

Bioinformatic analysis of peptide microarray immunoassay data for serological diagnosis of infectious diseases

Kate Zoe Nambiar

A thesis submitted in partial fulfillment of the requirements of the
University of Brighton and the University of Sussex
for the degree of Doctor of Philosophy

April 2017

Abstract

Understanding antibody - antigen interactions occurring in infectious diseases is important in understanding aetiology, can help facilitate diagnosis, and could offer potential targets for vaccine or therapeutic antibody development. Peptide arrays – collections of short peptides immobilised on solid planar supports – offer a high throughput and highly parallel method of identifying immunogenic epitopes and relating patterns of antibody identification to clinical disease states. As technology advances, so the density and complexity of peptide arrays of becomes ever higher. Managing the large volume of data that modern high density microarrays generate requires sophisticated bioinformatics in order to minimise errors and biases.

In this thesis I introduce a new software package, `pmpa`, that uses R, the open source statistical programming platform and an object orientated framework from the Bioconductor project. The package facilitates analysis of peptide microarray data including functions for reading scanned data files, quality assessment and pre-processing. It is both flexible and modular – integrating with existing software in the Bioconductor repository.

Data pre-processing is key to any microarray analysis. Noise due to technical variation can obscure true biological effects if careful steps are not taken. The aim of pre-processing is to minimise noise while preserving biological variation. No consensus exists as to the optimal method of pre-processing making comparison between studies difficult. This thesis explores two key aspects of pre-processing: background correction and normalisation using two experimental datasets – a titration series of a monoclonal anti *C.difficile* Toxin B monoclonal antibody, and dataset with an anti-Toxin A antibody spiked into non immune sera to examine biases introduced by the pre-processing and whether they improve measures such as precision and differential identification.

Finally the analysis method is applied to two studies identifying antibody signatures in infectious diseases: the first investigating immune responses to *C. difficile* – a major hospital acquired infection and the leading identifiable cause of antibiotic associated diarrhoea, and the second characterising antibody signatures that define paediatric tuberculosis infection. The real world application of the methodology identifies signatures of immune responses characterising clinical disease eg. relapsing vs. single episode *C. difficile* infection, but also highlights a number of limitations of the technique such as batch confounding and response variability.

Contents

1	Introduction	1
1.1	Antibody - Antigen Interaction	4
1.2	Epitope Mapping	5
1.2.1	In silico epitope prediction	8
1.3	Peptide Arrays	9
1.3.1	Peptide array ligand binding detection	12
1.4	Analysis of microarray data	14
1.4.1	Image processing	14
1.4.2	Quality assessment	18
1.4.3	Data pre-processing	19
1.5	Applications of peptide microarray immunoassays	20
1.6	Thesis outline	22
2	PMPA: Peptide Microarray Pre-Processing Analysis. An R package for peptide microarray data	24
2.1	Introduction	24
2.2	The R Statistical Programming Environment	25
2.2.1	The Bioconductor Project	25
2.2.2	Object-Oriented Programming	26
2.3	Installing PMPA	27
2.4	Processing spotted peptide microarray data using PMPA	28
2.4.1	Reading Axon Text File (ATF) format data into PMPA	28
2.4.2	Representation of data in PMPA	29
2.4.3	Data annotation	31
2.4.4	Accessor methods	32
2.4.5	Data quality assessment and visualisation	33
2.5	Pre-Processing peptide array data	37
2.5.1	Background correction and transformation	37
2.5.2	Normalisation	39
2.5.3	Summarisation	39
2.6	Analysis of pre-processed data	40
2.7	Example pre-processing script using pmpa	40
2.8	Conclusions and future work	42

3	Background correction for peptide microarrays	43
3.1	Introduction	43
3.2	Estimating background signal	44
3.3	Background Correction Methods	46
3.3.1	Subtraction	46
3.3.2	Offset subtraction	47
3.3.3	Edwards correction	47
3.3.4	Signal to noise ratio correction	48
3.3.5	The normal-exponential convolutional model (Normexp)	48
3.4	Aim	49
3.5	Methods	49
3.5.1	Peptide microarray processing	49
3.5.2	Image acquisition and processing	51
3.5.3	Data pre-processing	51
3.6	Results	51
3.6.1	Assessing variation induced by background correction.	51
3.6.2	Assessment of precision	52
3.6.3	Assessment of differential identification	55
3.7	Discussion	60
4	Normalisation of peptide microarray data	62
4.1	Introduction	62
4.2	Normalisation Methods	65
4.2.1	Control Probe Normalisation	66
4.2.2	Global Normalisation	67
4.3	Aim	69
4.4	Methods	69
4.4.1	Data pre-processing	70
4.5	Results	70
4.5.1	Assessment of variation induced by normalisation	70
4.5.2	Assessment of precision and bias	70
4.5.3	Assessment of differential identification	72
4.6	Discussion	74
5	Identification of antibody signatures characterising <i>Clostridium difficile</i> infection	76
5.1	Introduction	76
5.2	Microbiology and clinical features	77
5.2.1	Recurrent <i>C. difficile</i> infection	78
5.3	<i>C. difficile</i> pathogenesis	78
5.3.1	<i>C. difficile</i> toxins (TcdA and TcdB)	80
5.3.2	<i>C. difficile</i> binary toxin (CDT)	82
5.4	The immune response to <i>C. difficile</i>	84

5.4.1	The adaptive immune response to <i>C. difficile</i> toxins . . .	84
5.4.2	The adaptive immune response to non-toxin antigens . .	85
5.5	Aims	86
5.6	Methods	87
5.6.1	Study participants and sampling – Brighton	87
5.6.2	Study participants and sampling – Liverpool	87
5.6.3	Microarray Design	88
5.6.4	Peptide synthesis and microarray printing	89
5.6.5	Microarray Processing	89
5.6.6	Peptide microarray scanning and data extraction	89
5.6.7	Data pre-processing	90
5.6.8	Data analysis	90
5.7	Results	91
5.7.1	Assessment of the sequence conservation of the repre- sented <i>C. difficile</i> peptides	91
5.7.2	Exploratory data analysis – Brighton dataset	91
5.7.3	Differential identification – Brighton dataset	93
5.7.4	Batch effect – Brighton dataset reanalysis	95
5.7.5	Demographics and clinical characteristics – Liverpool Dataset	95
5.7.6	Differential reactivity of sera from <i>Clostridium difficile</i> Infection (CDI) and Control groups	95
5.8	Discussion	99
6	Antibody signatures characterising paediatric tuberculosis	101
6.1	Diagnosis of tuberculosis	102
6.1.1	Serological diagnosis of tuberculosis (TB)	103
6.1.2	Serological diagnosis of tuberculosis in children	104
6.1.3	Evaluation of serodiagnostic antigens	106
6.2	Methods	107
6.2.1	Clinical cohort	107
6.2.2	Case Definitions	108
6.2.3	Sample collection and storage	108
6.2.4	Peptide microarray design	108
6.2.5	Microarray Processing	111
6.2.6	Data pre-processing	111
6.2.7	Data Analysis	111
6.3	Results	113
6.3.1	Training Set Description	113
6.3.2	Feature Selection	115
6.3.3	Classification and Cross-Validation	116
6.3.4	Test Set Description	116
6.3.5	Comparison of training and test sets	122

6.3.6	Model validation using the test set	122
6.3.7	Post Hoc Analyses	122
6.4	Discussion	123
7	Conclusions	127
	References	131
	Appendices	152

List of Figures

1.1	Ribbon diagram of Immunoglobulin G (IgG) structure	3
1.2	Cartoon diagram of Immunoglobulin G (IgG) structure	4
1.3	Antibody binding for discontinuous and continuous epitopes . .	6
1.4	Schematic diagram of peptide array ligand binding detection methods	12
1.5	Overview of the peptide microarray immunoassay procedure . .	13
1.6	Overview of typical peptide microarray analysis workflow	15
1.7	Microarray image processing	17
2.1	PMPA package data representation schematic	30
2.2	Block (print-tip) boxplots from two peptide microarrays	34
2.3	Foreground and background image plots	36
2.4	PMPA quality assessment plots - arrayQApilot	38
3.1	Background image plots and foreground / background scatter plots	45
3.2	Background image plots for two arrays incubated with identical sera	52
3.3	MA-plots of background correction methods for a self-self comparison	53
3.4	Boxplots of log residual variances of quadratic model fits and log fold change compared to negative control array	55
3.5	Effect of offset values on log residual variances and log fold change	56
3.6	Multiple sequence alignment of the PA1-85042 polyclonal antibody (pAb) antibody immunogen to array peptides	58
3.7	ROC plots for differential identification of PA1 binding peptides from antibody spiked into normal rabbit serum	59
4.1	MA-plots of normalisation methods for a self-self comparison . .	71
4.2	Boxplots of log residual variances of quadratic model fits and log fold change compared to negative control array	72
4.3	ROC plots for differential identification of PA1 binding peptides from antibody spiked into normal rabbit serum	73

5.1	Primary domain structure of <i>Clostridium difficile</i> toxins A (TcdA) and B (TcdB)	79
5.2	The <i>C.difficile</i> Pathogenicity Locus (PaLoc)	81
5.3	The <i>C.difficile</i> Binary Toxin Locus	83
5.4	Exploratory data analysis of Brighton <i>C. difficile</i> dataset – Partition around medoids (PAM) clustering	94
5.5	Reanalysis of Brighton dataset – Partition around medoids (PAM) clustering	96
5.6	Differential Identification – Liverpool CDI dataset	98
6.1	MTB protein sequences in relation to H37Rv genome and regions of difference (RD).	109
6.2	MTB peptide reactivity measured by Z-score (training set) . . .	118
6.3	Heatmap of selected peptides after Z-score and ROC filtering . .	119
6.4	MTB peptide reactivity measured by Z-score (test set)	124
6.5	Heatmap of selected peptides after Z-score and ROC filtering . .	125

List of Tables

3.1	AUROC and 95% confidence intervals for PA ₁ spike-in series . .	57
4.1	AUROC and 95% confidence intervals for PA ₁ spike-in series (Normalisation)	74
5.1	<i>C. difficile</i> proteins used for peptide microarray	88
5.2	Summary of pairwise sequence alignments of Strain 630 and Strain R20291 proteins with publically available sequences from the Uniprot / TrEMBL database.	92
5.3	Summary descriptive statistics for the Brighton CDI dataset . . .	93
5.4	Summary descriptive statistics for the Liverpool CDI dataset . .	97
5.5	Peptide differential identification in recurrent CDI (rCDI) . . .	97
6.1	<i>Mycobacterium tuberculosis</i> proteins used for peptide array immunoassay	110
6.2	Summary descriptive statistics for the training set	114
6.3	MTB peptides selected after Z-score and ROC filtering	117
6.4	Training errors of classification algorithms applied to the training dataset	120
6.5	Cross-validation errors of classification algorithms applied to the training dataset	120
6.6	Summary descriptive statistics for the test set	121

List of Abbreviations

ATF Axon Text File.

AUROC Area Under ROC Curve.

CDI *Clostridium difficile* Infection.

cDNA Complementary Deoxyribonucleic Acid.

CDR Complementarity Determining Region.

CV Coefficient of Variation.

DNA Deoxyribonucleic Acid.

EBV Epstein Barr Virus.

ELISA Enzyme Linked Immunosorbent Assay.

FliC Flagellin C.

FliD Flagellar Cap Protein.

GAL GenePix® Array List.

GAPDH Glyceraldehyde 3-Phosphate Dehydrogenase.

GFP Green Fluorescent Protein.

GMM General Mixture Model File.

GPR GenePix® Results File.

HIV Human Immunodeficiency Virus.

HPLC High-Performance Liquid Chromatography.

IGRA Interferon Gamma Release Assay.

KLH Keyhole Limpet Haemocyanin.

LCT Large Clostridial Toxin.

mAb Monoclonal Antibody.

MIAME Mimimum Information About a Microarray Experiment.

mRNA Messenger Ribonucleic Acid.

MTB Mycobacterium tuberculosis.

NAAT Nucleic Acid Amplification Test.

NMR Nuclear Magnetic Resonance.

OOP Object-Orientated Programming.

pAb Polyclonal Antibody.

PaLoc Pathogenicity Locus.

PAM Partition Around Mediods.

PCR Polymerase Chain Reaction.

PLAC-1 Placental Specific Protein 1.

PPD Purified Protein Derivative.

QA Quality Assessment.

qPCR Quantitative Polymerase Chain Reaction.

RMA Robust Multi-Array Average.

RNA Ribonucleic Acid.

ROC Receiver Operator Characteristics.

rRNA Ribosomal Ribonucleic Acid.

SNR Signal to Noise Ratio.

SPPS Solid-Phase Peptide Synthesis.

SPR Surface Plasmon Resonance.

SVM Support Vector Machine.

TB Tuberculosis.

TcdA Toxin A.

TcdB Toxin B.

TIFF Tagged Image Format File.

TST Tuberculin Skin Testing.

VSN Variance Stabilisation Normalisation.

WAZ Weight Age Z-Score.

WHO World Health Organisation.

Acknowledgements

There are so many people that I want to thank for their help and inspiration during the long journey completing this PhD. I have had the good fortune to have been surrounded by so many incredibly talented and supportive individuals without whom I would have surely struggled.

First and foremost, I must thank my supervisors, Dr. Martin Llewelyn and Prof. Florian Kern for their guidance and untiring patience. Thank you to Dr. Chris Finan for introducing me to the strange world of R and for setting me firmly on the path to bioinformatics geekery! In the lab I must thank Matt Pope for all your help with the peptide arrays – you will always be the Beaker to my Bunsen! Meep!! A huge thanks must go to Dr. Natalie Chaplin who not only helped me to set up the lab in CIRU but who has become a very dear friend whose support has been invaluable. Thank you to Dr. Jasmin Islam for all her help with the clinical aspects of the study.

The *C.difficile* and tuberculosis studies would not have been possible without the participation of many patients and their families; I will always be grateful to them for taking part in the research. I am extremely grateful as well to Prof. Brian Eley and the staff of the Red Cross War Memorial Hospital, Cape Town, South Africa for their assistance.

Finally, I could not have completed this PhD without the help of my friends and family. In particular I want to thank Karyn Chapell and Zoë Brooke who have unwavering in their support and belief in me. I don't tell you enough how fantastic you are! Zoë in particular I must thank for proof-reading several chapters of the thesis, for all your encouragement and for being there when I needed someone to chat to. Dude, I raise this Crumpet of Courage to you! I want to thank Ciara who has shared more of my PhD journey than anyone else – I know I haven't exactly been easy to live with but I am grateful for your patience and understanding. I want to thank my parents for always believing in me even when I didn't think I could finish. Dad, I wish you could be here to see me complete this – I know you would have been proud of me.

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to these or any other university for a degree, and does not incorporate any material already submitted for a degree.

Signature :

Date :

Chapter 1

Introduction

Infectious diseases are one of the leading causes of mortality and morbidity worldwide. In 2012 the World Health Organisation (WHO) estimated that just over 12.5 million deaths (24% of all deaths) worldwide were attributable directly to one or more infectious diseases (World Health Organization (WHO), 2015). In the developing world the mortality burden is even higher, with more than half of all reported deaths being due to infection. Early and accurate diagnostic techniques are not only important in reducing mortality and morbidity, but also in preventing onward transmission of the infectious agent, screening of asymptomatic patients, surveillance of disease prevalence and evaluating the efficacy of treatment. As a result there is now an increasing interest in finding novel biomarkers that can be used for the specific diagnosis of infectious diseases.

The term biomarker is used to describe a measurable indicator of a disease or physiological state. However, this is an extremely broad definition. For example, blood pressure can be considered as a physical biomarker for cardiovascular disease, and C-Reactive Protein is a biomarker for inflammation. However, traditional markers such as these suffer from an inherent lack of specificity and so much of the focus of current biomarker research is aimed at finding biological macromolecules (e.g. Nucleic acids and proteins) that hold the promise of detecting specific diseases (Baker, 2005). In infectious diseases biomarkers can originate either from the infecting pathogen itself or from the host's response. Although pathogen derived markers may be highly specific, they can be difficult to detect as they may be tissue specific and present in low concentration especially

in the early stages of disease. In contrast, the host immune response is capable of amplifying the signal of infection and providing it in a form that can be readily detected in the circulation in the form of antigen specific cellular markers or antibodies.

Antibodies are particularly attractive as potential biomarkers. They are produced in high concentrations by the immune system and specifically target molecules expressed by the pathogen. The antibody response is incredibly diverse; the immune system is capable of generating a vast array of antibodies that can target an enormous number of targets. Yet these antibodies are capable of binding their target antigens with high affinity. Moreover, although they possess such diversity the antibody molecule is, for the most part, highly conserved with common regions that are important for their effector function. Antibodies are also highly stable over time, particularly in serum, facilitating testing of stored and historical samples (af Geijersstam et al., 1998).

The structure of antibody proteins, known as immunoglobulins, was first discovered by Gerald Edelman and Rodney Porter in the 1960s earning them the Nobel Prize for medicine in 1972. Immunoglobulins are glycoproteins with each monomer having a molecular weight of 150kDa. Immunoglobulin can be cell surface associated as a B-lymphocyte receptor or secreted in mono, di, tetra or pentameric forms. Each monomer is comprised of 4 polypeptide subunits – 2 identical heavy chains and 2 identical light chains. In mammals there are 5 classes of heavy chain designated γ , δ , α , μ and ϵ . These define the immunoglobulin classes IgG, IgD, IgA, IgM and IgE respectively. There are 2 classes of light chain, designated κ and λ . Each light chain is comprised of an N-terminal variable domain (V_L) and a constant domain (C_L). Similarly the heavy chains are comprised of an N-terminal variable domain (V_H), a ‘hinge’ region and 3 or 4 constant domains (C_{H1} to C_{H3} or C_{H4}). The protein is assembled with each light chain being associated with the V_H and C_{H1} domains of one heavy chain, and with the remaining heavy chain C_H domains being associated together. Thus the protein is ‘Y’ shaped with each N-terminal end at the top tips of the Y and the C-terminals at the base (Figure 1.1). Antigen binding occurs at the N-terminal ends where the V_H and V_L domains are brought together. It is the variability of these regions that allow the molecule to recognise such a diverse range of targets.

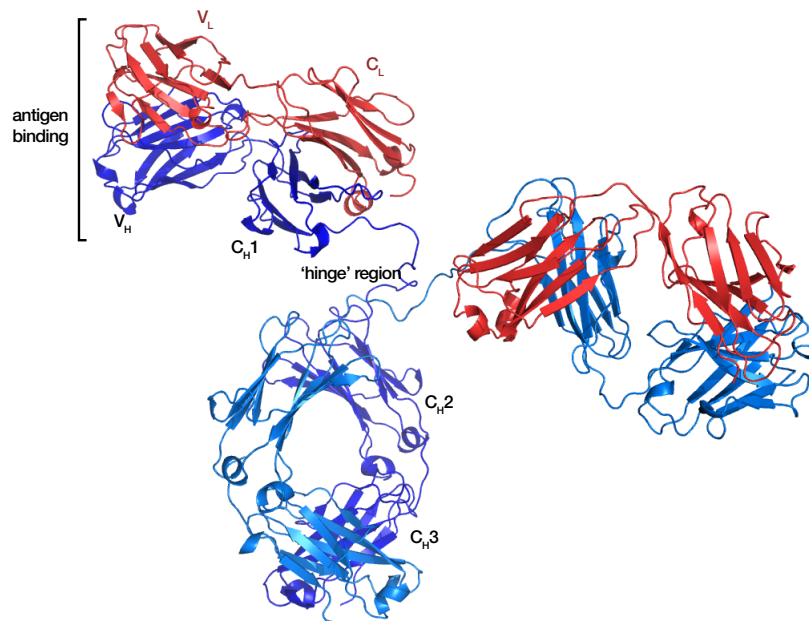


Figure 1.1: Ribbon diagram of Immunoglobulin G (IgG) Structure

IgG has a tetrameric structure comprising two identical heavy chains (shown above in blue and two light chains (shown in red). The two heavy chains are linked to each other and to a corresponding light chain by disulphide bonds to form a Y shaped structure.

The genetic basis for the diversity seen in immunoglobulins was, for several decades, a puzzle to immunologists. The problem was that, on the one hand a single individual could produce antibodies capable of recognising a huge range of antigenic targets, yet each of the antibodies from a particular class was serologically identifiable in a manner suggesting a single mendelian trait (Tonegawa, 1983). It was originally believed that each antibody variable (V) region was encoded in the germline and that B-lymphocytes (the immune cells that are responsible for antibody production) express only one of those genes. However, given the sheer number of specificities a huge amount of the genome would have had to be given over just to encoding those domains.

A competing theory then emerged suggesting that there was only one gene for each V region encoded in the germline and that the variation seen was a result of a process of somatic mutation. We know today that the genetic basis of antibody diversity takes elements of both theories; each immunoglobulin V region is encoded by a number of gene segments in the germline, but these segments undergo somatic modification in a number of different ways to produce the

eventual protein product. As a result of these processes the theoretical naïve antibody repertoire has been calculated to span between 10^7 to 10^{10} different specificities (Berek and Milstein, 1988; Cohn and Langman, 1990); experimental evidence points to an extremely diverse repertoire of at least 10^9 IgG antibody species (Nobrega et al., 1998).

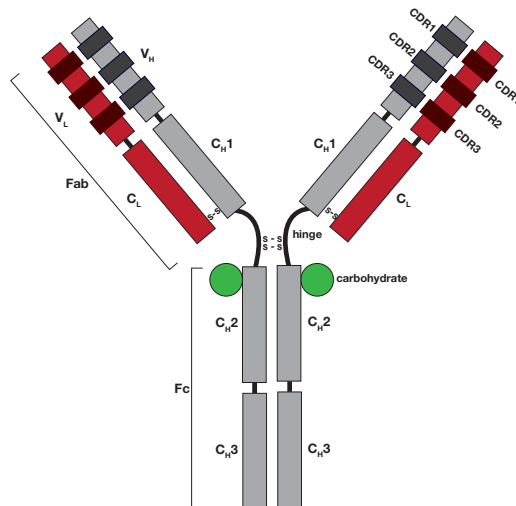


Figure 1.2: Cartoon diagram of Immunoglobulin G (IgG) structure

The heavy and light chains that make up immunoglobulin are comprised of a single variable (V) domain located at the N-terminal at the tip of each 'arm' of the molecule, and a series of constant (C) domains – 3 for the heavy chain and 1 for the light chain. The variable domains are then comprised of three regions of hypervariability known as the complementarity determining regions (CDRs) where antigen binding occurs.

1.1 Antibody - Antigen Interaction

The immunoglobulin variable (V) regions, although differing from one antibody molecule to another, concentrate their sequence variability in three short hypervariable regions. In between these are four lower variability regions. Structurally the conserved regions correspond to the beta sheets within the V domain whereas the hypervariable regions are three loops on the outer edge of the beta barrel that are brought together in the folded structure of the protein. The antigen binding region of immunoglobulin is formed by the juxtaposition of the hypervariable regions – also known as complementarity determining regions (CDRs), in the V_H and V_L domains (Figure 1.2).

ORIGINAL IN COLOUR

The surface formed by the six CDRs (three from each of V_H and V_L) forms a complementary shape upon which antigen can bind. Binding of antibodies to antigens occurs by a number of non-covalent interactions – electrostatic forces between polar side chains, hydrogen bonding between electronegative atoms, Van der Waals forces and hydrophobic forces. The contribution of each of these forces to the binding interaction varies between different antibodies and antigens but for the most part, short range forces (Van der Waals and hydrophobic forces) predominate necessitating a close fit in the shape of the antigen to the antibody binding site. The interaction is then strengthened by additional electrostatic and hydrogen bonds between specific residues in the binding site.

The majority of antibodies are directed against large biological macromolecules – most commonly proteins but also some polysaccharides and lipids. Antibodies typically recognise a small portion of the antigen known as an epitope or antigenic determinant. The majority of these epitopes are comprised of non-contiguous amino acids that are brought together when the protein is folded into its native conformation – known as discontinuous or conformational epitopes (Barlow et al., 1986). Epitopes can also be part of a continuous sequence from the protein's primary structure sometimes called a linear epitope (Figure 1.3). The term 'linear' epitope is rather misleading as contiguous sequences within a folded protein also occupy a specific conformation which may be constrained by regions distant to it. In essence *all* epitopes are conformational but some may be continuous within the primary sequence and others discontinuous (Barlow et al., 1986). For most proteins an epitope may comprise only 12-15 amino acids (Kringelum et al., 2013). However, only 5 – 7 of the amino acids making up the epitope dominate the binding energy of the antigen interaction. Changes in these residues greatly affect the binding affinity – much more so than changes in the remaining residues.

1.2 Epitope Mapping

Localising epitopes, a process known as epitope mapping, is important in guiding vaccine and therapeutic and diagnostic antibody development. X-ray crystallography of the antigen-antibody complex is generally considered to be the

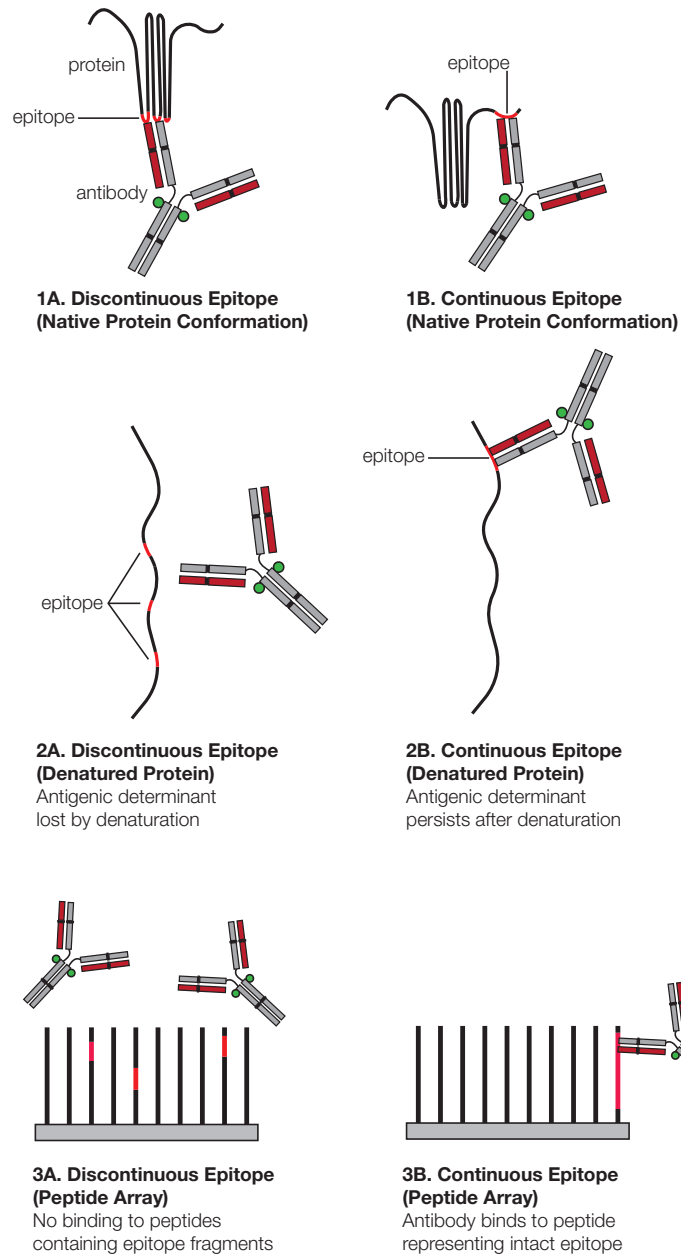


Figure 1.3: Antibody binding for discontinuous and continuous epitopes

Cartoon representations of antibody binding to a conformational and linear epitope (1A and 1B). Denaturation of the protein leads to loss of binding for conformational epitopes (2A) whereas linear epitopes retain binding activity even if the protein native conformation is lost (2B). If the protein primary structure is parsed into peptide fragments many conformational epitopes will not be identified as the constituent parts of the epitope are too short to be identified (3A). By contrast linear epitopes will retain binding activity if the epitope sequence is represented as a peptide (3B).

ORIGINAL IN COLOUR

‘gold standard’ for characterising epitopes (Saul and Alzari, 1996). It allows definition of the molecular interactions and key binding residues in the complex and gives precise 3-dimensional structural data about the epitope. However, X-ray crystallography requires the production of highly pure proteins and their co-crystallisation can be extremely difficult and time consuming. An alternative structural method of epitope mapping involves multidimensional nuclear magnetic resonance (NMR) spectroscopy (Rosen and Anglister, 2009). Similar levels of structural resolution to X-ray crystallography can be achieved with antibody-antigen complexes in solution thereby avoiding the need for crystallisation (Bardelli et al., 2015). Unfortunately NMR epitope mapping suffers with problems of increasing spectral overlap and decreasing signal attenuation as the antigen size increases. In practical term this limits its usefulness only to small protein or peptide antigens.

Because of the difficulty in applying structural methods such as X-ray crystallography or NMR spectroscopy to epitope mapping, a considerable amount of interest has been directed to ‘functional’ methods involving antigen modification, antigen fragments or synthetic peptides. Antigen modification methods rely on the production of altered proteins either by site directed mutagenesis (Benjamin and Perdue, 1996) or by direct chemical modification. These altered forms are compared to native antigen to identify key epitope residues.

An antigen fragmentation approach relies on exploiting binding between identifiable antigen fragments and antibody to identify epitopes. A simple example of this could involve partial protease digestion or chemical cleavage of the antigen followed by Western blotting or high-performance liquid chromatography (HPLC). Fragments binding antibody can then be identified by N-terminal protein microsequencing or by mass spectrometry. More commonly though, fragmentation techniques exploit expression of overlapping antigenic fragments from recombinant DNA to create ‘epitope libraries’. These antigenic fragments can be expressed on the surface of filamentous bacteriophages and then probed for antigen binding – so called ‘phage display’ (McCafferty et al., 1990; Smith, 1985). After multiple rounds of target binding, elution and reinfection in *E. coli* (a process known as panning), a selection of phages remain that show the highest binding affinities for the antibodies in the sample being tested. These phages can then be isolated and sequenced to find the peptides that they were

displaying. A modification of this technique known as 'mRNA display' utilises the aminonucleoside antibiotic Puromycin which is able to form a covalent link between the 3' end of an encoding messenger ribonucleic acid (mRNA) and its translated polypeptide (Wilson et al., 2001). Antibody bound peptide can then be identified by reverse transcription and sequencing of the attached nucleotide tag. Phage and mRNA display methods have the advantage of being able to screen vast numbers of peptides simultaneously. Often this is exploited by using combinatorial libraries of random sequence peptides – typically up to 10^{10} different peptides for phage display and up to 10^{13} for mRNA display (Wilson et al., 2001). Random peptide libraries produce numerous sequences that are similar to the epitope sequence thereby allowing identification of key binding residues. They also allow identification of peptides that do not share significant sequence identity with the antigenic protein but instead mimic the shape and physiochemical properties of a conformational epitope – so called 'mimotope' peptides (Geysen et al., 1986). Once a set of peptides with high affinity to the antibody being tested has been identified, bioinformatic algorithms such as PepSurf or Mapitope can then be used to map possible mimotopic peptides to regions of the antigen under investigation (Bublil et al., 2007; Mayrose et al., 2007).

Although phage or mRNA display can be very successful at characterising protein epitopes, they depend on the successful generation of combinatorial libraries followed by the successful identification of bound ligands – the techniques for achieving both remain difficult and time consuming to execute. An alternative method of generating peptide libraries is by direct chemical synthesis on solid supports such as polystyrene pins (Geysen et al., 1984), beads (Lam et al., 1991) or cellulose membranes (Frank, 1992). By using a solid support the location of any given peptide can be identified by its position on the support. The use of synthetic peptide libraries in a array format is discussed in further detail in section 1.3.

1.2.1 In silico epitope prediction

Due to the difficulty and expense of epitope mapping a considerable amount of interest has been devoted to computational methods of identifying potential epitope sites that can then be characterised experimentally in more detail. Early

efforts at prediction used sequence averaged hydrophobicity measurements to delineate possible epitopes (Hopp and Woods, 1981). The principle used was that antigenic determinants are found in solvent accessible regions and are often associated with charged and polar residues. Further efforts to predict epitopes based on more sophisticated amino acid propensity scores have not been shown to be at all useful in a study comparing them to experimentally verified epitopes (Blythe and Flower, 2009). These algorithms were primarily focused on linear epitopes with no means of accounting for the conformational shape of the protein. Newer epitope prediction algorithms use a combination of amino acid propensity scales alongside structural information such as side chain orientation and solvent accessibility (Sweredoski and Baldi, 2008; Rubinstein et al., 2009). Unfortunately accuracy, although better than simple propensity scales, still remains modest.

1.3 Peptide Arrays

A peptide array is a set of peptides immobilised on a solid support that can be used to investigate a wide variety of biological reactions from protein-protein interactions to enzymatic reactions and antibody-antigen binding. The concept of immobilisation of biologically active molecules on a solid support is not new. Indeed peptide arrays can be thought of as an extension of the technology used to produce nucleic acid arrays (investigating mRNA transcripts hybridising to immobilised DNA) or protein arrays where whole proteins are immobilised in an array format and probed for biological interactions. Peptide arrays offer a number of advantages over their protein array counterparts. Peptides can be easily and efficiently synthesised in a standardised manner whereas purified or recombinant proteins are typically much more complex to produce. The process of immobilising a protein on a solid support such as a glass slide can also be problematic especially in preserving its conformational integrity. The peptide array also has the great advantage that for an immunoassay it can potentially reveal the antigenic epitopes (albeit predominantly linear epitopes) of a protein. The ability to interrogate multiple peptides in parallel to investigate the presence or absence of antibodies directed against specific epitopes is a potentially extremely valuable tool in probing the immune response to infection.

The earliest peptide arrays owe much to the development of solid-phase peptide synthesis (SPPS) by R. Bruce Merrifield (Merrifield, 1963). The principle of SPPS is one of repeated cycles of coupling and deprotection. A peptide attached to a polymeric solid support is coupled to a single N-terminal protected amino acid unit. The amino acid is then de-protected revealing a free N-terminal amine for the next amino acid to be coupled. Thus, unlike ribosomal protein synthesis, SPPS proceeds in a C-terminal to N-terminal fashion.

The first utilisation of SPPS for the display of multiple peptides for parallel analysis was performed by Mario Geysen in the early 1980s (Geysen et al., 1984). He used SPPS to display peptides on 4mm diameter polyethylene rods with the same format and spacing as a microtitre plate. Using overlapping 6mer peptides spanning the primary structure of the foot and mouth viral VP1 envelope protein he then tested the pin-coupled peptides for antigenicity using rabbit sera. This method was successful at characterising an immunodominant epitope but also highlighted the variability of the antibody response to a given antigen. At around the same time Ronald Frank was developing a method of parallel oligonucleotide synthesis on cellulose discs packed in a column (Frank et al., 1983). He later modified the technique for peptide synthesis using a cellulose support – the so-called SPOT synthesis method (Frank, 1992). SPOT synthesis uses the same principle as SPPS but by applying droplets of reactants to a porous membrane without cleaving off the completed chain he was able to produce a planar array with peptides synthesised in situ. Although cellulose membrane arrays (macroarrays) are still widely used their major disadvantage is that their size means that they require a relatively large volume of reactant per probe on the array. This makes them impractical for large scale panning of epitopes from multiple proteins. Hence a major focus of array development has been in increasing the density of probes and miniaturising the platform.

An alternative technology for the parallel in situ synthesis of peptides was developed by Fodor et al. (1991). They developed a photolabile protecting group that would allow the SPPS reaction to only occur at an illuminated region. Then using photolithographic masks they were able to repeatedly deprotect selected regions of a glass array for coupling of new amino acids. Although this allowed a high density array to be produced, Fodor's original technique suffered from being time-consuming and expensive. However, although the technique was not

widely adopted at the time for peptide array manufacture, photolithography has been adopted as a standard process for production of high density oligonucleotide arrays.

