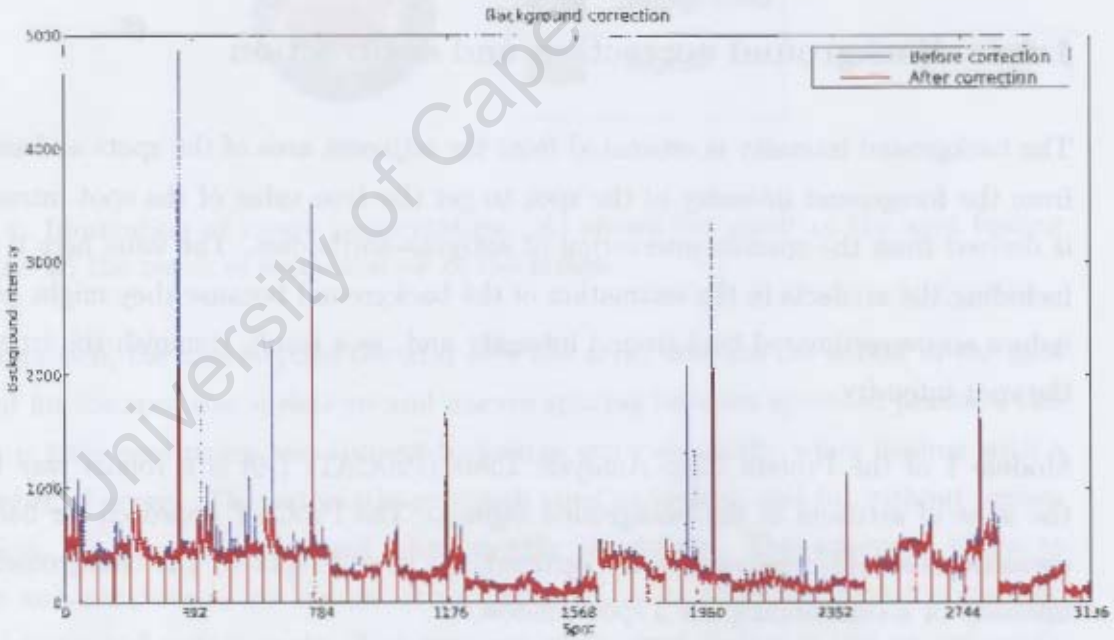


Figure 1.5: The background correction corrects the arbitrary peak of the background intensity in a three by three spot window. The trend lines in the graph come from eight custom arrays of 392 spots each.

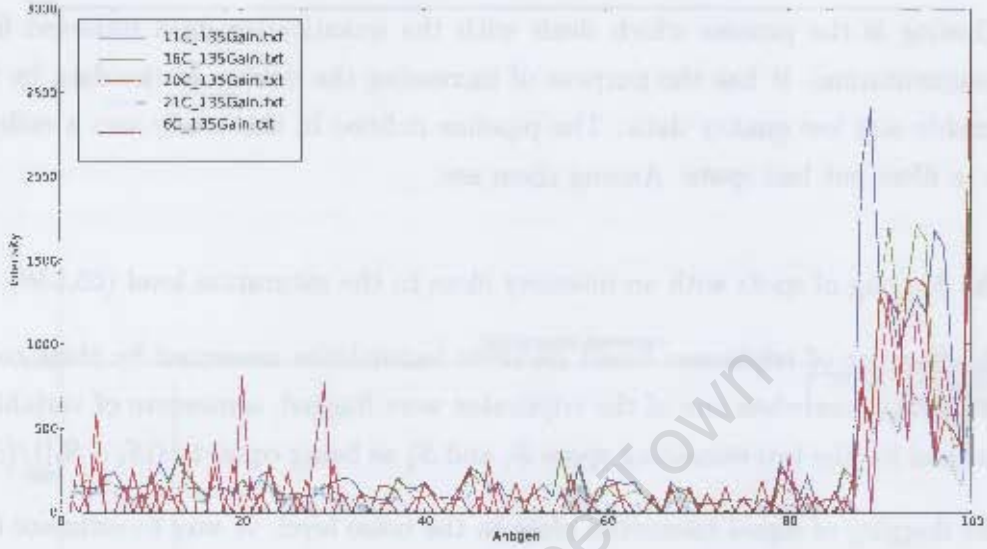
1.5 Data filtering

Data filtering is the process which deals with the quantitative data retrieved from the image segmentation. It has the purpose of increasing the quality of the data by flagging questionable and low quality data. The pipeline defined in this study sets a collection of criteria to filter out bad spots. Among them are:

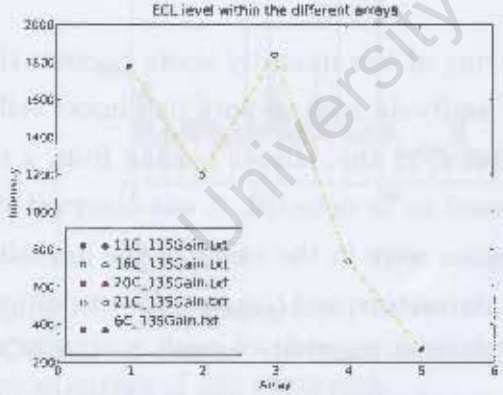
- the flagging of spots with an intensity close to the saturation level (65,536);
- the flagging of triplicates based on their variabilities measured by their coefficient of variation or when one of the triplicates were flagged, a measure of variability was defined for the two remaining spots S_1 and S_2 as being equal to $(|S_1 - S_2|)/(S_1 + S_2)$;
- the flagging of signal intensities close to the noise level. A way to estimate the level of noise in the neighborhood of a spot is to measure the standard deviation of the background of that spot [21];
- etc.

The negative controls might also enable the filtering of low intensity spots because they might reflect the cross-reactivity of the detecting antibody with co-purifying insect cell or *E.coli* proteins or with the BCCP tag. For instance, in the dataset coming from a non responding group where no antibodies were supposed to be detected, it was observed that for the antigens cloned within insect cells, intensities were in the range of the intensities measured in the spots printed with only insect empty vectors, and the same for the antigens cloned in *E.coli* (see Fig. 1.6). To correct for the cross reactivity caused by the BCCP tags from insect cells (ICL) or *E.coli* (ECL), the use of a negative array is advisable since the number of tags contained within a spot depends on the size of the probe antigens.

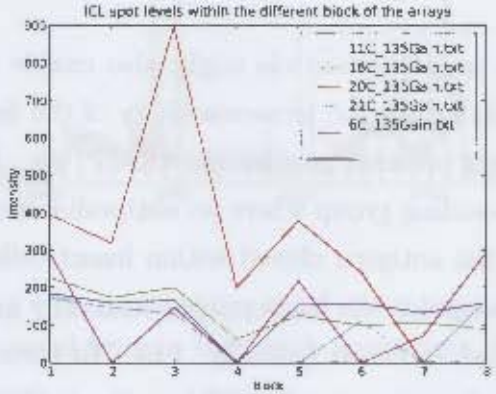
The data used in the following chapters were extracted using an automatic spot finding algorithm implemented in ArrayPro (version 4.5) and a gain setting of 135. A pixel threshold of 200 was applied for the segmentation of the images and different data filtering criteria further outlined in appendix B were applied to the data sets to clear the assay from questionable spots.



(a) Trendlines of the antibody responses against the 100 CT antigens for 5 non-responding patients identified by the IDs : 6,11,16,20 and 21.



(b) Average intensity of the triplicate spots only printed with *E.coli* empty vector per array.



(c) Intensities of the spot only printed with insect cell empty vector on each of the 8 blocks per array.

Figure 1.6: Antigens 1 to 87 were cloned and expressed using insect cells vectors and antigens 88 to 100 in *E.coli* vectors. a) and b) together reveal the relationship between the intensity obtained for the antigen cloned within insect cells and the unspecific binding to the insect cell empty vectors; while a) and c) together reveal the relationship between the intensity obtained for the antigen cloned within *E.coli* and the unspecific binding to the *E.coli* empty vectors.

The study presented in this thesis addresses two main issues: the normalization of the CT100 array data and their clustering. In addition, it aims to implement a pipeline to handle the CT100 data from their production to their normalization and clustering. As mentioned in section 1.2, the normalization of the data enables their analysis by correcting a certain range of variations in their measurements due to processes other than the biological activity targeted by the array experiment. The design of the CT100 accounts for the pin-to-pin and array-to-array systematical variations by the presence of housekeeping spots on each block and orientation signals on each array. However, the challenge is that the custom or boutique arrays gather relatively small number of probes which are all equally likely to find binding partners and have significant intensities which result in the difficulty to develop a robust biological hypothesis to normalize the data from two different samples. Chapter 2 addresses in a useful way both the issue of normalizing the data with a relatively small number of housekeeping controls and the issue of having a robust method able to deal with the flagging of some positive controls. Chapter 3 addresses the issue of defining a qualitative clustering method able to capture qualitatively the antibody profiles. And finally, Chapter 4 presents some conclusions. The pipeline has been implemented in Python and details are provided in Appendix A, while the name of the CT antigens are provided in Appendix B with regard to their annotations (Antigen 001, Antigen 002,... and Antigen 100.).

Chapter 2

Normalization method

The purpose of normalization is to correct microarray data from variations in their measurements due to other processes than the targeted biological activity in order to improve the quality of the data and make data from different experiments comparable [22, 23].

2.1 Background

Classical methods for microarray normalization have been published for two colour DNA microarrays. In two colour arrays, two samples - control and target - are assayed on the same array, and strong biological assumptions linking the two samples enable normalization of the data. In most cases, protein arrays are single colour arrays due to the size of the molecules being assessed, which does not allow for assaying two samples on the same array; and they lack strong biological assumptions to carry out their normalization [19]. However, the methods published for DNA microarrays can still inspire and support the normalization of protein arrays.

The classical normalization methods can be classified into three categories [12, 14, 24] :

1. Scaling methods: These assume that the chips being normalized share a common statistical measure, such as the mean, the median, etc. of their spot intensities or even the total intensities of their spots. m_j is a statistical measure on the chip j supposed to be equalized across the chips after normalization. The scaling factor α_j

for the chip j is given by,

$$\alpha_j = \frac{m}{m_j}$$

where m is the final value of m_j after normalization. The scaling normalization consists of multiplying every spot intensity on the chip j by the factor α_j [24].

2. Transformation methods: These rely on assumptions which allow quantitative mappings of two sets of spot intensities. The most popular are the curve fitting, the LOWESS and the quantile normalization methods. The curve fitting method assumes the distribution of the normalized datasets is known and tries to identify the parameters of the distribution model; for instance, Lu *et al* (2005) suggested adapting Zipf's law for the normalization of two or single colour DNA arrays [4, 22]. The LOWESS or LOcally WEighted polynomial regreSSion method maps the data from two datasets using a polynomial regression within overlapping intervals [4]. The LOWESS normalization is more effective when most of the spots within the two arrays keep the same intensities or have their intensities balanced [23]. The quantile normalization can be applied where the assumption of a common underlying distribution seems to be justified. It is fast, easy to implement and does not require any statistical modelling on the data [24, 25].