Two major methods exist for producing high density peptide microarrays - in situ synthesis typically using photolithography or laser printing technology, and immobilisation of pre-synthesised peptides using robotic spotters. The process of photolithographic synthesis has been refined considerably since Fodor's original report in 1991. Rather than using photo-labile protection groups and masks, the technique now utilises a photo-generated acid to create a localised acidic environment that deprotects a di-*tert*-butyl dicarbonate (*t*-Boc) protected amino acid. Laser printing technology offers a novel alternative method of in situ synthesis. It utilises solid amino acid particles which are deposited on a chip by a laser directed electrostatic pattern. This is directly analogous to deposition of toner particles on paper in a conventional laser printer. The amino acid 'toner' particles are then melted to initiate coupling followed by washing and then a deprotection step to allow coupling to the next round of particles deposited by the printer (Breitling et al., 2009; Beyer et al., 2007).

Despite these extremely high density methods, the commonest method of peptide microarray production involves immobilisation of pre-synthesised peptide to a solid support - typically glass. The advantage of this is that SPOT synthesis is widely available, relatively cheap and can produce highly purified peptides. Moreover it is relatively easy to introduce various modifications to the peptide for example phosphate groups, citrullination, lipidation etc. that allows a much wider range of peptide interactions to be studied. There are two main methods of peptide immobilisation. Firstly by dissolving cellulose bound peptide and spotting the peptide-cellulose conjugate on to a coated glass slide, and secondly by incorporating a linker group to the N-terminal, cleaving the peptide from its cellulose support and then forming a covalent bond between the N-terminal linker and the coated glass surface. The former method displays peptides attached by the C-terminal as with cellulose macroarrays and the latter displays N-terminal attached peptides.

1.3.1 Peptide array ligand binding detection

Numerous methods have been developed for identifying ligands – typically proteins or antibodies, bound to peptide arrays – from direct detection using surface plasmon resonance (SPR) through to indirect labelling with labelled secondary antibodies (Figure 1.4).

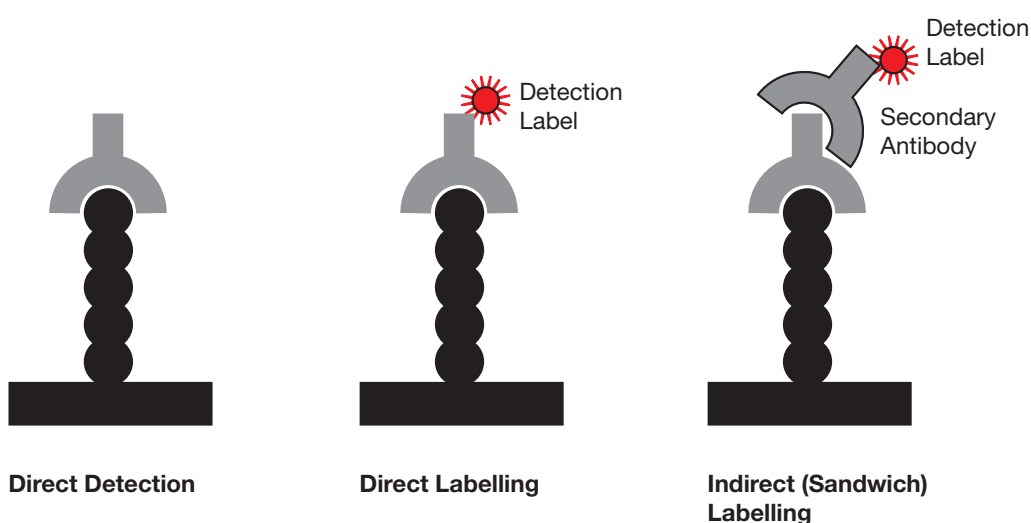


Figure 1.4: Schematic diagram of peptide array ligand binding detection methods

Three methods for peptide array ligand detection are shown above in the cartoon diagrams. The immobilized peptide chain and array surface are illustrated in black with primary and secondary antibodies depicted in grey. Direct detection methods rely on being able to distinguish bound antibody by a change in physicochemical properties eg. Surface refractive index – for SPR measurements. Labelling methods utilize a detection label such as a radioisotope or an conjugated fluorophore. Direct labelling involves coupling of the primary antibody to a detection label whereas indirect labelling requires an additional antibody binding step using a labelled anti-immunoglobulin antibody to detect the primary antibody.

The commonest method of peptide array ligand detection is by means of a detection label – typically either a radioisotope label or more commonly a fluorophore conjugate. Fluorophores are molecules that can absorb light of a particular wavelength causing excitation of an electron. As the electron returns to its ground state it releases the energy as a photon of a different wavelength (always longer due to the loss of energy). The difference in wavelength between excitation and emission is known as the Stokes shift and it underlies the principle of fluorescence-based assays – excitation at one wavelength and detection of the emitted light at another. Numerous fluorophores have been developed. Some

are derivatives of naturally occurring proteins such as Green Fluorescent Protein (GFP) from the *Aequorea victoria* jellyfish but the majority are synthetic organic dyes or semiconductor nanocrystals (Quantum dots) (Kairdolf et al., 2013). The organic cyanine dyes Cy3 and Cy5 are historically the most common fluorophores used on microarrays. In nucleic acid arrays the fluorophore is directly conjugated to the cDNA being hybridised to the array. Direct labelling of antibodies or an antibody mixture is possible using reactive fluorophore derivatives. This is achieved by covalent bonding of reactive groups to amino acid side chains e.g. N-hydroxysuccinimide esters to lysine side chains. In practice though, indirect labelling using a labelled anti-immunoglobulin secondary antibody is more common and avoids the problem of variable labelling efficiency for different antibodies in a mixture and non-specific labelling increasing background signal. An example schematic of a peptide array immunoassay using indirect labelling with a fluorophore tagged secondary antibody is shown in figure 1.5 below.

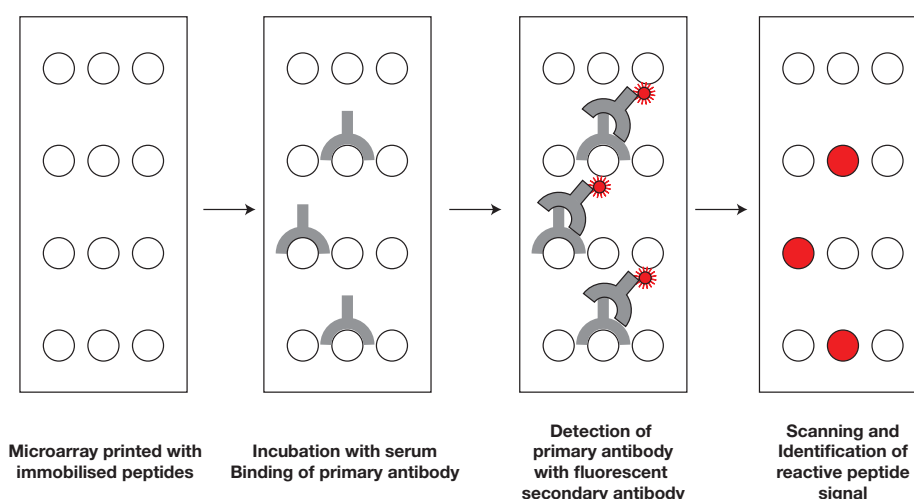


Figure 1.5: Overview of a typical peptide microarray immunoassay procedure

The process starts with incubation of the array with the primary antibody eg. a purified or monoclonal antibody or complex mixture such as serum. Bound primary antibody is detected by a fluorophore tagged secondary antibody. After washing and drying the array is scanned to visualise the fluorescence signal.

Labelling methods can potentially be associated with some problems. Direct labelling can interfere with the binding site or sterically hinder binding to the epitope. It can also increase non-specific background by binding to other proteins or antibodies in a complex mixture. Indirect labelling is also almost always complicated by cross-reactivity of the secondary antibody to peptides

ORIGINAL IN COLOUR

on the array. Moreover, using multiple secondary antibodies (for example, one specific to IgG and another specific to IgA) runs the risk of cross reactivity of the secondary antibodies to each other. Hence, there is increasing interest in developing label free detection methods that can directly identify bound proteins and antibodies.

Surface plasmon resonance is an example of a direct detection method. The technique uses peptides immobilised on a gold surface with microfluidic channels allowing the delivery of an antibody solution. The SPR measurements can then measure the real time binding of antibody to its ligand. Although these arrays can give very detailed binding information their construction inherently limits the throughput of peptides that can be interrogated on such a platform. Other label free methods of ligand detection such as atomic force microscopy, electrochemical impedance spectroscopy and mass spectrometry are yet to be used for peptide arrays but may become more commonplace in the future (Yu et al., 2006).

1.4 Analysis of microarray data

A typical microarray workflow is depicted in figure 1.6. Several computational steps are typically employed from quantisation of a scanned image, through to assessment of array quality and preprocessing to remove unwanted non-biological effects. These steps are important if valid conclusions are to be drawn from the data. A major theme of this thesis is the investigation of these data processing steps in their application to peptide array analysis.

1.4.1 Image processing

A typical microarray scanner records fluorescence data as a 16 bit TIFF (Tagged Image File Format) file (i.e. pixel intensity is a value in the range $0 - (2^{16} - 1)$). The pixel size typically varies between 10 to $1\mu\text{m}$ square resulting in a high resolution image. The first step of any microarray analysis is to quantise that image into a set of intensities that can be used for downstream analysis. This is accomplished by software that performs by three separate tasks:

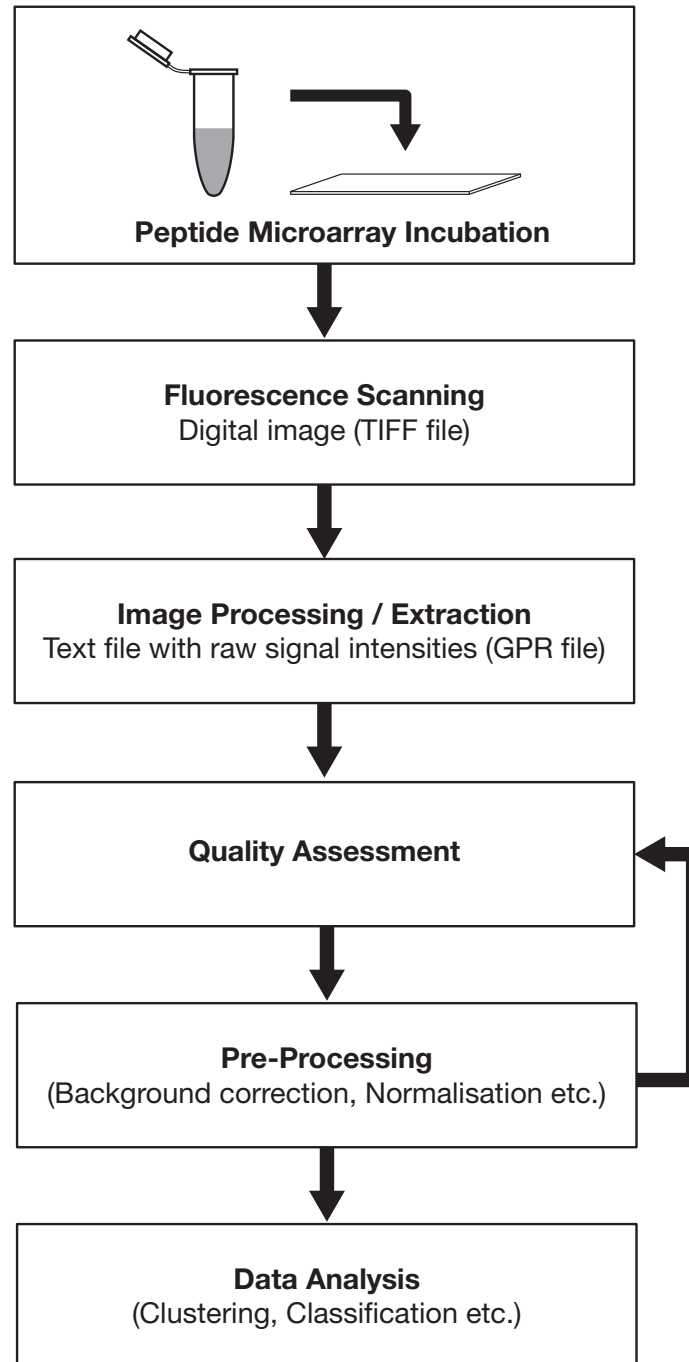


Figure 1.6: Overview of typical peptide microarray analysis workflow showing the computational steps required to scan, quantise, pre-process and analyse a microarray.

1. Addressing / Gridding: Assigning an identity and co-ordinates to pixels within regions of the array corresponding to the probe features.
2. Segmentation: Partitioning of the image by determining if a pixel is 'foreground' – within the area of a feature spot or 'background' – outside the feature boundary.
3. Extraction: Calculation of a summarised foreground and background intensity (and quality control measures) from the segmented pixels.

Figure 1.7 shows part of a scanned peptide array with its segmentation grid overlay. The addressing co-ordinates of each peptide or control feature is provided by the manufacturer in the form of a tab delimited text file. The co-ordinates for each feature are known from the printing process but are presented as an 'ideal' grid containing uniformly distributed blocks and spots. In reality a number of parameters need to be determined from the scanned image in order to fit the idealised grid to the actual features: The overall position of the array and individual blocks will need to be aligned, small translations of spot positions due to print head variations need to be addressed, and variations in spot size due to the physiochemical properties of the peptide or technical variation in spotting volume need to be accounted for. Although this can be achieved automatically by image analysis software, for peptide arrays this commonly fails to accurately align the array and block positions. Peptide arrays unlike deoxyribonucleic Acid (DNA) arrays tend to have a large proportion of low intensity features making accurate determination of the block edges difficult. Some manufacturers have attempted to overcome this by spotting bright control features around the edge of array blocks.

Image segmentation is by necessity an automated procedure due to the large number of spots on a microarray. Essentially the procedure partitions the image into a set of foreground pixels (within a feature spot) and background pixels (outside of the feature spot). Numerous methods have been developed for segmenting array images (Yang et al., 2011, 2001) but all essentially rely on producing a spot mask of a certain shape and size. The oldest segmentation method known as fixed circle segmentation fits a circular spot mask with a fixed diameter to each of the features on the array. This was first implemented by Eisen

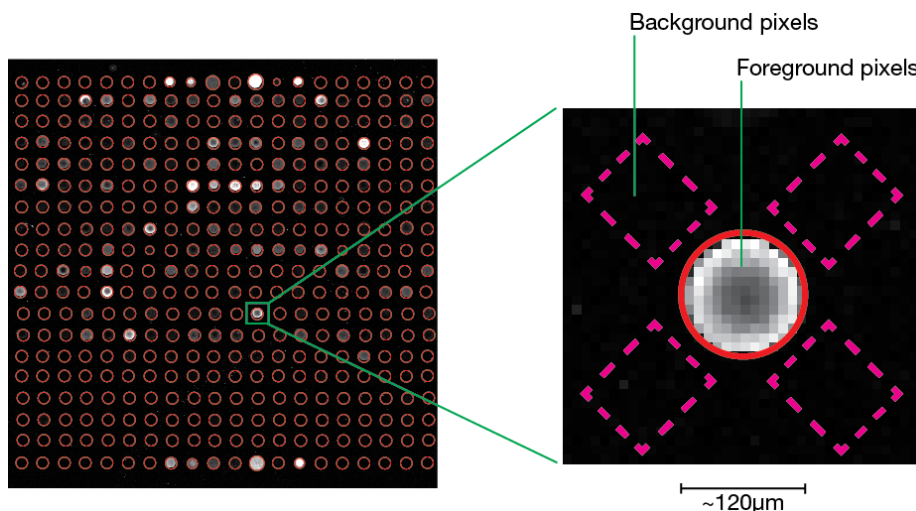


Figure 1.7: Example of a single block and feature from a scanned peptide microarray image

Peptide locations are identified from the grid overlay. The overlay itself is aligned to landmark features on the array and the spot size adjusted to delineate the image foreground (the peptide spot). On the right is an enlarged single feature with the foreground segmentation area outlined in red and the background sampling areas in pink (hatched line).

in the ScanAlyse software package (Eisen, 1999). Although fast and simple it does not account for variation in spot size; obtained intensities would be biased with smaller spots including more of the background pixels and larger spots only reading the central area of the feature.

A more accurate procedure is known as the adaptive circle method. This is implemented in the Genepix[®] software package (Molecular Devices, USA). Like the fixed circle method it fits a circular spot mask to the array features. However, by using an edge finding algorithm it can detect the boundary of features and adjust the mask diameter to better fit it. The limitation of this method is that it relies on features being of sufficient brightness to be able to distinguish the boundary from the background. Hence for peptide arrays many of the low intensity spots will not be automatically adjusted. In this case usually a default fixed circle segmentation is applied. The threshold for boundary detection can be adjusted in the software – a high threshold only adjusts the mask diameter of bright spots but a low threshold, while capturing more low intensity spots, risks including more artefact spots (scratches, dust spots etc.) within the spot mask.

Both of the above methods make the assumption that microarray spots will be

ORIGINAL IN COLOUR

circular. For the majority of commercial arrays this should be true. Some array features may however be irregularly shaped due to uneven deposition of the printing solution, or smearing in the manufacturing or processing steps. Fitting spot masks to irregular features can be achieved (adaptive shape segmentation) although is not commonly employed in software packages. Careful quality assessment of data segmented in this way is required as it is much more likely that artefactual spots will be included especially if there is a high background intensity and many low intensity features.

The final step of image analysis involves extraction of at least a set of foreground intensities for each array feature. This is usually given as the median pixel intensity in the spot mask region – although the mean or sum of pixel intensities can sometimes be used. In practice other data characteristics can also be extracted such as the pixel intensity variance and the spot mask diameter which may be useful for quality control. In addition to a spot intensity (foreground) usually a background intensity will also be extracted – this may be global (average background across the entire array), local (background in the vicinity of each feature) or regional (average background in particular array areas).

1.4.2 Quality assessment

Quality Assessment (QA) is a vital step in any microarray analysis. It can help identify arrays with significant artefacts and technical variation. Several quality assessment software packages are available for particular microarray types eg. *affyQCReport* for Affymetrix arrays (Parman et al., 2005), or as a more general QA package covering several array types eg. *arrayQualityMetrics* (Kauffmann et al., 2009), or as a module within established packages for array analysis eg. *limma* (Ritchie et al., 2015). These and the majority of other QA packages were designed for the needs of gene expression or other nucleic acid based array technology. Although some of the functions may be applicable and useful for peptide arrays many such as Ribonucleic Acid (RNA) degradation plots or NUSE (Normalised Unscaled Standard Error) plots are only specific to particular array platforms – in this case Affymetrix oligonucleotide arrays. The *pmpa* package introduced in this thesis introduces a number of QA functions designed for use with peptide microarrays. A more detailed account of QA applicable to peptide

arrays is given in Chapter 2.

1.4.3 Data pre-processing

High density microarrays by their very nature generate large and complex data sets necessitating sophisticated bioinformatics to draw meaningful conclusions. Alongside the huge number of signals that these arrays generate comes a considerable amount of noise which, if not properly controlled for can mask true biological variation. Hence computational methods are needed to detect systematic errors and quantify stochastic effects. This ‘low-level’ processing – often called ‘pre-processing’ – of raw signal intensities is an essential step in the analysis of microarray data. Pre-processing of microarray data typically involves a number of separate steps such as background correction, normalisation and summarisation. The overall aim is to take a set of raw intensities, remove non-biological effects while minimising the impact on true variation, and produce a set of probe-ligand binding estimates. There are a number of pre-processing algorithms in the current literature however the vast majority of them have been based on studies of DNA arrays (Quackenbush, 2002; Press et al., 2002). Hence the assumptions made for many of the computational processes may not necessarily be applicable to peptide arrays.

Background correction

The microarray background intensity is a measure of the non-specific contribution to the measured signal intensity of a given array feature. Background intensity is estimated from pixels of the array image that are not designated as feature pixels. Typically this is measured either globally with a single average value for the entire array; regionally with values representing defined areas of the array surface or locally with a one value for each of the array features. The typical assumption of background correction is that the observed signal intensity of a feature i i.e. the foreground signal ($Y_{(f)i}$) is comprised of the sum of the true signal (μ_i) and the observed background ($Y_{(b)i}$) intensity. Hence subtraction of the background from foreground signal is the usually the default method of background correction applied to many microarray experiments.

$$\mu_i = Y_{(f)i} - Y_{(b)i} \quad (1.1)$$

There are several problems associated with simple subtractive correction. In particular, for some low intensity features, the background intensity can exceed the foreground signal leading to negative corrected signals that are difficult to interpret. As a result several adaptations to subtractive correction have been developed (Ritchie et al., 2007). Chapter 3 of this thesis discusses background correction methods and the assumption of background additivity in its application to peptide microarray preprocessing.

Normalisation

Normalisation is a process of removing or minimising systematic non-biological variation in a microarray experiment by applying mathematical transformations to the dataset. Normalisation may be applied to each array in turn (within array normalisation) to account for spatial variation across an array e.g. as a result of uneven incubation or uneven spot printing, or it may be applied between several arrays to account for overall signal differences between different samples (Press et al., 2002). Many of the assumptions made for normalisation of a gene expression arrays are not necessarily valid for peptide arrays. In chapter 4 of this thesis I discuss in greater detail the various normalisation algorithms and their application to preprocessing peptide arrays.

1.5 Applications of peptide microarray immunoassays

Since their original inception peptide arrays have developed into a widely used and valuable proteomics tool (Reineke et al., 2001). Although they have been successfully used for characterising protein-protein interactions (Katz et al., 2011) and for understanding enzyme-substrate recognition (Thiele et al., 2011), the focus of this thesis is on their use as immunological tool for understanding antibody reactivity.

One of the earliest uses of peptide arrays was for the mapping of continuous epitopes using overlapping peptides derived from the primary sequence of a putative antigenic protein (Geysen et al., 1984). Possible epitopes can be identified where reactive peptides share a common sequence motif and key binding residues further characterised using a substitution array (peptides comprising the putative epitope with each residue substituted in turn for Alanine) (Frank, 1992; Jin et al., 1992). Understanding the relationship of a reactive peptide signal to the existence of an epitope on a protein of interest is a key concept in any peptide array analysis. Unlike nucleic acid microarrays where binding to the array probes occurs only between molecules of complementary sequence, multiple antibodies with specificities for different proteins could bind to any one given peptide. This occurs because peptides can mimic natural discontinuous epitopes (mimotope) as discussed in section 1.2, and because the antigen binding region (the paratope) is comprised of amino acids from multiple overlapping CDRs. Other amino acid residues may form a paratope with a specificity for a different, possibly unrelated epitope. Thus any one given antibody may be able to recognise several different epitopes. This effect can be clearly seen when monoclonal antibodies are incubated on a high density array containing random sequence peptides; binding occurs to peptides that have no similarity to the known epitope (Halperin et al., 2012; Stafford et al., 2012).

Given that serum or other complex antibody mixtures contain up to 10^9 specificities (Nobrega et al., 1998), we might expect that any specific response would be drowned out by the overwhelming cross-reactivity of all the different antibodies. However, Stafford et al. (2012) demonstrated that a specific signal from a monoclonal antibody was still distinguishable even when diluted with a large excess of serum immunoglobulin. This suggests that naïve serum (i.e in the absence of infection or a chronic disease state) is comprised of a large number of relatively low affinity, low concentration antibodies. Hence any specific signal such as high-affinity antibodies produced in the context of infection would be distinguishable against the background signal.

Using this principle, peptide arrays can be used to characterise patterns of antibody reactivity – sometimes called an ‘antibody signature’ (Schutkowski et al., 2009). This approach has been successfully applied to infectious diseases such as Tuberculosis (Gaseitsiwe et al., 2008), Echinococcus (List et al., 2010), Toxoplas-

mosis (Maksimov et al., 2012, 2013), HIV (Gallerano et al., 2015; Stephenson et al., 2015), JC virus infection (Lagatie et al., 2014) and Influenza (Ambati et al., 2015). Similar approaches have been used to characterise auto-antibodies (Hecker et al., 2012), allergens (Shreffler et al., 2004, 2005; Severance et al., 2011; Lin et al., 2012; Perez-Gordo et al., 2011) and responses to vaccination (Price et al., 2013).

The above studies all used peptides derived from the primary sequence of putative antigenic proteins similar to arrays used for classical epitope mapping. However because of the complex nature of serum and the polyclonal nature of immune responses in these conditions, it is impossible to be certain if the signals being identified represent reactivity from continuous epitopes of the protein of interest or are cross-reactive signals from another antibody. A logical extension from using known peptide sequences is to look purely for the cross-reactive signal by using large arrays of random sequence peptides. This process was given the name ‘immunosignaturing’ (Sykes et al., 2013). The array peptide sequences in immunosignature experiments typically have no sequence similarity to any likely antigenic proteins yet are able to distinguish remarkably consistent patterns of antibody reactivity. This approach has proved to be successful at identifying clinical conditions in infectious diseases (Stafford et al., 2012; Navalkar et al., 2014), neurodegenerative disease (Restrepo, 2013), and cancer (Hughes et al., 2012; Stafford et al., 2014).

1.6 Thesis outline

This thesis explores the computational challenges involved in analysing high density peptide microarray immunoassay data. Chapter 2 describes the development of the `pmpa` open-source software package and its use in reading and pre-processing scanned peptide array data. Chapter 3 and 4 investigate two key aspects of pre-processing - background correction and normalisation respectively. No consensus exists on which pre-processing methods if any should be applied to peptide array data making comparison between studies difficult. The work presented in chapters 3 and 4 compares many of the existing methods using a ‘spike-in’ antibody dilution series to assess performance. The final two chapters

look at applying the optimised preprocessing methodology in two studies of antibody responses to infection. Chapter 5 demonstrates the use of peptide arrays for characterising outcome and relapse risk in patients with *Clostridium difficile* infection, and chapter 6 discusses using a peptide array for identifying an antibody signature that characterises paediatric tuberculosis – an infection where current diagnostic techniques are limited.

Chapter 2

PMPA: Peptide Microarray Pre-Processing Analysis. An R package for peptide microarray data

2.1 Introduction

The work presented in this chapter describes the development of an open-source software package (`pmpa`) for the analysis of peptide microarray data. The package is written in the R programming language (Ihaka and Gentleman, 2012; R Development Core Team, 2014) and utilises several components from the Bioconductor project (Gentleman et al., 2004; Huber et al., 2015). A key feature of the `pmpa` package is the integration of several background correction and normalisation algorithms. This allows the parallel pre-processing of data using different methods and facilitates the comparison of these methods in order to obtain an optimised analysis pipeline. Several open source software packages already exist for microarray analysis. However, many are specific to particular microarray platforms e.g. the `affy` package for Affymetrix[®] oligonucleotide arrays (Gautier et al., 2004) or `beadarray` for Illumina bead based arrays (Dunning et al., 2007) and so are not appropriate for use with peptide arrays where the

underlying technology is very different. Although packages designed for spotted complementary deoxyribonucleic Acid (cDNA) arrays e.g. `limma` (Ritchie et al., 2015) or `marray` (Yang et al., 2009) have been adapted for use with peptide arrays because they are designed for use with two colour arrays where the outcome measure is a log ratio, they require some care to ensure an interpretable result. To date only two published software packages have been developed specifically for peptide array analysis: `rapmad` (Renard et al., 2011) which is based on `marray` but unfortunately is no longer being maintained, and `PepStat` (Imholte et al., 2013) which, like `pmpa` is based on the Bioconductor project.

2.2 The R Statistical Programming Environment

R is both a programming language and an environment for statistics and data visualisation. It is a free, open source and cross-platform implementation of the S language which was originally developed by John Chambers at Bell Labs (Chambers, 1999). Its popularity as a statistical package has increased substantially since its original release in 1993. One of R's biggest strengths is its open source nature and the active development community that has sprung up around it. Thousands of packages now exist written in the R language itself or by integrating code written in other languages such as C++ or Java which extend its base capabilities and allow its use in a large variety of situations.

2.2.1 The Bioconductor Project

The Bioconductor project is an online repository of open source software for the analysis of high throughput data (eg. genomics, proteomics, flow cytometry etc.) based primarily on the R programming language. Bioconductor is comprised of several hundred software packages each designed to work with a particular technology or to implement a new methodology. Many Bioconductor packages utilise a common data structure. This facilitates the integration of methods or functions from several packages allowing code to be reusable. Moreover, the open-source nature of the packages promotes modification of these methods in order to customise a workflow. An online mailing list also exists

(<https://support.bioconductor.org/>) with active participation of developers and end-users providing support and serving as a platform for proposing improvements and reporting bugs.

2.2.2 Object-Oriented Programming

Object-orientated programming (OOP) is a model for programming based around data structures known as objects. Objects encapsulate both data (known as attributes) and behaviour (known as methods) into a self-contained entity. OOP therefore differs from the more traditional procedural programming paradigm where functions that operate on data are considered separately from the data itself. The concept of OOP arose in the early 1960's and is now commonplace and intrinsic to many programming languages. R is often thought to be a procedural language but it also supports OOP albeit in a less strict manner than many other languages. R was originally developed with an OOP instruction set known as S3 which is a rather informal way of implementing OOP. A more formal S4 style was implemented with the *methods* package in R version 1.4.0 onwards (Chambers, 2001). Both S3 and S4 are used in Bioconductor but the *pmpa* package exclusively uses S4 OOP. It utilises OOP in a 'behind the scenes' manner so that the end user can write simple scripts to import a dataset, use functions to manipulate it and obtain an output in a procedural manner but taking advantage of two key features of OOP – *encapsulation* in order to ensure data security and integrity, and *inheritance* in order to facilitate integration to other packages and code reusability.

Objects within OOP are considered to be an instance of a class. A class is essentially a blueprint or schematic for how the object should look and behave. To use a real world analogy: if we wanted to build a car we might firstly draw up a schematic – it would define a set of attributes about the car – engine capacity, colour, transmission type etc. and it would define a set of functions that the car could do – accelerate, brake, change gear etc. By itself the schematic does not constitute a car rather it is a set of instructions on how to create one. In the same way a class sets out what attributes and what methods the resulting object should have. When we create or *instantiate* the object those attributes now represent actual pieces of data that the methods can operate on.

If we now consider a microarray experiment with m peptide features and n samples, we can define a class with a set of attributes that contains the relevant data. For example, we would need 2 $m \times n$ matrices to hold the foreground intensities and background intensities. Each sample would have a set of annotation describing the phenotype of the experimental subject – a matrix of n columns and as many rows as required for each item of annotation. Similarly each feature would have a set of annotation describing the peptide e.g. sequence, protein, position etc. – a matrix of m rows and as many columns as needed. We might also need additional data attributes to hold data about the experimental conditions for each array or about the experiment as a whole. The advantage of OOP is that all of these attributes can be encapsulated into a single object. We can then define methods that obey strict rules in handling this data. For example if we want to subset our data to remove control samples we would need to subset the foreground data, background data, the phenotypic data and the array protocol data simultaneously. The process of doing this or other operations is substantially simplified taking an OOP approach.

2.3 Installing PMPA

The `pmpa` package was written entirely in R and depends on the `Biobase` package Huber et al. (2015) to provide the core Bioconductor classes and methods. Other package dependencies include `plyr` (Wickham, 2011), `limma` (Ritchie et al., 2015) and `preprocessCore` for the `normalise.quantiles` (Bolstad et al., 2003) function. Bioconductor (and other packages) can be download within R by

```
source("https://bioconductor.org/biocLite.R")
biocLite("Biobase")
```

Once installed, `pmpa` is downloaded using

```
install.packages("pmpa", contriburl="https://github.com/katenambiar/pmpa")
```

This downloads the current stable release as a precompiled binary. Development builds can be downloaded as source code directly from GitHub (requires the `devtools` package).

```
library(devtools)
devtools::install_github("pmpa", "katenambiar")
```

Once installed, `pmpa` can be loaded by the command `library(pmpa)` in common with all R packages.

2.4 Processing spotted peptide microarray data using PMPA

2.4.1 Reading Axon Text File (ATF) format data into PMPA

The `pmpa` package can be used for reading and analysing peptide array data files obtained after image processing using GenePix® software (Molecular Devices LLC, USA). Currently only files in the Axon text file (ATF) format such as GenePix® results files (GPR) can be used. The ATF format is a tab delimited text file that is comprised of a header portion and a data records section. The data records section has one line for each feature on the array with columns recording feature specific data such as the foreground signal intensity, background intensity, spot location, spot diameter etc. The header section contains data about the scanner parameters such as the scan date, PMT gain, laser power etc. Data is imported using the `readArrays` function

```
rawdata <- readArrays(samplename, filename, path, wavelength)
```

where *samplename* is a vector containing unique names for each of the samples, *filename* is a vector containing the file names and extensions of the GPR file corresponding to the sample name and *path* is a character vector containing the path to the directory containing the files. If all the files are in a single directory only one value needs to be specified for *path* and it will be recycled in the standard manner for R functions. *wavelength* contains a single integer value corresponding to the laser excitation wavelength used to scan the array. Multi-colour arrays – i.e. those scanned in more than one wavelength, need to be imported as separate objects for each wavelength. The `readArrays` function operates in a

similar way to other microarray input functions from established packages e.g. `read.maimages` from `limma` or `read.marrayRaw` from `marray`.

A convenient way of passing the arguments to `readArrays` is by reading them in to a data frame from an external delimited text file with a column for *samplename*, *filename* and *path* e.g.

```
> files <- read.delim("./Data/arrayFileNames.txt")
> rawdata <- readArrays(files$samplename,
+                       files$filename,
+                       files$path,
+                       wavelength = 635
+                       )
```

2.4.2 Representation of data in PMPA

`readArrays` will read each of the array files specified and instantiate a new object of the `MultiSet` class – in the example above the object is named `rawdata`. `MultiSet` is an extension of the parent Bioconductor `eSet` class and it inherits the same data structure but allows multiple elements to occupy the *assayData* slot. This is ideal for raw peptide microarray data as it allows it to be populated with several data matrices with dimensions m (features) $\times n$ (samples) each with a different data attribute. The default `readArrays` function writes 9 matrices to the *assayData* slot: Median foreground pixel intensity, mean foreground pixel intensity, morphological background intensity (if measured), local median background pixel intensity, local mean background pixel intensity, total foreground pixels, feature diameter, and quality control flags (features identified during image processing as ‘bad’ – default value -100, which will be excluded from analysis).

In addition all of the array specific data found in the `gpr` file header is written to the *protocolData* slot, a basic sample annotation is written to the *phenoData* slot (just containing the sample names) and finally a feature annotation is written to the *featureData* slot comprising feature IDs, names, and the block, column and row positions to localise the spot on the array. This creates a single data object with a schematic as in figure 2.1.

The data import can be checked by the following command

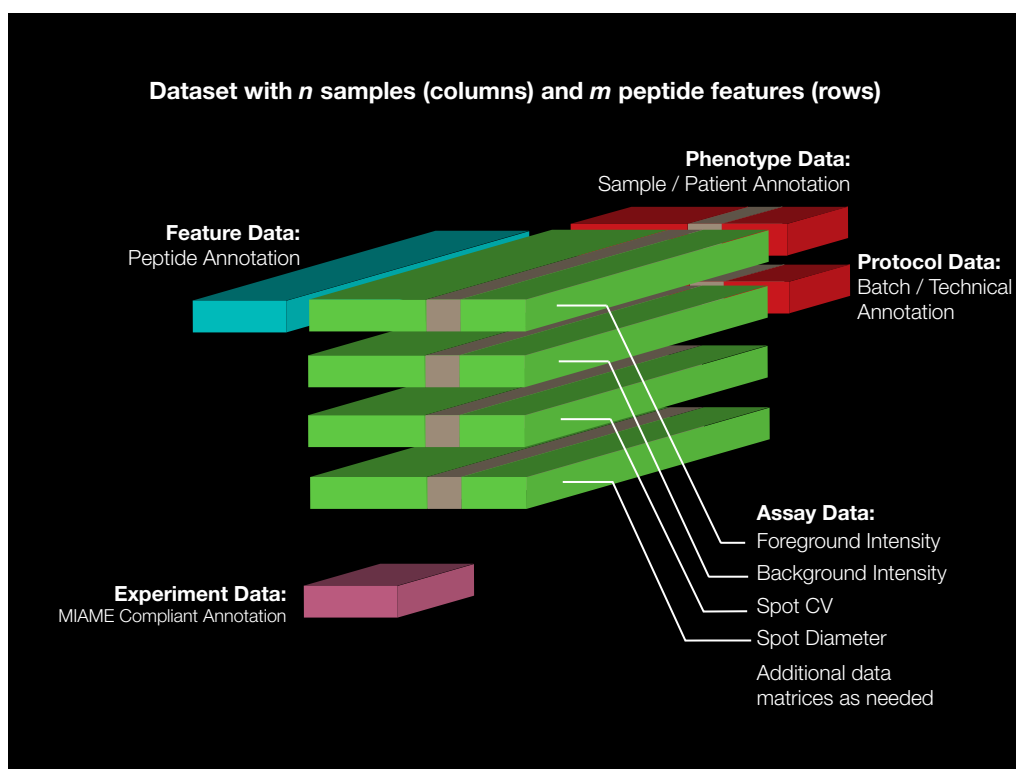


Figure 2.1: PMPA package data representation schematic

Objects of the MultiSet class created by the PMPA package are represented in the schematic shown above. The main microarray intensity and feature quality data is held within the assayData slot. This shown as green blocks with m rows – one per array feature, and n columns – one for each sample analysed. Each item in the assayData slot has the same dimensions. Annotation information for each feature is shown as a blue block with m rows corresponding to features and as many columns as required to hold the annotation information. Similarly the sample annotation is shown as red blocks with n columns corresponding to the samples in the experiment.

```
> show(rawdata)

MultiSet (storageMode: lockedEnvironment)
assayData: 15552 features, 28 samples
  element names: bg, bMean, bMedian, bSD, dia, fCV, flags, fMean ...
protocolData
  rowNames: LEAR0011065 WALDA011057 ... CDI022 (28 total)
  varLabels: ATF.1.0 HeaderLines ... ATF (35 total)
  varMetadata: labelDescription
phenoData
  sampleNames: LEAR0011065 WALDA011057 ... CDI022 (28 total)
  varLabels: fileName
  varMetadata: labelDescription
featureData
  featureNames: LIYSQVLFKGGCPS_1_1_1 HVLLTHTISRIAVSY_1_2_1 ... (15552 total)
  fvarLabels: ID Block ... Name (5 total)
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
```

The output shown above summarises the object class and the data slots (attributes) with examples of the data contained within.