3. Invariant set method: This approach relies essentially on the ability to identify a suitable set of non-differentially expressed probes or housekeeping probes. The selection of the set of invariant spots might be experiment-dependent, and an uncritical choice of housekeeping probes can lead to bias in results [24, 26].

Custom antigen arrays are not amenable to standard normalization approaches [23]. Typically a relatively small selection of specific probes show strong signals for any one sample and the identity of these probes varies between samples. This breaks down most of the assumptions based on a common ground between the distribution of the signals across the arrays being compared, unless the samples being compared display special features [23]. Therefore, two methods are widely used; the first relies on housekeeping spots and the second on Microarray Sample Pool control (MSP) [22, 23, 27]. The housekeeping controls are supposed to keep a consistent signal across the experimental conditions and across different samples. In the case of the CT100, antigen arrays assessing antibody profiles

in blood serum, the housekeeping controls should ideally be serum independent to enable sample comparisons. The MSP method selects probes from a heterogeneous pool library and dilutes them at different concentrations to cover ranges similar to those covered by the probes used in the experiment; they must be printed in a number large enough to enable the assumption of non differential expression between samples [23]. Transformation methods such as LOWESS can subsequently be applied for normalization. However, the size of the custom arrays can be a limiting factor in the usage of the MSP method.

2.2 Method

The normalization method suggested in this study aims to make more efficient, effective and robust usage of the relatively small number of positive controls or housekeeping spots to correct for systematical bias in pin-to-pin and array-to-array variations. Robustness is understood here as the ability of the method to deal with the flagging of some positive controls, whilst still being based on sound biological principles.

The assumption made here, for the normalization, is that the housekeeping spots share a common underlying distribution across the chips (block, arrays, etc.) where they are printed. This way of looking at the housekeeping spots seems to provide more flexibility than assuming that the housekeeping spots keep the same intensities across the chips. Therefore, a composite normalization method combining quantile and total intensity normalization modules was considered to correct for systematical bias among the chips and to provide more flexibility when dealing with flagged positive control spots [16, 25].

Quantile based module

Since the housekeeping spots - human IgG samples - are replicate experiments on different chips, it seems reasonable to assume they share an underlying distribution across the arrays and the quantile approach can be used to identify, across the different chips, the corresponding housekeeping spot intensities based on their intensity distributions (see Tables 2.1 and 2.2).

Bolstad *et al* (2002) describes the algorithm to carry out spot identification within the

same quantile. Let S_{ij} be the intensity of housekeeping spot i on the chip j,

1. Load the housekeeping spot intensities S_{ij} into an $I \times J$ matrix X;
2. Sort the spot intensities in each column j of X to get X_{sort} ;
3. Take the means across the rows i of X_{sort} and get $\bar{X}_{i.}$.

$\bar{X}_{i.}$ is considered to be the underlying distribution of the housekeeping spot intensities across the chips [25]. This reorganization enables more flexibility to handle outliers or flagged spots within the housekeeping dataset (see Tables 2.1 and 2.2 for illustrations). The next step is implementing the total intensity based module.

Control	Probe	Chip 1	Chip 2	Chip 3
Housekeeping 1	1	4.7	5	14
Housekeeping 1	2	3.5	6	11
Housekeeping 1	3	0.5	8	8
Housekeeping 2	1	7	11	15
Housekeeping 2	2	12	10	19
Housekeeping 2	3	10	9	16

Table 2.1: Illustration of the intensity distributions before normalization. Housekeeping 1 and 2 denote the housekeeping controls printed in triplicate at two different concentrations.

Control	Probe	Chip 1	Chip 2	Chip 3	Underlying
Housekeeping 1	1	0.5	5	8	4.5
Housekeeping 1	2	3.5	6	11	6.83
Housekeeping 1	3	4.7	8	14	8.9
Housekeeping 2	1	7	9	15	10.33
Housekeeping 2	2	10	10	16	12
Housekeeping 2	3	12	11	19	14
	Total	37.2	44	75	52.07

Table 2.2: Identification of the corresponding spots in the underlying distribution, and identification of a potential outlier in Chip 1,Housekeeping 1, probe 1.

Total intensity based module

This module assumes that after normalization, the housekeeping spot intensities printed onto the different chips should be balanced in such a way that their sums on each chip should be the same [16,28]. Therefore, it is expected that all the normalized chips have the same value of the total intensity of their housekeeping spots which is given by $\sum_{i=1}^{N_{spots}} \bar{X}_{i.}$ and the normalization factor α_k to normalize the chip k is given by,

$$\alpha_k = \frac{\sum_{i=1}^{N_{spots}} \bar{X}_{i.}}{\sum_{i=1}^{N_{spots}} \bar{X}_{ik}}$$

This is in fact a scaling normalization method where it is assumed that the different chips share a common total intensity of their housekeeping spots, whilst taking in account the potential existence of flagged spots within the housekeeping spots. As illustrated in Table 2.3, when a probe is identified as an outlier on a chip the corresponding probes are flagged across all chips prior to normalization.

Control	Probe	Chip 1	Chip 2	Chip 3	Underlying
Housekeeping 1	1	Flagged	Flagged	Flagged	Flagged
Housekeeping 1	2	4.9	7.1	7.64	6.83
Housekeeping 1	3	6.58	9.47	9.72	8.9
Housekeeping 2	1	9.8	10.65	10.41	10.33
Housekeeping 2	2	14	11.83	11.11	12
Housekeeping 2	3	16.8	13.02	13.19	14
	Total	52.07	52.07	52.07	52.07

Table 2.3: Scaling normalization where it is assumed that chips share a common underlying distribution of their housekeeping spots.

2.3 Evaluation of the method

Two approaches were selected for the evaluation of this new normalization method. The first used a simulation based on the Rocke and Durbin error model to generate multiplicative bias on to the different chips [29, 30], and the second used an experimental reproducibility dataset.

2.3.1 Simulation

The simulation enabled the comparison between this new normalization method and the classical methods using the assumption of the equality of the mean intensities of the housekeeping spots across the chips [16, 26]. For that purpose the error model of Rocke and Durbin (2001) was used because of its proportional error term which can model efficiently systematical bias introduced by the pin-to-pin or array-to-array variations [29].

$$y = \alpha + xe^{\eta} + \epsilon$$

where y is the measured spot intensity, α the mean background signal of the spot, x the true value of the spot intensity, e^{η} the proportional error term and ϵ the standard deviation of the background. In the simulation, the signal intensity was considered to be significantly above the background and noise, which simplifies the equation to,

$$y \approx xe^{\eta}.$$

Therefore the focus was on the proportional error term with η following a normal distribution of mean μ_η and standard deviation σ_η.

Eight chips with exactly the same spot intensities were generated, then each submitted to proportional errors or bias characterized by a given μ_η and σ_η randomly selected in the intervals $0 < \mu_\eta < Sup_{\mu_\eta}$ and $0 < \sigma_\eta < Sup_{\sigma_\eta}$, where Sup_{μ_η} and Sup_{σ_η} represent respectively the highest value μ_η and σ_η can take. Every chip contained 16 triplicates; two of these were housekeeping spots at two different concentrations.

Normalization was carried out using three methods. The first scales the spots in the chips i by the normalization factor α_i used to balance the mean intensity of the low concentration housekeeping spots across the eight chips. The second scales the spots in the chips i by the normalization factor α_i used to balance the mean intensity of the high concentration housekeeping spots across the eight chips. The third one, described in the section 2.2, scales the spots in the chips i by the normalization factor α_i used to balance the total intensities of the housekeeping spots across the eight chips.

After normalization, the coefficient of variation (CV) of a random antigen was calculated across the eight chips. CV_1, CV_2 and CV_3 denote respectively the CV values obtained

using the normalization method one, two and three; while CV_{BN} denotes the CV value of the randomly selected antigens before normalization. For each given Sup_{μ_η} and Sup_{σ_η}, the simulations were run 2000 times.

The μ_η component of the systematical bias accounts for the major part of the CV_{BN}. For the pin-to-pin variations, μ_η stays very close to zero since all the pins share the same physical properties and print antigen spots of roughly the same diameters. Calibration experiments were run at the CPGR for the CT100 array and showed that pin-to-pin variations were more or less 15% in the worst cases. For the array-to-array variations μ_η might depend on more experimental variables and might vary in larger scales, but is still located in the neighborhood of zero, and for a well calibrated experiment σ_η should stay very small.

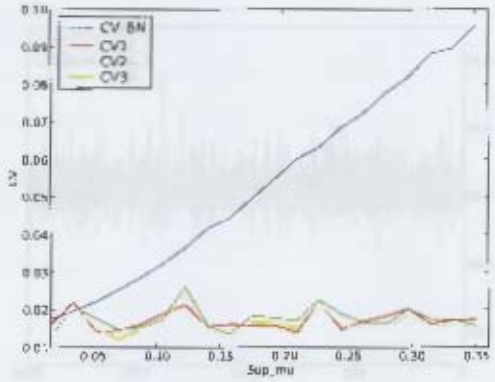
2.3.2 Results and discussion

The simulation showed that the CV_{BN} increases faster with Sup_{μ_η} than Sup_{σ_η}, and that the normalization methods handle the systematical bias caused by the parameter Sup_{μ_η} well. However, the final CVs are mostly dependent on the parameter Sup_{σ_η} which translates the range of noise variations in the systematical bias (see Fig. 2.1).

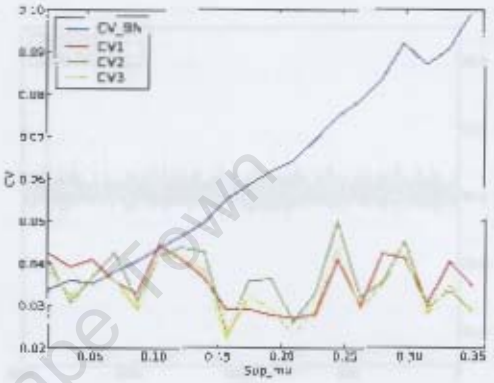
The simulation also showed that for a given Sup_{σ_η}, the method developed in this study yields better or comparable results than the other two methods (see Fig. 2.2).

On average, 43% of the trials yielded better results for our new method than the other two methods; while in 32% it only produced better results than method 1 and in 25% better than method 2. Importantly, there were no cases where the our method was the worst of the three.