In addition to the data slots populated by `readArrays` a further slot, *experimentData* can be used to include information regarding the overall experiment to ensure MIAME compliance. MIAME – the Minimum Information About a Microarray Experiment is a standard that was established to ensure that sufficient annotation data is made available in order to interpret the results and to independently verify the analysis (Brazma et al., 2001). MIAME is widely adopted as a standard for gene expression data publication and for allowing datasets to be accessible on public databases. Although many of the elements of the MIAME definition are more applicable to gene expression arrays than to peptide arrays, it remains a useful standard and one which has, to date, not widely been used for peptide array experiments.

2.4.3 Data annotation

Although `readArrays` creates a basic annotation set for the samples and features it is almost always desirable to add in additional annotation information. For example, a microarray experiment using patient samples may include pheno-

typic data to describe each patient (disease state, age, sample collection date etc.). Similarly annotation of each feature with information such as the peptide sequence, origin protein and sequence position will almost always be essential for interpreting the analysis. The method `annotateArrays` allows additional information to be written to the *phenodata*, *protocoldata* or *featuredata* slots of a *MultiSet* or *ExpressionSet* object.

```
annotateArrays(rawdata, pheno, protocol, feature)
```

where `rawdata` is the *MultiSet* (or *ExpressionSet*) object to be annotated and `pheno`, `protocol` and `feature` are data frames containing the annotation information for the respective slot. The `pheno` and `protocol` data frames must contain one column called `sampleNames` and this must match with the sample names from the object to be annotated. Similarly the `feature` data frame must have a `featureNames` column that matches the feature names of the data object. Feature names must be unique and so to allow replicate features to be indexed a composite feature name comprising the four mandatory columns of the GenePix[®] array list (GAL) file – block, column, row and ID, is created when the GPR file is read.

2.4.4 Accessor methods

`pmpa` inherits the standard accessor methods for Bioconductor `eSet` objects: The feature unique identifying name can be accessed by `featureNames()`

```
> featureNames(rawdata)[1:5]
[1] "LIYSQVLFKGGQCPS_1_1_1" "HVLLTHTISRIAVSY_1_2_1"
[3] "TKVNLLSAIKSPCQR_1_3_1" "TPEGAEAKPWYEPIY_1_4_1"
[5] "GGVFQLEKGDRLSAE_1_5_1"
```

Sample unique identifiers are accessed by `sampleNames()`

```
> sampleNames(rawdata)[1:5]
[1] "NCTRL2" "LVP086" "LVP109" "LVP074" "LVP085"
```

The phenotype data annotation is accessed by `pData()` and the feature annotation by `fData()`

```
> pData(rawdata)

> fData(rawdata)[1:5, ]
      ID Block Column Row      Name
1 LIYSQVLFKGGQGCPS  1     1   1 LIYSQVLFKGGQGCPS-TNF-a_133
2 HVLLTHTISRIAVSY  1     2   1 HVLLTHTISRIAVSY-TNF-a_149
3 TKVNLLSAIKSPCQR  1     3   1 TKVNLLSAIKSPCQR-TNF-a_165
4 TPEGAEAKPWYEPIY  1     4   1 TPEGAEAKPWYEPIY-TNF-a_181
5 GGVFQLEKGDRLSAE  1     5   1 GGVFQLEKGDRLSAE-TNF-a_197

      Annotation protein
1 TNF-a_133 TNF-a
2 TNF-a_149 TNF-a
3 TNF-a_165 TNF-a
4 TNF-a_181 TNF-a
5 TNF-a_197 TNF-a
```

Accessing individual columns of the phenotype data can be achieved by using the \$ operator

```
> rawdata$Cases_Controls
 [1] NA  0  1  0  1  1  0  1  0  0  1  1  0  0  0  0  0  1  0
[20]  1  0  1  0  0  0  1  1  1  0  1  0  1  1  1  1  1  1
[39]  0  0  0  1  1  0  0  0  1  1  0
```

In addition to these standard methods a number of custom functions are available to access the *assayData* slots. The *fg*, *bg*, *bmedian*, *bmean*, *dia* and *flags* methods access the foreground, morphological background, median local background, mean local background, spot diameter and spot quality flag matrices. Summarised data that is written to an *ExpressionSet* object can be accessed using the Bioconductor standard *exprs* function.

2.4.5 Data quality assessment and visualisation

An essential stem in any microarray analysis is an assessment of the data quality. *pmpa* provides a number of methods to visualise the raw and pre-processed data to assist quality assessment:

plotSubarrayBlocks produces a boxplot of intensities from a single array organised by print block. Conventional spotted arrays are printed using a pin

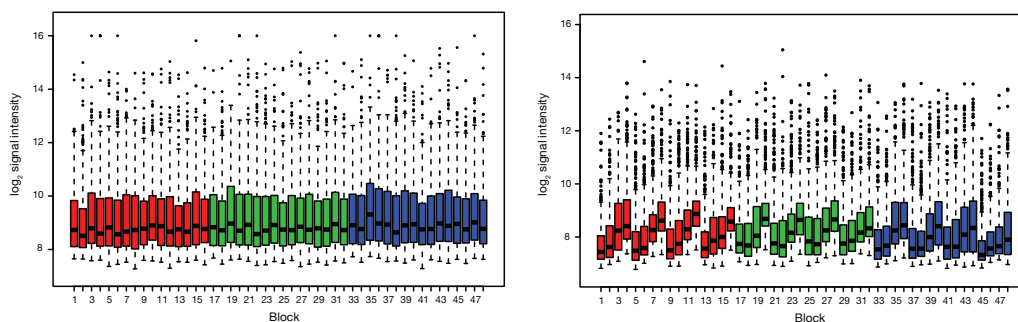


Figure 2.2: Block (print-tip) boxplots from two peptide microarrays

The above plots show the \log_2 transformed signal intensity from features on two peptide arrays organised by block (print-tip group). On this array the blocks are organised into a 4 x 4 subarray – therefore each colour represents a replicate subarray, and each group of four blocks represents one row of blocks across the array. The left plot shows relatively even distribution of signal across each of the blocks and between the subarrays. In contrast the right plot shows a trend of increasing intensity from left to right across each row of the microarray.

spotter – the print head comprises one or more metal points which sequentially dip into a solution containing the peptide probe and then deposit it on to the array surface. Each pin is responsible for printing one block – hence an array with an arrangement of 4 blocks x 4 blocks would be printed with a print head with a 16 x 16 arrangement of pins. These blocks – sometimes called print-tip groups should have a similar distribution and mean if, as is typical practice, the array is designed with evenly distributed probes. The boxplot identifies any between block intensity trends (Figure 2.2). The plot is called by

```
plotSubarrayBlocks(rawdata, arr = 1, transform = "log2")
```

where *rawdata* is our *MultiSet* object and the argument *arr* refers to the microarray to be plotted (1 is the 1st array in the series, 2 the 2nd etc). Finally a transform function allows the use of any valid function call or expression to be applied to the data prior to plotting. The transform argument defaults to \log_2 i.e. a base 2 logarithmic transformation. To plot without any data transformation set `transform = "none"`.

`plotImage` reconstructs the scanned microarray image using the intensity values read into the foreground and background *assayData* slot (Figure 2.3). This method is similar to the `imageplot` function from the *limma* package (Ritchie et al., 2015) or the `image` method implemented in the *marray* package (Yang et al.,

2009). The reconstructed microarray image allows direct visualisation of spatial trends in the foreground and background and is a fast way to identify any gross inconsistencies in the array particularly if access to the original scanned images is not possible.

`plotSubarrayScatter` produces a set of scatter plots with the intensity data of each subarray plotted against each other (Figure 2.4). As the subarrays are technical replicates a hypothetical perfect array would have a distribution of points along the plot diagonal. Any deviation from that indicates a spot that varies from its replicate. Similarly `plotSubarrayClosestValues` produces a scatter plot but uses the closest two values from a standard triplicate subarray high density peptide array (Figure 2.4 bottom right plot). This can be used to assess the effect of removing outliers; gross deviation from the main diagonal suggests that the array should be excluded from further analysis. Simple linear modelling is also included on each plot giving the modelled slope (β) and coefficient of determination (R^2) This implementation of between subarray scatterplots is essentially the same as described in Stephenson et al. (2015) with optimisation of the code for use with the `pmpa` data structure.

```
> plotSubarrayScatter(rawdata, arr = subarray = c(1,2), flagval = -100,  
+ transform = "log2")
```

```
> plotSubarrayClosestValues(rawdata, arr = , transform = "log2")
```

The argument *arr* indicates the array to be plotted. The argument *subarray* indicates the subarrays to be plotted; subarrays are numbered sequentially from the top of the array to the bottom. The function also allows flagged spots to be highlighted using the *flagval* argument – for GenePix® processed arrays any spots flagged ‘bad’ are given the default flag value of -100.

`plotSubarrayDensity` plots the kernel density estimation for each of the subarrays (Figure 2.4 bottom left plot). A hypothetical perfect array should have density plots of each subarray perfectly overlying each other. Deviations suggest a change in distribution or overall intensity between subarrays. In many cases small variations can be corrected by within-array normalisation but density plots should be rechecked after pre-processing.

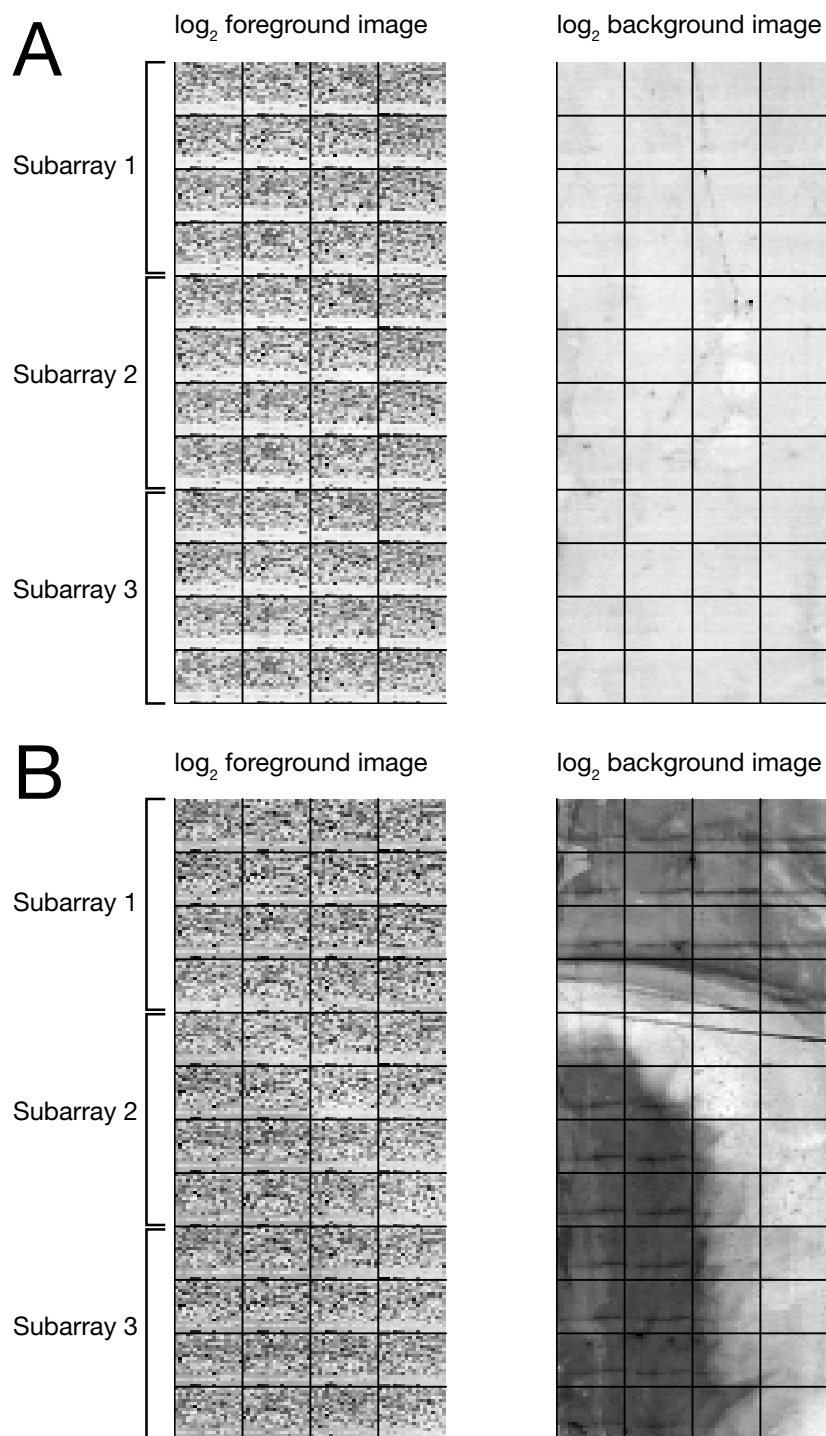


Figure 2.3: Foreground and background image plots

The above plots show reconstructed intensity data arranged by spatial location on the array slide. The left plots show the signal originating from the peptide spot itself (foreground data) and the right plots show the signal from a sampled region surrounding each spot (background data). Array A – (upper images) has a low and even background with no obvious inconsistencies in the foreground data. Array B (lower images) in contrast, has a significant background variation suggesting some inconsistency during the processing or manufacture. The foreground variation appears subtle and may be amenable to background correction or within array normalisation.

A convenience function `arrayQAp1ot` groups together the above functions on to a single output plot (Figure 2.4). Similar user defined plot collections can be easily created using the standard R functions `layout` or `par(mfrow=c(nrows, ncols))`.

```
> arrayQAp1ot(rawdata, arr = 1, transform = "log2", flagval = -100)
```

2.5 Pre-Processing peptide array data

2.5.1 Background correction and transformation

Fluorescence microarray data is typically logarithmically transformed to stabilise intensity dependent variance (Quackenbush, 2002). In *pmpa* transformation is combined with background correction in the `arrayBGcorr` method. The default arguments of `arrayBGcorr` are no background correction and a base 2 log transformation

```
> bgcorrdata <- arrayBGcorr(rawdata, method = "none",  
+                           transform = "log2"  
+                           )
```

Background correction of peptide array data is discussed in greater detail in chapter 3 but the *pmpa* package provides a number of methods listed below. All methods are implemented before data transformation:

1. Subtraction of background from foreground (any resulting negative intensities are set to 1) – `method = "subtract"`
2. Edward's method (Edwards, 2003) – `method = "edwards"`
3. Normal-Exponential convolution (Ritchie et al., 2007; Silver et al., 2009) – `method = "normexp"`
4. Multiplicative (Ratio) correction (Nahtman et al., 2007; Reilly and Valentini, 2009) – `method = "ratio"`

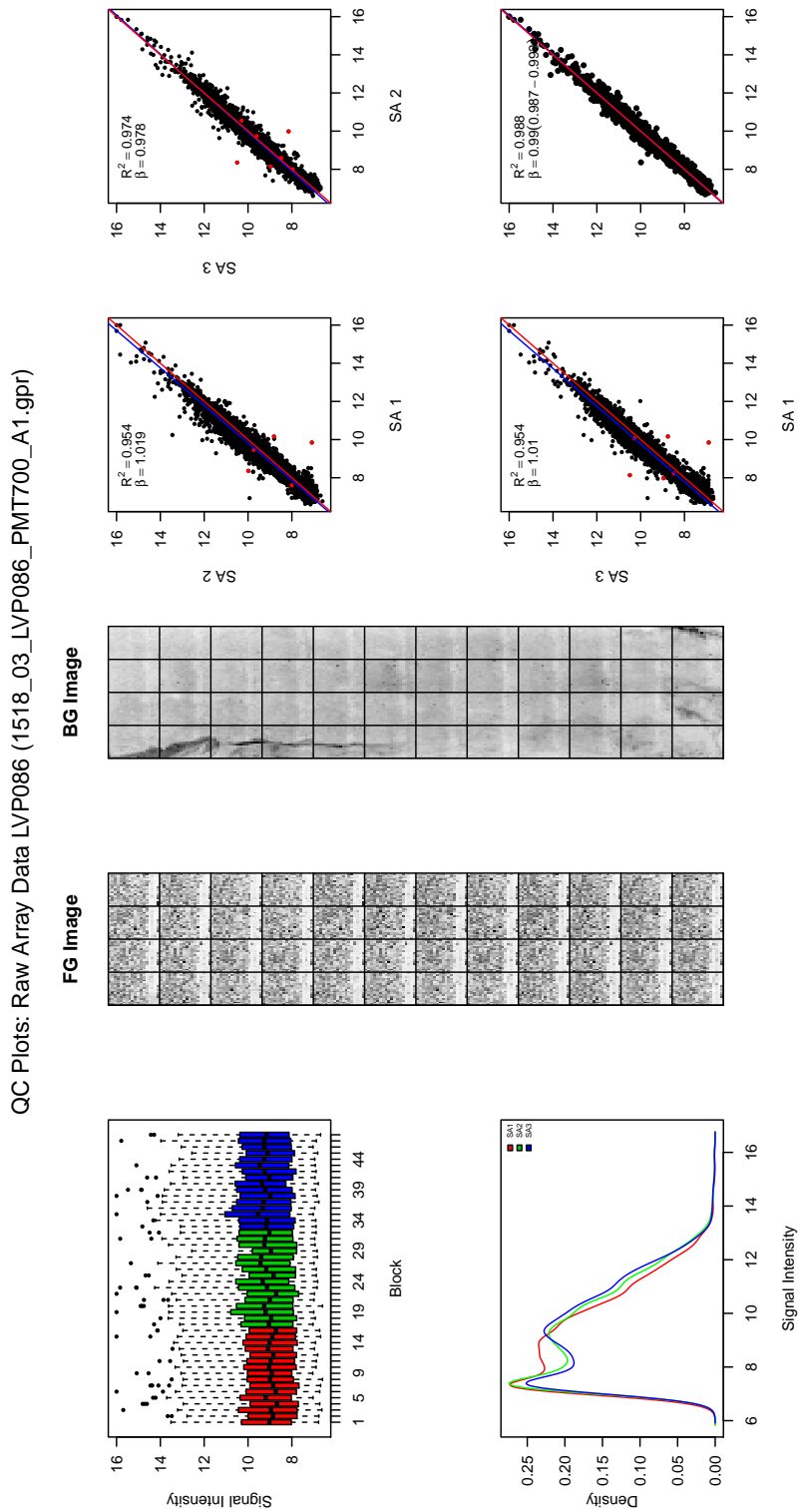


Figure 2.4: PMPA quality assessment plots - arrayQAplot

A collection of plots designed to visualise raw peptide microarray data to help quality assessment. In the upper left the boxplots show the log transformed signal intensity for each block (print-tip) on the array. Below that is kernel density plot for each subarray. Moving right are the log transformed foreground and background intensity image plots, and on the right are between subarray scatterplots (flagged values highlighted in red) and a closest value scatter plot with outlier values removed.

ORIGINAL IN COLOUR

2.5.2 Normalisation

Peptide array normalisation is discussed in chapter 4 but can be grouped into within-array or spatial methods (where intensity values are normalised on each array independently), and between array methods (where overall intensities are adjusted to be comparable across arrays).

Within-array and between-array normalisation is achieved using `normWithinArray` and `arrayNorm` respectively e.g.

```
normdata.wa <- normWithinArray(bgcorrdata,
                              method = "block.scale"
                              )

normdata <- arrayNorm(normdata.wa,
                      method = "quantile"
                      )
```

2.5.3 Summarisation

Summarisation of values for each peptide is achieved by calculating the mean (or median) of all replicates using `arraySummary`.

```
> summarydata <- arraySummary(normdata,
+                             method = "mean.closest",
+                             cv.threshold = 0.3
+                             )
```

The function calculates summaries based on the peptide sequence or probe name in the ID field of the *phenoData* slot. Typically there will be as many replicates as subarrays on the array. However, in some cases, an array may be designed with additional replicates (e.g. control features) which will be summarised to single values by the `arraySummary` function. Flagged values are discarded and are not used in the summary calculations. Non-flagged outliers can be identified using a Coefficient of Variation (CV) filter as described by Stephenson et al. (2015). If the CV of replicate values exceeds a threshold (defaulting to 0.3 in *pmpa*) then the most extreme value is discarded and the mean calculated from the remaining replicates. After summarisation all data and annotation is written to a new *ExpressionSet* object.

2.6 Analysis of pre-processed data

Representation of pre-processed data as a standard Bioconductor *ExpressionSet* allows for existing analysis methods to be applied easily to the peptide array data. Although the *ExpressionSet* class was originally designed for use with gene expression arrays as its name suggests, it works very well at holding summarised peptide array data. More importantly, many existing analysis functions that are applicable to peptide array analysis operate on *ExpressionSet* objects.

For example, analysis of differential identification by empirical Bayes moderated t-statistics implemented in *limma* is carried out by the `lmFit` function. `lmFit` can use the signal intensity data from the *ExpressionSet* as well as native *limma* MAlist objects. Similarly filtering by variance or by intensity can be a useful operation prior to statistical analysis - this can be accomplished easily using the functions provided by the *genefilter* package (Gentleman et al., 2013). This is a key difference of *pmpa* to the *pepStat* package - although both use Bioconductor methods, the latter defines a custom *PeptideSet* class making integration with external packages more difficult.

2.7 Example pre-processing script using pmpa

The following is an example of a script that would be executed by the end user. The script imports a set of text files containing GPR file and annotation information, and then imports a set of microarray data from those GPR files. It saves a document containing QA plots and then pre-processes the data using a log transformation, intra-array normalisation and mean summarisation.

```
library(pmpa)

# Read file data from external text file
files <- read.delim("./Data/fileNames.txt",
                   stringsAsFactors = FALSE
                  )

# Read phenotype annotation
pheno.annot <- read.delim("./Data/sampleAnnotation.txt",
```

```

        stringsAsFactors = FALSE
    )

# Read feature annotation
feature.annot <- read.delim("./Data/featureAnnotation.txt",
    stringsAsFactors = FALSE
)

# Read GPR Files
rawdata <- readArrays(samplename = files$sampleName,
    filename = files$fileName,
    path = files$path,
    wavelength = 635
)

# Add annotation
rawdata <- annotateArrays(rawdata,
    pheno = pheno.annot,
    feature = feature.annot
)

# Check QA plots - write to pdf file for review
pdf("./QA_Plots.pdf", paper = "a4r")
arrayQAPlot(rawdata, transform = "log2", flagval = -100)
dev.off()

# BG correction and log transform data
bgcorrdata <- arrayBGcorr(rawdata,
    method = "none",
    transform = "log2"
)

# within array block normalisation
normdata <- normWithinArray(bgcorrdata,
    method = "block.scale"
)

# Summarisation - mean of replicates with CV filter
finaldata <- arraySummary(normdata,
    method = "mean.closest",
    cv.threshold = 0.3
)$
    
```

2.8 Conclusions and future work

Peptide arrays are an important tool for understanding antibody - antigen interactions. However studies involving peptide arrays have suffered from a lack of specific tools to assist analysis. The `pmpa` package utilises existing methods and data structure from the Bioconductor project allowing integration with other packages, yet is designed specifically for a peptide array platform. Many peptide array studies have been published using customised analysis scripts or using existing genomic microarray technology software. However, often this means that scripts will be specific to a particular manufacturer's microarray design and comparison between studies can be difficult when many different methods are in use.

At present the `pmpa` package has only been tested with microarrays developed by JPT Peptide Technologies GmbH (Berlin, Germany) and only imports GPR files created by or compatible with Genepix® software. Planned future work includes additional work on the `readArrays` function to allow compatibility with other image processing software and testing with other array platforms. `pmpa` remains in active development but I plan to submit the package to Bioconductor for publication and distribution in due course.

Chapter 3

Background correction for peptide microarrays

3.1 Introduction

The measured signal from a peptide microarray does not only originate from fluorescence given off by labelled antibody binding to a peptide probe, but also from a number of other sources known as the 'background'. These may include non-specific binding of fluorescent secondary antibody to the array surface, auto-fluorescence of the glass or deposits remaining after processing, or electronic or optical noise from the scanner itself. Removal of this non-specific signal from the foreground intensity, termed 'background correction' is important step in pre-processing. Comparisons of background adjustment methods have been addressed for cDNA microarrays (Ritchie et al., 2007), and high density oligonucleotide arrays (Bolstad et al., 2003) but no such studies have been undertaken to establish an optimal method applicable to peptide arrays which have very different physiochemical properties and differing sources of non-specific background fluorescence.

3.2 Estimating background signal

The estimation of the microarray background is performed by image analysis software. There are three main methods for estimating background:

1. Global background – with one value for the whole array
2. Local background – each feature has an associated background value
3. Regional modelled background – background values are estimated for regions of the array encompassing several features.

Global background measures are simple to implement and were often used by older software that only sampled pixels from the feature segmentation area. Typically the value of the global background would be estimated from the feature intensity (foreground intensity) of negative control spots on the array. It has the disadvantage of disregarding spatial variations in the background intensity that are typically present on microarrays. Figure 3.1 shows image plots of the background signal of 2 microarrays from the mouse monoclonal antibody dataset described below. These show heterogeneity in the background signal suggesting that a global measure would over or underestimate the true background of many features.

As a result most image processing software utilise local background estimation. This is calculated from the mean or median pixel intensity sampled from an area surrounding each feature spot mask. Local background can capture the full range of spatial variation seen on microarrays. However, this can lead to corrected intensities having undesirable statistical properties with certain correction methods. In particular the background intensity, in some cases, can exceed that of the foreground leading to negative corrected intensities after subtractive correction. As log transformation is usually applied after background correction, these negative values become truncated from the dataset. Various correction methods are available that avoid this effect, however they are essentially dealing with a problem that was introduced by the increased background variance that is captured using a local calculation method.

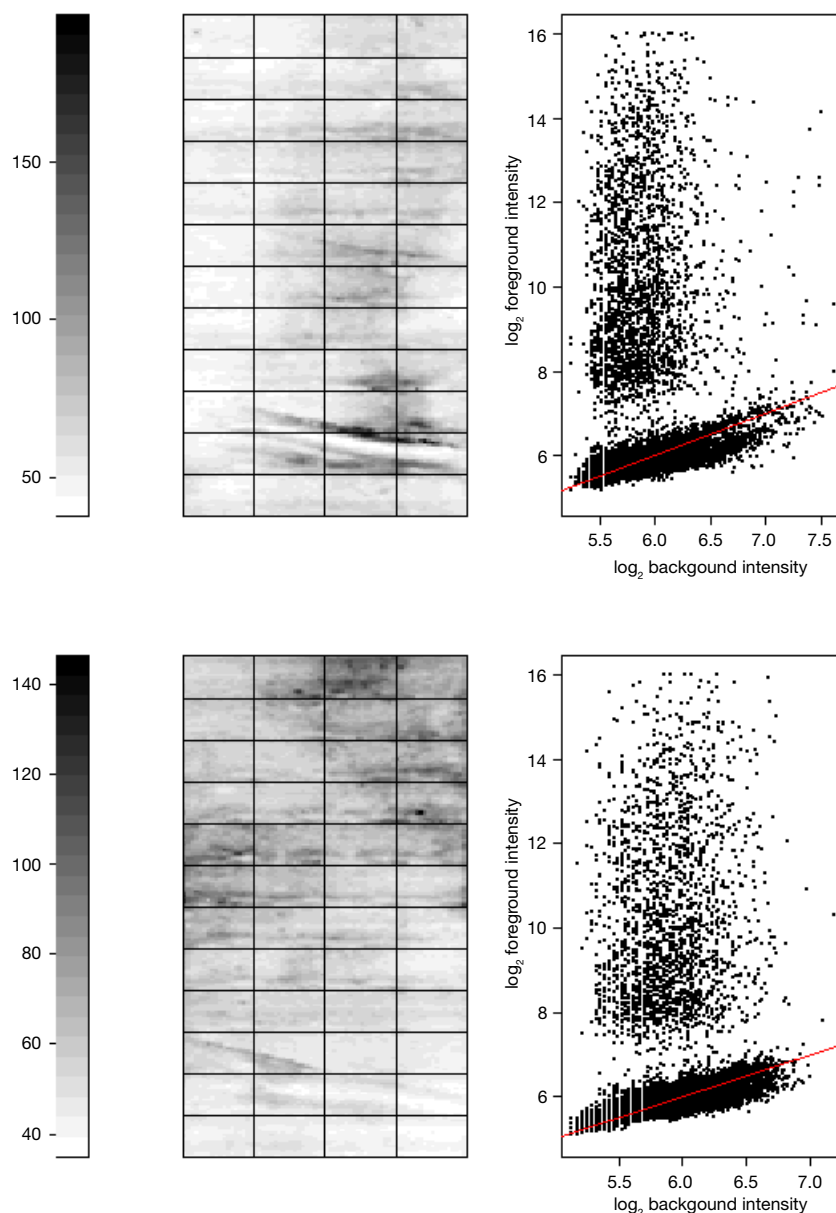


Figure 3.1: Background image plots and foreground / background scatter plots

Two arrays from the mouse monoclonal antibody (mAb) dataset are shown above. The imageplots demonstrate the spatial heterogeneity of the background (darker signals denote higher background intensity). The scatter plots show a population of features closely associated with the background (the red line shows foreground = background) and a separate population of responding peptides that do not clearly track the background values.

ORIGINAL IN COLOUR

The third group of methods for estimating background intensity do so by applying non-linear filtering algorithms to model the background regionally – i.e. not necessarily one background value per feature, rather one background value for a group of spatially related features. The morph background estimation method implemented in Spot® software (CSIRO, Australia) and GenePix®Pro (version 6.0 and later) samples the pixel intensities over a fixed window size – typically a square window with a side length at least twice that of the feature to feature distance. It then applies an erosion function (i.e. the output value is set the minimum value of all the pixels in the window), followed by a dilation function (i.e. the output value is set the maximum value of all the pixels in the window). This generates an estimated background for the whole slide. In general morph and other modelled backgrounds such as Total Variation (TV) + L^1 (Yin et al., 2005) or rank filtering tend to produce a more precise and less biased estimate of the background compared to local methods (Bengtsson et al., 2004).

For the purposes of this thesis all microarray images were acquired using Genepix®Pro software (Molecular Devices, USA) and background estimates used for the following analyses are based on local median values from the segmentation areas shown in pink on figure 1.7.

3.3 Background Correction Methods

3.3.1 Subtraction

The traditional assumption for microarray data is that the background signal is additive to the true signal. Hence by subtracting the raw background intensity $Y_{(b)i}$ from the foreground signal $Y_{(f)i}$ one can obtain an estimate of the true signal μ .

$$\mu_i = Y_{(f)i} - Y_{(b)i} \quad (3.1)$$

As described above, when used with local background estimation this method suffers from producing negative corrected intensities. These are either truncated prior to transformation which has the disadvantage of losing information

(sometimes from numerous features over a whole dataset) or can be set to a local minimum. This retains the features in the data but does introduce additional bias. Another disadvantage of subtraction is that as low intensity features are shifted towards the lower end of the measured intensity range, log transformation tends to dramatically inflate the variance of these values. This so-called ‘fanning’ phenomenon (so called because of the fan pattern on an MA plot – see figure 3.3) is well recognised and has been reported in the literature for cDNA arrays (Bolstad et al., 2003; Bengtsson et al., 2004).

3.3.2 Offset subtraction

A very simple method of stabilising the variance of low intensity features is to add a small positive offset, k , to the data before subtracting the background signal. This is analogous to the started-log approach described by Rocke et al. (2003). A secondary advantage of this method is that if the value of k is greater than the minimum foreground intensity then no negative values are produced by subtraction.

3.3.3 Edwards correction

The Edwards algorithm is designed to avoid the ‘overcorrection’ of low intensity foreground signals giving rise to negative corrected values (Edwards, 2003). In this model background subtraction is applied when the foreground signal is above a threshold value (δ) and a smooth monotonic function applied when it is below. δ was set arbitrarily at 1 for this analysis.

$$\mu_i = \begin{cases} Y_{(f)i} - Y_{(b)i} & \text{if } Y_{(f)i} - Y_{(b)i} > \delta \\ \delta e^{(1 - (Y_{(b)i} + \delta / Y_{(f)i}))} & \text{if } Y_{(f)i} - Y_{(b)i} < \delta \end{cases} \quad (3.2)$$

Although producing strictly positive corrected values, the Edwards algorithm still suffers from inflated variance of low intensity features.

3.3.4 Signal to noise ratio correction

Background correction using the signal to noise ratio (SNR) – i.e. the log ratio of foreground signal to background signal - was proposed by Nahtman et al. (2007) as an optimal method of background correction for peptide arrays. SNR correction has the inherent advantage of avoiding any negative corrected values. However, unlike other methods, it assumes that the observed signal varies with the measured background in a multiplicative fashion. In fact, the observed signal (at least for low intensity features) varies linearly (i.e. in an additive fashion) with observed background as shown in the scatter plots in figure 3.1. For arrays with fairly uniform backgrounds, using this method should work well, however if the background is variable and noisy (as is typical of peptide arrays), SNR correction will result in amplification of those stochastic effects.

3.3.5 The normal-exponential convolutional model (Norm-exp)

The normexp model is a very common background correction method and is often applied to genomic microarrays as part of the RMA algorithm (Irizarry et al., 2003). The model proposes that the observed foreground signal $Y_{(f)i}$ is comprised of not only the true signal (modelled as $\mu \sim \text{Exponential}(a)$) and the measured background $Y_{(b)i}$ but also a normally distributed residual signal $B \sim \mathcal{N}(m, \sigma^2)$. Thus the conditional expectation of μ given that the observed background subtracted intensity $Y_i = x$ is given by

$$E(\mu|Y_i = x) = m_{\mu,Y} + \frac{\sigma^2 \phi(0; m_{\mu,Y}, \sigma^2)}{1 - \Phi(0; m_{\mu,Y}, \sigma^2)} \quad (3.3)$$

where $m_{\mu,Y} = x - m - \sigma^2/a$, ϕ denotes the normal probability density function and Φ is the normal cumulative distribution function. A commonly used variant of the normexp algorithm is to add an offset prior to subtraction as in method 3.3.2 in order to stabilise the variance of low intensity features. Typically the offset is much lower than the minimum foreground intensity as the algorithm by definition only gives a positive corrected value.

3.4 Aim

To evaluate background correction algorithms applied to peptide microarray immunoassay data.