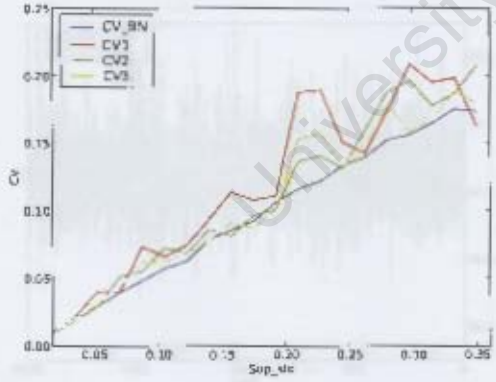
Moreover, the simulation assessed the robustness of our method when some positive controls are flagged. Indeed, when one of the positive controls is flagged, methods 1 or 2 rely automatically on the average of two spot intensities which is, from a statistical point of view, not advisable. The robustness of our method was assessed using 4 different cases. Case 1 considered a flagged spot within the low concentration housekeeping spots, case 2 considered a flagged spot within the high concentration housekeeping spots, case 3 considered one flagged spot within both low and high concentration housekeeping spots and case



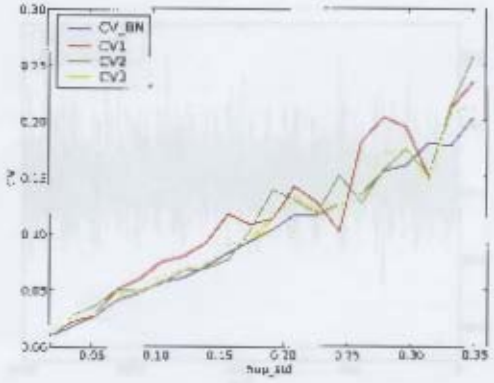
(a) Sup_{σ_η} set to 0.0175 (here $Sup\_mu = Sup_{\sigma_\eta}$)



(b) Sup_{σ_η} set to 0.035 (here $Sup\_mu = Sup_{\mu_\eta}$)

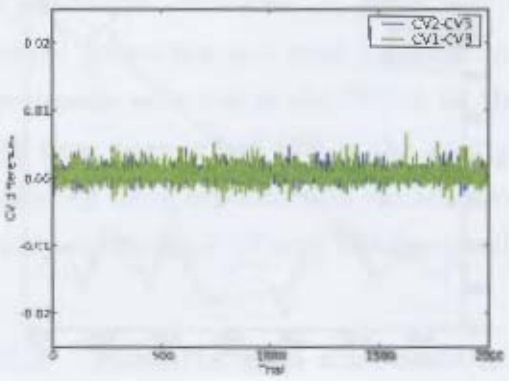


(c) Sup_{μ_η} set to 0.0175 (here $Sup\_std = Sup_{\sigma_\eta}$)

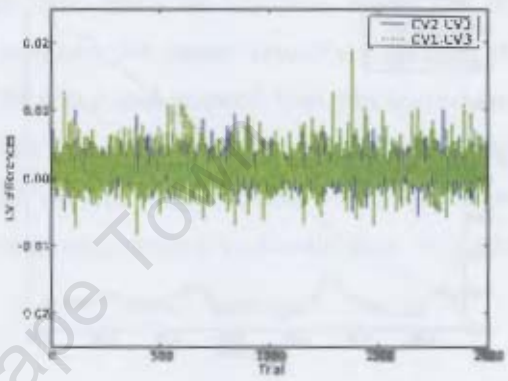


(d) Sup_{μ_η} set to 0.035 (here $Sup\_std = Sup_{\sigma_\eta}$)

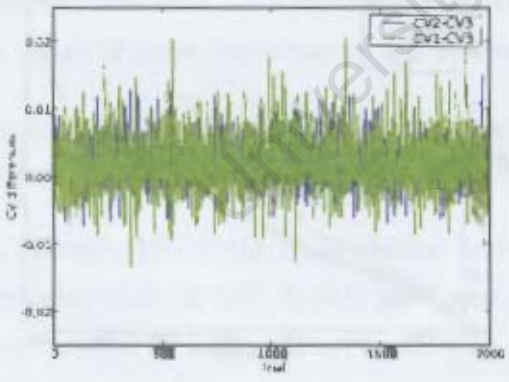
Figure 2.1: Figures a) and b) show the independence of the normalization methods to the range of systematical bias (Sup_{μ_η}) ; and figures c) and d) point out the dependence of the final CVs on the noise in the systematical bias.



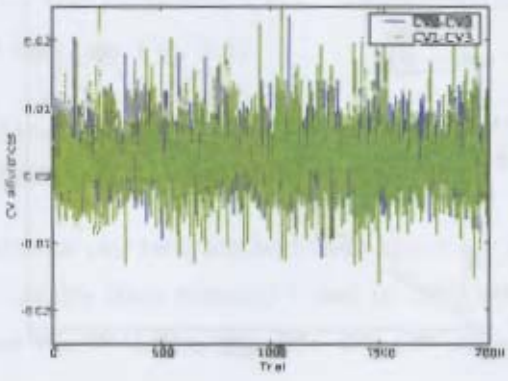
(a) $Sup_{\sigma_n} = 0.0175$



(b) $Sup_{\sigma_n} = 0.035$



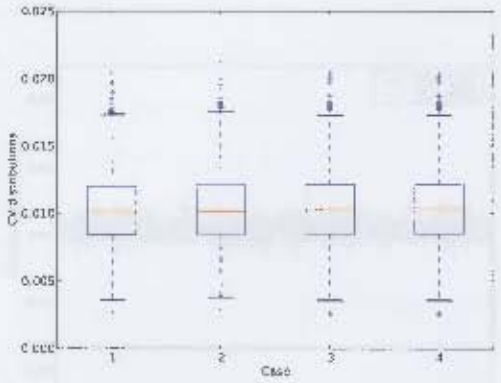
(c) $Sup_{\sigma_n} = 0.0525$



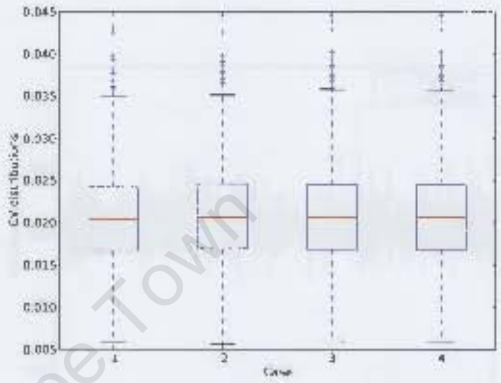
(d) $Sup_{\sigma_n} = 0.07$

Figure 2.2: Normalization method comparisons for different values of Sup_{σ_n}. The comparison showed that the differences between CV_1 and CV_3, as well as CV_2 and CV_3 increases with Sup_{σ_n}. In most of the cases, CV_3 is lower than the two other CVs.

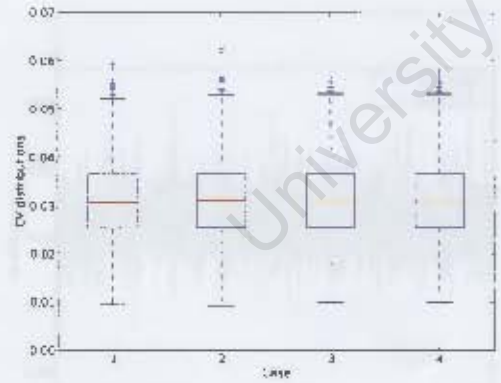
4 considered no flagged spots among the housekeeping spots. Boxplots allowed comparison of the distribution of the CV values obtained for the different cases after 2000 trials and different values of Sup_{σ_η}, and revealed that the CV distribution of the different cases were in the same ranges (see Fig. 2.3).



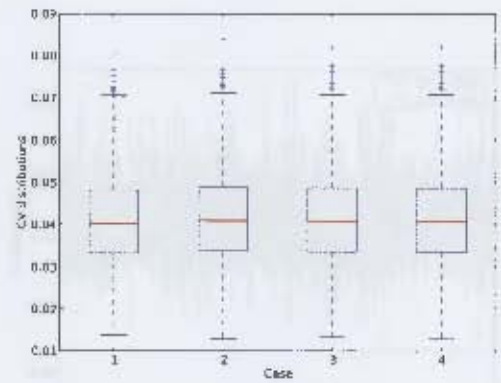
(a) $Sup_{\sigma_n} = 0.0175$



(b) $Sup_{\sigma_n} = 0.035$



(c) $Sup_{\sigma_n} = 0.0525$



(d) $Sup_{\sigma_n} = 0.07$

Figure 2.3: The CV distributions for the 4 different cases considered (1: one flagged spot within the low concentration housekeeping spots, 2: one flagged spot within the high concentration housekeeping spots, 3: one flagged spot within both low and high concentration housekeeping spots and 4: no flagged spots among the housekeeping spots) seem to be quite similar for different values of Sup_{σ_n}. This suggests that the method proposed in this study shows flexibility to deal with flagged housekeeping control spots.

2.3.3 Reproducibility experiment

The reproducibility experiment was performed on two different real-life slides containing 4 arrays each. The coefficient of variations were compared for the same antigens across all the eight arrays and a significant improvement was observed after normalization (see Table 2.4). A general improvement in the similarity of the arrays is demonstrated by the increase in the correlation between the arrays, and the scatter plots of the arrays taken two by two (see Table 2.5 and Fig. 2.4).

	Before normalization	After normalization
$0.00 \leq CV < 0.20$	0	7
$0.20 \leq CV < 0.30$	0	18
$0.30 \leq CV < 0.40$	1 (0.39)	6
$0.40 \leq CV < 0.50$	6	1(0.44)
$0.50 \leq CV$	25	0

Table 2.4: This table shows the CV distribution per interval of 32 antigens out of 100 which showed a significant positive signal across the 8 replicate arrays after spot filtering. The CV values were calculated from all the replicates across the 8 replicate arrays. If there is only one antigen in an interval the CV value is given in brackets.

	Before normalization	After normalization
$0.00 \leq r < 0.60$	1 (0.59)	0
$0.60 \leq r < 0.65$	1 (0.60)	0
$0.65 \leq r < 0.70$	0	1 (0.68)
$0.70 \leq r < 0.75$	1 (0.71)	0
$0.75 \leq r$	25	27

Table 2.5: This table shows the Pearson correlation coefficient (r) distribution per interval for the 28 pairwise associations of 8 replicate arrays.

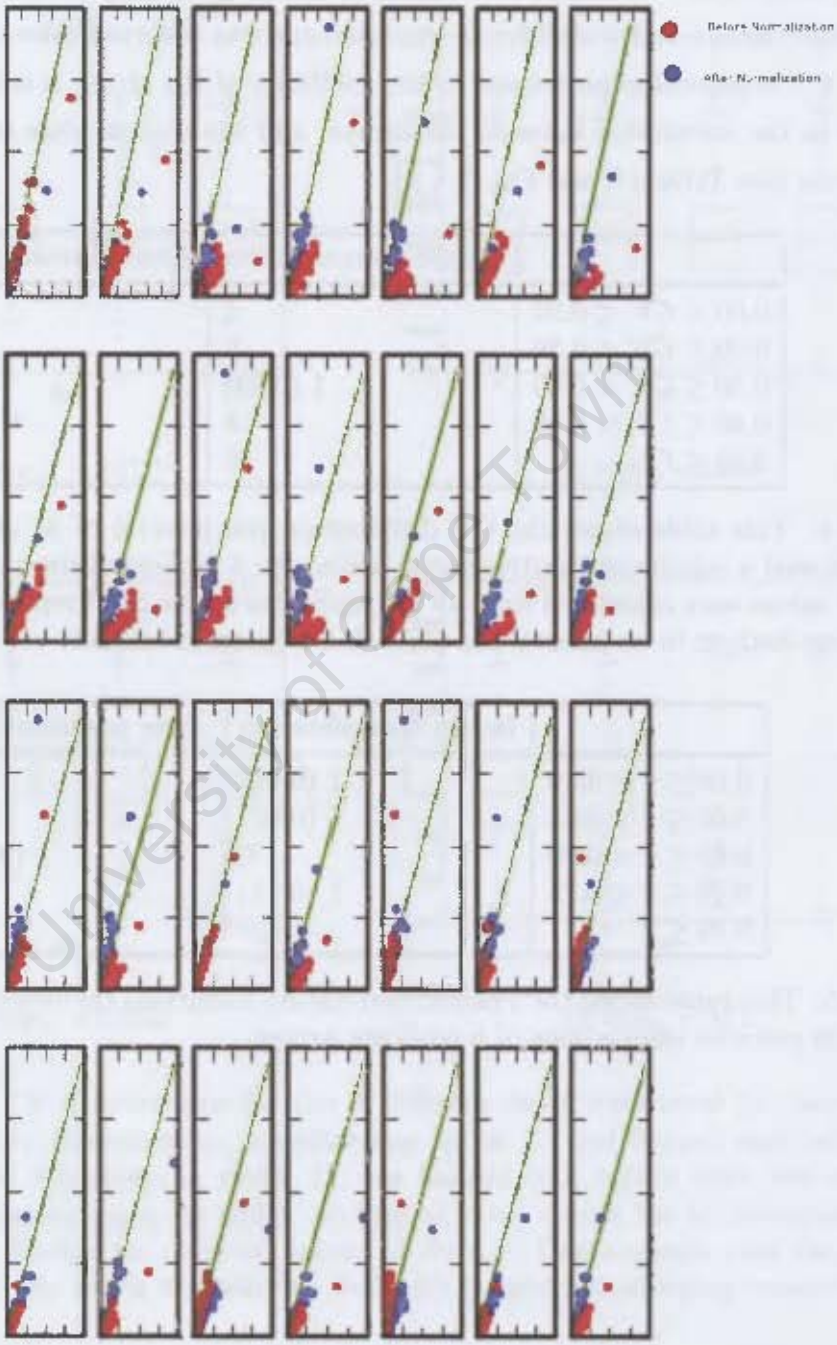


Figure 2.4: The scatter plots illustrate how the normalization processes increase the similarity between the arrays. In red are the arrays before normalization and in blue after normalization.

Chapter 3

Qualitative Clustering

The purpose of all clustering methods is to group or segment a set of items by taking into consideration a criterium of similarity or dissimilarity. The clustering can reveal something about the data structures, enabling the reduction of their dimensions as well as revealing outliers; an outlier being an item not sufficiently similar to any other items. An additional goal of clustering can be to infer hierarchical order between clusters [4, 16, 31–33].

Before choosing a clustering algorithm, one has to think about the type of cluster which might be expected from the dataset and, the most appropriate measure of similarity to capture the clusters of interest. Another consideration is the performance of the selected clustering algorithm [31, 32].

In this chapter, the focus is on the qualitative clustering of antibody profiles defined quantitatively against one hundred Cancer Testis antigens. For the reasons explained within section 1.3, the qualitative clustering should be based on the trendline similarities between antibody profiles of the patient samples. To this end, two clustering methods are compared; the factor analysis method and the K-mean method using a Pearson correlation metric.

The Factor Analysis method is an unsupervised method used to investigate or explore the intrinsic factors summarizing the correlations among variables or observations [31]. Factor analysis has been widely used in intelligence research to explain a variety of results based on different tests by identifying groups of correlated results. For instance, the performance

at running, weight lifting and jumping could be explained by the general athletic ability [33–35].

The K-mean method is among the most popular clustering algorithm. It is an iterative method relying on the minimization of an objective function defining a measure of dissimilarity between the items or of their K centroids, a centroid here being a measure by which the dissimilarity between clusters or between clusters and items can be summarized by one value [16,31,32]. The K-mean method requires a couple of inputs to achieve the clustering such as the number of clusters K, the maximum number of iterations, the selection of the similarity metric and an initial clustering which is improved iteratively until a steady state or the maximum number of iterations is reached [4].

3.1 Factor analysis method

3.1.1 Description

Let Y_1, Y_2,...,Y_n be n variables or antibody profiles under study. The factor analysis method aims to cluster the variables according to their intrinsic trendlines by identifying a set of regression lines called factors, then measures the strength of the correlations between the variables and each of the factors. Further investigation of the factors provides information about the unobservable reason explaining the patterns within the measurements [31,35].

An analytical description of the model is given below,

$$\begin{aligned} Y_1 &= \alpha_{10} + \alpha_{11}F_1 + \alpha_{12}F_2 + \cdots + \alpha_{1m}F_m + e_1 \\ Y_2 &= \alpha_{20} + \alpha_{21}F_1 + \alpha_{22}F_2 + \cdots + \alpha_{2m}F_m + e_2 \\ &\cdots \\ Y_m &= \alpha_{n0} + \alpha_{n1}F_1 + \alpha_{n2}F_2 + \cdots + \alpha_{nm}F_m + e_n \end{aligned}$$

where the $\{e_k\}_{1\leq k\leq n}$ are the error terms translating the limitation of the factorial decomposition to explain respectively the variables $\{Y_k\}_{1\leq k\leq n}$; and α_{ij}, called *loading* of the variable Y_i on the factor F_j is the measure of the Pearson correlation between Y_i and F_j.

The factor analysis method relies on two assumptions, the first is that the factors $\{F_j\}_{1\leq j\leq n}$ have to be uncorrelated and independent from each other and from the error terms $\{e_i\}_{1\leq i\leq n}$ in addition to being normally distributed with a mean of zero and a variance of one. The second is that the error terms $\{e_i\}_{1\leq i\leq n}$ have to be normally distributed with a mean of zero and a variance σ_i [35].

From those two assumptions, the variance of the variables Y_i can be expressed as being equal to,

$$\begin{aligned} Var(Y_i) &= Var(\alpha_{i0} + \alpha_{i1}F_1 + \alpha_{i2}F_2 + \cdots + \alpha_{im}F_m + e_i) \\ &= \sum_{j=1}^{m} Var(\alpha_{ij}F_j) + Var(e_i) \\ &= \sum_{j=1}^{m} \alpha_{ij}^2 Var(F_j) + \sigma_i^2 \\ &= \sum_{j=1}^{m} \alpha_{ij}^2 + \sigma_i^2 \end{aligned}$$

where $\sum_{j=1}^{m} \alpha_{ij}^2$ is called *communality* and translates as the part of the total variance of Y_i explained by the factorial decomposition; and σ_i^2, called *specific variance*, translates as the part of the variance Y_i not accounted for by the factorial decomposition.

Both *communality* and *specific variance* can be interpreted as the degree of explanation related to the data structures yielded by the factorial decomposition.

The description of the factorial analysis method leaves some ambiguities about the selection of the best factors. Indeed, different sets of factors can fill the requirements of being uncorrelated and independent from each other, and the selection of one set or another might provide different loadings. At this stage it is important to stress the fact that the type of cluster investigated guides the selection of the optimal factors to be used.

In this chapter, the clustering investigation is about non overlapping qualitative clusters, and the factoral decomposition is also expected to bring the communality as close as possible to the variance $Var(Y_i)$. The principal component analysis (PCA) should provide uncorrelated and independent factors after *varimax* rotation [31, 35]; which should yield

the closest communalities to the total variance of the variables $\{Var(Y_i)\}_{1\leq i\leq n}$. For PCA to capture the correlation structures of the variables rather than their covariance, the variable should first be standardized to equilibrate the influences of variables with high and low variances.

The standardization of the variables, also called in statistics *standard score* [36], can be described as below,

$$Z_{ij} = \frac{Y_{ij} - \bar{Y}_i}{S_i}$$

where $\bar{Y}_i$ is the mean of the variable Y_i, Y_{ij} the j^{th} component of Y_i before standardization, Z_{ij} the j^{th} component of Y_i after standardization and S_i the standard deviation of Y_i.

After standardization, all the variables should have a mean of zero and a variance of one.

The issue of the number of factors to be retained as a potential cluster can be answered using the *Kaiser criterion* which is the most widely used [37]. This states that only the factors associated with eigenvalues greater than one should be retained. The *Kaiser criterion* relies on the fact that the eigenvalues show the proportions of the total variance explained by the factors. Therefore, since the variables have been standardized, the interest is on the factors which are likely to explain at least one variable.

3.1.2 Results and discussion

Factor analysis as a qualitative clustering method was assessed against two different datasets. The first one contained the data from a reproducibility experiment made of 8 CT100 arrays assayed with the same patient serum. The second one contained the data from a vaccine response experiment comprising 33 CT100 arrays assayed with samples taken at three different time points from each of 11 patients.

Reproducibility experiment

Based on prior knowledge of the reproducibility experiment, it is expected that the factor analysis method will give one factor or cluster to whom the antibody profiles of the 8 arrays would be strongly correlated. The results of the factor analysis clustering are summarized

in Table 3.1 below.

i	Standardized Y_i	$Var(Y_i)$	α_{i1}	$\sum_{j=1}^{1} \alpha_{ij}^2$	Variance explained (%)
1	Array 1	1	0.98	0.96	96.24
2	Array 2	1	0.96	0.92	92.33
3	Array 3	1	0.98	0.97	96.66
4	Array 4	1	0.91	0.83	82.97
5	Array 5	1	0.94	0.89	88.54
6	Array 6	1	0.97	0.95	95.02
7	Array 7	1	0.98	0.95	95.49
8	Array 8	1	0.99	0.99	98.58

Table 3.1: Reproducibility experiment clustering using factor analysis

As expected, the results in Table 3.1 revealed one underlying factor within the 8 arrays and high correlations or *loadings* of the antibody profiles to that unique factor which are not less than 0.91. The results also showed that the factorial decomposition explains at least 82.97% of the variance within the antibody profiles.