3.5 Methods

Two datasets were used for background evaluation. The first set allows the corrected signal intensities to be compared against a known monoclonal antibody concentration in order to assess the precision of the background correction models. The second set allows the comparison of corrected signal to a control antibody spiked into normal serum. This allows an assessment of differential identification. Preparation and design of the *Clostridium difficile* peptide microarrays used in this chapter is described in detail in chapter 5.

3.5.1 Peptide microarray processing

Microarray slides were placed into individual humidified hybridization chambers (Camlab, UK). A microarray ‘sandwich’ was prepared by placing plastic spacers on the ends of the array and placing a dummy slide on top. Prior to use the dummy slide was washed firstly in Milli-Q water and then in 95% Isopropanol, dried and dust removed under a nitrogen stream. Sera were diluted in blocking buffer (Pierce Superblock T20, Thermo, UK) as described below. 300 μ l of the diluted sera was then pipetted into the space between the dummy slide and the array. The space fills evenly by capillary action avoiding air bubbles and filling artefacts. The chamber was then incubated at 37°C for 2 hours. After incubation the hybridization chamber and dummy slide were removed and the array slide was placed into a slide holder. Arrays were washed with 5 times with TBST buffer (Tris-buffered saline, pH 7.4, Na 150mM, K 2mM, 0.05% Tween-20) for 5 minutes with continual agitation.

Secondary antibodies – goat anti-human IgG AlexaFluor™ 647 (Invitrogen, USA – cat no. A-21445), and rabbit anti-human IgA (α chain) Cy3 (Jackson ImmunoResearch, USA – cat no 309-165-011)- were diluted to a concentration of 1 μ g/ml

in blocking buffer. Incubation with the secondary antibodies was carried out for 1 hour at room temperature with continual agitation. Following secondary incubation the slide was washed for a further 5 times in TBST buffer and then 5 times in Milli-Q grade water. Slides were dried by centrifugation for 5 minutes at 200g. The processed arrays were stored in the dark at 4°C until data acquisition. No longer than 48 hours elapsed between completion of processing and data acquisition.

Mouse mAb titration dataset

A titration series of mouse anti *Clostridium difficile* toxin B (TcdB) monoclonal antibody (MCA7438 – ABD Serotec, UK) was diluted in blocking buffer in the following concentrations: buffer only control, 100 fg/ml, 1pg/ml, 10pg/ml, 100pg/ml, 1ng/ml, 10ng/ml, 100ng/ml. These were incubated on a peptide microarray containing *C. difficile* sequences (see chapter 5). Arrays were probed with fluorescent labelled goat anti-mouse IgG Cy5 secondary antibody.

Rabbit pAb spike-in dataset

A rabbit anti *C. difficile* toxin A (TcdA) pAb preparation (PA1-85042, Pierce Biotechnology, IL, USA) was spiked into normal rabbit serum (Pierce Biotechnology, IL, USA) diluted to 1 part in 100 in blocking buffer. The rabbit pAb was prepared using a synthetic peptide immunogen (C)TIDGKKYYFN conjugated to keyhole limpet haemocyanin (KLH) and corresponds to a conserved repeating sequence within the C-terminal domain of TcdA. The polyclonal rabbit sera was purified by Protein-G antibody affinity chromatography. Spike-in concentrations were 2µg/ml, 4µg/ml and 8µg/ml. These were incubated on a *C. difficile* peptide array as previously described. Buffer only negative controls and a serum only control were also incubated. Arrays were probed using a goat anti-rabbit IgG Cy5 fluorescent secondary antibody. Image acquisition and analysis were performed as described above.

3.5.2 Image acquisition and processing

Array tagged image format file (TIFF) images were acquired using a Genepix® 4300A scanner and image analysis performed with Genepix®Pro version 7.2 software (Molecular Devices, USA). Default image segmentation was applied with foreground and background intensities taken as the median pixel intensity in the sampling area (see section 1.4.1 and figure 1.7). GPR files were analysed using the `pmpa` package for R version 3.0.1 (R Development Core Team, 2014). Following acquisition slides were archived at 4°C in sealed boxes in a dry nitrogen atmosphere.

3.5.3 Data pre-processing

Raw GPR files were imported using the `readArrays` function of `pmpa`. Each dataset was then background corrected using the methods described above. The final corrected intensity was given by the mean of the signal intensity from the 3 replicate probes in each subarray with additional filtering to remove outliers as described in Chapter 2.

3.6 Results

3.6.1 Assessing variation induced by background correction.

The background intensity image plots for two arrays from the rabbit pAb dataset are shown in figure 3.2 below. Both of these arrays were incubated with normal rabbit serum only (no spike in) at the same concentration under identical conditions. They show a typical pattern of random background variation.

Figure 3.3 shows the variation in signal intensities for a comparison of these two arrays. As the arrays are biologically identical there should be no differential identification seen i.e. M values should be 0. The MA plots show a marked difference in the amount of variation induced by the different background correction

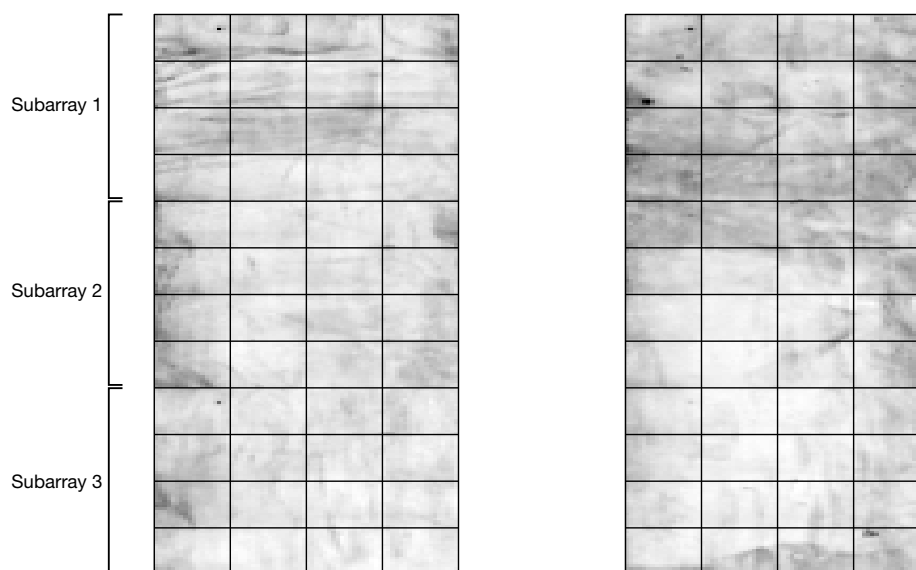


Figure 3.2: Background image plots for two arrays incubated with identical sera

Each image plot is a spatial reconstruction of the background signal originating from the array. The gridlines on the plot delineate blocks of peptides printed by a single print-tip. The array is comprised of 3 identical sub-arrays (each 4 x 4 set of blocks). Darker areas of the image indicate higher background signal. The plots show that even for two arrays incubated under identical conditions with the same biological material, there is a random, yet spatially coherent, variation of the background signal.

methods. In particular, the subtraction and Edwards methods produce M-values that vary substantially at low values of A.

In addition, figure 3.3 also demonstrates that background correction methods with less variable M values also compress the range of A-values offsetting them further away from 0. Although for a self-self comparison as shown in this figure this is desirable as there is no biological difference between the arrays, for samples where there should be genuine differential identification, compression of A-values may make that difference harder to detect. The following analyses aim to determine which method achieves the best trade-off between low variability and high detection sensitivity.

3.6.2 Assessment of precision

Each array in the mouse mAb dataset was analysed by fitting peptide-wise 2nd order polynomial (quadratic) regression models to the data across the range of measured antibody concentrations. Thus the measured signal intensity Y_i for the

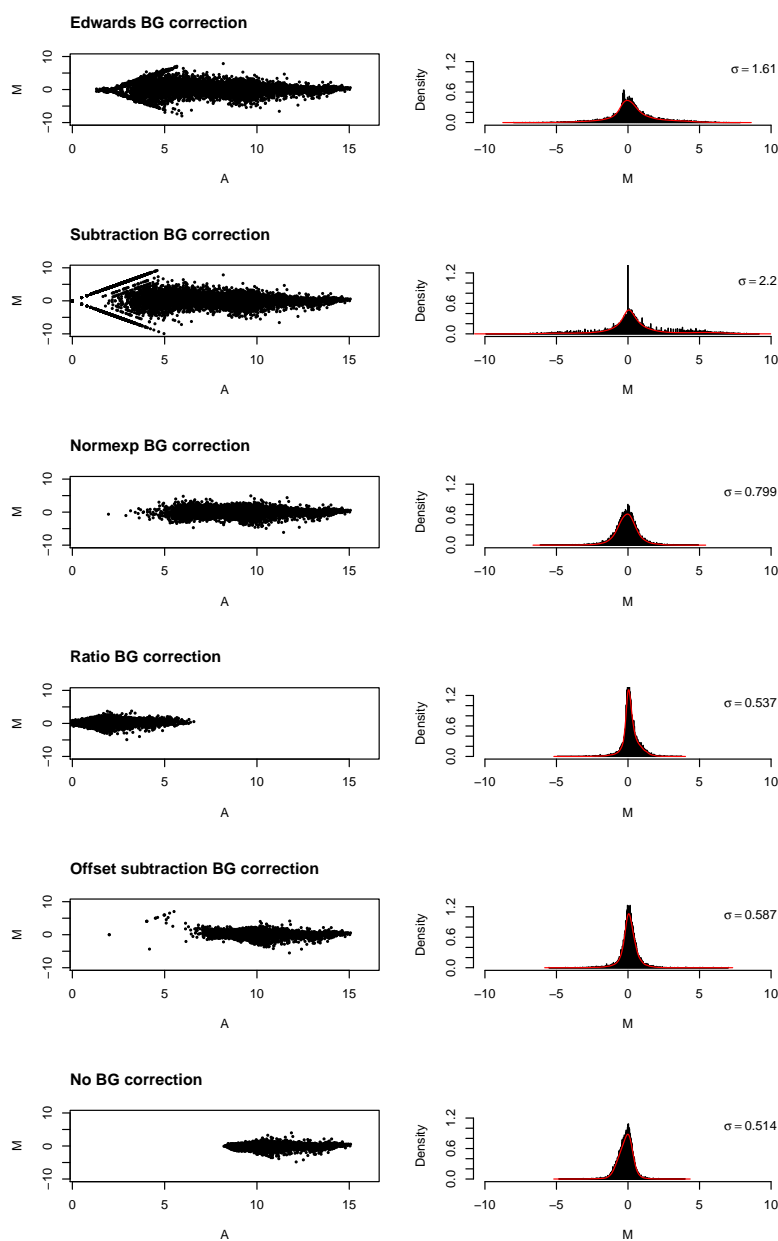


Figure 3.3: MA-plots of background correction methods for a self-self comparison. Each plot shows the log ratio (M) of a comparison between subarray 1 and subarray 2 of a single array on the y axis, and the log average peptide signal (A) of comparison on the x axis. The subarray signals should be biologically identical hence the true M value should be zero. The amount of M variance is shown graphically as a histogram and kernel density (red line) in the plots on the right (the σ value denotes the standard deviation of the distribution).

i th probe is modelled as

$$Y_i = \beta_2 x_i^2 + \beta_1 x_i + \beta_0 + \varepsilon_i \quad (3.4)$$

where β_1 and β_2 are the regression coefficients, ε_i represents the measurement error of the model for the i th probe and x_i is the antibody concentration. Fitting these regression models allows assessment of how precisely the observed background corrected data fits the predicted statistical model. This can be enumerated by the residual variance (σ_i^2). Fitting this model to all peptides in the dataset yields a comparison of variances for the different background correction methods (figure 3.4). The vertical scale in the figure is the \log_2 transformed variance – hence each unit increase on this scale represents a doubling of the variance (i.e a halving of statistical information).

The subtraction, Edwards and normexp methods appear to produce variances greater than no correction at all, whereas an offset subtraction method gives the lowest overall variances of all the methods tested. However, this higher precision comes at the cost of dynamic range. Figure 3.4 shows the spread of log fold changes for arrays in the mouse mAb dataset. The spread of fold changes is less for the offset subtraction method compared to the others. It is notable that the experimental model in this case is comparing a monoclonal antibody to a buffer only negative control. Hence we would only expect a fold change greater than or equal to 0. The observation that the methods giving large dynamic range (subtraction, Edwards and normexp) also produce a number of negative fold changes suggests that these methods are introducing additional bias. Given that the difference between the best performing method (offset subtraction) and the worst performing method (subtraction) is merely the addition of a constant positive offset, I compared the effect of varying offset values on the precision and bias. Figure 3.5 summarises the findings; that precision is essentially a function of offset – the greater the offset the greater the precision gained. However, this is at the expense of a rapid falloff in the dynamic range of the data even with relatively small offsets being applied. Little difference is seen between the subtraction and normexp methods with offsets applied. Normexp tends to a higher precision for a given offset but loses more dynamic range.

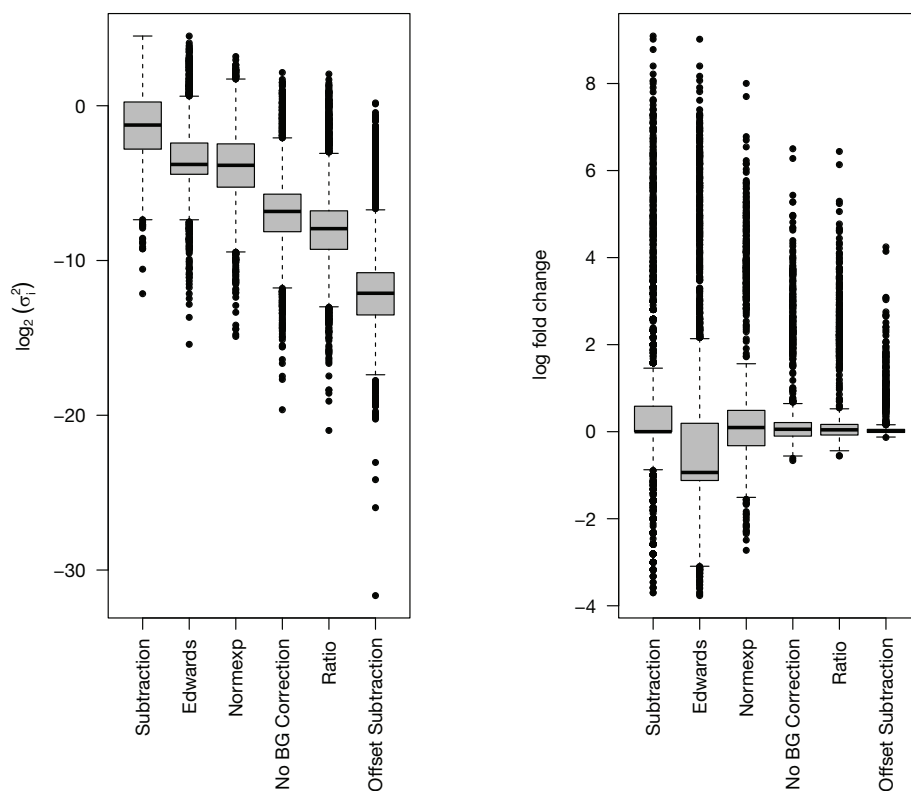


Figure 3.4: Boxplots of log residual variances of quadratic model fits (left plot) and log fold change compared to negative control array (right plot) for to all peptides from the mouse mAb titration dataset.

For each background correction algorithm the signal intensity change was modelled by a quadratic function and the residual variance (σ_i^2) calculated. The lower σ_i^2 , the more precisely the measured intensities follow the predicted model. Signal intensity fold changes were calculated for arrays in the mouse mAb dataset for each of the tested background correction methods. The methods are ordered from left to right by the spread of fold changes

3.6.3 Assessment of differential identification

Peptide microarrays used for serological analysis are typically employed in order to find differentially reactive peptide probes from a series of sera. Hence it is critically important that any pre-processing methods are optimised to minimise bias to differential identification. In order to assess this, the rabbit pAb dataset was used. The spike-in antibody identifies a peptide sequence from the C-terminus (receptor binding domain) of *C. difficile* Toxin A (TcdA). This region of the toxin is comprised of several repeating oligopeptide sequences and so the

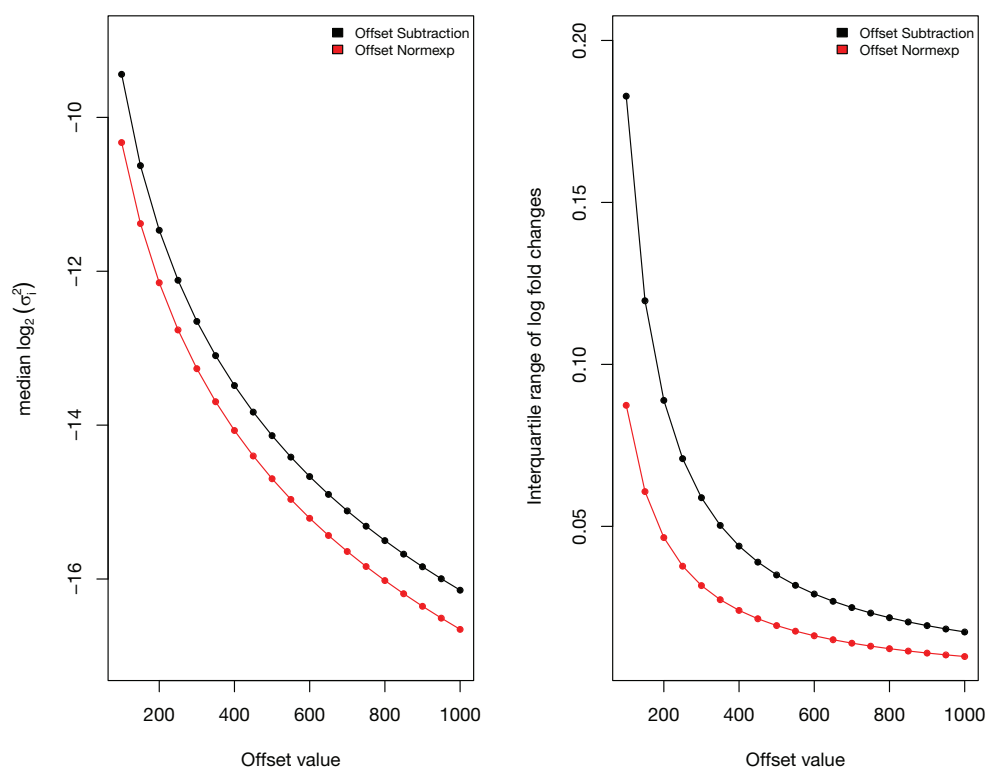


Figure 3.5: Plots of median log residual variances of quadratic model fits (left plot) and interquartile ranges of log fold change compared to negative control array (right plot) for varying amounts of added offset.

The residual variance and log fold changes were computed as for figure 3.4. However, varying positive offsets were added to the data prior to correction by subtraction or using the normexp model. The median log residual variance is shown on the left and the interquartile range (IQR) for fold changes on the right.

pAb immunogen sequence (or sequence homologue) is displayed several times.

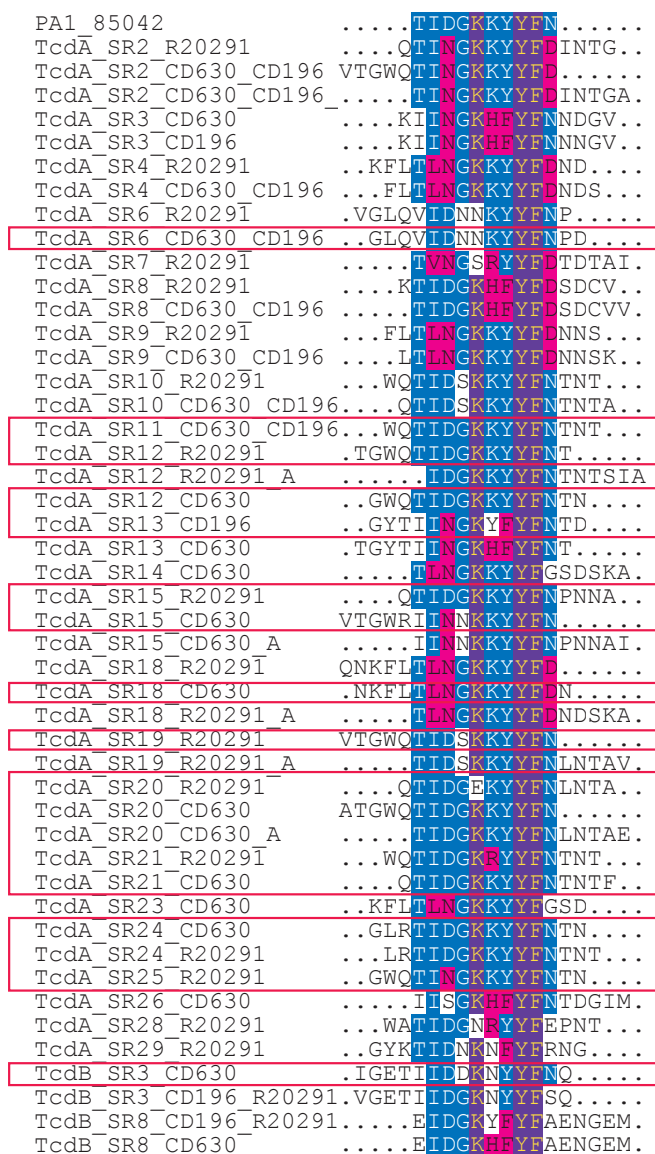
As the array peptides were comprised of overlapping sequences, there were a number of peptides spanning this region with only partial representation of the immunogen sequence. Hence in order to determine a set of genuinely differentially identified peptides an array was incubated with just the PA1 antibody. Peptides showing significant signal (fold change > 2 log compared to buffer negative control) excluding peptides reactive to normal rabbit serum alone and to the secondary antibody alone, were then aligned in a pairwise fashion to the immunogen sequence using the Smith – Waterman Algorithm (Smith and Waterman, 1981). Peptides with a significant alignment score ($p < 0.001$ after 1000 bootstrap replications) were selected. Figure 3.6 shows the array peptides

	2µg/ml Spike In	4µg/ml Spike In	8µg/ml Spike In
No BG correction	0.53 (0.40 – 0.66)	0.76 (0.62 – 0.91)	0.96 (0.93 – 0.95)
Ratio	0.51 (0.38 – 0.64)	0.78 (0.63 – 0.92)	0.96 (0.93 – 1.00)
Subtraction	0.61 (0.48 – 0.74)	0.83 (0.74 – 0.92)	0.94 (0.92 – 0.95)
Normexp	0.64 (0.45 – 0.75)	0.85 (0.73 – 0.97)*	0.97 (0.94 – 0.97)
Offset Subtraction	0.58 (0.42 – 0.74)	0.87 (0.74 – 1.00)*	0.99 (0.99 – 1.00)

Table 3.1: AUROC and 95% confidence intervals for PA1 spike-in series

with significant sequence similarity to the immunogen peptide. Only a subset of those peptides are significantly reactive to the antibody (shown in the red boxes) in a series of arrays incubated with just the PA1 antibody in a titration series. This set of 18 peptides was then used to assess differential identification.

A receiver operator characteristics (ROC) analysis was conducted in order to ascertain the performance of each background correction method for detection of the 18 peptides predicted to be differentially identified as shown in figure 3.6. The PA1 antibody was spiked into normal (non-immune) rabbit serum at three concentrations 2µg/ml, 4µg/ml and 8µg/ml. ROC plots of each correction method for each concentration are shown in fig 3.7. The diagonal line (slope = 1, intercept = 0) denotes no discrimination – i.e. a ROC curve on this slope shows the assay is no better than chance at finding the predicted differentially identified peptides. A perfectly discriminating assay will have a ROC curve that rises immediately to Sensitivity = 1, Specificity = 1 and then retains this sensitivity value for all measured specificities. This discriminating capacity can be captured by the area under the ROC curve (AUROC). An area of 0.5 corresponds to no discrimination whereas a perfectly discriminating assay will have an AUROC = 1.0. Table 3.1 summarises the AUROCs for the methods used. At a spike-in concentration of 2µg/ml the peptide array data is no better than chance for identifying the predicted peptides. In contrast at 8µg/ml almost perfect discrimination is achieved irrespective of the method chosen. This indicates that peptides giving high signals are less subject to the non-specific binding effects measured by the background intensity. However at 4µg/ml we see that offset subtraction and normexp correction both show a statistically significant difference ($p < 0.01$) between the method and no background correction at all.



- non conserved
- similar
- ≥ 50% conserved
- ≥ 80% conserved

Figure 3.6: Multiple sequence alignment of the PA1-85042 pAb antibody immunogen to array peptides

All array peptides were aligned against the immunogen sequence (Smith – Waterman algorithm) and sequences chosen if the alignment score was significant ($p < 0.001$ after 1000 bootstrap replications). Peptide sequences are labelled by protein (TcdA – Toxin A, TcdB – Toxin B), short repeat position, and *C. difficile* strain. The peptides highlighted in the red boxes correspond to those with significant signal on an array incubated with just the PA1-85042 antibody.

ORIGINAL IN COLOUR

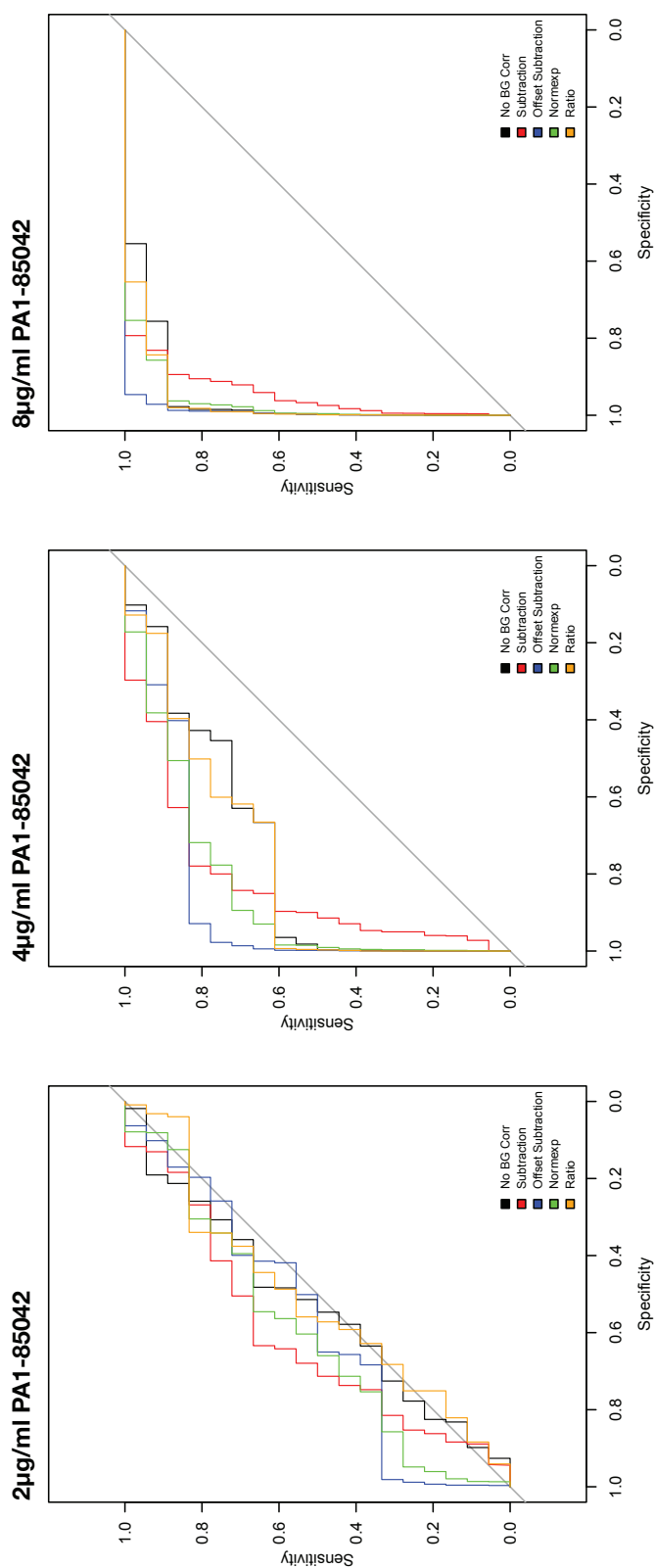


Figure 3.7: ROC plots for differential identification of PA1 binding peptides from antibody spiked into normal rabbit serum
The plots represent the three spike-in concentrations (from left to right 2 µg/ml, 4 µg/ml and 8 µg/ml). At 2 µg/ml no statistically significant differential identification can be found irrespective of the correction method used. In contrast, at 8 µg/ml differential identification is reliably found with any correction method.

ORIGINAL IN COLOUR

3.7 Discussion

The work presented in this chapter compares six methods of background correction for peptide microarray data. The precision and bias of the methods were assessed by comparison of the observed values from a monoclonal antibody titration series to predicted (modelled) intensities; and the power to find differentially identified peptides was assessed by the ability of the array to detect the signal from an antibody spiked in to a normal, non-immune serum.

This study includes a signal to noise ratio method as proposed by Nahtman et al. (2007) and Reilly and Valentini (2009). Essentially this method assumes that background signal exerts a multiplicative effect on the measured foreground signal. This is in contrast to the other methods used that assume additivity of the background signal. However, I could not find evidence of a multiplicative relationship. Rather low intensity foreground signals varied in a linear fashion (suggesting additivity) with the feature specific background.

The results of the precision study mirror the findings seen in cDNA arrays (Ritchie et al., 2007) where higher offset methods achieve increasing precision. The one exception to this finding is for the ratio method which actually gives the lowest offset value. Assay precision for ratio corrected data was comparable to no correction at all and better than the methods with no offset (subtraction, Edwards and normexp). Apart from this method, I found that precision is essentially a function of offset with the difference between correction methods being relatively smaller than the effect of changing the offset itself. Part of the reason why higher offsets achieve higher precision is due to the log transformation of the data. Although this is done to minimise the inherent heteroscedasticity of microarray data, variance stability is only effectively achieved at higher intensity values. As the intensity tends to zero, so the variance increases dramatically. Offset methods prevent this variance inflation by increasing the overall intensity. Another way this could be achieved is by using a different data transformation. The variance stabilisation normalisation (VSN) algorithm was developed in order to achieve exactly this (Huber et al., 2002). It uses an inverse hyperbolic sine (asinh) transformation to stabilise the intensity data across the entire measurable range. However, it is computationally more complex to implement and has not been utilised before for peptide array data.

In general, higher offsets achieve better precision at the expense of losing dynamic range. A lower dynamic range risks losing discrimination ability making the array less sensitive for lower intensity signals. We therefore compared the effect of each correction method on the ability of the array to find a set of differentially reactive peptides in a spike-in antibody experiment.

Contrary to expectation, the offset subtraction method actually shows the best discrimination of the spike-in probes despite having the lowest dynamic range. This method and the normexp method (with no offset) both showed a statistically significant increase of AUROC compared to no correction at all. Interestingly, although the ratio method showed better precision than the no offset methods, it did not perform better than no correction at all for differential identification. This suggests that it may perform adequately for a low noise assay such as the mAb titration dataset, but for noisier data such as a serum series, the amplification of stochastic effects may hinder its ability to discriminate truly differentially identified peptides.

I therefore conclude that background correction is important to peptide array pre-processing. As the microarray background appears to exert an additive effect on signal intensities, a subtractive method is optimal for correcting this bias. Offset subtraction was chosen as the method for all subsequent analyses in this thesis as it is simple to implement and the offset effectively prevents both data truncation from negative corrected values, and variance inflation of lower intensity signals causing a loss in overall precision.

Chapter 4

Normalisation of peptide microarray data

4.1 Introduction

Normalisation of a microarray dataset is a mathematical transformation of the signal intensities in order to remove or minimise any non-biological variation. Non-biological differences can include spatial variation in feature printing (print-tip variation), non-uniformity in the primary incubation or labelling, and surface artefacts, variable ambient conditions during processing, variation of labelling efficiency between process batches, and scanner biases.

In this sense, background correction as discussed in Chapter 3 can be considered a form of normalisation. However background correction adjusts for non-specific binding of antibody to the microarray – an effect that can be estimated from the signal intensity of the region surrounding each peptide feature. Because it corrects for a specific measurable source of error, it is usually considered separately to normalisation. For the purposes of this thesis normalisation is defined as a data transformation intended to correct for non-biological variation that is not accurately measurable or that originates from an unknown source.

There are several methods that have been developed in order to correct for these biases. Although the details of each method may differ, they all rely on a single fundamental assumption – that there should always be at least one measurable

property on each array or each region to be normalised that remains constant. For example, normalising a set of arrays to a control probe relies on each array being printed with a probe that has a constant biological reactivity irrespective of the conditions being tested. This form of normalisation is often used for quantitative PCR (qPCR) or low density transcription arrays by relying on the constant expression of a 'housekeeping gene' such as β -actin or Glyceraldehyde 3-Phosphate Dehydrogenase (GAPDH). If we assume that the expression of the gene is constant, any variation seen in the intensity of the housekeeping gene probe must therefore be due to a non-biological source. Normalisation is achieved by adjusting the overall signal intensity of the arrays such that the signals from the housekeeping probes are rendered equal. However, the assumption that housekeeping gene expression is constant is not always true and may vary significantly between comparator groups (Glare et al., 2002). This can cause significant confounding in the normalised signal intensities. Other methods of normalisation may not require control probes but rely on other properties being constant – for example, global scaling methods rely on the mean or median signal from all the probes on the array being equal, quantile normalisation makes the assumption that not only the median intensity but the signal distribution should also be equal and constant (Quackenbush, 2002).

The overwhelming majority of normalisation methods were developed for pre-processing gene expression microarray data. Although many have been applied to peptide array data, there are a number of potential problems associated with the application of these methods to peptide array immunoassay data. Global normalisation methods rely on the assumption that overall signal intensity should be constant between arrays. For high content gene expression arrays where typically an entire genome may be represented on one array, it is a reasonable assumption. Experimental steps ensure that the total amount of mRNA used from each biological sample is equal, and in general we expect only a small proportion of genes to be differentially expressed with roughly equal numbers being up-regulated as down-regulated. For high content peptide arrays this assumption may be true when dealing with a monoclonal antibody or purified primary antibody it would be reasonable to expect a small number of signals binding events relative to the number of probes on the array. Hence as the remaining peptides are not differentially identified we would not expect the median signal

on each array to differ significantly. With different sera though there may be very different low level binding signal between different arrays. Although these signals may be of relatively low intensity, because they affect a large number of peptides they have a dramatic effect on the median signal. Hence global rescaling of the array signal to equalise medians may unduly increase or decrease high intensity signals that we are most interested in. Moreover, usually peptide arrays are customised to include sequences from proteins of interest. Typically we would expect increased reactivity in one group compared to the other. This is particularly true for low-content arrays (and indeed low content DNA arrays – sometimes called ‘boutique’ arrays). Global normalisation in this situation can potentially destroy true biological variation and normalisation to an internal or external control is more commonly employed.

As normalising serological peptide arrays can present a number of difficulties, one option favoured by some is not to normalise the data at all. In this case we are making the assumption that any non-biological variation is small and that normalisation is likely to introduce significant artefactual signal. No consensus exists in the literature as to the optimal way of normalising peptide array serology data. However, two studies have looked at comparing normalisation methods (Renard et al., 2011) compared Z-scale normalisation (ie. conversion of intensity values to Z-scores – the mean intensity of each array is then set to zero; essentially equivalent to scaling to a global mean) to a between and within array linear model proposed by (Nahtman et al., 2007), and a within-array linear model normalisation using control peptide sequences within each block.

They used a Placental Specific Protein 1 (PLAC-1) antibody spiked into a pooled serum from six patients and compared sensitivities and specificities for identifying PLAC-1 peptides on the array at two concentrations. They found sensitivities and specificities were approximately equal for all of the methods at a high sensitivity and as expected, fell at the lower concentration. However they found that the control peptide based normalisation method showed less of a fall in sensitivity and specificity for the lower concentration. It should be pointed out that their comparison was not just for the normalisation alone but included the entire pre-processing pipeline. They also proposed a General Mixture Model (GMM) for filtering secondary antibody binding and for signal calling (not used for the comparator methods) which may have affected their conclusions. They

also treated non-reactive PLAC-1 peptides as false negatives for the sensitivity and specificity calculations. While we would expect some PLAC-1 peptides to be reactive it is extremely unlikely that all would show a reaction to the antibody. The proportion of false negatives is therefore likely to be overestimated in their calculations.