Vaccine response experiment

Three different time points were considered in the vaccine response experiment; 2 weeks before vaccination (time A), 4 weeks after vaccination (time B) and 16 weeks after vaccination (time C). Prior knowledge suggested that two clusters, responder and non-responder, should be expected. Indeed, two groups were found experimentally at the end of an anonymous vaccine response study from which the data were taken; the responder group consisted of 5 patient samples and the non-responder group of six patient samples (see Table 3.2).

Responder IDs	Non-responder IDs
1	6
5	8
7	11
14	16
25	20
	21

Table 3.2: IDs of the arrays per response to a vaccine

At time point A (see Table 3.3), the factor analysis method revealed three underlying factors. All the arrays showed a high correlation to the factor $F1$ with the exception of array 14 which showed a high correlation to $F3$, and arrays 1, 5 and 25 which showed a weak correlation to all three factors.

i	Standardized Y_i	$Var(Y_i)$	α_{i1}	α_{i2}	α_{i3}	$\sum_{j=1}^{3} \alpha_{ij}^2$	Variance explained (%)
1	11A_135Gain.txt	1	0.86	0.3	0.04	0.82	82.26
2	14A_135Gain.txt	1	0.29	0.45	0.77	0.88	87.92
3	16A_135Gain.txt	1	0.94	0.16	0.04	0.91	91.39
4	1A_135Gain.txt	1	0.65	0.47	0.16	0.66	66.45
5	20A_135Gain.txt	1	0.77	0.26	0.18	0.7	69.64
6	21A_135Gain.txt	1	0.95	0.13	0.01	0.92	91.86
7	25A_135Gain.txt	1	0.65	0.33	0.19	0.57	57.08
8	5A_135Gain.txt	1	0.33	0.52	0.55	0.68	68.36
9	6A_135Gain.txt	1	0.9	0.14	0.11	0.85	84.59
10	7A_135Gain.txt	1	0.8	0.35	0.03	0.77	76.6
11	8A_135Gain.txt	1	0.83	0.41	0.02	0.87	86.65

Table 3.3: Vaccine response experiment clustering using factor analysis at time point A. αA_135Gain.txt refers to Patient id α at time point A, with the array scanned at a gain setting of 135 and saved in a text file.

At time point B (see Table 3.4), the factor analysis method revealed two underlying factors. All the arrays were strongly correlated to F1 with the exception of array 14, which was strongly correlated to factor $F2$, and array 5 which was weakly correlated to both factors $F1$ and $F2$.

At time point C (see Table 3.5), the factor analysis method revealed two underlying factors. Most of the arrays were strongly correlated to F1 with the exception of array 14, which was strongly correlated to factor $F2$, and arrays 1 and 5 which were weakly correlated to both factors $F1$ and $F2$.

In summary, across all the time points a set of arrays (ie patient antibody responses measured on the arrays) seemed to stay clustered together and correlated to the same factor $F1$, while the other arrays seem to be either singly correlated to one factor or weakly correlated to all the factors. That can mean that since antibody profiles of responders and non-responders are analyzed, the non responders have a consistent antibody profile across the samples while the responders exhibit different antibody profiles characterizing their

i	Standardized Y_i	$Var(Y_i)$	α_{i1}	α_{i2}	$\sum_{j=1}^{2} \alpha_{ij}^2$	Variance explained (%)
1	11B_135Gain.txt	1	0.81	0.25	0.72	71.68
2	14B_135Gain.txt	1	0.35	0.86	0.86	85.89
3	16B_135Gain.txt	1	0.96	0.12	0.94	94.45
4	1B_135Gain.txt	1	0.83	0.23	0.73	73.47
5	20B_135Gain.txt	1	0.89	0.09	0.81	80.73
6	21B_135Gain.txt	1	0.95	0.05	0.91	90.81
7	25B_135Gain.txt	1	0.95	0.07	0.91	90.68
8	5B_135Gain.txt	1	0.63	0.19	0.43	42.81
9	6B_135Gain.txt	1	0.94	0.04	0.88	88.5
10	7B_135Gain.txt	1	0.86	0.19	0.77	77.5
11	8B_135Gain.txt	1	0.84	0.27	0.77	76.95

Table 3.4: Vaccine response experiment clustering using factor analysis at time point B. αB_135Gain.txt refers to Patient id α at time point B, with the array scanned at a gain setting of 135 and saved in a text file.

i	Standardized Y_i	$Var(Y_i)$	α_{i1}	α_{i2}	$\sum_{j=1}^{2} \alpha_{ij}^2$	Variance explained (%)
1	11C_135Gain.txt	1	0.83	0.09	0.71	70.57
2	14C_135Gain.txt	1	0.18	0.7	0.53	52.7
3	16C_135Gain.txt	1	0.97	0.03	0.93	93.25
4	1C_135Gain.txt	1	0.6	0.53	0.64	64.23
5	20C_135Gain.txt	1	0.8	0.05	0.64	64.18
6	21C_135Gain.txt	1	0.93	0.09	0.88	88.11
7	25C_135Gain.txt	1	0.88	0.11	0.78	77.84
8	5C_135Gain.txt	1	0.51	0.24	0.32	31.62
9	6C_135Gain.txt	1	0.89	0.21	0.84	84.02
10	7C_135Gain.txt	1	0.87	0.29	0.84	83.71
11	8C_135Gain.txt	1	0.85	0.34	0.83	83.44

Table 3.5: Vaccine response experiment clustering using factor analysis at time point C. αC_135Gain.txt refers to Patient id α at time point C, with the array scanned at a gain setting of 135 and saved in a text file.

individual humoral responses.

The fact that arrays 7 and 25 were found to cluster within the responder group can have two different explanations; one can be that patients 7 and 25 had not yet reached their humoral response equilibrium at time point C and the clinical test saying they were responders was carried out at a later time point. The other reason could be that patients 7 and 25 were just false positive responders - ie their recovery was not in response to vaccination.

A look at the trendlines of arrays 7 and 25 within the non-responder group (see Fig. 3.1) seems to suggest that factor analysis does capture the antibody profiles sharing the same trends. As suggested by the factor analysis results (see Table 3.5), the responder patients exhibit various kind of trendlines (see Fig. 3.2).

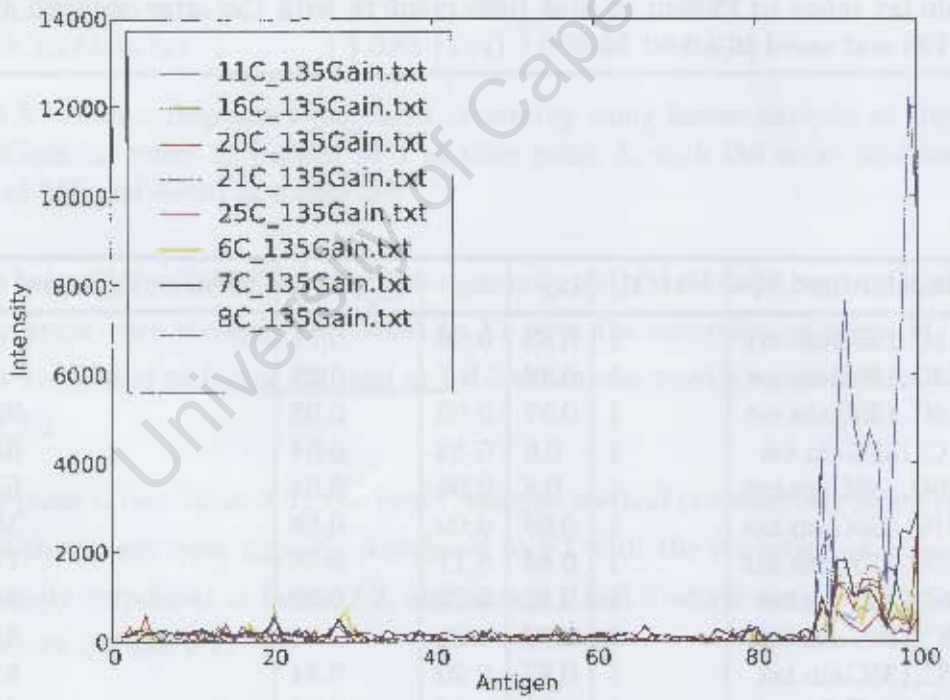


Figure 3.1: Trendlines of the non-reponder's antibody profiles including 7C and 25C.

An assessment of the likelihood of generating random clusters of arrays was then carried out. 2 to 200 arrays were randomly generated, one thousand times each, and clustered using the factor analysis. As the method uses standardized variables, the simulation results

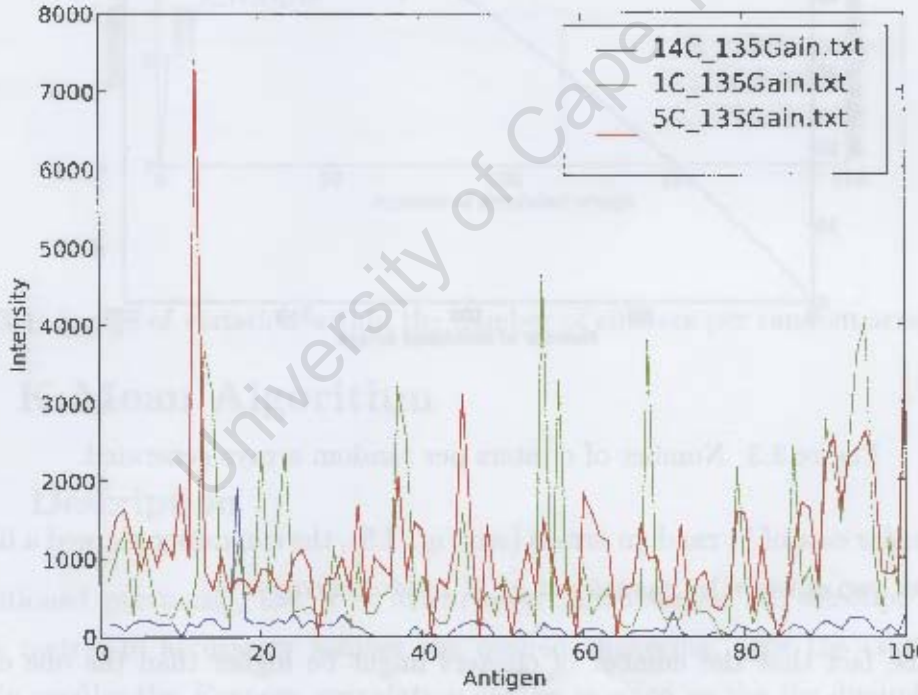


Figure 3.2: Trendlines of the reponder's antibody profiles excluding 7C and 25C.

were independent of the range of the signal intensity generated. Figures 3.3 and 3.4 show consistent numbers of clusters for a given number of random arrays.

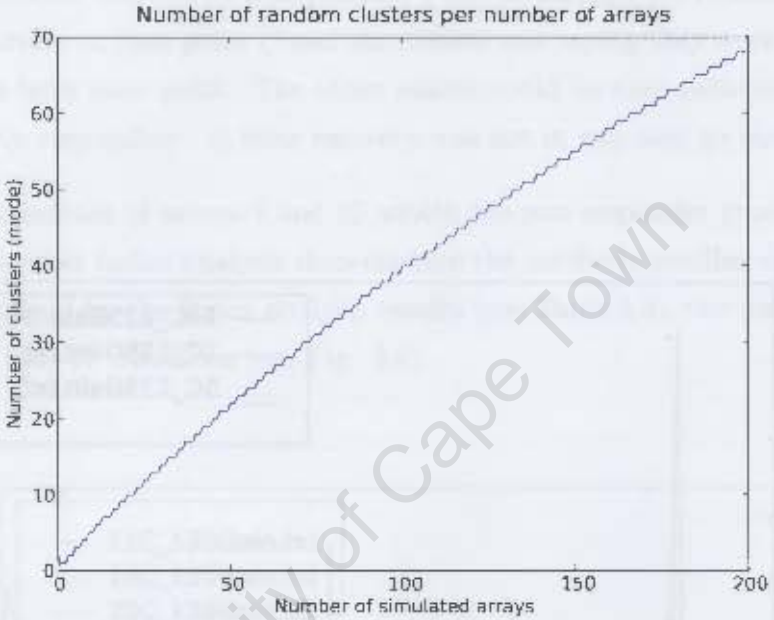


Figure 3.3: Number of clusters per random arrays generated.

In the particular case of 11 random arrays (see Fig. 3.5), the simulation showed a likelihood of zero to get two clusters by chance out of 11 random arrays.

Inspite of the fact that the number of clusters might be higher than the one expected, further analysis of the loadings might enable the reduction of the number of clusters to a reasonable number or the detection of outliers (see Table 3.3).

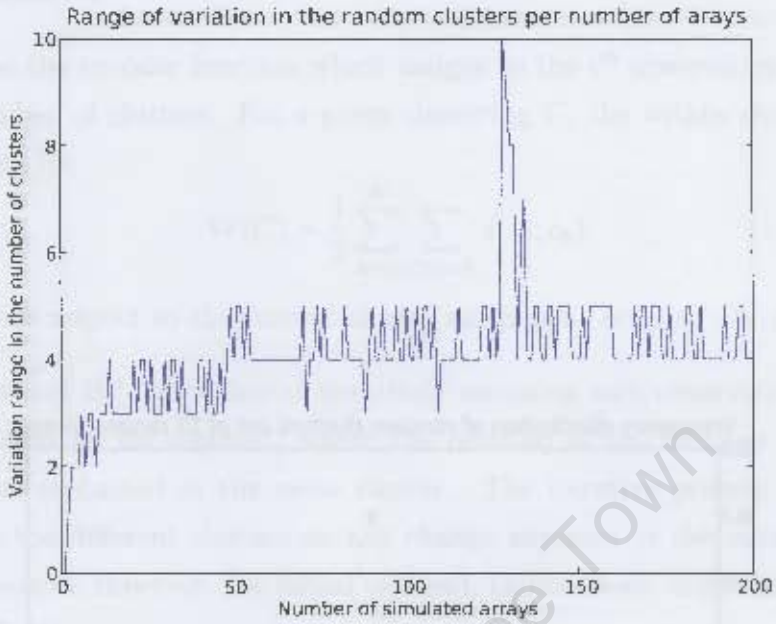


Figure 3.4: Range of variation within the number of clusters per random arrays generated

3.2 K-Mean Algorithm

3.2.1 Description

As mentioned previously, the $K - Mean$ algorithm requires the selection of the most suitable metric to accurately achieve the desired clustering. For the clustering of the antibody profiles the *Pearson correlation* can be selected as the the dissimilarity metric since it enables the capture of the similarities in shapes between two profiles by measuring their correlations.

The Pearson correlation metric is defined as being [38],

$$d_{ij} = 1 - \rho_{y_i,c_j}$$

where ρ_{y_i,c_j} is the Pearson correlation between antibody profiles y_i and a cluster centroid c_j.

Thus, the metric d_{ij} would be tending to 1 for uncorrelated antibody profiles or 0 if they

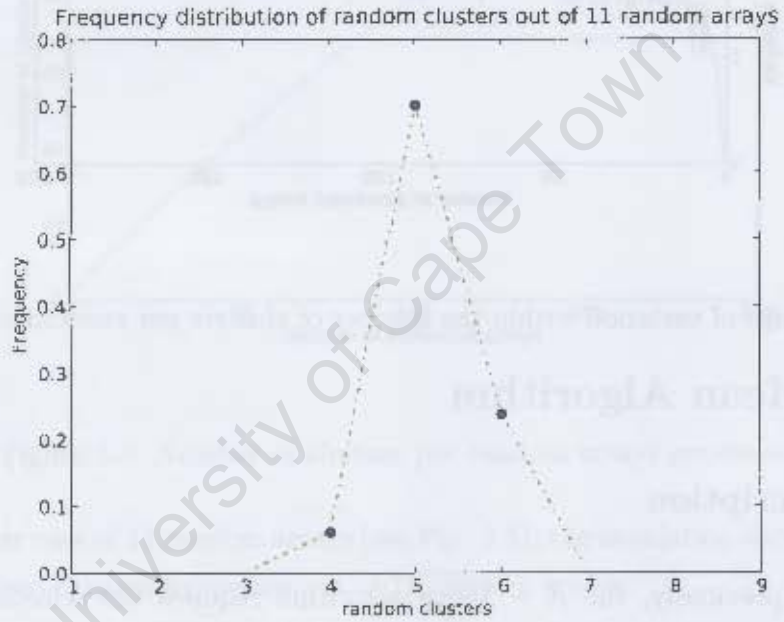


Figure 3.5: Random number of cluster frequency distribution for 11 randomly generated arrays.

are strongly correlated.

Let $C(i) = k$ be the encoder function which assigns to the i^{th} observation the k^{th} cluster, and K the number of clusters. For a given clustering C, the within cluster observation scatter [31] given by,

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k} d(x_i, c_k)$$

is minimized with respect to the current cluster assignment centroid $\{c_1, c_2, \ldots, c_k\}$.

The minimization of $W(C)$ consists of iteratively assigning each observation to the closest cluster represented by its centroid, where the centroid is the measure of the mean of the observations contained in the same cluster. The iterative process stops when the assignments to the different clusters do not change anymore or the maximal number of iterations is reached. However, the initial centroid, called *seeds*, might influence the final partitioning [16, 31].

3.2.2 Results and discussion

MultiExperimentViewer - version 4.5 was used for the clustering of the same vaccine experiment dataset described in section 3.1.2 by using the *K-Mean* algorithm. According to prior knowledge, the number of clusters was set to two; the selected metric is the Pearson correlation and the maximum number of iterations was arbitrarily set to one million (see Fig. 3.6).

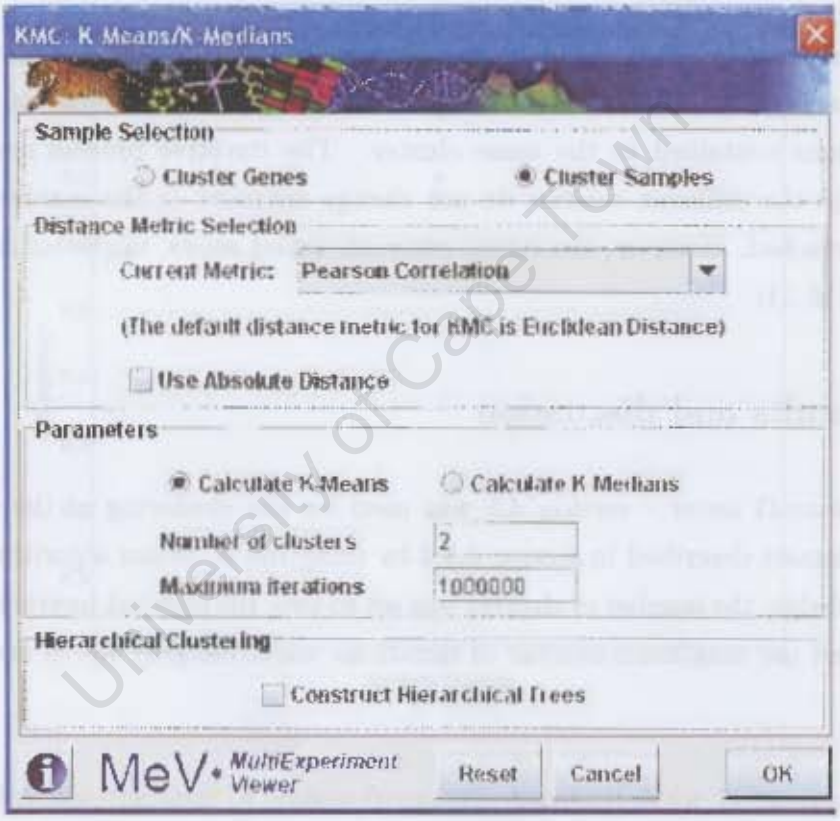


Figure 3.6: The dialogbox to set the K-mean algorithm parameters

The dataset was first standardized for the same reasons described in section 3.1.1 and at each time point, the clustering algorithm was run 10 times. The results are presented in table 3.6.

Time A	Responders	Time B	Responders	Time C	Responders
Trial 1	14;25;5;1	Trial 1	14;5	Trial 1	14;5
Trial 2	14;20;25;6	Trial 2	11;14;8	Trial 2	1;25;5
Trial 3	14;20;5	Trial 3	14;20;7;8	Trial 3	
Trial 4	14;25;5	Trial 4	14;5	Trial 4	14;1
Trial 5	14;21;25;6;7;8	Trial 5	14;1;5;20;21	Trial 5	14;1
Trial 6	14;20	Trial 6	14;5	Trial 6	
Trial 7	25;5	Trial 7	14;5	Trial 7	14;1
Trial 8	14;20;1;5;6	Trial 8	14;1	Trial 8	1;25;5
Trial 9	14;20;5	Trial 9	14;1;20;21;5	Trial 9	14;5
Trial 10	11;1;5;8	Trial 10	14;1;20;21;25;6	Trial 10	11;14;8

Table 3.6: Time A, B and C responder cluster using K-mean for 10 trials.

The differences between the cluster compositions at each trial are a result of the heuristical component of the *K-Mean* algorithm and the *seeds* considered. However, by looking at the higher frequency of the cluster members across the trials (see Table 3.7), the clusters obtained using the *K-Mean* algorithm seem to exhibit the same compositions as those obtained using the factor analysis (see Tables 3.3, 3.4 and 3.5).