Imholte et al. (2013) compared quantile normalisation, a linear model normalisation as described above and no normalisation with a novel physiochemical model normalisation. They proposed that the non-specific binding to array peptides is influenced by the physiochemical properties (molecular weight, hydrophobicity, polarity etc.) of the constituent amino acids in the peptide. The amino acids in each peptide were scored using a physiochemical properties score described by Sandberg et al. (1998). The summated scores for each peptide on five composite physiochemical properties were then used to fit a linear model. The residuals of the model fit were used as the normalised values. The same background correction (Normexp), summarisation and signal call was used for all methods. They used a set of sera from participants in an HIV-1 vaccination study with vaccinated participants compared to a placebo group (Karasavvas et al., 2012). Like Renard et al. (2011) they calculated specificities and sensitivities but did so as a Receiver-Operator Characteristics (ROC) analysis to evaluate overall test performance. Reactive peptides in the Gp120 sequence that were expected to be immunogenic (C1 region, V2/V3 loop and C-terminus) were counted as true positives. The authors acknowledge that this is not a true reference truth – some participants will have generated antibodies to sequences outside these regions and some may not have responded at all. Nevertheless comparisons between the normalisation methods show a marginal benefit for the physiochemical model in one dataset with linear model normalisation being slightly better in the other. The difference in area under ROC curve (AUROC) for all of the methods was small and unfortunately confidence intervals were not stated.

4.2 Normalisation Methods

Normalisation methods can be broadly categorised into two groups: those requiring control probes, and those that rely on the global properties of all the probes

on the array.

4.2.1 Control Probe Normalisation

Endogenous controls

Normalisation to a housekeeping gene as previously described is an example of the use of an endogenous control. This relies on a constant biological signal endogenous to the samples being analysed. In practice, although housekeeping genes are believed to be constitutively expressed, their expression levels can actually vary between tissue types. Hence this may result in significant confounding of the normalisation process. The problem is even greater for serological peptide arrays as specific antibody levels may vary substantially between subjects meaning it is almost impossible to define a 'housekeeping' antibody. Nevertheless, normalisation to an endogenous control (typically immunoglobulin printed on the array) has been employed for previous peptide microarray studies (Hueber et al., 2007). Immunoglobulin control spots typically give a constant high level signal and so are an attractive target for normalisation controls. However, there are a number of problems associated with the use of immunoglobulin controls (Ngo et al., 2009). As they are proteins, they are more sensitive to degradation than peptide probes. Hence significant variation in signal intensities can occur between batches or over time which will confound the normalisation process. In addition, as the signal from the immunoglobulin probes is almost entirely due to the reaction with the secondary antibody, they cannot capture variation occurring in the primary antibody source.

Exogenous (spike-in) controls

Exogenous controls were first used for normalising DNA arrays. They are particularly useful for low density arrays where only a small number of genes are represented and where the overall expression might be expected to be uniformly up or down regulated. Using a global normalisation would remove that variation and so give misleading results. The technique involves using selected reference RNA (spike RNAs) which is added to the experimental RNA. The spike RNAs are

selected to have no sequence similarity to the genome being investigated and will have an analogous cDNA or oligonucleotide printed on to the microarray. Thus we expect the spike signal to always react in a consistent manner independent of any biological variation in the experimental RNA. Any variations seen in the spike signals must therefore represent undesired technical variation and the intensities of the other probes can be adjusted to compensate (Fardin et al., 2007). Spike-in normalisation relies on no reactivity between the experimental RNA and the complementary spike-in array probes (and the spike in RNA itself). Because DNA hybridisation is a relatively specific process with little cross-reactivity this tends to work well. However, for antibody-antigen interactions, where cross reactivity is common and may yield strong signals it is difficult to ensure that a spike-in antibody will be unreactive to serum components (serum proteins / other antibodies) and that the serum constituents will be unreactive to the complementary spike-in array probes. This significantly limits the usefulness of spike-in normalisation to small scale experiments where sufficient control incubations can be performed to ensure that no cross-reactivity is occurring.

4.2.2 Global Normalisation

Scaling to a global mean or median

This algorithm applies a constant scaling factor to the signal from every probe so that the mean or median intensity is the same on all arrays. Because a constant scaling factor is applied, the relative intensity between features in the same channel is unchanged. The algorithm for global median scaling is shown below:

- 1: Let \mathbf{X} be a $p \times n$ matrix of \log_2 transformed signal intensities with rows representing peptide features and columns representing arrays
- 2: Choose a column of \mathbf{X} to be the baseline array (column j)
- 3: Let $\tilde{X}_j = \text{median of column } j$
- 4: **for** $i = 1$ **to** $i = n, (i \neq j)$ **do**
- 5: Let $\tilde{X}_i = \text{median of column } i$
- 6: $\beta_i = \tilde{X}_j / \tilde{X}_i$
- 7: $X_{norm,j} = X_i \cdot \beta_i$
- 8: **end for**

This results in normalised arrays having a median of 0. Often, in order to return the normalised intensities to the same range as the original signal, a constant is added to each array. The `pmpa` package used for all the analyses in this thesis uses the geometric mean of all signals from all arrays was used as the constant.

Quantile normalisation

Quantile normalisation was first developed for the pre-processing of high density oligonucleotide microarrays (Bolstad et al., 2003). It is used within the robust multi-array average (RMA) protocol for processing Affymetrix[®] oligonucleotide microarray data. Quantile normalisation works by replacing the largest signal for each array by a median value of the largest signals. Then the second largest signal is replaced by a median value of the second largest signals, and so forth. This results in not only a uniform median for all arrays but an identical distribution.

- 1: Let \mathbf{X} be a $p \times n$ matrix of \log_2 transformed signal intensities with rows representing peptide features and columns representing arrays
- 2: Order each column of \mathbf{X} to give \mathbf{X}_{sort}
- 3: Calculate the row means of \mathbf{X}_{sort} to give a vector of length p called \mathbf{x}_{sort}
- 4: Duplicate each column of \mathbf{x}_{sort} n times to give the matrix \mathbf{X}'
- 5: Rearrange each column of \mathbf{X}' to have the same ordering as the original matrix \mathbf{X}

Linear model normalisation

The underlying assumption of linear model normalisation is that the observed signal intensities can be represented as a linear regression model, the covariates of which account for all the variance seen in the data. The measurement of the array signal intensity can be perturbed by a between array effect A_j , a between subarray effect on any one given slide S_k and between block effects that are generated by spatial variation during processing or variation in the printing process B_l . The effect of the actual peptide can then be denoted by μ_i . In addition to these one can add in any other effects that can be quantified e.g a row and column effect to account for more detailed spatial variation. So for a peptide i from block l on subarray k on array j the observed intensity can be modelled as:

$$Y_{i,j,k,l} = \mu_i + A_j + S_k + B_l + \varepsilon_{i,j,k,l} \quad (4.1)$$

where the residual is modelled as $\varepsilon_{i,j,k,l} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

These controls allow estimation of the parameters of the model and normalisation is achieved by subtracting them to leave the peptide effect μ_i .

4.3 Aim

To evaluate four normalisation algorithms (scaling to global median, quantile normalisation, linear model normalisation with array and subarray effects and no normalisation) applied to peptide microarray immunoassay data.

4.4 Methods

The previous two datasets used for background evaluation (mouse mAb anti-TcdB and rabbit pAb anti-TcdA datasets) were used to compare the performance of normalisation algorithms. Microarray processing, scanning and image processing was as previously described in chapter 3.

4.4.1 Data pre-processing

The raw data was preprocessed by applying an offset subtraction background correction. The offset value was chosen as 1 + the minimum signal intensity in the dataset in order to minimise loss to signal dynamic range. The array data was then normalised using global median scaling, quantile normalisation and linear model normalisation (no normalisation was included as a comparator).

4.5 Results

4.5.1 Assessment of variation induced by normalisation

Figure 4.1 shows the variation in signal intensities for a comparison of two identical arrays. These arrays are technical replicates i.e. biologically identical hence there should be no differential identification seen i.e. M values should be 0. The MA plots show only minimal differences in the amount of variation from the different normalisation methods as measured by the standard deviation (σ) of M values. The between-array M variance is, for these arrays, comparable to the within array variance reported in chapter 3 (section 3.6.1) suggesting very little residual bias after background correction. As a result median values and distribution of the original arrays were very similar, and the effect of any normalisation process was very subtle.

4.5.2 Assessment of precision and bias

In chapter 3 I demonstrate a method for determining the precision and bias of a peptide array titration series. The assumptions made are that as the antibody concentration changes so the signal intensity varies in a consistent and mathematically modellable manner. Thus variation outside the predicted model must represent technical variation which normalisation seeks to minimise. Comparing the amount of variation from the model therefore allows assessment of the performance of each of the normalisation methods.

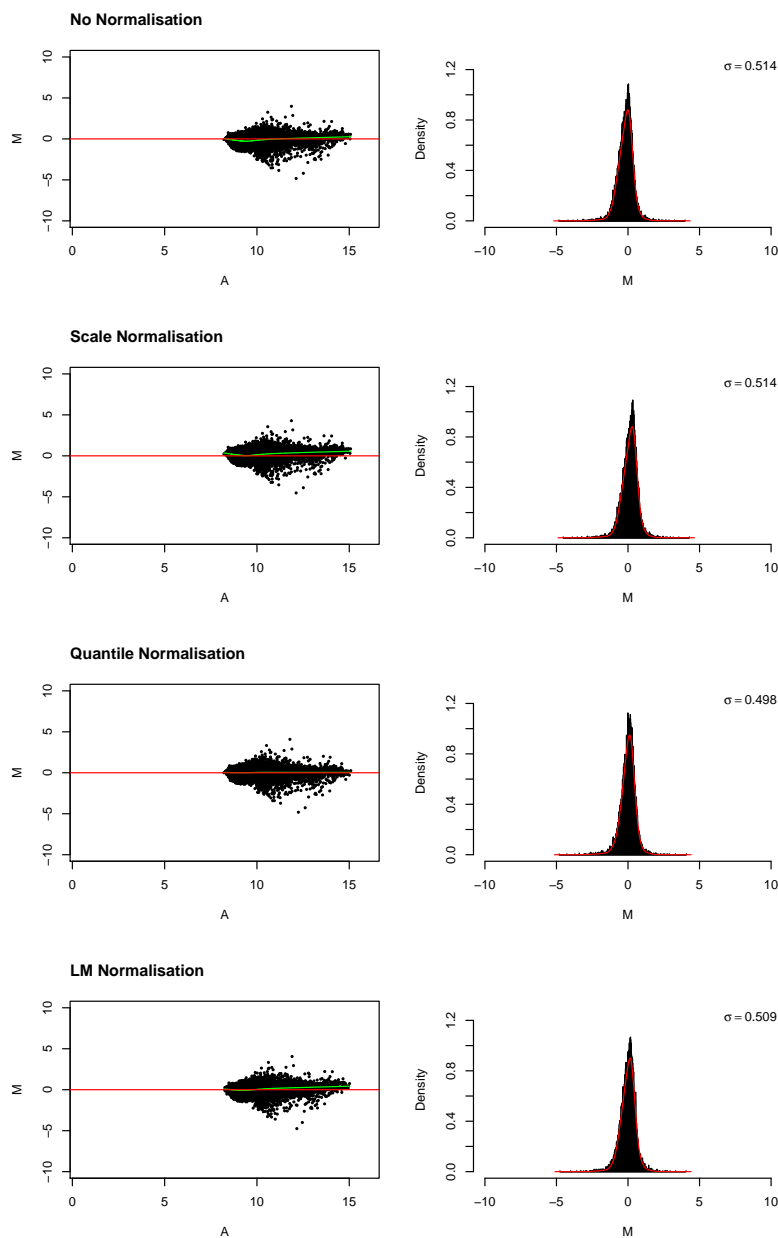


Figure 4.1: MA-plots and M distribution obtained using different normalisation methods for a self-self comparison

Each plot shows the log ratio (M) of a comparison between two arrays incubated with the same serum under identical conditions. The MA plots on the left show the log ratio (M) on the y axis with the mean log peptide signal (A) on the x axis representing the dynamic range measured by the array. As the arrays are biologically identical an ideal MA plot should show no variation around $M=0$. The amount of M variance is shown graphically as a histogram and kernel density (red line) in the plots on the right (the σ value denotes the standard deviation of the distribution).

ORIGINAL IN COLOUR

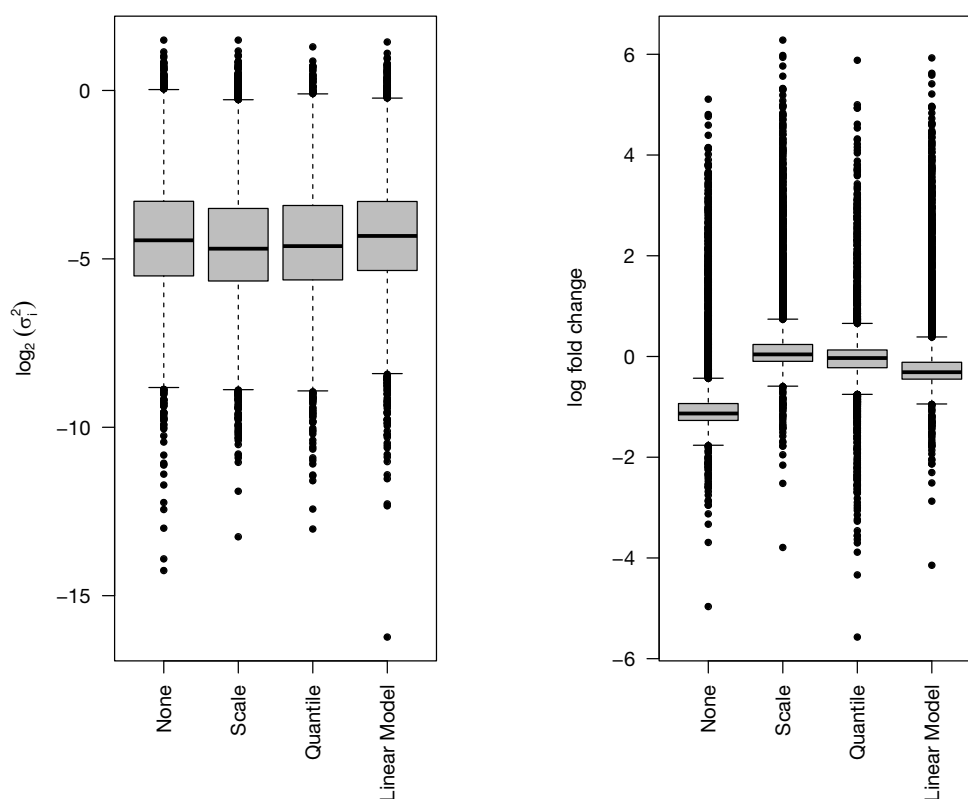


Figure 4.2: Boxplots of log residual variances of quadratic model fits (left plot) and log fold change compared to negative control array (right plot) for to all peptides from the mouse mAb titration dataset.

For each normalisation method the signal intensity change was modelled by a quadratic function and the residual variance (σ_i^2) calculated. The lower σ_i^2 , the more precisely the measured intensities follow the predicted model. Signal intensity fold changes were calculated for arrays in the mouse mAb dataset for each of the tested normalisation methods.

4.5.3 Assessment of differential identification

In order to assess the effect of normalisation on differential identification, the rabbit pAb dataset was used. As described previously in Chapter 3, the spike-in antibody identifies a repeated peptide sequence from the C-terminus (receptor binding domain) of *C. difficile* Toxin A (TcdA). A ROC analysis was conducted in order to evaluate the performance of each normalisation method for detection of the peptides predicted to be differentially identified at three concentrations of the spike in antibody - 2 μ g/ml, 4 μ g/ml and 8 μ g/ml – see Chapter 3 for details of the analysis. ROC plots of each correction method for each concentration are shown in fig 4.3 and the AUROC indicating test performance is shown in table 4.1.

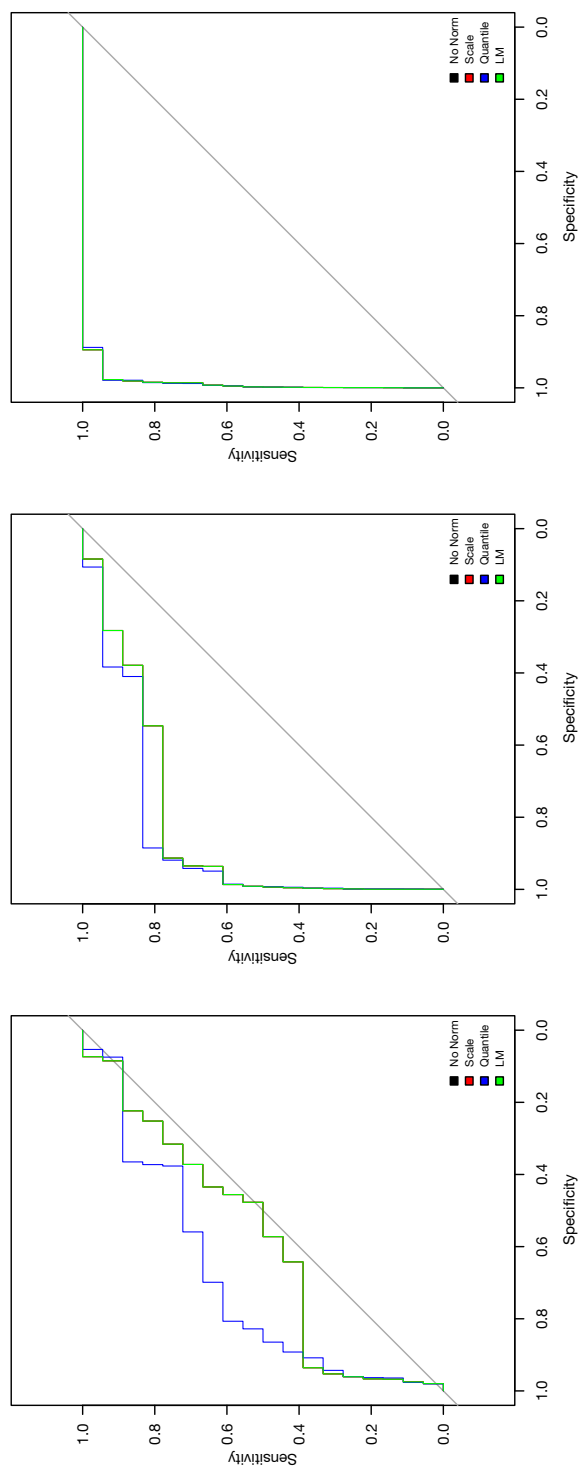


Figure 4.3: ROC plots for differential identification of PA1 binding peptides from antibody spiked into normal rabbit serum

The plots represent the three spike-in concentrations (from left to right 2µg/ml, 4µg/ml and 8µg/ml). For this dataset At 2µg/ml no statistically significant differential identification can be found irrespective of the correction method used. In contrast, at 8µg/ml differential identification is reliably found with any correction method.

ORIGINAL IN COLOUR

	2μg/ml Spike In	4μg/ml Spike In	8μg/ml Spike In
No Normalisation	0.59 (0.44 – 0.75)	0.83 (0.70 – 0.97)	0.99 (0.98 – 1.00)
Scale Normalisation	0.59 (0.44 – 0.75)	0.84 (0.70 – 0.97)	0.99 (0.98 – 1.00)
Quantile Normalisation	0.70 (0.55 – 0.85)	0.86 (0.74 – 0.99)	0.98 (0.98 – 1.00)
Linear Model Normalisation	0.59 (0.44 – 0.75)	0.84 (0.70 – 0.97)	0.99 (0.98 – 1.00)

Table 4.1: AUROC and 95% confidence intervals for PA1 spike-in series

Area under the ROC curve for each of the normalisation methods and for each spike-in concentration. Values in bold type show a significant discrimination of the test for detecting the spike-in antibody. With the exception of Quantile normalisation at the lowest concentration, none of the tested methods demonstrate statistically significantly better discrimination than any other (including no normalisation).

4.6 Discussion

Although technical (non-biological) variation undoubtedly can affect the observed signal seen from peptide arrays, the array datasets presented in this chapter demonstrate that following background correction, none of the normalisation methods employed significantly improve the performance of the array assay beyond no normalisation at all (with the exception of quantile normalisation applied to the 2 μ g/ml spike in dataset). It is likely that most of the technical variation is captured by the background correction and variance stabilisation by log transformation and offsetting. Hence any residual effect is small compared to the overall biological effect being measured. A major limitation of this study is that the peptide array data used to compare normalisation methods is already very similar with only a very small number of peptides varying between samples. This is likely to be very different to ‘real world’ samples where sera may be comprised of very different antibody mixtures. However, it is important to note that none of the methods used here caused significant worsening of bias or loss of differential recognition.

Another limitation of this study is that only three methods were considered. Normalisation to immunoglobulin control spots was not considered despite being previously used in published peptide array studies as the *C. difficile* arrays used only had human immunoglobulin immobilised on them. Although moderate cross reactive signal were seen with both the anti mouse IgG and anti rabbit IgG secondary antibodies, the reliability of such a signal for normalisation was

questionable. A post hoc review of technical replicate arrays from the *C. difficile* study (Chapter 5) showed that immunoglobulin control spots were at or close to saturation in all arrays suggesting that normalising to this signal would have little effect. This finding is similar to that expressed by Ngo et al. (2009) who found that responses to immunoglobulin controls were often inconsistent and varied significantly between array manufacturing batches.

Global normalisation methods such as mean or median scaling or quantile normalisation rely on the assumption that a relatively small number of peptides will be differentially identified and that variations seen in the majority of non-differentially identified peptides represent non-biological effects. This is likely to be true for large high density peptide arrays where there will be a large proportion of features with low level signal – for example a high density random peptide immunosignaturing array is likely to show significant binding activity in only a very small number of its features.

Given the limitations of this study, it is difficult to draw conclusions about the optimal method of normalisation that one should apply to a peptide array dataset. Quantile normalisation was chosen for the subsequent studies into *C. difficile* and *Mycobacterium tuberculosis* immune responses (Chapters 5 and 6) on the basis of modest improvement in differential identification but additional study is warranted to verify if that is a more generalisable finding.

Chapter 5

Identification of antibody signatures characterising *Clostridium difficile* infection

5.1 Introduction

Clostridium difficile is the commonest identifiable cause of antibiotic-associated diarrhoea. It rose to notoriety at the start of the last decade, with a dramatically increasing rate of infection and several high profile outbreaks raising public awareness of it as a major healthcare associated infection. This increase in frequency was associated with the emergence of 'hypervirulent' and antibiotic resistant strains (Warny et al., 2005). In the UK the incidence of CDI peaked in 2007 at 55,500 cases per year but since then rates of CDI have fallen significantly, primarily due to improved infection control measures and better antibiotic stewardship. However, this decline in rates has now stopped and there are still some 20,000 cases annually in the NHS with an all-cause mortality of over 20%. The burden on the health economy is considerable with each case costing around £5000 and relapses costing around £15,000 (Ghantoji et al., 2010). With infection control efforts apparently exhausted and the advent of expensive novel treatments aimed at 'at-risk' patients there is an urgent need to better understand patients' susceptibility to CDI.

5.2 Microbiology and clinical features

C. difficile is a gram-positive, anaerobic rod-shaped bacterium. The organism can exist in a vegetative or spore form. The vegetative form is an obligate anaerobe and is killed by even brief exposure to oxygen. The spore form, by contrast, is heat stable and able to survive in a wide array of adverse environmental conditions including gastric acidity. Moreover it may resist some commercial disinfectants (Wilcox et al., 2003) making it difficult to eradicate. *C. difficile* has also been isolated from a variety of sources: hands of health care workers, fomites such as toilet seats and sinks, endoscopy equipment, and other instruments. Transmission occurs by the faecal-oral route, from person to person, and fomite / instrument to patient.

The clinical presentation of *C. difficile* can be extremely variable ranging from asymptomatic carriage right through to fulminant colitis. *C. difficile* is commonly found in the faecal flora of healthy children but disappears during childhood and is not generally found in the faeces of healthy adults. After the age of around 60 *C. difficile* starts to appear again the faeces especially among patients with chronic disease and frequent hospital contact. Only around one third of elderly people with *C. difficile* in their faeces will develop symptomatic disease and usually in the days immediately following acquisition. The cardinal feature of symptomatic CDI is diarrhoea typically following antibiotic use. The onset of diarrhoea follows the initiation of antibiotics from as early as a few days right through to 8 weeks following cessation of treatment (Bartlett et al., 1978). Stools are typically watery with a characteristic offensive odour, however mucoid and bloody diarrhoea is also recognised especially with severe disease. Other clinical features include abdominal pain, fever, leucocytosis and hypoalbuminaemia. Systemic symptoms are found more commonly as the severity of the diarrhoea increases. Typically leucocytosis occurs in 50%, fever in 28% and abdominal pain in 22% of affected patients (Bartlett et al., 1978). Very severe disease can present with paralytic ileus evolving into toxic megacolon. Paradoxically, in this situation diarrhoea may be absent with the clinical presentation being more akin to bowel obstruction.

However, exposure to *C. difficile* rarely causes symptomatic infection in healthy adults. The reason for this is thought to be a 'colonization resistance' imparted

by the normal gut microbial flora. This colonization resistance is dramatically impaired by antibiotic treatment which disrupts the gut microbiome. This explains the almost ubiquitous association of CDI with preceding antibiotic treatment.

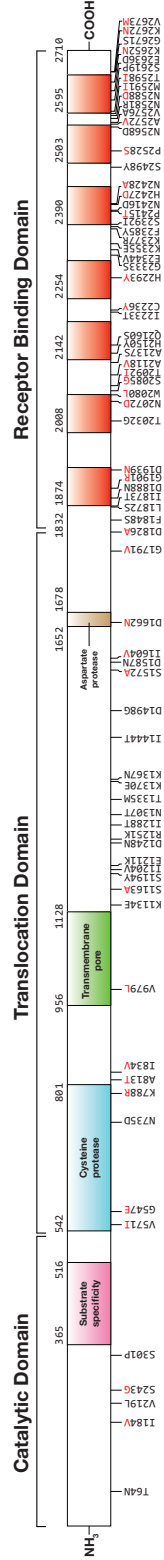
5.2.1 Recurrent *C. difficile* infection

Recurrent CDI is defined as the relapse of CDI following clinical resolution of disease after completion of appropriate treatment. The majority of patients treated with antibiotics (typically metronidazole or vancomycin) are able to clear *C. difficile* but approximately 20% will develop recurrent CDI within 180 days of treatment completion (Olson et al., 1994; Bartlett, 2006). Recurrent CDI may occur by failure to clear the original infection, recrudescence of *C. difficile* spores, or by re-infection with another strain (McFarland et al., 1999; Kamboj et al., 2011).

5.3 *C. difficile* pathogenesis

Over the last decade advances in animal models and molecular biology have allowed the pathogenesis of CDI to be understood in great detail. Colonisation of the intestine appears to depend on two processes. Firstly attachment to host tissues occurs by interaction with several bacterial adhesins: the flagella – comprising Flagellin C (FliC) and the Flagellar Cap Protein (FliD) that are involved in chemotaxis, as well as cell and mucus attachment (Tasteyre et al., 2001), the cell surface proteins from the S-layer (Cerquetti et al., 2000; Fagan et al., 2009), cell wall proteins Cwp66 (Waligora et al., 2001), and Cwp84 (Kirby et al., 2009), and a Fibronectin binding protein (Fbp68 or FbpA) (Hennequin et al., 2003). This process of adhesion has been shown to be associated with increased virulence in animal models (Borriello et al., 1988). The second process important for colonization is the release of two toxins – Toxin A (TcdA) and Toxin B (TcdB).

Toxin A (TcdA)



Toxin B (TcdB)

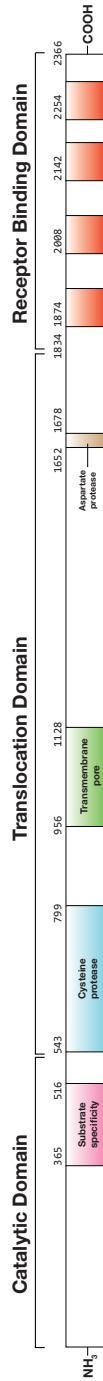


Figure 5.1: Primary domain structure of *Clostridium difficile* toxins A (TcdA) and B (TcdB)

TcdA and TcdB are large clostridial toxins with substantial sequence homology. Polymorphic residues in TcdA are shown under the sequence schematic. The toxin binds to carbohydrate receptors on target gut epithelial cells using multiple binding regions in the toxin C-terminus (shown in red). The binding regions are comprised of a short repeat (SR) sequence and a longer repeat (LR) sequence. TcdA is comprised of 32 SRs interspersed with 7 LR, whereas TcdB has a shorter binding domain of 19 SRs and 4 LR. Receptor binding is followed by internalisation of the toxin into an endosome. Acidification of the endosome induces a conformational change allowing insertion of the hydrophobic transmembrane pore (shown in green) into the endosomal membrane and activation of autoproteolysis (protease shown in blue). The N-terminal catalytic domain functions as a glucosyltransferase enzyme. Cleavage of the toxin releases this enzyme which then can interact with membrane bound Rho-GTPases inactivating them.

ORIGINAL IN COLOUR

5.3.1 *C. difficile* toxins (TcdA and TcdB)

TcdA and TcdB are members of the family of large clostridial toxins (LCTs) which include *Clostridium sordellii* lethal toxin (TcsL) and haemorrhagic toxin (TcsH), *Clostridium novyi* alpha toxin (TcnA) and a *Clostridium perfringens* toxin called TpeL. All are monomeric proteins with three functional domains (Figure 5.1): A C-terminal domain that mediates binding to a host cell-surface receptor, an N-terminal with catalytic glucosyl transferase activity and a translocation domain comprising a cysteine protease which serves to cleave the N-terminal portion from the rest of the protein and a hydrophobic region which is important in carrying the holotoxin into the cytosol.

The first step of toxin action is the binding of the C-terminal domain to carbohydrate moieties expressed on the host cell surface. In humans a number of carbohydrate moieties appear to operate as the functional TcdA receptor. The blood group antigens LewisX – Gal β (1-3)[Fuc(1-4)]GlcNAc, LewisY – Gal β (1-4)GlcNAc β 1 and LewisI – Gal β (1-14)GlcNAc β (1-3)Gal β (1-4)Glc are all present on the brush border of intestinal epithelial cells and have been shown to function in vitro as a receptor for TcdA (Tucker and Wilkins, 1991). Moreover, the glycoprotein gp96 was also shown to function as a receptor for TcdA in human cells (Na et al., 2008). The receptor for TcdB is currently unknown although it does appear to be distinct from the TcdA receptor. TcdB appears to bind to a receptor located on the basolateral side of colonic cells in culture whereas TcdA appears to bind apically (Stubbe et al., 2000).

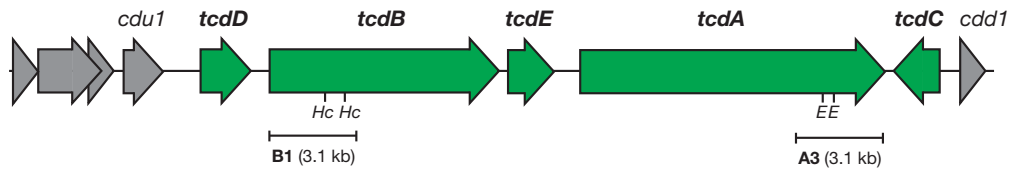
TcdA and TcdB, in common with other LCTs, target the Rho family of small GTPases (RhoA, RhoB, RhoC, RhoG, Rac1 to 3, Cdc42 and TC10) that play varied roles in cell signalling pathways (Jank et al., 2007). By irreversibly glycosylating these proteins the toxins cause disruption of the cytoskeleton. This leads to disruption of tight junctions, loss of epithelial integrity, production of inflammatory mediators and cell death (Nusrat et al., 2001).

The pathogenicity locus and toxinotyping

Both TcdA and TcdB are encoded on a 19.6kb genetic element called the pathogenicity locus (PaLoc) – Figure 5.2 In addition to *tcdA* and *tcdB* three other open read-

ing frames are present on the PaLoc (*tcdR*, *tcdE* and *tcdC*) which are implicated in toxin regulation and release from the cell.

Strain 630 - toxinotype 0



Strain R20291 - toxinotype III

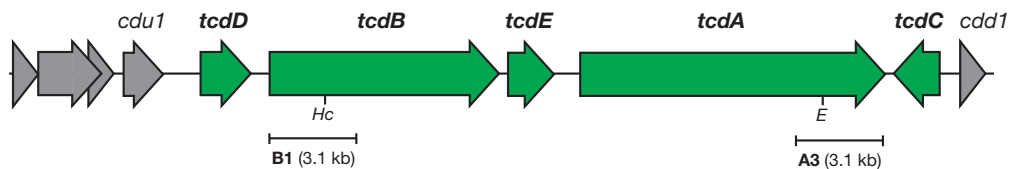


Figure 5.2: The *C. difficile* Pathogenicity Locus (PaLoc)

Schematic representation of a 19.6kb element from the *C. difficile* genome known as the pathogenicity locus (PaLoc). It encodes the large clostridial toxins on two genes *tcdA* and *tcdB*, and contains three regulatory genes (*tcdR*, *tcdE* and *tcdC*). Restriction site locations for the HincII and EcoRI enzymes are shown in italics below the ribbon diagram (*Hc* and *E* respectively). The amplified regions that are used for toxinotyping (B1 and A3) are shown in bold. The resulting pattern of restriction site polymorphisms is then used to define the toxinotype.

Toxinotyping is a method of classifying *C. difficile* strains based on similarities within the PaLoc. It is accomplished by polymerase chain reaction (PCR) amplification of the PaLoc region in 10 fragments followed by the determination of restriction site polymorphisms. Rupnik et al. (1998) proposed a modification of the process in which only the B1 and A3 fragments are amplified (corresponding to the 5' end of the *tcdB* gene and the 3' end of the *tcdA* gene respectively). A *C. difficile* strain is then grouped into a toxinotype based on the HincII/AccI (B1) and EcoRI (A3) restriction patterns. Currently 31 toxinotypes are defined [<http://www.mf.uni-mb.si/tox/>]. In the majority of cases these toxinotypes correlate well to PCR ribotypes – another method of *C. difficile* typing based on the pattern of PCR products from the 16S-23S ribosomal ribonucleic acid

(rRNA) intergenic spacer region (Rupnik et al., 2001). Ribotyping is a more discriminative classification process than toxinotyping so strains from several ribotypes may be found within a single toxinotype but only very rarely will a particular ribotype contain strains with different toxinotypes (Rupnik et al., 2001).

5.3.2 *C. difficile* binary toxin (CDT)

C. difficile binary toxin was first discovered in 1988 by Popoff et al. (1988) when they identified a *C. difficile* strain that produced a toxin with ADP-ribosylating activity. Other bacterial ADP-ribosylating toxins were already known at this time but it was not until 1997 that the sequence of CDT was published and shown to be part of the C2-like binary toxin family (Perelle et al., 1997). The family includes the *Clostridium botulinum* C2 toxin, *Clostridium perfringens* iota toxin, *Clostridium spiriforme* toxin and *Bacillus cereus* vegetative insecticidal protein.

Binary toxin positive *C. difficile* strains are becoming increasingly prevalent in epidemiological surveys of human infection. Binary toxin is expressed in many of the variant (non-toxinotype o) toxinotypes but overall is only found in a minority of ribotypes. However, the finding that the epidemic 027/BI/NAP₁ and 078/BL/NAP_{7,8} strain types both express binary toxin has stimulated interest in its role as a possible enhanced virulence factor in CDI (McDonald et al., 2005).

Binary toxin is encoded within a 6.2 kb region of the *C. difficile* genome known as the Cdt locus (CdtLoc). This region includes two genes encoding the two subunits of the toxin (*cdtA* and *cdtB*) and a gene encoding a regulatory protein (*cdtR*). In contrast to the multiple toxinotypes of the PaLoc, the CdtLoc appears in three forms: a whole locus, a truncated locus with *cdtR* and a *cdtAB* pseudogene, and an absent locus with a unique 68bp sequence occupying its location (Carter et al., 2007), (Figure 5.3).