Array-Time A	Frequency	Array-Time B	Frequency	Array-Time C	Frequency
14	0.8	14	1.0	14	0.8
5	0.7	5	0.6	1	0.5
25	0.5	20	0.4	5	0.4
20	0.5	1	0.4	25	0.2
1	0.3	21	0.3	8	0.1
6	0.3	8	0.2	11	0.1
8	0.1	25	0.1		
11	0.1	7	0.1		
		11	0.1		
		6	0.1		

Table 3.7: Frequency of arrays within the responder group out of 10 trials.

Therefore, the issue of determining the number of trials necessary to provide enough confidence is raised. For instance, according to the results at time point A (see Table 3.6), the

patient ID:20 is more likely to be a responder than the patient ID:1, which is contrary to the knowledge available about the final results of the experiment.

The result of the *K-Mean* algorithm are represented either by heatmaps (see Fig. 3.7) or trendlines of the antibody profiles contained within a cluster plus the trendline of their mean values (see Fig. 3.8). Neither heatmaps nor trendlines allow a quantitative analysis of the clustering results in order to screen the outlier profiles within the clusters.

Figure 3.7: Heatmap of the non-responding patient samples at time point C when 14 and 5 where clustered as responder samples.

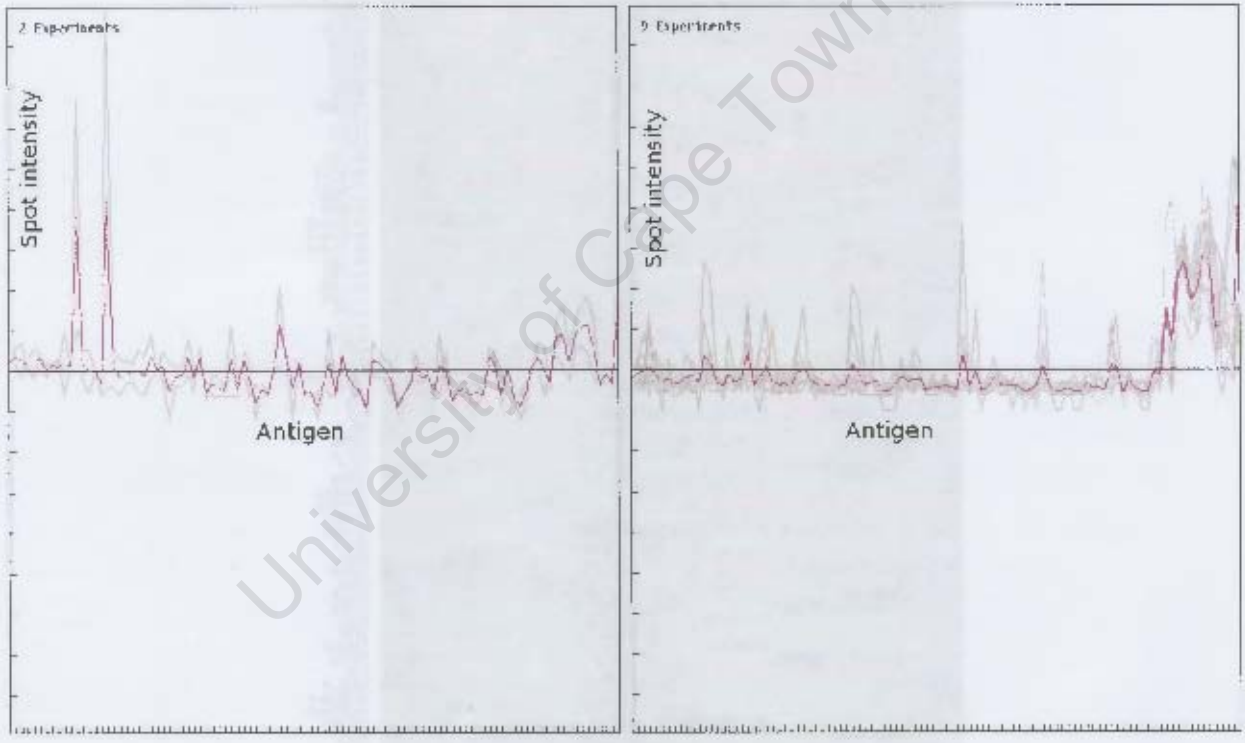


Figure 3.8: Trendlines of the 2 responding and the 9 non-responding patient samples at time point C for one trial of K-Mean clustering.

Chapter 4

Conclusion

This study highlighted the importance of lab protocols to reduce, from an early stage, irrelevant variations within the microarray datasets due to nonspecific binding, smears, artifacts or high background resulting from insufficient washing of the chips which can, individually or combined, compromize the validity of an experiment.

The study also underlined the deterministic role of the experiment design, and suggests the importance of the selection of specific controls, to clear or normalize the data from bias of different origins (such as the plasmids from which the recombinants were cloned) or even the systematical bias in the measurements. Through a simulation based on the Rocke and Durbin error model [29], the normalization method developed in this study, based on the assumption that the housekeeping spots share a common underlying distribution across the chips, was shown to perform better in roughly 71% of the cases than the methods founded on the assumption that the housekeeping spot should have the same mean expression across the chips. Our new method also demonstrated robustness and flexibility when dealing with flagged housekeeping spots.

In addition, this study addressed the issue of the qualitative clustering of the patient samples based on their antibody profiles. Indeed, as in the case of the CT100 array, the characterization of the antibody profiles against the cancer testis antigens are more qualitatively informative than the detection of antibodies against individual cancer testis antigens [18]. The factor analysis method allowed us through an unsupervised approach to cluster the samples and provided a quantitative measure saying by how much each

profile correlates to a given cluster. It also enabled the detection of the outlier profiles within the clusters. Compared to the *K-mean* equivalent using a Pearson correlation metric, the factor analysis is more straight forward, since it estimates itself the number of potential clusters and does not proceed by iterative steps. The factor analysis also provides reproducible results, which is not always the case with the *K-Mean*, where the clustering might depend on the initial cluster assignment - which is randomly generated - and a sufficient number of iterations.

However, the relatively limited number of control spots printed onto the custom arrays proved to be a challenge when finding the appropriate controls to generate a valuable normalization hypothesis. The particular case of antigen arrays assaying serum samples highlighted the fragility of the assumption of positive control serum independence, since serum compositions are highly patient dependent and antibody binding is not always antigen specific. Therefore, the future of the custom arrays relies on their design, which is intimately linked to the knowledge available on the research question being addressed in each particular experiment.

Bibliography

[1] Thongboonkerd V, Klein J: **Proteomics in Nephrology**. *Contrib Neprol. Basel* 2004, **141**:1–10.

[2] Gloerich J, Wevers R, Smeitink J, Engelen B, Heuvel L: **Proteomics Approaches to study Genetic and Metabolic Disorders**. *Proteome Research* 2007, **6**:506–512.

[3] Hardiman G: **Microarray Technologies - An Overview**. *Conference Scene* 2002.

[4] Draghici S: In *Data Analysis Tools for DNA Microarrays*, 2nd edition, Chapman & Hall/CRC 2003.

[5] Hall D, Ptacek J, Snyder M: **Protein Microarray Technology**. *Mech Ageing Dev* 2007, **128(1)**:161–167.

[6] Phizicky E, Bastiaens P, Zhu H, Snyder M, Fields S: **Protein analysis on a proteomic scale**. *Nature* 2003, **422:208-15**.

[7] Gray M, Colot H, Guarente L, Rosbash M: **Open reading frame cloning : Identification, cloning, and expression of open reading frame DNA**. *Proc Natl Acad Sci USA* 1982, **24**:6598–6602.

[8] MacBeath G, Schreiber S: **Printing Proteins as Microarrays for High-Throughput Function Determination**. *Science* 2000, **289**:1760–1762.

[9] Bussow K, Konthur Z, Lueking A, Lueking H, Walter G: **Protein Array Technology Potential Use in Medical Diagnostics**. *Pharmacogenomics* 2001, **1(1)**.

[10] Schweitzer B, Predki P, Snyder M: **Microarrays to characterize protein interactions on a whole-proteome scale**. *Proteomics* 2003, **3**:2190–2199.

[11] Ingvarsson J, Larsson A, Sjholm A, Truedsson L, Jansson B, Borrebaeck C, Wingren C: **Design of Recombinant Antibody Microarrays for Serum Protein Profiling: Targeting of Complement Proteins**. *Journal of Proteome Research* 2007, **6(9)**:3527–3536.

[12] Smyth G, Speed T: **Normalization of cDNA Microarray Data**. *Elsvier* 2003, **31**:265–273.

[13] Hultschig C, Kreutzberger J, Seitz H, Konthur Z, Bussow K, Lehrach H: **Recent advances of protein microarrays**. *Current Opinion in Chemical Biology* 2006, **10**:4–10.

[14] Steinhoff C, Vingron M: **Normalization and quantification of differental expression in gene expression microarrays**. *Briefings in Bioinformatics* 2006, **7**(2):166–177.

[15] Altman N: **Replication, Variation and Normalization in Micoarray Experiments**. *Applied Bioinformatics* 2005, **4(1)**:33–44.

[16] Causton H, Quackenbush J, Brazma A: In *Microarray Gene Expression Data Analysis. A beginner's Guide*, 1st edition, Blackwell Publishing 2004.

[17] Sanchez-Carbayo M: **Antibody Arrays: Technical Considerations and Clinical Applications in Cancer**. *Clinical Chemistry* 2006, **52**(9).

[18] Casiano CA, Mediavilla-Varela M, Tan EM: **Tumour-associated Antigen Arrays for the Serological Diagnosis of Cancer**. *Molecular and Cellular Proteomics* 2006, **5**:1745–1759.

[19] Zhu X, Gerstein M, Snyder M: **ProCAT: a data analysis approach for protein microarrays**. *Genome Biology* 2006, **7**(R110).

[20] Robinson W: **Antigen arrays for antibody profiling**. *Current Opinion in Chemical Biology* 2006, **10**:67–72.

[21] Tecan LS Series Laser Scanner: *How to Set the Correct Gain in the LS Scanner.* [`http://www.tecan.com`].

[22] Lu T, Costello C, Croucher P, Hasler R, Deuschl G, Schreiber S: **Can Zipf's law be adapted to normalize microarray ?** *BMC Bioinformatics* 2005, **6**(37).

[23] Oshlack A, Emslie D, Corcoran L, Smyth G: **Normalization of boutique two-colour microarrays with a high proportion of differentially expressed probes.** *Genome Biology* 2007, **8**(2).

[24] Freudenberg J: **Comparison of background correction and normalization procedures for high-density oligonucleotide microarrays.** *PhD thesis*, Universitat Leipzig 2004.

[25] Bolstad B, Irizarry R, Astrand M, Speed T: **A comparison of normalization methods for high density oligonucleotide array data based on variance and bias.** *Bioinformatics* 2002, **19**(2):185–193.