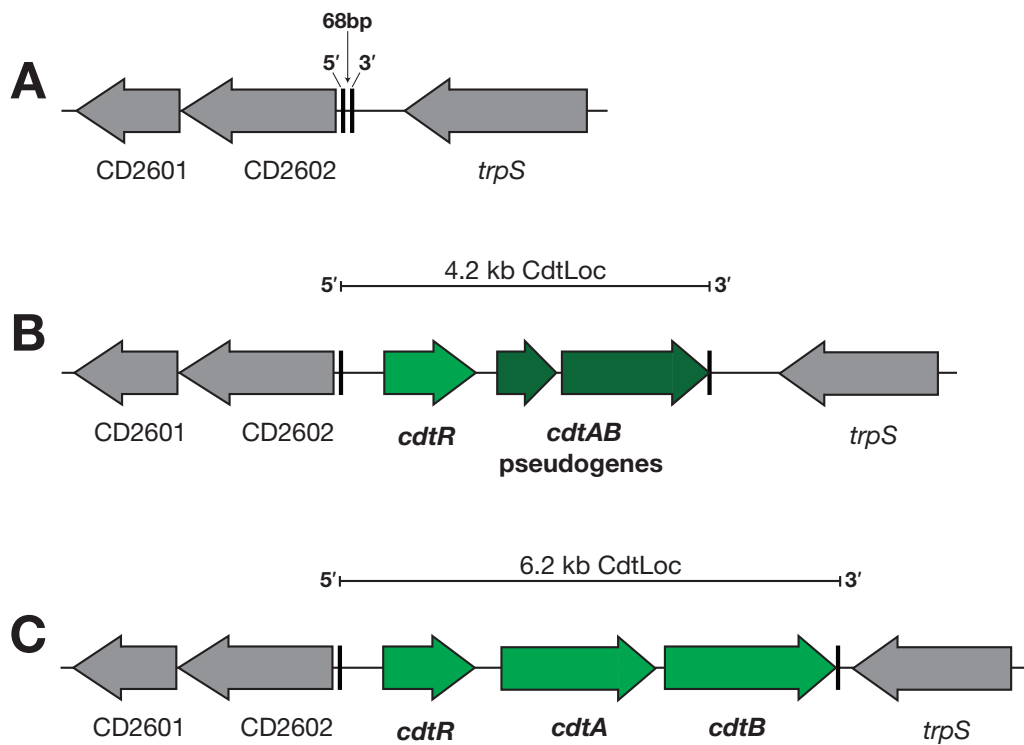


Figure 5.3: The *C.difficile* Binary Toxin Locus

Schematic representation of an element from the *C. difficile* genome known as the Cdt locus (CdtLoc). The CdtLoc can be found in three forms – Subfigure A: an absent locus with a short non-coding 68bp sequence in its location. Subfigure B: a truncated locus containing the regulatory element *cdtR* and a pseudogene *cdtAB*. Subfigure C: a complete locus containing the *cdtR* regulatory element and the genes encoding both subunits of binary toxin *cdtA* and *cdtB*.

5.4 The immune response to *C. difficile*

Despite a clear pathogenic mechanism there is a considerable variation in the clinical outcome of patients infected with *C. difficile*. Following acquisition of CDI only around 1/3 of patients develop symptoms. Among those that do, antibiotic treatment results in clearance of infection in around 70% of cases. In a further proportion, carriage may persist. Overall about 30% of patients will develop symptomatic relapses of infection following treatment and among these around half are genuine relapses of the same infection and half are new infections resulting from persistent abnormalities in the gut microflora or immune response. Although factors such as gut microflora, concomitant diseases and medication play a role in determining the phenotype of CDI there is a considerable amount of evidence that the host adaptive immune response to *C. difficile* is a crucial determinant of disease outcome. Indeed this is the rationale for treatments including injection of pooled immunoglobulin and (currently experimental) vaccination.

5.4.1 The adaptive immune response to *C. difficile* toxins

Systemic IgG and IgA antibodies against TcdA and TcdB can be found in up to 60% of healthy adults (Viscidi et al., 1983; Bacon and Fekety, 1994). This probably relates to exposure in infancy and the ubiquitous nature of the bacterium as an environmental pathogen. Kyne et al. (2000) showed that development of higher levels of serum IgG antibodies against TcdA are associated with asymptomatic carriage rather than symptomatic disease in patients who are initially colonised with *C. difficile*. It was also shown that patients who had pre-existing asymptomatic colonisation (and presumably a high antitoxin antibody titre) were much less likely to develop symptomatic CDI than patients exposed to *C. difficile* in hospital. In addition to its association with asymptomatic carriage antibodies against TcdA (Kyne et al., 2001), TcdB (Leav et al., 2010), and other non toxin antigens: Cwp66, Cwp 84, FliC, FliD and the surface layer proteins (Péchiné et al., 2005; Drudy et al., 2004) have all been associated with protection against recurrent disease.

5.4.2 The adaptive immune response to non-toxin antigens

Although the toxins secreted by *C. difficile* are the major virulence factors, a number of other proteins have been implicated in the pathogenesis of CDI and colonisation. These proteins include flagellin (FliC) and the flagellar cap protein (FliD) (Tasteyre et al., 2001), Fibronectin Binding Protein (FbpA or Fbp68) (Hennequin et al., 2003), the adhesin Cwp66 (Waligora et al., 2001), Surface Layer Proteins (SlpA) (Calabi et al., 2002) and the cysteine proteases required for surface layer maturation (Cwp84 and Cwp13) (de la Riva et al., 2011). These proteins have also been shown to be immunogenic; patients with CDI have been shown to have increased levels of antibody to these proteins (Pechine, 2005).

The observation that the antibody response against toxin and non-toxin antigens influences the outcome of CDI has been the driving force in the development of vaccines and antibody products. The efficacy of passive vaccination (administration of exogenous antibody) for protection against CDI has been demonstrated in animal models: *Clostridium sordellii* anti-toxin has been shown to have cross-reactive protection against *C. difficile* toxin challenge in hamsters (Allo et al., 1979), and IgG monoclonal antibodies directed against the receptor binding C-terminal domain of toxin A was protective against death after toxin challenge in gnotobiotic mice (Corthier et al., 1991). This protective effect was further enhanced by using a combination of anti-toxin A and anti-toxin B monoclonal antibodies (Babcock et al., 2006). The first trials of passive vaccination in humans utilized pooled immunoglobulin derived from a large number of donors (Leung et al., 1991). The response in human studies was rather mixed with some rather limited reports of protection and improved outcome (Beales, 2002; Salcedo et al., 1997). However, all of the studies to date suffered from being small with no standardization of protocol and preparation used. Possibly a more promising advance in the field of passive immunisation for CDI was made by Lowy et al. (2010) who showed that human monoclonal antibodies, produced against *C. difficile* TcdA and TcdB and administered intravenously in conjunction with standard antibiotic treatment, were effective at protecting patients against recurrent infection in a randomised, placebo controlled trial (Lowy et al., 2010; Leav et al., 2010).

5.5 Aims

Despite this body of evidence there is still much that is unknown about humoral immunity to CDI.

1. Most of the existing data predates the advent of hypervirulent strains of *C. difficile* which synthesise additional toxins (e.g. Binary Toxin) and toxin variants.
2. Data indicating a central role for immune responses to Toxin A have been undermined by the advent of highly virulent strains of CDI which do not express this toxin.
3. The majority of studies have focused primarily on toxin A and B and not addressed proteins more recently identified as of crucial importance in colonisation phase of infection with precedes disease.
4. Very little is known about the precise antigenic determinants (epitopes) that generate the immune response in CDI. Epitopes are regions of proteins (or other biological macromolecules) that are recognised by the immune system. Being able to characterise the antibody response to *C. difficile* at the epitope level potentially allows identification of novel targets for immune based treatments and vaccination. Moreover, as outlined above, changes in epidemiology and the treatments available now demand better understanding of patient susceptibility for treatment and infection control purposes. Being able to characterise the evolution of epitope specificity during the course of natural infection and the pattern of epitopes that are generated during asymptomatic colonisation or recurrent infection could allow us to potentially predict the outcome of *C. difficile* infection early in the disease course.

The work described in this chapter attempts to address these knowledge gaps by designing and applying a *C. difficile* peptide microarray to elucidate the humoral immune responses present in patients with acute symptomatic *C. difficile* infection. My hypothesis is that differences in humoral immune responses detected by microarray may predict outcome of *C. difficile* infection (including severity and recurrence).

5.6 Methods

5.6.1 Study participants and sampling – Brighton

Ethical approval was obtained to recruit patients diagnosed with *C. difficile* associated diarrhoea from the South East Research Ethics Committee (Reference no. 09/H1102/63). Participants were deemed to be eligible if they had been diagnosed by a positive stool Enzyme Linked Immunosorbent Assay (ELISA) for TcdA, TcdB or both toxins and new-onset active diarrhoea. Diarrhoea was defined as a change of stool habit with greater than two Bristol Type 6 or 7 stools in the preceding 24 hours – appendix 2). Patients with known immunodeficiency were excluded from participation. A control group was selected from hospitalised patients being treated with antibiotics but diarrhoea free as part of a sub-study from a larger clinical trial of *Lactobacillus casei* probiotic – Probiotic NU278 study. Stool specimens were taken from the control participants to ensure they were negative for *C. difficile* (by stool toxin ELISA and stool PCR). For the CDI+ group, clotted blood (serum) and stool samples were taken at entry to the study (within 4 days of development of symptoms). Samples were frozen at -80°C until use.

5.6.2 Study participants and sampling – Liverpool

A second set of patient samples were obtained from the University of Liverpool Biobank. Sera for the case group were collected from patients diagnosed with *C. difficile* infection in the Royal Liverpool and Broadgreen University Hospitals NHS Trust within 72 hours of development of symptoms. CDI cases were all symptomatic with diarrhoea. Cases were followed up for a minimum of 4 weeks to determine relapse rates. Sera from a comparator group was obtained from hospitalised patients who were experiencing diarrhoea but were found to be negative for *C. difficile* (by stool toxin ELISA). All sera aliquots were stored at -80°C until use. Samples were labelled with a unique identifier code and investigators were blinded to the clinical status of each sample until after all laboratory processing had been completed.

5.6.3 Microarray Design

Protein sequences from two strains of *C. difficile* were used to synthesise the array peptides: Strain 630, a ribotype 012 strain isolated in Switzerland in 1982 from a patient with severe pseudomembranous colitis (Sebahia et al., 2006), and Strain R20291(SM), a ribotype 027 strain isolated in the UK in 2006 from a severe outbreak of CDI at Stoke Mandeville Hospital (Stabler et al., 2009). Strain 630 was the first strain to be fully characterised by its genome sequence (Sebahia et al., 2006; Monot et al., 2011). The sequence of 10 known immunogenic proteins from these strains (The toxins TcdA and TcdB, binary toxin – CdtA and CdtB, the surface layer complex SlpA and Cwp84, the adhesins Cwp66 and FbpA, and the flagellar proteins FliC and FliD) were parsed into 15mer peptides overlapping by 10 residues. In addition to these *C. difficile* proteins the array contains overlapping 15mer peptide sequences of the outer membrane protein (OmpA) of three Bacteroides species (*B. vulgatus*, *B. ovatus* and *B. fragilis*) typically found as gut commensals. Overlapping peptides representing the sequence of human TNF- α , IFN- γ , and the heavy chain of *Clostridium tetani* toxin (Tetanus toxin) were also included on this array (Table 5.1).

Accession No. (UniProt/Tr-EMBL)	Strain	Protein	Gene	Length (aa)	Mass (Da)
Q189K5	CD630	Toxin A	TcdA	2710	308219
C9YJ37	SM (R20291)	Toxin A	TcdA	2710	308311
Q189K3	CD630	Toxin B	TcdB	2366	269712
C9YJ35	SM (R20291)	Toxin B	TcdB	2366	269140
C9YPH7	SM (R20291)	Binary Toxin – Subunit A	CdtA	463	53250
A8DS70	SM (R20291)	Binary Toxin – Subunit B	CdtB	876	98797
Q182T2	CD630	Fibronectin Binding Protein	FbpA	591	67641
C9YPG5	SM (R20291)	Fibronectin Binding Protein	FbpA	591	67617
Q18CX7	CD630	Flagellin component C	FliC	290	30773
C9YI47	SM (R20291)	Flagellin component C	FliC	319	34375
Q18CX9	CD630	Flagellar Cap Protein	FliD	507	56077
C9YI45	SM (R20291)	Flagellar Cap Protein	FliD	507	56336
Q183M7	CD630	Cwp66 Cell Wall Protein	Cwp66	610	66779
C9YQ13	SM (R20291)	Cwp66 Cell Wall Protein	Cwp66	611	66967
Q183M1	CD630	Cwp84 Cell Wall Protein	Cwp84	803	87281
C9YQ11	SM (R20291)	Cwp84 Cell Wall Protein	Cwp84	803	87282
Q183M8	CD630	Surface Layer Precursor Protein	SlpA	719	76133
C9XP98	SM (R20291)	Surface Layer Precursor Protein	SlpA	758	80428

Table 5.1: *C. difficile* proteins used for peptide microarray

5.6.4 Peptide synthesis and microarray printing

Amino-oxyacetylated peptides were synthesised on a cellulose membrane support using SPOT synthesis by JPT, Berlin, Germany. The peptide spots were punched out of the cellulose membrane and treated with aqueous triethylamine to cleave the peptide from its support. After filtration and evaporation of the solvent the peptides were dissolved in a printing buffer (70%DMSO, 25% 0.2M sodium acetate pH 4.5, 5% glycerol) and spotted on to epoxy-functionalised glass slides. Printed arrays were washed in Saline Sodium Citrate (SSC) buffer before drying and storage at 4°C.

Microarray slides were printed with three identical subarrays each with 5184 spots arranged in 16 blocks comprising 18 columns x 18 rows. Each subarray was printed with 3534 peptides from the *C. difficile* proteins, 349 peptides from the three *Bacteroides* OmpA sequences and 640 peptides covering the primary structure of TNF α , IFN- γ and *Clostridium tetani* toxin. The 18th row of each block was reserved for printing control peptides, immunoglobulin control spots and Cy3 spots for array orientation. The remaining array spots were printed with the print buffer only and contained no peptide.

5.6.5 Microarray Processing

Microarrays were processed as previously described in Chapter 3. Sera were thawed immediately prior to processing and diluted 1 part in 100 with blocking buffer. After primary incubation and washing, arrays were incubated with two secondary antibodies – goat anti-human IgG – AlexaFluor™647 (Invitrogen, USA – Cat. No. A21445) and rabbit anti-human IgA α chain Cy3 (Jackson ImmunoResearch, USA – Cat. No. 309-169-099) each diluted to 1 μ g/ml in blocking buffer. A negative control array (primary incubation with buffer only) was included with each processing batch.

5.6.6 Peptide microarray scanning and data extraction

All arrays were scanned within 48 hours of processing on a Genepix®4300A microarray scanner at excitation wavelengths of 635nm (red channel) and 532nm

(green channel). Pixel resolution was set to 10 μ m, laser power to 100% and PMT gain optimised and fixed for all arrays in the series. The raw images were saved in a 16 bit multi-image TIFF format. Image processing was performed with Genepix®Pro version 7.2 software. Block alignment was performed manually prior to automated feature alignment. A final manual check of alignment and segmentation was performed before data extraction. The resulting data was saved as GenePix Results (GPR) files.

5.6.7 Data pre-processing

All pre-processing was carried out using the R statistical programming environment (R Development Core Team, 2014) with the *pmpa* package as described previously. Background signal was subtracted from foreground with a positive offset of 128 (2^7) applied. Arrays were quantile normalised and mean summarised with outliers filtered if the per-peptide CV exceeded 30%. One array per processing batch was reserved for a negative control incubation (primary incubation with blocking buffer and secondary incubation with the fluorophore conjugated secondary antibody (2 slides for the Brighton dataset and 3 for Liverpool although one was discarded due to processing artefacts)).

5.6.8 Data analysis

Differential identification was assessed by fitting a linear model to the data from each peptide and applying an empirical Bayes moderated t-test (Limma package for R/Bioconductor). To allow for multiple testing, P values were adjusted using the Benjamini and Hochberg method for controlling the false discovery rate Benjamini and Yekutieli (2001). Peptide signals were considered to be significantly differentially identified if they had a mean fold change of at least 2 log and an adjusted P value of < 0.01 .

5.7 Results

5.7.1 Assessment of the sequence conservation of the represented *C. difficile* peptides

Given the falling prevalence of CDI caused by ribotype 027 strains and the extremely low prevalence of ribotype 012 infections in the UK at the time of sample collection, the ability of the array to identify antibody raised to linear sequences from non 012/027 strains was investigated. The Uniprot/TrEMBL database (Apweiler et al., 2014) was queried for *C. difficile* protein sequences matching the protein name and/or gene name of the proteins used for the peptide array. Fragment sequences (less than full length) were not included. Pairwise alignment to the strain 630 and strain R20291 full length protein sequences were performed and the percentage identity calculated (Raghava and Barton, 2006) – Table 5.2. >90% sequence homology is seen for the alignments of TcdA, TcdB, FbpA and Cwp84 to any of the publicly available sequences in the database. FliD and FliC sequences showed a lower sequence identity for the alignment of strain 630 (89.6% and 86.8% respectively) than for the alignment of strain R20291 (97.5% and 93.9% respectively). By contrast the two most highly variable protein sequences Cwp66 and SlpA show the lowest overall sequence similarity. Multiple sequence alignment of the SlpA sequences (not shown) confirms the majority of variation is seen in the highly antigenic low molecular weight peptide region.

Therefore arrayed peptides from the highly conserved proteins listed above should be reliable at identifying possible linear epitopes from antibody raised against proteins from non-012/027 strains. Given the variability of Cwp66 and SlpA only conserved regions of those proteins identified by multiple alignment were used to identify possible epitopes.

5.7.2 Exploratory data analysis – Brighton dataset

Unsupervised clustering was performed for all 3534 *C. difficile* peptide probes. This is an exploratory data analysis procedure which does not classify the samples (or peptides) into groups prior to the analysis but attempts to find structure in the

Protein	Accession no. of reference sequence (Strain 630)	Mean Percentage Identity (SD)
TcdA	Q189K5	99.6 (0.58)
FbpA	Q182T2	99.4 (0.51)
Cwp84	Q183M1	99.0 (1.91)
TcdB	Q189K3	98.8 (2.83)
FliD	Q18CX9	89.6 (3.7)
FliC	Q18CX7	86.8 (6.12)
Cwp66	Q183M7	69.0 (14.6)
SlpA	Q183M8	61 (13.1)

Protein	Accession no. of reference sequence (Strain R20291)	Mean Percentage Identity (SD)
FbpA	C9YPG5	99.4 (0.51)
CdtA	C9YPH7	99.4 (1.16)
CdtB	A8DS70	98.9 (1.16)
Cwp84	C9YQ11	98.9 (1.83)
TcdA	C9YJ37	98.3 (0.45)
FliD	C9YI45	97.5 (3.51)
FliC	C9YI47	93.9 (7.12)
TcdB	C9YJ35	92.3 (1.99)

Table 5.2: Summary of pairwise sequence alignments of Strain 630 and Strain R20291 proteins with publicly available sequences from the Uniprot / TrEMBL database

The upper table shows pairwise alignments to the protein sequences from strain 630 and the lower shows strain R20291. Binary toxin (CdtA, CdtB) is not expressed by strain 630 and so is omitted from the upper table. Global pairwise alignments using the Needleman-Wunsch algorithm were performed using the BLOSUM80 matrix. A gap opening penalty of 10 and gap extension penalty of 4 was used. Percentage identity was calculated as the ratio of identical positions to the minimum of the sum of residues and gap positions from the two sequences considered.

data which can then be examined. Hierarchical agglomerative clustering (using average linkage) segregates the CDI positive (CDI+) and the non-CDI control (CDI)- group into two clusters except for 3 CDI+ arrays (CDI006, CDI008, CDI007 for the IgG data, and CDI016, CDI006, CDI008 for the IgA data).

To examine if these unclustered arrays are likely to be outliers the partition around medioids (PAM) algorithm was applied with $k = 2$ (two clusters specified) (Figure 5.4). This is a semi-supervised method where the number of clusters can be specified *a priori* but the analysis does not take into account any assignment to a category. The IgA (Cy3) signal segregates into clusters which define the clinical status (CDI+ vs. CDI-) although CDI016 does not neatly sit with any of the other

	CDI+ (n=18)	CDI- (n=18)	P value*
Age in years (Median, IQR)	85 (65-88)	77.5 (65.8-81.8)	0.39
Male gender (%)	11 (55%)	12 (67%)	0.39
White Cell Count 10 ⁹ /L (Median, IQR)	11 (7-12)	13 (9-15.5)	0.25
C-Reactive Protein mg/L (Median, IQR)	48 (15.6-91.6)	72.5 (23.3-144.3)	0.31
Urea mmol/L (Median, IQR)	6 (4-8)	7 (6-8)	0.43
Creatinine μmol/L (Median, IQR)	78 (64-127)	80 (70.5-104.5)	0.92
Albumin g/L (Median, IQR)	33 (28-37)	39 (37-41)	0.004

Table 5.3: Summary descriptive statistics for the Brighton CDI dataset

Demographics and baseline laboratory measures of disease severity are shown for the analysed cases categorised by CDI status.

**p* values determined by Wilcoxon Rank Test or Chi Squared Test as appropriate

data points. The IgG signal also segregates clearly into two clusters although the analysis groups one CDI+ case (CDI008) into the CDI- cluster. Examination of the cluster plot shows that it lies between the two clusters on the axis of component 1 suggesting it may be better considered as an outlier. Silhouette plots were created for values of *k* from 2 to 5 (data not shown) with the maximal average silhouette width achieved at *k* = 2 (IgG 0.27, IgA 0.35) indicating that the optimal structure of this data is achieved with two clusters.

5.7.3 Differential identification – Brighton dataset

Given that the exploratory analyses point to a data substructure that differentiates CDI+ from CDI-, a classification analysis was conducted to ascertain which peptides contributed to defining that substructure. 68 peptides were differentially identified in the IgG data compared to 72 from IgA. 54 peptides were commonly identified in both the IgG and IgA dataset. Plotting this filtered data as a heatmap and repeating the hierarchical clustering shows, as expected, the demarcation of CDI+ from CDI-, but also reveals that the ‘signature’ of differential identification

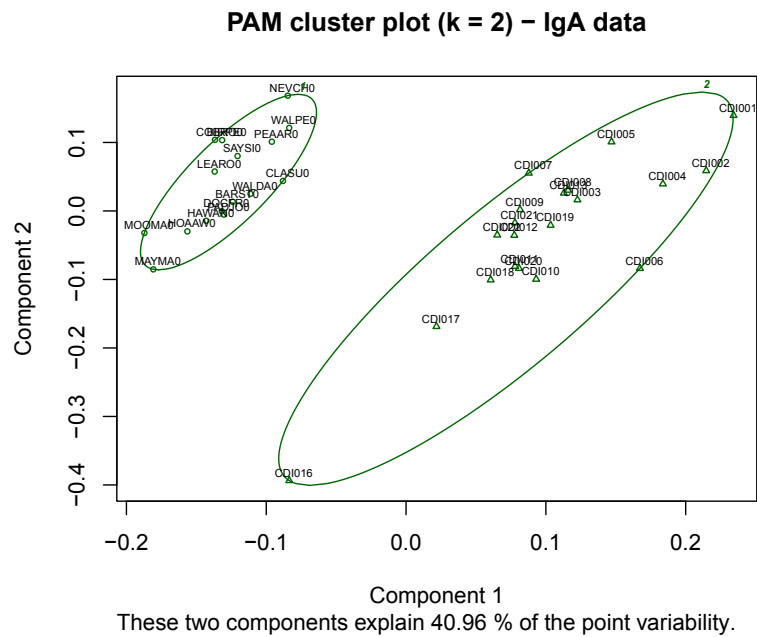
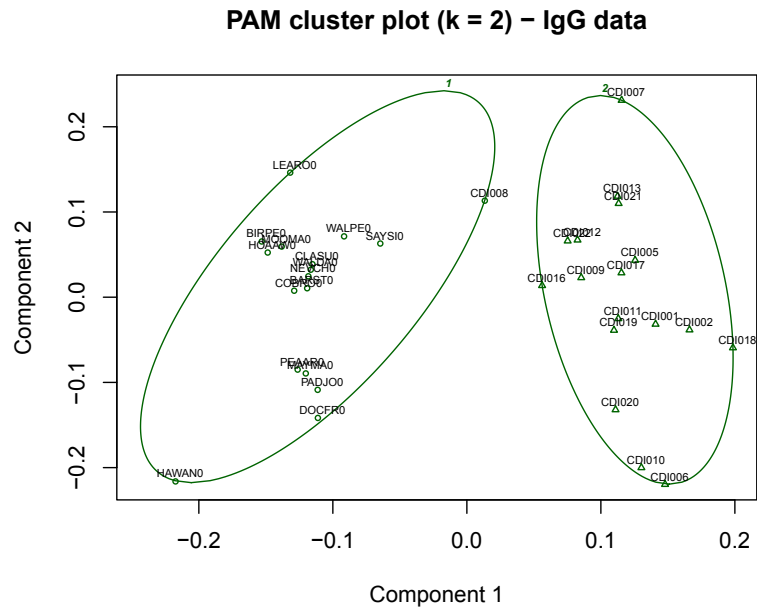


Figure 5.4: Partition around medioids (PAM) clustering – Brighton dataset
 PAM defines two clusters corresponding to CDI status. The upper plot shows the analysis performed on the IgG signal and the lower plot shows the IgA data. In both the CDI+ cases are denoted with the prefix CDI. This is a semi-supervised classification method where the number of clusters is specified *a priori* but segregation into each cluster is performed by the algorithm.

comprises peptides that show increased recognition in the CDI- group compared to CDI+, and also peptides that are more highly recognised in the CDI+ group compared to the CDI-.

5.7.4 Batch effect – Brighton dataset reanalysis

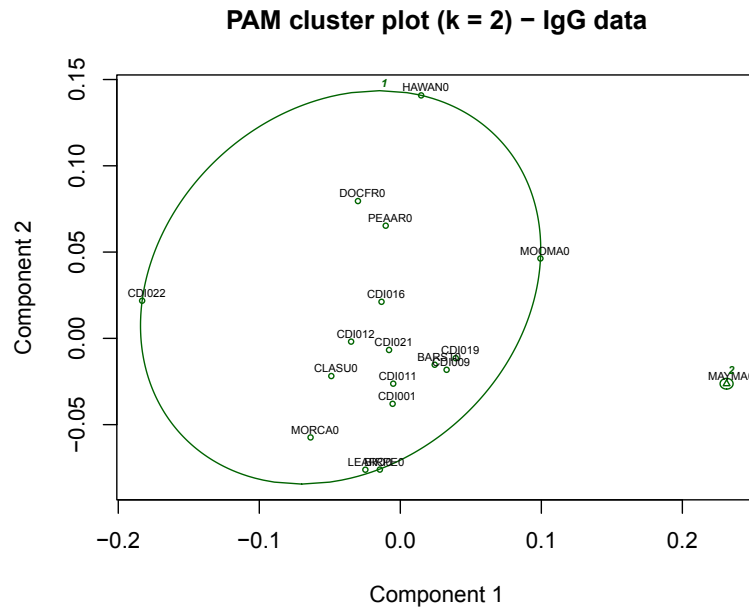
Concerns were raised as to the validity and reproducibility of the clustering and differential identification results. Differential identification was seen in many non *C. difficile* peptides and importantly, because the the CDI+ and CDI- samples were processed separately (although an identical protocol and reagents were used), the possibility of batch confounding was raised. Hence a random subset of the Brighton samples were re-processed in a single batch and re-analysed. Figure 5.5 shows the repeated PAM clustering showing no separation by CDI status. Subsequent reanalysis of differential identification failed to show any peptides that were significantly identified in the groups.

5.7.5 Demographics and clinical characteristics – Liverpool Dataset

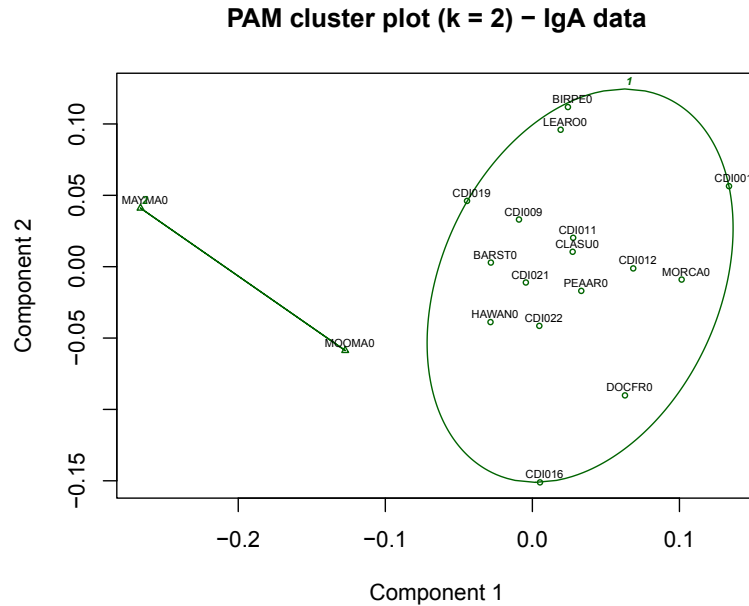
51 samples were analysed from a total of 60 supplied sera. 8 were rejected due to poor data quality after microarray processing and analysis, and 1 was excluded due to the sample identifier not matching the clinical information. No statistically significant differences were seen in the age, gender distribution and body mass index of the CDI cases and non-CDI controls (Table 5.4). Overall the median age of participants was 69.5 years (IQR 61.1 – 81.6). Increasing age was not significantly associated with severe CDI ($p = 0.3$) but was associated with relapsing disease (median age for relapse 84.7, $p = 0.005$).

5.7.6 Differential reactivity of sera from CDI and Control groups

No significant differential reactivity in either the IgG or IgA signal was seen comparing all CDI cases from either the Brighton or the Liverpool cohort to their



These two components explain 46 % of the point variability.



These two components explain 52.59 % of the point variability.

Figure 5.5: Reanalysis of Brighton dataset – Partition around medioids (PAM) clustering
 Reanalysis of the dataset in a single process batch fails to replicate the earlier finding of two clear clusters separating the CDI+ and CDI- groups. It is likely the earlier finding was due to batch confounding as the two groups were processed separately.

	CDI (n=28)	Control (n=23)	P-value*
Age (Median, IQR)	72.9 (60.5 – 84.1)	65.2 (61.2 - 76.4)	0.38
Body Mass Index (Median, IQR)	26.1 (21.3 – 29.3)	27.0 (24.0 – 31.2)	0.32
Gender			
females	20 (71%)	15 (65%)	0.86
males	8 (29%)	8 (35%)	
Severe CDI			
Baseline severe	8 (29%)	N/A	
4wk severe	4 (14%)	N/A	
Total severe	12 (43%)	N/A	
Relapsing Disease			
Early relapse (4 wk)	9 (32%)	N/A	
Late relapse (12 wk)	3 (11%)	N/A	
Total relapse	12 (43%)	N/A	

Table 5.4: Summary descriptive statistics for the Liverpool CDI dataset

Demographics are shown for the analysed cases categorised by CDI status. Severity and relapse data was gathered for the CDI cases in the Liverpool dataset.

**p* values determined by Wilcoxon Rank Test or Chi Squared Test as appropriate

respective CDI negative controls (Figure 5.6). However recurrent CDI (rCDI) cases alone (n=12) from the Liverpool cohort did show significant differential identification from the IgG signal only. (Figure 5.6). After correction for multiple testing, 9 peptides had a statistically significant differential reactivity (Table 5.5).

Sequence	Protein	Position	logFC	t	adj P value
GKVVYFMPDTAMAAA	TcdA	2671	-1.33	-5.51	0.003
LELNQMRNQMNAMAAA	<i>B. ovatus</i> OMP	245	-1.51	-5.40	0.003
ELANMRRQMNDMAAAA	<i>B. fragilis</i> OMP	245	-1.21	-5.23	0.003
LSKMNILVQASQSML	FliC	261	-2.02	-5.01	0.005
EKFYINNFMMVSGSL	TcdB	1821	-1.53	-4.94	0.005
MYQLEYAGLSKIMSN	Cwp84	336	-1.23	-4.82	0.006
MASAGITTASIGSMK	FliC	176	-1.27	-4.55	0.009
RNNFYFDANNESKMV	TcdA	2260	-1.04	-4.38	0.013
ETENLDFSKIMMLP	TcdA	1211	-1.94	-4.37	0.013

Table 5.5: Peptide differential identification in recurrent CDI (rCDI)

9 peptides show IgG differential identification in the rCDI group compared to patients with single episode CDI. *P* values in the far right column are adjusted to account for multiple testing using the Benjamini and Hochberg method (Benjamini and Yekutieli, 2001)

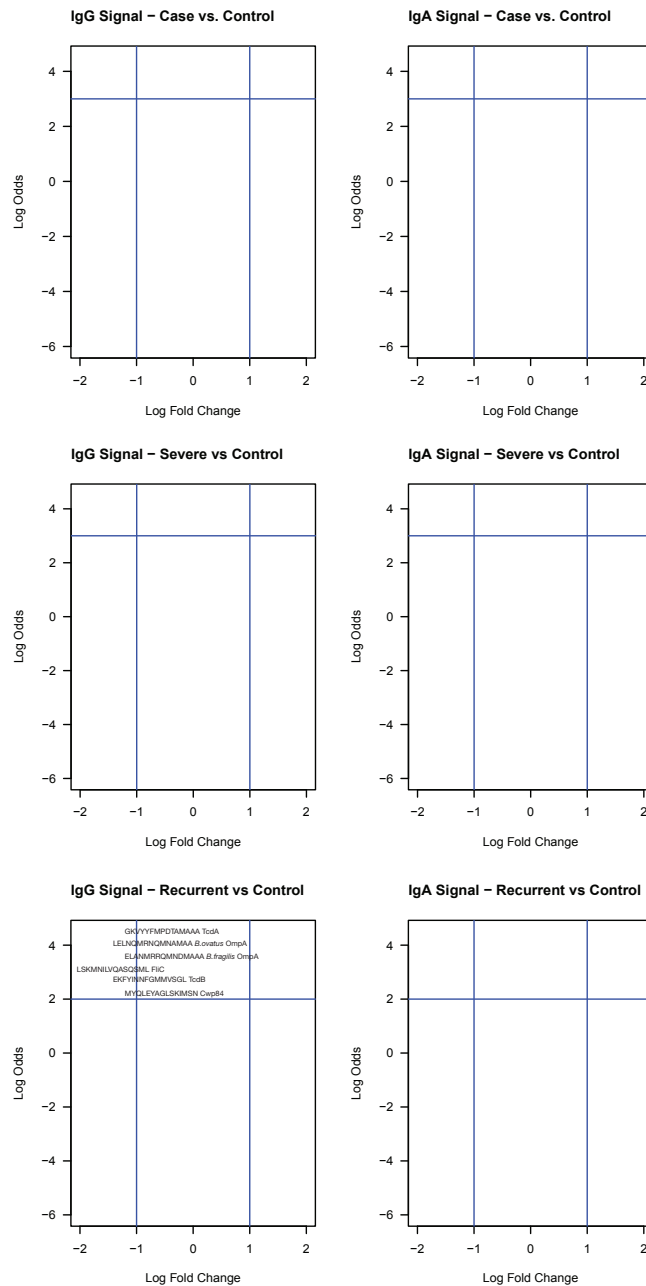


Figure 5.6: Volcano plots showing differential identification analysis of the Liverpool CDI dataset

The volcano plots show the mean difference between case and control on the x axis (log fold change) for each peptide plotted against the $-\log P$ -value on the y axis. Plots in the left column show the IgG antibody response signal and plots on the right show the IgA signal. The uppermost plots show a comparison for all cases (CDI+) vs. control (CDI-). No peptides are found that reach a statistically significance intensity difference between groups. The middle row of plots shows a subset of CDI+ cases selected by disease severity compared to all controls and the lowest row shows the subset of CDI+ patients that developed recurrent disease compared to all controls. The IgG signal in the recurrent disease subset shows a signature of 9 peptides that are significantly differentially identified (Table 5.5)

ORIGINAL IN COLOUR

5.8 Discussion

The data presented in this chapter applies the approaches I have developed in my research to investigate the differences in patterns of serum IgG and IgA reactivity to *C. difficile* peptides between patients diagnosed with CDI and hospitalised patients without CDI. Patients with CDI in both the Brighton and Liverpool cohorts underwent blood sampling early in the disease course (within 72 hours of diagnosis) – at this time the circulating IgG and IgA antibody profiles reflect the pre-existing immune response rather than a reactive response resulting from clonal selection and proliferation of antibody secreting B-cells and plasma cells. Hence, the hypothesis being tested is that there is a *pre-existing* difference in antibody reactivity that underlies susceptibility to *C. difficile* infection.

An important finding in this study is that the effect of process batching may bias the interpreted outcome despite rigorous pre-processing designed to minimise technical variation. In the case of the Brighton dataset arrays for the CDI+ group were processed separately to the CDI- group leading to an intractable source of bias. This phenomenon has been noted in genomic and transcriptomic array experiments and was apparent even following normalisation (Leek et al., 2010). No such effect has been described for peptide array studies previously but this may become much more apparent as larger sample groups are being analysed requiring multiple process batches. More sophisticated normalisation methods that take into account array processing variables may have to be employed to counter batch effects (Johnson et al., 2007). Because the outcome was intractably confounded with batch in the initial Brighton dataset no such process could be applied post hoc to the initial data. As a result I would recommend that care is taken to ensure sample randomisation occurs prior to processing and that sufficient control arrays are included in each process batch to be able to correct for such biases.

The first major finding of this study is that no antibody reactivity signature can be found using this peptide array that differentiates the CDI group as a whole from a non-CDI comparator group in either the Brighton or Liverpool dataset accounting for batch confounding. This is perhaps not a surprising finding given the ubiquity of *C. difficile* in the environment. The earliest studies of *C. difficile* noted that it was frequently found as constituent of the intestinal microflora in infants and so it would be extremely unlikely that the patients in this study had

not been exposed at some point to *C. difficile* in the environment even if they had not developed symptoms. An alternative explanation for this finding is that the array only contains peptides from two strains of *C. difficile* – 630, a ribotype 012 (toxintype 0) strain that was first to be fully sequenced, and R20291, a ribotype 027 (toxintype III) strain. However, my analysis of the conservation of proteins selected to include in the array suggests that all of the array proteins apart from the cell wall and surface proteins SlpA and Cwp66 are highly conserved and so any strain specific variation is likely to be very minor.

The second major finding of this chapter however is that differences in immune response to toxins do exist between single-episode and recurrent disease. This is in keeping with previous studies which have found immune responses to toxin B in particular to determine risk of recurrence, and are supportive of research which seeks to prevent recurrence with toxoid vaccination. It is important to note though, that the recurrent cases were a relatively small subset of the Liverpool dataset and that as none of the Brighton patients went on to develop recurrent disease in the study time period, I was not able to test this finding in another group.

However, this is the first study to apply the unconstrained approach of peptide arrays to detect immune responses in CDI. The robust analysis approaches and statistics I have applied demonstrates both the importance of immune responses to toxins in determining outcome but also the utility of my approach to array analysis.

Chapter 6

Antibody signatures characterising paediatric tuberculosis

If the importance of a disease for mankind is measured by the number of fatalities it causes, then tuberculosis must be considered much more important than those most feared infectious diseases, plague, cholera and the like.

ROBERT KOCH, 1882

It is a sobering thought that over a century has passed since Robert Koch uttered those words in his seminal lecture on the aetiology of TB, yet it remains one of the most common causes of death due to an infectious agent, second only to the Human Immunodeficiency Virus (HIV). The TB pandemic is now one of the biggest global health problems facing mankind. In 2011 the global TB prevalence was estimated at a total of 12 million cases (170 cases per 100,000 population) with 8.7 million new cases being diagnosed in 2011 alone. TB was responsible for 1.4 million deaths in 2011 (990,000 in HIV negatives and 430,000 HIV-associated TB deaths) (World Health Organization, 2014). This is despite the fact that TB is, in the majority of cases, a treatable disease. Globally, 85% of new TB cases (87% of sputum smear positive cases) were successfully treated in 2010 (World Health Organization, 2014). One of the biggest problems of the pandemic

remains the identification of TB cases. Although there have been significant improvements over the last decade the case detection rate (CDR) – defined as the number of notified TB cases divided by the estimated number of incident infections remains low – 66% globally in 2011. This figure hides the considerable global variation in CDR with the highest rates of case detection in Brazil (91%), China (89%), Kenya (81%), the Russian Federation (81%) and Tanzania (76%); by contrast, the lowest rates were reported from Mozambique (34%), Bangladesh (45%), Nigeria (45%) and Afghanistan (46%) (World Health Organization, 2014). The consequences of delayed or missed diagnoses are severe - impacting not only the affected individuals but also contributing to the onward transmission of the infection. Given the improvements made in treatment delivery, case detection is now the 'rate limiting step' in the control of the TB pandemic (Perkins and Kritski, 2002).

6.1 Diagnosis of tuberculosis

TB is spread by droplet infection through the respiratory system. It is the identification of the characteristic acid-fast bacilli on microscopy of a sputum specimen followed by isolation of the organism in culture that has stood at the frontline of TB diagnosis for over a century. However, sputum microscopy has a low sensitivity, requires special laboratory facilities to be carried out safely and effectively and is cumbersome for the patient, often requiring several specimens over a period of time to make a diagnosis. It is also useless for diagnosis of extrapulmonary disease and its already poor sensitivity is further reduced in paucibacillary cases typical of paediatric TB and in HIV co-infected individuals. Mycobacterial culture on the other hand, remains the most sensitive method for diagnosis of TB and is widely regarded as the 'gold standard'. However, it too requires special laboratory facilities and even with modern liquid culture techniques, it can take several weeks to confirm a result (Drobniewski et al., 2003). As a result, a considerable amount of attention has now turned to the development of new diagnostic methods that have good sensitivity and specificity, can be performed rapidly and require a minimum of specialised laboratory equipment and training. In particular, there is considerable interest in developing tests that

can be used at the point of care, as the overwhelming burden of TB occurs in resource poor settings where laboratory access may be difficult or impossible (McNerney and Daley, 2011).

6.1.1 Serological diagnosis of TB

Serological diagnosis has, for many decades, been seen to be an attractive method for the diagnosis of TB. Serology is well established for the diagnosis of viral infections such as HIV, Epstein Barr Virus (EBV), and viral Hepatitis, and bacterial infections such as Mycoplasma and Group A Streptococci. Moreover, the expertise and technology to perform such tests is widespread. However, perhaps the most attractive feature of serology, especially for a disease such as TB where the overwhelming burden is located in resource poor settings, is that it offers the potential for development of a point of care (POC) test (McNerney and Daley, 2011). A POC test is one that can be performed without needing to transport the sample to a laboratory for processing. Typically the tests are simple to perform, without requiring extensive laboratory training, and the results can be obtained in a matter of minutes.

As a result, several commercial and 'in-house' serological diagnosis kits have been developed using a wide variety of methods and diagnostic antigens. Moreover, a number of these assays have been adapted into POC tests using lateral flow immunochromatography or other similar technology (Grenier et al., 2012). However, despite the availability of these tests, their clinical validation in many cases lacks rigor and often show poor diagnostic sensitivity and specificity (Steingart et al., 2007b,a, 2011). As a result the WHO now recommends that existing commercial serological testing kits should no longer be used for the routine diagnosis of TB and that further research needs to be done in this area.

The major obstacle hampering development of an effective serological test for TB is that the antibody response to TB is directed to a large number of antigens, and that the response is extremely variable from one person to another (Lyashchenko et al., 1998). There are a number of reasons why such variability is seen. Foremost among those is the immunogenetic background of the host. In a mouse model, vaccination with BCG results in a different pattern of antibody production

between different mouse strains (Huygen et al., 1990). Similarly in humans, there is some evidence that HLA haplotype can influence production of specific antibodies to *Mycobacterium tuberculosis* (MTB) (Bahr et al., 1988). Another factor influencing antibody production is the stage of disease. In animal models of infection different antibodies are produced at different stages of infection (Min et al., 2011; Lyashchenko et al., 1998) In human disease, diagnosis is made at varying time points, often with no clear idea of the duration of infection. Aside from these intrinsic reasons for person to person variability, there may be additional confounding factors that may amplify the differences seen. The variation in human immune response with disease state was demonstrated in a high throughput screening study using a custom MTB proteome protein microarray (Kunnath-Velayudhan et al., 2010). The authors screened sera from over 500 patients and showed an evolving antibody response pattern correlating with bacillary load. They identified an ‘immunoproteome’ of approximately 500 proteins that were identified by one or more of the sera in the study but confirmed the enormous variability of the antibody response seen in previous studies with no one protein showing significant diagnostic potential.

6.1.2 Serological diagnosis of tuberculosis in children

In 2011 it was estimated that there were 490,000 new cases (6% of all incident cases) of TB in children under 15 years of age (World Health Organization, 2014). Often because of the atypical presentation of paediatric TB, many cases do not present in the early stages to health care facilities. In fact finding cases of TB infected children typically occurs after screening of exposed household contacts after diagnosis of an adult case. The difference in clinical presentation is also reflected in the ability of conventional diagnostic tests to detect TB infection in children. Sputum smear microscopy has an extremely poor yield – less than 10% in some studies (Cruz and Starke, 2010). This is partly because cavitating pulmonary disease does not often develop (Marais et al., 2006), and also because expectorating sputum is an unrealistic expectation for young children. For similar reasons nucleic acid amplification tests (NAATs) also perform poorly when using respiratory specimens. A study of the Cepheid Gene Xpert MTB test in South African children demonstrated a sensitivity of only 13% compared to 6% for

sputum smear microscopy. Even mycobacterial culture, the 'gold-standard' in adults, performed poorly in this setting, with a sensitivity of only 16% (Nicol et al., 2011). This is typical of other studies using culture, where the maximal detection rate is 30 to 40% and often below 20% of cases in most settings (Cruz and Starke, 2010). Sampling extrapulmonary tissue, although potentially useful, is much more invasive and difficult particularly for children (Achkar and Ziegenbalg, 2012).

As a result, there is a considerable amount of interest in finding immunological based diagnostics for paediatric TB. The immune system is capable of amplifying the signal from TB infection and presenting it in a manner that can allow collection at a site distant from the infection. Cellular immunity detection tests – the tuberculin skin testing (TST) and the interferon gamma release assay (IGRA) have been used for TB diagnosis in children with some success. A meta-analysis of 67 paediatric TB studies showed a diagnostic sensitivity of between 40-100% for various IGRAs alone, with an increase in sensitivity if the tests were used alongside TST (Chiappini et al., 2012). However, the accuracy cellular immunity tests drops dramatically in very young children (less than 5 years old) most likely due to the immaturity of T-cell immune response. Similarly, their effectiveness is also diminished by HIV co-infection. Furthermore, just as in adults, neither TSTs nor IGRAs can reliably distinguish latent disease from active TB.

In contrast, serology has, in certain cases, been shown to be able to differentiate between latent and active infection (Steingart et al., 2009). This is potentially much more useful for a diagnostic test where the focus is on identifying active cases requiring treatment. Despite this, just as in adult disease, the accuracy of diagnostic serology in paediatric TB remains poor with highly variable responses to the antigens used in tests (Steingart et al., 2007a, 2011). Many of the reasons for this variability are the same as for adult disease but for children there is an important additional confounder – the effect of age on the maturation of the humoral immune system. A recent review of serological studies in paediatric TB showed that the majority evaluated a very broad age range of children without finer stratification into narrower age categories (Achkar and Ziegenbalg, 2012).

6.1.3 Evaluation of serodiagnostic antigens

Many fewer different antigens have been tested in children than in adult disease. The earliest studies used extracts of whole-cell lysates (mycobacterial sonicates), or purified protein derivative (PPD) – also called tuberculin, as diagnostic antigens. PPD is prepared from the culture filtrate of MTB or other mycobacteria. Both these preparations are comprised of a large variety of protein, glycolipid and polysaccharide antigens. Hence standardisation is extremely difficult and it is impossible to know the relative contribution of any one given antigen for the preparations used in the study. Moreover, many of the antigens represented in these extracts are not specific to MTB and are represented in a variety of environmental mycobacteria and BCG. Unsurprisingly, an extremely broad range of sensitivities (20% to 63%) and specificities (40% to 97%) were reported in studies using these preparations (Barrera et al., 1989; de Larrea et al., 2006; Rosen, 1990; Zheng et al., 1994).

The A60 complex is a lipopolysaccharide – protein complex found in the cell wall and cytoplasmic fractions of MTB. It is comprised of approximately 30 components – some of which have been definitively identified. These include Lipoarabinomannan (LAM) a cell wall glycolipid, Rv0440 (65 kDa antigen), Rv2780 (40 kDa antigen), Rv0934 (38 kDa antigen), Rv2744c (35 kDa antigen), Rv3763 (19 kDa antigen) and Rv2031c (16 kDa antigen). A60 is used in two commercially available tests: The Anda-TB kit (Anda Biologicals, Strasbourg, France) and Immunozyne Mycobacterium (Assay Designs, Ann Arbor). Because many of the components of the A60 complex are found in a variety of different mycobacteria, in a similar way to the crude extracts, it is likely that there is a considerable amount of cross-reactivity in the serological responses detected. Indeed, in studies using the Anda-TB kit in children, the reported sensitivities varied between 14 to 71% with specificities ranging between 50 to 100% (Delacourt et al., 1993; Gupta et al., 1997; de Larrea et al., 2006).

A number of studies have reported the antibody response to specific antigens (protein, glycolipid or polysaccharide) in childhood TB. These studies have been performed either using commercial assays or to ‘in-house’ developed assays. The two most commonly tested antigens are Rv0934 (Ag5 / 38 kDa antigen) and Rv2031c (16 kDa antigen). Neither of these proteins are specific to MTB – Rv0934

is expressed by *Mycobacterium bovis* BCG as well as other mycobacteria, and Rv2031c is a member of the heat shock protein family and is expressed by a large number of different mycobacteria. Sensitivities and specificities in adult studies using these antigens is lacking (Steingart et al., 2007a, 2011) and studies in children shows a large variation in reported accuracy. 4 proteins have previously shown modest diagnostic utility in paediatric populations: Rv0934 (Ag5 / 38 kDa antigen), Rv2031c (16 kDa antigen), Rv3874 (CFP-10), Rv3875 (ESAT-6). The remaining 37 proteins have only shown diagnostic utility in adult populations.

6.2 Methods

6.2.1 Clinical cohort

Study participants were recruited as part of a European Union funded (EU action for diseases of Poverty Program grant, Sante/2006/105-061), multi-centre study to identify biomarkers for diagnosis of paediatric TB in Africa. This project included paediatric sample collection in South-Africa, Malawi and Kenya.

For this analysis, only samples from one study centre were used (Red Cross War Memorial Hospital, Cape Town, South Africa). Patients were enrolled from February 2008 to June 2010. A total of 100 children contributed samples for this analysis – 50 cases with acute pulmonary ± extrapulmonary TB, and a comparator group of 50 with other infections (predominantly lobar or bronchopneumonia) where TB had been suspected but definitively excluded. Exclusion criteria included children under the age of 6 months, HIV infected children, children with severe hypoxia (Oxygen saturation by pulse oximetry < 90% on room air), weight < 5kg and children who had completed TB treatment within 4 weeks of potential enrolment.

Samples were split randomly into two sets: a training set of 60 samples (30 TB cases / 30 Non-TB), and a test set of 40 samples (20 TB cases / 20 Non-TB). Randomisation was performed by the clinical team in South Africa and training set samples were sent for processing first before the test set was sent.

6.2.2 Case Definitions

TB cases were defined by a positive culture of MTB complex obtained from an induced sputum specimen. Cases with additional extrapulmonary disease were defined on the basis of a positive culture of MTB complex from a non-pulmonary site (eg. Cerebrospinal fluid or lymph node aspirate). The comparator population was defined as TB negative on the basis of two negative sputum cultures, a non-reactive Mantoux and negative baseline and 3 month follow-up IGRA.

6.2.3 Sample collection and storage

Serum samples were collected prior to the initiation of anti-tuberculous treatment. Whole blood was collected without anticoagulation. After centrifugation, the serum supernatant was aliquotted and frozen at -80C. No sample underwent more than one freeze-thaw cycle prior to analysis. Samples were anonymised and blinded with respect to the case definition and other phenotypic characteristics until after array processing and data extraction had been completed.

6.2.4 Peptide microarray design

Peptide synthesis and custom microarray printing was undertaken by JPT Peptide Technologies GmbH, Berlin, as described in Chapter 3. The array peptides were derived from overlapping 15mer sequences from 41 proteins of MTB strain H37Rv (table 6.1) selected on the basis of their performance in previous serological studies in paediatric and/or adult populations as described above.

Each array was comprised of 3 replicated subarrays of 3600 features (total 10800 features). The full array surface was used to incubate each serum sample and data from each subarray was collected to give triplicate results for each subject. 2749 peptide features per subarray were derived from MTB protein sequences. In addition to these there were 490 other peptide features: 465 derived from Dengue virus sequences and 25 random peptide sequences. 12 features per subarray were printed with human immunoglobulin – 3 for each subtype (IgG, IgM, IgA, IgE). Finally 330 features were left ‘empty’ and printed only with the spotting buffer and no peptide or protein.

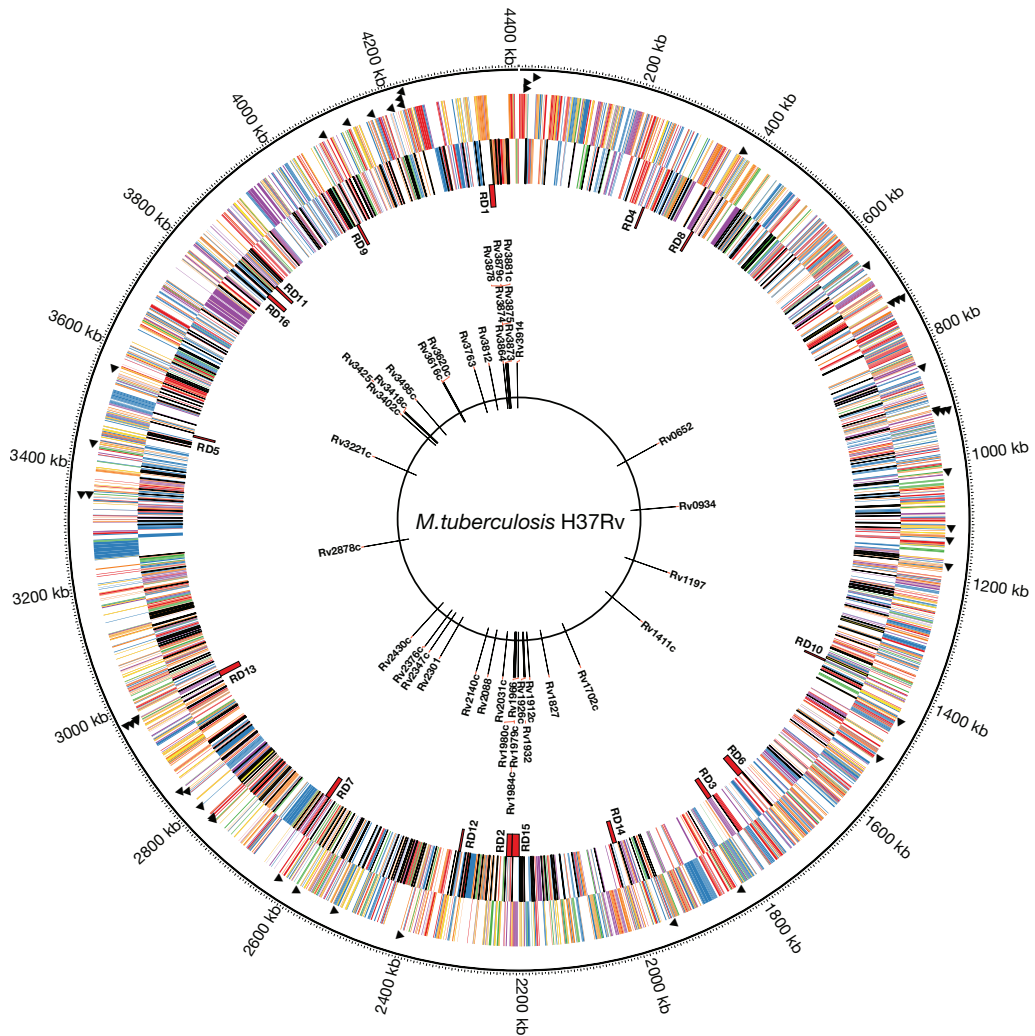


Figure 6.1: MTB protein sequences in relation to H37Rv genome and regions of difference (RD).

The outer ideogram displays coding sequences on the forward strand and the inner ideogram shows coding sequences on the reverse strand. tRNAs are indicated by the black triangles. Coding sequences are colour coded by protein function (Red – macromolecule metabolism, Blue – small molecule metabolism, Green – cell processes, Purple – other function including virulence factors, Orange – conserved hypotheticals, and Yellow – unknown). RD regions – coding sequences that differ substantially from *M. bovis* (BCG) are shown in red on the innermost ideogram. The peptide microarray proteins are shown on the central ring. The majority of protein sequences represented are from the RD1 and RD2 regions although other proteins that have shown significant antigenicity from previous studies are also included.

ORIGINAL IN COLOUR

Accession	Gene	RD	Protein Name	Length (aa)
Rv0652	<i>rplL</i>	0	PROBABLE 50S RIBOSOMAL PROTEIN L7/L12 RPLL	130
Rv0934	<i>pstS1</i>	0	PERIPLASMIC PHOSPHATE-BINDING LIPOPROTEIN PSTS1 (38kDa protein)	374
Rv1197	<i>esxK</i>	0	ESAT-6 LIKE PROTEIN ESXK	297
Rv1411c	<i>lprG</i>	0	PROBABLE CONSERVED LIPOPROTEIN LPRG	236
Rv1702c	<i>rv1702c</i>	0	CONSERVED HYPOTHETICAL PROTEIN	454
Rv1827	<i>garA</i>	0	CONSERVED PROTEIN WITH FHA DOMAIN, GARA	162
Rv1912c	<i>fadB5</i>	0	POSSIBLE OXIDOREDUCTASE FADB5	334
Rv1926c	<i>mpt63</i>	0	IMMUNOGENIC PROTEIN MPT63	159
Rv1932	<i>tpx</i>	0	PROBABLE THIOL PEROXIDASE TPX	165
Rv1966	<i>mce3A</i>	15	MCE-FAMILY PROTEIN MCE3A	425
Rv1979c	<i>rv1979c</i>	2	POSSIBLE CONSERVED PERMEASE	481
Rv1980c	<i>mpt64</i>	2	IMMUNOGENIC PROTEIN MPT64	228
Rv1983	<i>PE_PGRS35</i>	2	PE-PGRS FAMILY PROTEIN PE_PGRS35	558
Rv1984c	<i>cfp21</i>	2	PROBABLE CUTINASE PRECURSOR CFP21	217
Rv1986	<i>rv1986</i>	2	PROBABLE CONSERVED INTEGRAL MEMBRANE PROTEIN	199
Rv1987	<i>rv1987</i>	2	POSSIBLE CHITINASE	142
Rv2031c	<i>hspX</i>	0	HEAT SHOCK PROTEIN HSPX (16kDa protein)	144
Rv2088	<i>pknJ</i>	0	TRANSMEMBRANE SERINE/THREONINE-PROTEIN KINASE J PKNJ	589
Rv2140c	<i>TB18.6</i>	0	CONSERVED PROTEIN TB18.6	176
Rv2301	<i>cut2</i>	0	PROBABLE CUTINASE CUT2	230
Rv2347c	<i>esxP</i>	7	PUTATIVE ESAT-6 LIKE PROTEIN ESXP (ESAT-6 LIKE PROTEIN 7)	98
Rv2376c	<i>cfp2</i>	0	LOW MOLECULAR WEIGHT ANTIGEN CFP2	168
Rv2430c	<i>PPE41</i>	0	PPE FAMILY PROTEIN PPE41	194
Rv2878c	<i>mpt53</i>	0	SOLUBLE SECRETED ANTIGEN MPT53 PRECURSOR	173
Rv3221c	<i>TB7.3</i>	0	BIOTINYLATED PROTEIN TB7.3	71
Rv3402c	<i>rv3402c</i>	16	CONSERVED HYPOTHETICAL PROTEIN	412
Rv3418c	<i>groES</i>	0	10 KDA CHAPERONIN GROES	100
Rv3425	<i>PPE57</i>	11	PPE FAMILY PROTEIN PPE57	176
Rv3495c	<i>lprN</i>	0	POSSIBLE MCE-FAMILY LIPOPROTEIN LPRN	384
Rv3616c	<i>espA</i>	0	ESX-1 SECRETION-ASSOCIATED PROTEIN A	392
Rv3620c	<i>esxW</i>	9	PUTATIVE ESAT-6 LIKE PROTEIN ESXW (ESAT-6 LIKE PROTEIN 10)	98
Rv3763	<i>lpqH</i>	0	19 KDA LIPOPROTEIN ANTIGEN PRECURSOR LPQH	159
Rv3812	<i>PE_PGRS62</i>	0	PE-PGRS FAMILY PROTEIN PE_PGRS62	504
Rv3864	<i>espE</i>	0	ESX-1 SECRETION-ASSOCIATED PROTEIN ESPE	402
Rv3873	<i>PPE68</i>	1	PPE FAMILY PROTEIN PPE68	368
Rv3874	<i>esxB (cfp10)</i>	1	10 KDA CULTURE FILTRATE ANTIGEN ESXB (LHP) (CFP10)	100
Rv3875	<i>esxA (esat6)</i>	1	6 KDA EARLY SECRETORY ANTIGENIC TARGET ESXA (ESAT-6)	95
Rv3878	<i>espJ</i>	1	ESX-1 SECRETION-ASSOCIATED PROTEIN ESPJ.	280
Rv3879c	<i>espK</i>	1	ESX-1 SECRETION-ASSOCIATED PROTEIN ESPK.	729
Rv3881c	<i>espB</i>	0	SECRETED ESX-1 SUBSTRATE PROTEIN B, ESPB.	460
Rv3914	<i>trxC</i>	0	THIOREDOXIN TRXC (TRX) (MPT46)	351

Table 6.1: *Mycobacterium tuberculosis* proteins used for peptide array immunoassay

Full length amino acid sequences from strain H37Rv for each of the proteins listed above were parsed into overlapping 15mer peptides for immobilisation on to the peptide array as previously described in Chapter 3. Each peptide sequence was represented in triplicate on the array. Accession numbers for the protein sequences refer to the TubercuList database (<http://tuberculist.epfl.ch/>)

6.2.5 Microarray Processing

Microarrays were processed as previously described in chapter 3. Sera were thawed immediately prior to processing and diluted by 1 part in 100 with blocking buffer. After primary incubation and washing, arrays were incubated with a single secondary antibody: goat anti-human IgG – AlexaFluor™ 647 (Invitrogen, USA – Cat. No. A21445) diluted to 1µg/ml in blocking buffer. Two arrays were reserved for negative controls (primary incubation with buffer only). Images were acquired within 48 hours of processing using a GenePix® 4300 microarray scanner. Laser power was set to 100% with a PMT gain of 600V for all slides in this series. Image processing was carried out using GenePix®Pro 7.2.

6.2.6 Data pre-processing

All pre-processing was carried out using the R statistical programming environment (R Development Core Team, 2014) with the `pmpa` package as described previously. Background signal was subtracted from foreground with a positive offset of 128 (2^7) applied. Arrays were quantile normalised and mean summarised with outliers filtered if the per-peptide CV exceeded 30%. Two slides were reserved for negative control incubations (primary incubation with blocking buffer and secondary incubation with the fluorophore conjugated secondary antibody

After examination of array slides processed with just the secondary antibody 11 peptides were filtered as being strong non-specific secondary antibody binders. Finally all non-MTB peptides were excluded to give a final array of 2575 features that was used for data analysis.

6.2.7 Data Analysis

Filtering

A Z-score filter was used as described by Kunnath-Velayudhan et al. (2010). A Z-score $Z_{i,j}$ for each pre-processed intensity value was calculated as

$$Z_{i,j} = \frac{\log(I_{i,j}) - \mu_{i_{ctrl}}}{\sigma_{i_{ctrl}}} \quad (6.1)$$

where $I_{i,j}$ is the signal intensity of peptide i in TB sample j , and $\mu_{i_{ctrl}}$ and $\sigma_{i_{ctrl}}$ are the mean and standard deviation respectively of peptide i from the non-TB control group. The upper tailed p -values were calculated from the Z-scores and after adjustment for multiple testing using a Benjamini - Hochberg false-discovery rate (Benjamini and Yekutieli, 2001) (threshold <0.01) a set of peptides was identified with significant reactivity in one or more TB sera.

Peptides identified from the Z-score filter were then filtered again by fitting a ROC curve to each peptide in turn. The area under ROC curve (AUROC) and bootstrapped 95% confidence intervals were calculated. Peptides were selected if the lower bound of the confidence interval was >0.5 .

Classification

Binary classification into TB cases vs. control was performed on the filtered training set using four classifiers: Linear discriminant analysis (LDA), a Support Vector Machine (SVM) classifier with linear and radial kernels (Guyon et al., 2002), and a Random Forest classifier (Breiman, 2001). 10-fold cross validation was used in order to assess classifier performance for the LDA and SVM classifiers. Cross validation involves random partition of the training set into equal sized subsamples with one subsample being used as a validation set and the remaining subsamples used to train the classifier. The process is repeated until all subsamples have been used for both training and validation. This gives a measure of the classifier's performance on the known dataset and estimates how it will perform on an unseen set. Random Forest classifiers essentially calculate a form of cross-validation during their computation known as the 'out of bag (OOB)' error. This was used instead of cross-validation to assess the performance of the Random Forest. The best performing classifier (lowest cross-validation error) was then validated against the test dataset.

6.3 Results

6.3.1 Training Set Description

Sixty samples were randomly allocated for training and cross validation of a classification algorithm (30 confirmed TB cases, and 30 with non-TB other infections). Randomization was performed by the clinical team in South Africa and samples for this set were sent and processed separately to the subsequent test set.

Five samples were excluded from the analysis (2 TB cases and 3 controls) due to poor data quality. A further 3 samples were excluded because of mislabeling – sample identifiers could not be matched to the clinical data. Demographic and baseline clinical details for the analysed study participants are shown in table 5.4. Of the 25 TB cases, all had pulmonary disease (positive induced sputum culture). Fourteen were diagnosed with co-existing extra-pulmonary TB (2 case of miliary disease, 4 with TB meningitis, 3 with tuberculous pleural effusions, 1 with splenic disease (and a pleural effusion), 2 with abdominal lymphadenitis and 2 with cervical lymphadenitis). Among the 27 controls there were 13 cases of lobar pneumonia, 5 cases of bronchopneumonia, 2 cases of meningitis, 3 cases of gastroenteritis, 2 with septicaemia (undefined infection source) and 1 cases of an upper respiratory tract infection.

Notably there was a statistically significant difference in age between the TB cases and the control group. The median age of TB positive children studied was 39 months (IQR 14.75 – 118), compared to 18 months (IQR 11.5 – 34), $p = 0.02$. The ages of the TB cases show a bimodal distribution with the two modal peaks at 12 months and 130 months. 11 children were under 1 year of age (4 TB cases and 7 controls), and 13 children were over the age of 5. As expected from the age distribution these were almost entirely from the TB cases (12 TB cases vs. 1 control).

As expected from the older median ages of the TB cases, there was also a statistically significant difference in weight (median weight of TB cases 13.4kg, median weight of controls 9.1kg, $p=0.03$). However, weight age Z-scores (WAZs) were not significantly different between the groups (TB cases -1.46, controls -

	TB (PTB ± EPTB) n = 27	Control n= 25	p-value *
Age in months (Median, IQR)	59 (12 - 119)	18 (12.5 – 31)	0.005
Weight (kg) (Median, IQR)	14.25 (8.9 – 25.8)	9.1 (8.1 – 12.1)	0.01
Weight Age Z-Score (WAZ) (Mean, SD)	-1.48 (1.73)	-1.56 (1.32)	0.84
Gender			
females	11 (39%)	9 (36%)	1
males	16 (59%)	16 (64%)	
Previous TB treatment			
yes	2 (7%)	1 (4%)	1
no	25 (93%)	24 (96%)	
Household TB contact			
yes	15 (60%)	10 (37%)	0.4
no	11 (40%)	17 (63%)	
TST induration (mm) (Median, IQR)	17 (15 – 20)**	0 (0 – 0)	<0.001
Baseline IGRA			
positive	20 (74%)	0 (0%)	<0.001
negative	7 (26%)	25 (100%)	
3 month IGRA			
positive	13 (48%)	0 (0%)	<0.001
negative	6 (22%)	25 (100%)	
unknown	8 (30%)	0 (0%)	

Table 6.2: Summary descriptive statistics for the training set

Demographic, TB exposure and clinical immunology data is shown for the 52 analysed training cases categorised by TB status. Of the 60 supplied samples for training 8 were excluded due to poor data quality (n=5) and mislabelling (n=3).

* p values determined by Wilcoxon Rank Test, Z test or Chi Squared Test as appropriate

** 5 non reactive TST cases not included

1.56) implying that for a given age the participants' weight was similar. Compared to aged matched population controls, the children in the study were underweight – overall mean WAZ was -1.50 (i.e. 1.50 standard deviations below the population mean). More males were recruited to the study (60%) than females, although the relative proportions of each gender were not significantly different between the TB cases and controls (TB cases 63%, controls 57%, $p=0.87$).

All subjects underwent Mantoux tuberculin skin testing and had blood drawn for IGRA at study screening. Seven patients from the TB case group had no measurable TST induration. Of the remaining 21 cases the median induration diameter was 17mm (IQR 15mm to 20 mm). The induration diameter was greater than or equal to 10mm for all of the responding cases. Baseline IGRAs were positive in 22 (79%) of the TB cases. Of the 6 negative cases at baseline, one had become positive at the 3 month follow up IGRA, 3 remained negative and two were lost to follow-up. Overall, 8 of the TB cases were lost to follow up at the 3 month point so no IGRA result was available (Baseline IGRA was positive in 6 of these cases). Two cases had a negative 3 month IGRA following a positive baseline test.

6.3.2 Feature Selection

Serum reactivity of each of the TB cases was assigned for each analysed peptide feature by calculating a Z-score filtering as previously described. 796 features (22% of all analysed array features) were retained after filtering. Figure 6.2 summarises the Z-scores (red radial bar plot) and the frequency that each peptide is selected (black radial bar plot) for all features analysed. 618 peptides (78%) were significantly reactive in only 1 serum sample, 155 in 2 samples, 30 in 3 samples, 7 in 4 samples and 2 peptides were reactive in 5 samples.

As a large proportion of peptides were only reactive in one or two sera, a secondary feature selection was conducted by ROC analysis of each of the selected peptides in turn. The area under the curve (AUC) of each ROC plot was computed along with its 95% confidence interval (calculated using 2000 bootstrap replications). Using a threshold for the lower bound of the confidence interval of 0.5, 41 peptides were selected and with a threshold of 0.55, 8 peptides were

selected (Table 5.3).

6.3.3 Classification and Cross-Validation

Binary classification using the selected features was performed using the four algorithms previously discussed (LDA, SVM-Linear, SVM-Radial and Random Forest). The results of the training set accuracy and 10-fold cross validation accuracy (for LDA, SVM-Linear and SVM-Radial) are shown in the tables below (Table 6.4 and 6.5). No cross-validation was performed for the random forest classifier, rather the out of bag (OOB) error was calculated as an unbiased test set error estimate. Classification was performed using the 40 peptide feature set obtained using the AUROC cutoff of 0.5, and the 8 peptide set using the cutoff of 0.55. SVM with a radial (Gaussian) kernel was the best performing algorithm with a cross-validation error of 23.6% (11.1 – 37.0) when used with the 41 feature dataset. This algorithm was therefore used for validation on the test set.