[26] Ploner A, Miller L, Hall P, Bergh J, Pawitan Y: **Correlation test to assess low-level processing of high-density oligonucleotide microarray data.** *BMC Bioinformatics* 2005, **6**(80).

[27] Wilson D, Buckley M, Helliwell C, Wilson I: **New nomalization methods for cDNA microarray data.** *Bioinformatics* 2002, **19**(11):1325–1332.

[28] Quackenbush J: **Comptational Analysis of Microarray Data.** *Macmillan Magazines* 2001, **2**:418–427.

[29] Rocke D, Durbin B: **A model for measurement error for gene expression arrays.** *Journal of Computational Biology* 2001, **8**:557–569.

[30] Motakis ES, Nason G, Fryzlewicz P, Rutter G: **Variance stabilization and normalization for one-colour microarray data using a data-driven multiscale approach.** *Bioinformatics* 2006, **22**(20):2547–2553.

[31] Hastie T, Tibshirani R, Friedman J: In *The Elements of Statistical Learning*, 1st edition, Springer 2001:437–508.

[32] Boutros P, Okey A: **Unsupervised pattern recognition : An introduction to the whys and wherefores of clustering microarray data.** *Briefings in Bioinformatics* 2005, **6**(4):331–343.

[33] Costello A, Osborne J: **Best Practices in Exploratory Factor Analysis : Four Recomendations for Getting the Most From Your Analysis**. *Practical Assessment, Research & Evaluation* 2005, **10**(7).

[34] Wikepidea: **Factor analysis in psychometrics.** 2010, [http://en.wikipedia.org/wiki/Factor_analysis].

[35] Tryfos P: **Notes on Factor Analysis** 2010, [http://www.yorku.ca/ptryfos/f1400.pdf].

[36] Wikepidea: **Standard score.** 2010, [http://en.wikipedia.org/wiki/Standard_score].

[37] StatSoft: **Principal Components and Factor Analysis.** 2010, [http://www.statsoft.com/textbook/principal-components-factor-analysis].

[38] Zhang M, Therneau T, McKenzie M, Li P, Yang P: **A Fuzzy C-Means Algorithm Using a Correlation Metrics and Gene Ontology**. *IEEE* 2008, :1–4.

Appendix A

Filtering and normalization pipeline

1. Pipeline components

The pipeline has been implemented in *python* and is running on the command line. It is composed of a module library (*custom_function.py*) and the main program (*run.py*) briefly described in table A.1.

custom_functionpy	A library containing the different modules described in the work flow (see fig. A.1), allowing modularity of the program.
run.py	The main program: 1. Edit the file *settings.txt* to allow the user to provide the settings; 2. process all the pipeline modules in the right sequence; 3. Command : *python run.py file(s)* (ex : the command *python run.py data/ * .txt* normalize all the files with *txt* extension contained in the directory data/). The results are outputed in the directory *OUTPUT*.

Table A.1: Pipeline components

The input dataset should start with the header line, and the two first columns should contain the annotation and name of the Cancer Testis antigens. Further details on the pipeline modules are given in section 2.

2. Work flow description

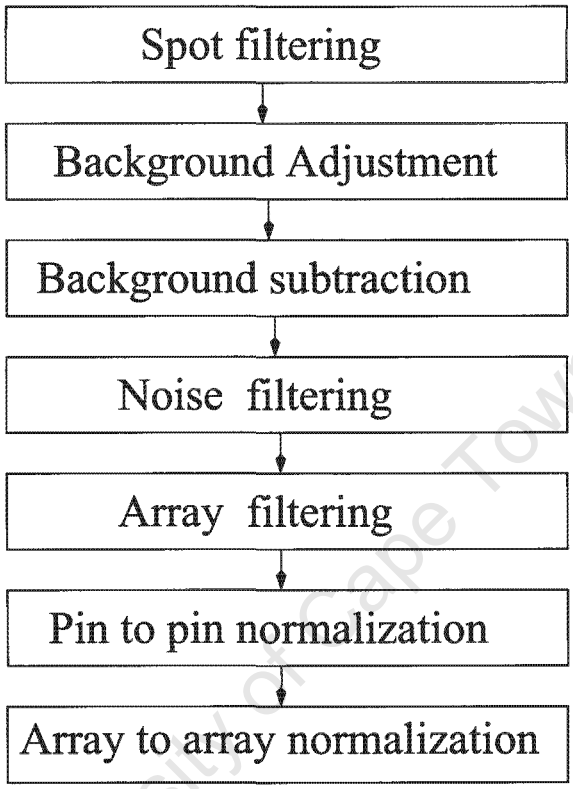


Figure A.1: Pipeline work flow

The different modules of the pipeline work flow are further detailed in table A.2.

SPOT FILTERING	OUTPUT directory and Function
- Spot area percentage - Spot saturation level	Directory : OUTPUT/FLAGGED_FILES Function : flagging(RAW_INT_1, SPOT_AR_PCTG, AREA_THRSLD, SATURATION, FILES)
BACKGROUND ADJUSTMENT	OUTPUT directory and Function
Replace the background value by the median of the background in the neighborhood.	Directory : OUTPUT/BCKGD_ADJUSTED Function : bckgd_adjust(GRID, ROW, COLUMN, BCKGD_INT_1, FILES)
BACKGROUND SUBTRACTION	OUTPUT directory and Function
Subtract the value of the corrected background from the signal of each particular spot	Directory : OUTPUT/BCKGD_SUBTRACTED Function : bckgd_substract(ANNOTATION,NAME, GRID, ROW, COLUMN, RAW_INT_1, BCKGD_INT_1,BCKGD_STDEV_1,FILES)
NOISE FILTRATION	OUTPUT directory and Function
Set to zero all the signals lower than 2 standard deviation of the background (since low intensities are often the background intensity residues of detecting antibodies still present on the slide after washing)	Directory : OUTPUT/BCKGD_SIGNAL_2_ZERO Function : set_bckg2zero(FILES)
ARRAY FILTERING	OUTPUT directory and Function
- Number of flagged positive controls for pin-to-pin normalization; - Number of remaining controls for array to array normalization. - CV of the positive controls on the array before pin-to-pin normalization.	Directory : OUTPUT/DISCARDED_FILES Function : discard_array(CV_THRSLD, FILES)
PIN to PIN NORMALIZATION	OUTPUT directory and Function
- To deal with the few and variable positive controls, a method was developed based on the total intensities of positives controls. - Hypothesis : The positive controls (housekeeping) would rather share a common distribution across the chips than the same intensities. - Method based on : Quantile normalization (Bolstad 2002) and total intensity used in cDNA microarrays.	Directory : OUTPUT/P2P_NORMALIZED_DATA Function : pin2pin(FILES)
ARRAY to ARRAY NORMALIZATION	OUTPUT directory and Function
Same approach as pin to pin except that the controls are taken across the arrays.	Directory : OUTPUT/A2A_NORMALIZED_DATA Function : array2array(FILES,A2A_CTRL).

Table A.2: Module descriptions

Appendix B

Cancer Testis antigen annotations.

Number	Annotation	Name
1	antigen 001	BAGE2
2	antigen 002	BAGE3
3	antigen 003	BAGE4
4	antigen 004	BAGE5
5	antigen 005	CCDC33
6	antigen 006	CEP290
7	antigen 007	COL6A1
8	antigen 008	COX6B2
9	antigen 009	CSAG2
10	antigen 010	CT47.11
11	antigen 011	CT62
12	antigen 012	CTAG2
13	antigen 013	CXorf48.1
14	antigen 014	DDX53
15	antigen 015	DSCR8/MMA1
16	antigen 016	FTHL17
17	antigen 017	GAGE1
18	antigen 018	GAGE2A
19	antigen 019	GAGE4
20	antigen 020	GAGE5
21	antigen 021	GAGE6
22	antigen 022	GAGE7
23	antigen 023	GRWD1
24	antigen 024	HORMAD1

Table B.1: Cancer Tesis antigens 1 to 24 and their annotations.

Number	Annotation	Name
25	antigen 025	LDHC
26	antigen 026	LEMD1
27	antigen 027	LIP1
28	antigen 028	MAGEA1
29	antigen 029	MAGEA10
30	antigen 030	MAGEA11
31	antigen 031	MAGEA2
32	antigen 032	MAGEA3
33	antigen 033	MAGEA4v2
34	antigen 034	MAGEA4v3
35	antigen 035	MAGEA4v4
36	antigen 036	MAGEA5
37	antigen 037	MAGEB1
38	antigen 038	MAGEB5
39	antigen 039	MAGEB6
40	antigen 040	MART1
41	antigen 041	MICA
42	antigen 042	NLRP4
43	antigen 043	NXF2
44	antigen 044	NYCO45
45	antigen 045	NY-ESO-1
46	antigen 046	OIP5
47	antigen 047	p53
48	antigen 048	PBK
49	antigen 049	RELT
50	antigen 050	ROPN1
51	antigen 051	SGY-1
52	antigen 052	SILV
53	antigen 053	SPAG9
54	antigen 054	SPANXA1
55	antigen 055	SPANXB1
56	antigen 056	SPANXC
57	antigen 057	SPANXD
58	antigen 058	SPO11
59	antigen 059	SSX1
60	antigen 060	SSX2A
61	antigen 061	SSX4
62	antigen 062	SYCE1
63	antigen 063	SYCP1
64	antigen 064	THEG
65	antigen 065	TPTE

Table B.2: Cancer Tesis antigens 25 to 65 and their annotations

Number	Annotation	Name
66	antigen 066	TSGA10
67	antigen 067	TSSK6
68	antigen 068	TYR
69	antigen 069	XAGE-2
70	antigen 070	XAGE3av1
71	antigen 071	XAGE3av2
72	antigen 072	ZNF165
73	antigen 073	AKT1
74	antigen 074	CDK2
75	antigen 075	CDK4
76	antigen 076	CDK7
77	antigen 077	FES
78	antigen 078	FGFR2
79	antigen 079	MAPK1
80	antigen 080	MAPK3
81	antigen 081	PRKCZ
82	antigen 082	RAF
83	antigen 083	SRC
84	antigen 084	CALM1
85	antigen 085	CDC25A
86	antigen 086	CREB1
87	antigen 087	CTNNB1
88	antigen 088	p53 S6A
89	antigen 089	p53 C141Y
90	antigen 090	p53 S15A
91	antigen 091	P53 T18A
92	antigen 092	p53 Q136x
93	antigen 093	p53 S46A
94	antigen 094	p53 K382R
95	antigen 095	p53 S392A
96	antigen 096	p53 M133T
97	antigen 097	p53 L344P
98	antigen 098	cytochrome P450 3A4
99	antigen 099	cytochrome P450 reductase
100	antigen 100	EGFR

Table B.3: Cancer Tesis antigens 66 to 100 and their annotations