6.3.4 Test Set Description

Forty samples were randomly allocated for validation of the trained classification algorithm (20 confirmed TB cases, and 20 with non-TB other infections). Four samples were excluded from the analysis (1 TB case and 3 controls) due to sample mislabelling. Demographic and baseline clinical details for the analysed study participants are shown in table 6.6. Of the 19 TB cases, all had pulmonary disease (positive induced sputum culture). Six were diagnosed with co-existing extra-pulmonary TB (1 case of miliary disease with a tuberculoma, 2 cases of cerebral tuberculomas, 2 with TB meningitis, and 1 with cervical lymphadenitis). Among the 17 controls there were 13 cases of lobar pneumonia, 2 cases of bronchopneumonia, and 2 cases of upper respiratory tract infections.

Comparing between the TB cases and the control group, there was no statistically significant difference in age, weight, gender (proportion of males), exposure to previous TB treatment, and the known presence of a household TB contact. Although the median age of the children in the study was 15.5 months, a wide

Peptide Sequence	Protein Accession	Sequence Position	AUC ROC	(95% CI)
SAYDLGTVIGFYGLT	Rv2088	449	0.756	(0.628 - 0.884)
QGGWMLSRASAMELL	Rv2376c	149	0.725	(0.588 - 0.863)
AIVSVGLAVSYDYR	Rv1979c	421	0.723	(0.586 - 0.86)
AVIGVAVLVTAVSFT	Rv1966	21	0.72	(0.584 - 0.856)
NVEAQISATTAFGAK	Rv1966	101	0.715	(0.574 - 0.855)
MPQNPSRARLSAGAV	Rv1966	121	0.71	(0.569 - 0.85)
NATFRTIIVVGALIS	Rv1979c	297	0.708	(0.568 - 0.849)
VTAVSFTGSLRSTVP	Rv1966	29	0.694	(0.55 - 0.837)
NFAVTNDGVIFFFNP	Rv1980c	190	0.688	(0.544 - 0.832)
QIIANQQVYWQQIAA	Rv3812	149	0.683	(0.538 - 0.828)
TAPTAVNVVLLSIPT	Rv1983	353	0.682	(0.539 - 0.824)
VASVGGQPSQATQLL	Rv3878	105	0.678	(0.53 - 0.825)
LSISGYSGLLYIFA	Rv1983	325	0.675	(0.528 - 0.822)
VFGPNPLPAPNVEVV	Rv1983	437	0.674	(0.525 - 0.822)
PMSGIAMAVVGGALL	Rv3864	137	0.674	(0.526 - 0.821)
NNKAALEPVNPPKPP	Rv3881c	265	0.672	(0.527 - 0.818)
AGWQTLAALDAQAV	Rv3873	29	0.67	(0.523 - 0.817)
QISATTAFGAKFVDL	Rv1966	105	0.668	(0.522 - 0.815)
LSDFVKKFEETFEVT	Rv0652	21	0.667	(0.52 - 0.814)
LDADERHTAINSLVT	Rv3879c	129	0.667	(0.522 - 0.812)
NARGDTIGGNWRSLK	Rv1966	197	0.666	(0.517 - 0.815)
DEDDIKATYDKGILT	Rv2031c	109	0.666	(0.52 - 0.812)
IAQQLRAQVMGDLDK	Rv3864	69	0.664	(0.518 - 0.811)
HITQAVLTATNFFGI	Rv3873	113	0.663	(0.517 - 0.81)
LAAPGMANPADDTPC	Rv1702c	229	0.66	(0.51 - 0.811)
QIGRIEWAQNGASLR	Rv1966	73	0.659	(0.512 - 0.806)
GKNRNHVNFQELAD	Rv3616c	65	0.659	(0.512 - 0.806)
CGAEGTENVMPASAF	Rv1932	93	0.656	(0.507 - 0.806)
HLHVATLANGTLALL	Rv3402c	85	0.656	(0.504 - 0.809)
PGDIFNTGSSLFKQI	Rv3864	29	0.656	(0.51 - 0.803)
QLASMGSQQAQLISS	Rv3864	341	0.656	(0.508 - 0.805)
MAEMKTAATLAQEA	Rv3874	1	0.656	(0.51 - 0.803)
RLHDAWWTLILFAVI	Rv1966	9	0.655	(0.508 - 0.802)
VSIAATVAGCSSGS	Rv1411c	17	0.654	(0.505 - 0.803)
FIVGGLWIITTQHVN	Rv1979c	177	0.654	(0.507 - 0.801)
GSGLPANTNIEVYT	Rv1983	489	0.651	(0.503 - 0.799)
CPFCNAEAPSLSQVA	Rv2878c	73	0.651	(0.504 - 0.799)
ERHTAINSLVTATHG	Rv3879c	133	0.651	(0.503 - 0.799)
NAAAASTTALAAAG	Rv1983	29	0.649	(0.5 - 0.797)
IKATYDKGILTVSVA	Rv2031c	113	0.649	(0.501 - 0.796)

Table 6.3: MTB peptides selected after Z-score and ROC filtering

Dataset of 41 peptides showing significant reactivity in patients with MTB compared to the control population. Feature selection was performed by firstly selecting peptides that showed significant reactivity by Z-score in at least one patient compared to the controls. This large set of 618 peptides was further filtered by computing the area under the ROC curve (AUROC) for each peptide in turn. Selection was based on the lower bound of the bootstrapped confidence interval being >0.5

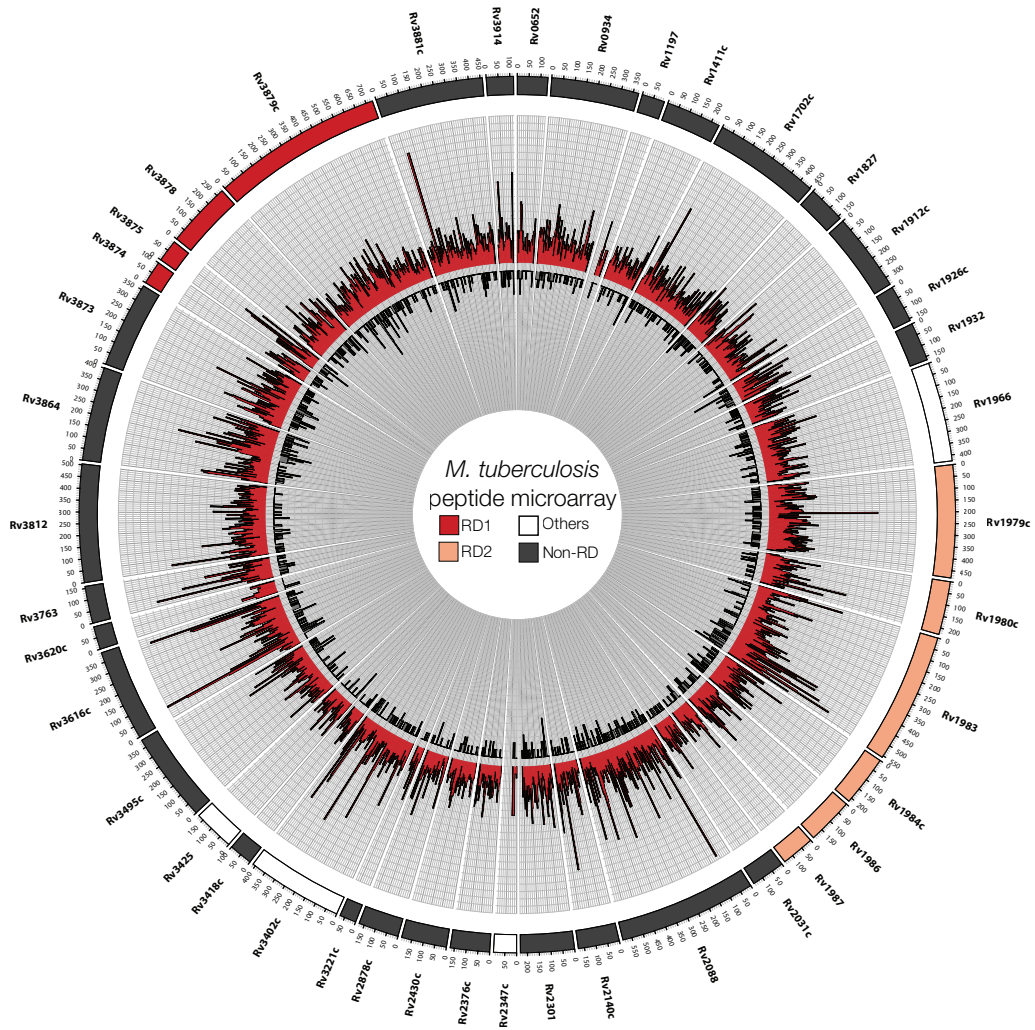


Figure 6.2: MTB peptide reactivity measured by Z-score (training set)
 The arrayed MTB proteins are arranged on the above plot as a circular track reading clockwise from the N-terminal to the C-terminal of the primary sequence. The proteins corresponding to the RD1 and RD2 regions are coloured on the outer track in red and salmon respectively. Non-RD proteins are shaded in dark grey. The red coloured radial bar plot shows the Z-score for the peptide sequence corresponding to the protein and start position on the outer track. The inner black radial histogram shows the frequency of active TB patients with a peptide significant peptide Z-score (corresponding to a p -value of <0.01 after multiple test adjustment – false discovery rate).

Number of peptide features	LDA	SVM-Linear	SVM-Radial	Random Forest
41	0 (0 – 6.5)	0 (0 – 7.0)	10.9 (4.2 – 22.3)	0 (0 – 7.0)
8	21.8 (11.8 - 35.1)	20 (10.4 - 33.0)	20 (10.4 – 33.0)	0 (0 – 6.5)

Table 6.4: Training error of classification algorithms applied to the training dataset

Classification error calculated for the full training set without cross validation. Each algorithm (Linear Discriminant Analysis (LDA), Support Vector Machine (SVM)-Linear, SVM-Radial and Random Forest) is shown in a separate column. The top row shows the errors for the 41 peptide dataset and the bottom for the 8 peptide set. 95% confidence intervals (calculated from 1000 bootstrap replications) are shown in parentheses. Out of Bag (OOB) error is shown for the random forest classifier.

Number of peptide features	LDA	SVM-Linear	SVM-Radial	Random Forest
41	32.8 (0 – 6.5)	30.9 (0 – 7.0)	23.6 (4.2 – 22.3)	30.9
8	29.1 (17.7 - 42.9)	29.1 (19.2 - 33.0)	20 (10.4 – 33.0)	29.1

Table 6.5: Cross-validation errors of training dataset

10 fold cross-validation was calculated for each of the classification algorithms and for both the 41 peptide and 8 peptide datasets as in table 6.4 above. Bootstrapped 95% confidence intervals and out-of bag (OOB) errors for the random forest are shown in parentheses.

range of ages were included; the youngest being 8 months, and the oldest 127 months (10.5 years). 11 children were under 1 year of age (5 TB cases and 6 controls), and 2 children were over the age of 5 (1 TB case and 1 control). The measured weight age Z-scores (WAZ) showed that the children in this study were underweight compared to aged matched population controls – overall mean WAZ -1.59 (i.e. 1.59 standard deviations below the population mean). Although the mean WAZ was numerically lower for TB cases (-1.60) compared to controls (-1.38), this difference was not statistically significant ($p = 0.65$). More males were recruited to the study (69%) than females, although the relative proportions of each gender were not significantly different between the TB cases and controls.

All subjects underwent Mantoux tuberculin skin testing (TST) and had blood drawn for IGRA at study screening. 5 patients from the TB case group had no measurable TST induration. Of the remaining 15 cases the median induration diameter was 18mm (IQR 15mm to 20 mm). The induration diameter was greater than or equal to 10mm for all of the responding cases. Baseline IGRAs were positive in 17 (89%) of the TB cases. Of the two negative cases at baseline, one had become positive at the 3 month follow up IGRA and the other remained negative. Three of the TB cases were lost to follow up at the 3 month point so no IGRA

	TB (PTB ± EPTB) n = 19	Control n = 17	p-value *
Age in months (Median, IQR)	16 (11.5 - 30)	15 (11 - 27)	0.62
Weight (kg) (Median, IQR)	8.75 (8.1 - 13.7)	9.0 (8.0 - 11.7)	0.88
Weight Age Z-Score (WAZ) (Mean, SD)	-1.60 (1.37)	-1.38 (1.53)	0.65
Gender			
females	7 (37%)	4 (23.5%)	0.61
males	12 (63%)	13 (76.5%)	
Previous TB treatment			
yes	1 (5%)	0 (0%)	1
no	28 (95%)	17 (100%)	
Household TB contact			
yes	11 (58%)	12 (71%)	0.66
no	8 (42%)	5 (29%)	
TST induration (mm) (Median, IQR)	18 (15 - 20)**	0 (0 - 0)	<0.001
Baseline IGRA			
positive	17 (89%)	0 (0%)	<0.001
negative	2 (11%)	17 (100%)	
3 month IGRA			
positive	12 (63%)	0 (0%)	<0.001
negative	3 (16%)	17 (100%)	
indeterminate	1 (5%)	0 (0%)	
unknown	3 (16%)	0 (0%)	

Table 6.6: Summary descriptive statistics for the test set

Demographic, TB exposure and clinical immunology data is shown for the 36 analysed training cases categorised by TB status.

* *p* values determined by Wilcoxon Rank Test, Z test or Chi Squared Test as appropriate

** 1 non reactive TST case not included

result was available (Baseline IGRA was positive in these three cases). Two cases had a negative 3 month IGRA following a positive baseline test, and one case had an intermediate result also following a positive baseline.

6.3.5 Comparison of training and test sets

Summary descriptive statistics for the training and test sets are shown in table 5.4. The most notable difference between the sets is in the age of the study participants. Overall the median age for the training set is 8.5 months greater than for the test set, $p = 0.04$. This difference is principally seen in the TB cases (training set 23 months older than test, $p = 0.02$), whereas the control cases show no significant age difference (18 months for the training set and 15 months in the test set, $p = 0.51$). Within the training TB cases there were 14 participants (50%) with co-existing extrapulmonary disease (EPTB). The test set by comparison had fewer cases 6 (31%). No significant differences are seen for other clinical or demographic parameters.

6.3.6 Model validation using the test set

Using the 41 peptide ‘signature’ previously selected, the trained SVM classifier was applied to the test set. Validation accuracy was low at 58.3% (40.8 – 74.5). Test sensitivity was 42.1% with a specificity of 76.5%. Positive and negative predictive values were 66.7% and 54.2% respectively. The test accuracy is considerably lower than the estimated accuracy from cross-validation the training set (estimated accuracy 76.4% by cross-validation). Interestingly, the cross-validation accuracy of the SVM classifier is not significantly better than using the most highly discriminatory peptide - SAYDLGTVIGFYGLT (Rv2088) selected from the ROC filtering of the training set - AUROC (test set) 0.68 (0.494 - 0.875).

6.3.7 Post Hoc Analyses

Given the poor test set classification accuracy, despite reasonable cross-validation on the training set, the test set was subjected to filtering and cross-validation to

examine its internal consistency. After Z-score filtering 418 peptides (12% of all array features) were retained (Figure 6.4). Following secondary filtering using the ROC criteria applied to the training set, a final ‘signature’ of 40 peptides was obtained (Figure 6.5).

6.4 Discussion

The analysis presented in this chapter demonstrates the use of a peptide microarray for the diagnosis of TB – a disease that is known to produce a highly variable serological response (Lyashchenko et al., 1998). To our knowledge, this is the first time that the diagnostic utility of peptide arrays have been tested in a paediatric TB population. The study design was conducted using a training and test dataset. These were drawn from the patients recruited to the study and selected randomly. In order to maintain an unseen test set, samples for training were processed first with the disease status blinded until analysis. The test set samples were processed after the training set. The data for the two sets was pre-processed separately but the same analysis methodology was used.

Feature selection was carried out prior to classification in order to reduce the dataset size, to reduce noise and to give a potentially informative set of peptides that could be studied further to identify potential linear epitopes or potentially useful diagnostic mimotope probes. The assumption made by the Z-score filtering algorithm is that in TB infection there will be an increase in the antibody binding (measured by feature signal intensity) in one or more sera for any given peptide probe compared to the mean intensity of that feature in the control sera. This differs from other microarray studies that look at two-way changes in the signal intensity. The peptide array study by Gaseitsiwe et al. (2008) looked at both where the feature signal intensity was increased and decreased in cases compared to controls. This is analogous to a gene expression microarray characterising up-regulation and down-regulation. However, in the context of TB infection, a high control signal intensity compared to cases might indicate a deficient immune response induced by TB, or perhaps more likely, that these responses were being driven by the other infections found in the control subjects.

Although 796 peptides were selected by the filtering algorithm, the majority

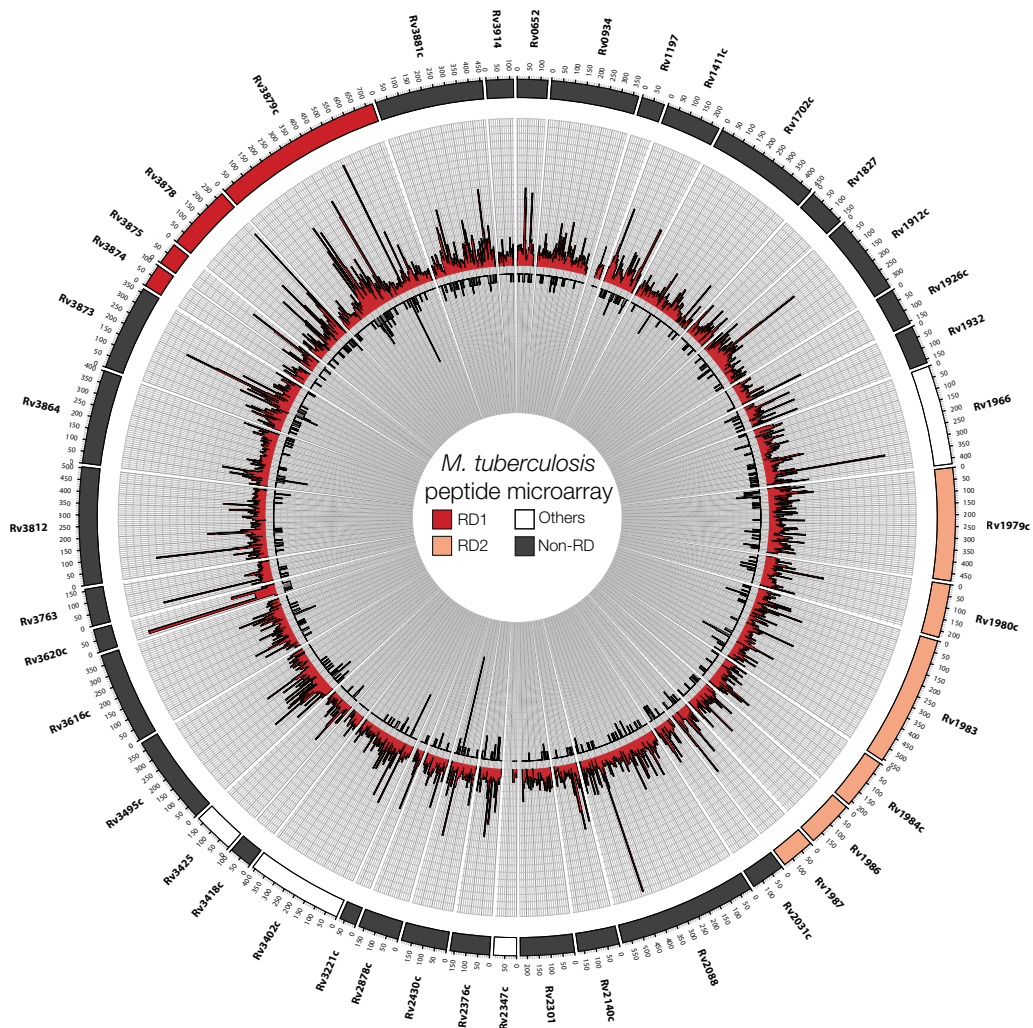


Figure 6.4: MTB peptide reactivity measured by Z-score (test set)

The arrayed MTB proteins are arranged on the above plot as a circular track reading clockwise from the N-terminal to the C-terminal of the primary sequence. The proteins corresponding to the RD1 and RD2 regions are coloured on the outer track in red and salmon respectively. Non-RD proteins are shaded in dark grey. The red coloured radial bar plot shows the Z-score for the peptide sequence corresponding to the protein and start position on the outer track. The inner black radial histogram shows the number of active TB patients with a peptide significant peptide Z-score (corresponding to a p -value of <0.01 after adjusting for multiple testing).

were identified only in single serum samples. This is a similar finding to the comprehensive protein microarray study by Kunnath-Velayudhan et al. (2010). They showed that sera from TB cases identified up to 10% of the bacterial proteome (approximately 500 proteins). However, this so-called ‘immunoproteome’ identified by individual patients varied enormously and the majority of proteins were selected by only in single samples. Although, the TB response can be highly variable, because each peptide probe can potentially be a mimotope for other non-TB antibodies, it is not possible to definitively say that these 796 peptides form a peptide immunome for TB. By calculating the AUROC for each of these peptides we were able to identify which peptides individually were useful at discriminating the TB cases from the controls. Of the 41 peptides identified from the ROC analysis only one, MAEMKTDAATLAQEA from the N-terminus of the CFP-10 antigen (Rv3874) originated from a protein that had been previously shown to be of use in the diagnosis of paediatric TB.

Previous studies have shown that serological tests based around single antigens or combinations of a few antigens perform poorly principally because of this highly variable response (Steingart et al., 2007b,a, 2011). Although protein antigens are able to identify antibody binding to native conformational targets, this relies on the accurate synthesis and folding of recombinant proteins, and the maintenance of their conformation on the microarray platform. This is inherently difficult (and expensive) to accomplish. Peptides, however, do not rely on an intrinsic conformation and can be cheaply produced in large volume by chemical synthesis. However, the disadvantage of using peptides as target molecules, is that they will only be recognised where a linear epitope is presented (approximately 10% of epitopes), or where the peptide acts as a mimotope – i.e. it mimics the conformational binding site on a protein. In this study we have taken the approach of looking primarily for linear epitopes by using overlapping peptide sequences from the primary structure of selected TB proteins. The disadvantage of using a linear epitope classifier is the potential sparsity of responses especially given the variability of protein specific responses. Although the array peptides may have a cognate protein origin, they will inevitably have a mimotopic function too and are likely to react to other serum antibodies. These could be TB specific antibodies directed against other proteins or could be other antibodies not related to the TB immune response.

Chapter 7

Conclusions

In this thesis I have discussed fundamental aspects of the analysis of spotted peptide microarray immunoassay data. The work presented here proposes an optimised methodology for this analysis based on objective comparisons of established methods of background correction and normalisation. This methodology is then applied, firstly to characterise disease severity and recurrence risk in *Clostridium difficile* infection, and secondly as a diagnostic tool for paediatric tuberculosis.

The `pmpa` software which was developed as part of this thesis is an integrated package for the importing of scanned raw array data and for the low level pre-processing of that data to minimise non-biological or technical variation that might obscure the true biological signal of interest. The package integrates several existing methods for background correction and normalisation allowing the user to compare and evaluate alternative methods. A novel aspect of the `pmpa` package is the integration of several QA methods for identifying problems that may have occurred during microarray processing. For example, spatial artefacts can be identified by image plots (microarray images reconstructed from the background or foreground intensity data), or by print-tip boxplots. Similarly, features with significantly higher or lower intensity values than their intra-array replicates can be identified by the subarray scatter plots. `pmpa` is an open source project and remains in active development. Using the modular framework of the highly regarded Bioconductor project allows integration of the data generated by `pmpa` with a large number of existing analysis packages. Currently work is under

way to expand the scope of the peptide array formats that can be used with `pmpa` including support for non-spotted arrays with variable numbers of features in each block.

Previous attempts to evaluate pre-processing methods for peptide array data have been limited by a lack of objective models comparing observed results to expected values. The approach presented in this thesis, using a spike-in antibody raised against a known continuous epitope, attempts to address this. This method generates a signal that should be unambiguously present on arrays incubated with the spike-in antibody with the signal intensity varying in a predictable manner with the antibody concentration. In conjunction with self-self array comparisons and assessment of signal intensity from a monoclonal antibody titration series, a comprehensive picture of array performance can be generated. Two key procedures in microarray pre-processing were assessed: background correction and normalisation. These procedures are mathematical transformations that aim to remove or compensate for technical or non-biological variation in the observed signal. The aim of preprocessing is to produce a set of signal intensities that reflect true biological variation whilst minimising any potential technical biases.

Assessment of background correction has been undertaken for cDNA and oligonucleotide arrays (Ritchie et al., 2007) but the work presented in this thesis is the first time that such an analysis has been undertaken for peptide array data. The observed microarray signal was found vary linearly with the measured background intensity implying an additive relationship. Moreover, traditional subtractive correction was shown to lead to significant variance inflation of low intensity signals. A simple additive offset combined with subtractive correction was sufficient to increase the precision of the data without obscuring the biological signal. This finding was similar to observations by Ritchie et al. (2007) where higher offset methods were found to be more favourable. Assessment of normalisation was performed *after* background correction by offset subtraction. No method showed significantly better performance at improving precision over no normalisation at all. However, quantile normalisation was found to improve discrimination of the spike-in signal at the lowest antibody concentration over all other methods. Although quantile normalisation controls for non biological variation by forcing the signal intensity distribution and median to be identical between arrays, it is still possible for biases to remain. A processing batch effect

was found that confounded the comparison of serological reactivity patterns between patients with *Clostridium difficile* infection and matched disease free controls. Similar batch effects have been documented in genomic and transcriptomic array experiments even following normalisation (Leek et al., 2010) but have not previously been documented in published peptide array studies.

In applying the proposed methodology to experimental and clinical datasets, I have presented in this thesis two analyses: one from patients and matched controls with *Clostridium difficile* infection, and another from a cohort of children with tuberculosis. The former study demonstrated that no antibody reactivity signature can be found using this peptide array that differentiates the CDI group as a whole from a non-CDI comparator group once batch effects were accounted for. However, a pattern of peptide reactivity was found that differentiated patients with recurrent disease from those with single episode CDI. This finding suggests that if this reactivity pattern were confirmed, it could be used to identify which patients are likely to benefit from additional treatment to help prevent relapse such as immunotherapy with pooled immunoglobulin or monoclonal antibodies.

The analysis of reactivity patterns characterising paediatric tuberculosis highlights a limitation of the high throughput immunoassay approach - namely that discovering consistent signal patterns in highly variable immune responses may be extremely difficult particularly with a limited sample size. In part this is due to the array design looking for linear epitopes which are in the minority compared to conformational epitopes. However to a certain extent the biological context contributes to this too - *M. tuberculosis* is known to generate a very variable humoral immune response. Hence it is impossible to know just from this analysis if the antibody recognising a given peptide has a cognate protein origin, or is responding to a mimotopic peptide. An important future step will be to try to verify these peptides in another low-throughput platform (eg. ELISA) and characterise the reactivity in more detail to determine if they represent genuine epitopes.

In conclusion, this thesis describes the development of a software tool for the analysis of peptide microarray data and the use of that tool for optimising preprocessing methodology and for 'real world' analysis of infectious diseases datasets. This work demonstrates the software's flexibility in assessing multiple

pre-processing methods in parallel, ability to integrate with existing statistical and machine learning packages, and utility in identifying signatures that characterise infectious diseases. Hopefully, the work of this thesis and the provision of the software as an open source package will help advance our understanding of peptide array immunoassay technology and the challenges it presents.

References

Achkar, J. M. and Ziegenbalg, A. (2012). Antibody responses to mycobacterial antigens in children with tuberculosis: Challenges and potential diagnostic value. *Clinical and Vaccine Immunology*, 19(12):1898–1906.

af Geijersstam, V., Kibur, M., Wang, Z., Koskela, P., Pukkala, E., Schiller, J., Lehtinen, M., and Dillner, J. (1998). Stability over time of serum antibody levels to human papillomavirus type 16. *Journal of Infectious Diseases*, 177(6):1710–1714.

Allo, M., Silva, J., Fekety, R., Rifkin, G. D., and Waskin, H. (1979). Prevention of clindamycin-induced colitis in hamsters by *Clostridium sordellii* antitoxin. *Gastroenterology*, 76(2):351–355.

Ambati, A., Valentini, D., Montomoli, E., Lapini, G., Biuso, F., Wenschuh, H., Magalhaes, I., and Maeurer, M. (2015). H1N1 viral proteome peptide microarray predicts individuals at risk for H1N1 infection and segregates infection versus Pandemrix vaccination. *Immunology*, 145(3):357–366.

Apweiler, R., Bateman, A., Martin, M. J., O'Donovan, C., Magrane, M., Alam-Faruque, Y., Alpi, E., Antunes, R., Arganiska, J., Casanova, E. B., Bely, B., Bingley, M., Bonilla, C., Britto, R., Bursteinas, B., Chan, W. M., Chavali, G., Cibrian-Uhalte, E., Da Silva, A., De Giorgi, M., Fazzini, F., Gane, P., Castro, L. G., Garmiri, P., Hatton-Ellis, E., Hieta, R., Huntley, R., Legge, D., Liu, W., Luo, J., MacDougall, A., Mutowo, P., Nightingale, A., Orchard, S., Pichler, K., Poggioli, D., Pundir, S., Pureza, L., Qi, G., Rosanoff, S., Sawford, T., Shypitsyna, A., Turner, E., Volynkin, V., Wardell, T., Watkins, X., Corbett, M., Donnelly, M., Van Rensburg, P., Goujon, M., McWilliam, H., Lopez, R., Xenarios, I., Bougueleret, L., Bridge, A., Poux, S., Redaschi, N., Aimo, L., Auchincloss, A., Axelsen, K., Bansal, P., Baratin, D., Binz, P. A., Blatter, M. C., Boeckmann, B., Bolleman, J., Boutet,

E., Breuza, L., Casal-Casas, C., De Castro, E., Cerutti, L., Coudert, E., Cuche, B., Doche, M., Dornevil, D., Duvaud, S., Estreicher, A., Famiglietti, L., Feuermann, M., Gasteiger, E., Gehant, S., Gerritsen, V., Gos, A., Gruaz-Gumowski, N., Hinz, U., Hulo, C., James, J., Jungo, F., Keller, G., Lara, V., Lemercier, P., Lew, J., Lieberherr, D., Lombardot, T., Martin, X., Masson, P., Morgat, A., Neto, T., Paesano, S., Pedruzzi, I., Pilbout, S., Pozzato, M., Pruess, M., Rivoire, C., Roechert, B., Schneider, M., Sigrist, C., Sonesson, K., Staehli, S., Stutz, A., Sundaram, S., Tognolli, M., Verbregue, L., Veuthey, A. L., Wu, C. H., Arighi, C. N., Arminski, L., Chen, C., Chen, Y., Garavelli, J. S., Huang, H., Laiho, K., McGarvey, P., Natale, D. A., Suzek, B. E., Vinayaka, C. R., Wang, Q., Wang, Y., Yeh, L. S., Yerramalla, M. S., and Zhang, J. (2014). Activities at the Universal Protein Resource (UniProt). *Nucleic Acids Research*, 42(D1):D191–D198.

Babcock, G. J., Broering, T. J., Hernandez, H. J., Mandell, R. B., Donahue, K., Boatright, N., Stack, A. M., Lowy, I., Graziano, R., Molrine, D., Ambrosino, D. M., and Thomas, W. D. (2006). Human monoclonal antibodies directed against toxins A and B prevent *Clostridium difficile*-induced mortality in hamsters. *Infection and Immunity*, 74(11):6339–6347.

Bacon, A. E. and Fekety, R. (1994). Immunoglobulin G directed against toxins A and B of *Clostridium difficile* in the general population and patients with antibiotic-associated diarrhea. *Diagnostic Microbiology and Infectious Disease*, 18(4):205–209.

Bahr, G. M., Rook, G. A., Al-Saffar, M., Van Embden, J., Stanford, J. L., and Behbehani, K. (1988). Antibody levels to mycobacteria in relation to HLA type: evidence for non-HLA-linked high levels of antibody to the 65 kD heat shock protein of *M. bovis* in rheumatoid arthritis. *Clinical and Experimental Immunology*, 74(2):211–215.

Baker, M. (2005). In biomarkers we trust? *Nature Biotechnology*, 23(3):297–304.

Bardelli, M., Livoti, E., Simonelli, L., Pedotti, M., Moraes, A., Valente, A. P., and Varani, L. (2015). Epitope mapping by solution NMR spectroscopy. *Journal of Molecular Recognition*, 28(6):393–400.

- Barlow, D. J., Edwards, M. S., and Thornton, J. M. (1986). Continuous and discontinuous protein antigenic determinants. *Nature*, 322(6081):747–8.
- Barrera, L., Miceli, I., Ritacco, V., Torrea, G., Broglia, B., Botta, R., Maldonado, C. P., Ferrero, N., Pinasco, A., and Cutillo, I. (1989). Detection of circulating antibodies to purified protein derivative by enzyme-linked immunosorbent assay: its potential for the rapid diagnosis of tuberculosis. *The Pediatric Infectious Disease Journal*, 8(11):763–767.
- Bartlett, J. G. (2006). Narrative review: the new epidemic of *Clostridium difficile*-associated enteric disease. *Annals of Internal Medicine*, 145(1539-3704 (Electronic)):758–764.
- Bartlett, J. G., Moon, N., Chang, T. W., Taylor, N., and Onderdonk, A. B. (1978). Role of *Clostridium difficile* in antibiotic-associated pseudomembranous colitis. *Gastroenterology*, 75(5):778–782.
- Beales, I. L. P. (2002). Intravenous immunoglobulin for recurrent *Clostridium difficile* diarrhoea. *Gut*, 51(3):456.
- Bengtsson, H., Jönsson, G., and Vallon-Christersson, J. (2004). Calibration and assessment of channel-specific biases in microarray data with extended dynamical range. *BMC Bioinformatics*, 5:177.
- Benjamin, D. C. and Perdue, S. S. (1996). Site-Directed Mutagenesis in Epitope Mapping. *Methods*, 9(3):508–515.
- Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4):1165–1188.
- Berek, C. and Milstein, C. (1988). The Dynamic Nature of the Antibody Repertoire. *Immunological Reviews*, 105(1):5–26.
- Beyer, M., Nesterov, A., Block, I., König, K., Felgenhauer, T., Fernandez, S., Leibe, K., Torralba, G., Hausmann, M., Trunk, U., Lindenstruth, V., Bischoff, F. R., Stadler, V., and Breitling, F. (2007). Combinatorial synthesis of peptide arrays onto a microchip. *Science*, 318(5858):1888.

Blythe, M. J. and Flower, D. R. (2009). Benchmarking B cell epitope prediction: Underperformance of existing methods. *Protein Science*, 14(1):246–248.

Bolstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193.

Borriello, S. P., Welch, A. R., Barclay, F. E., and Davies, H. A. (1988). Mucosal association by *Clostridium difficile* in the hamster gastrointestinal tract. *Journal of Medical Microbiology*, 25(3):191–196.

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., and Vingron, M. (2001). Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nature Genetics*, 29(4):365–71.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Breitling, F., Felgenhauer, T., Nesterov, A., Lindenstruth, V., Stadler, V., and Bischoff, F. R. (2009). Particle-based synthesis of peptide arrays. *ChemBioChem*, 10(5):803–808.

Bublil, E. M., Freund, N. T., Mayrose, I., Penn, O., Roitburd-Berman, A., Rubinstein, N. D., Pupko, T., and Gershoni, J. M. (2007). Stepwise prediction of conformational discontinuous B-cell epitopes using the Mapitope algorithm. *Proteins*, 68(1):294–304.

Calabi, E., Calabi, F., Phillips, A. D., and Fairweather, N. F. (2002). Binding of *Clostridium difficile* surface layer proteins to gastrointestinal tissues. *Infection and Immunity*, 70(10):5770–5778.

Carter, G. P., Lyras, D., Allen, D. L., Mackin, K. E., Howarth, P. M., O'Connor, J. R., and Rood, J. I. (2007). Binary toxin production in *Clostridium difficile* is regulated by CdtR, a LytTR family response regulator. *Journal of Bacteriology*, 189(20):7290–301.

- Cerquetti, M., Molinari, A., Sebastianelli, A., Diociaiuti, M., Petruzzelli, R., Capo, C., and Mastrantonio, P. (2000). Characterization of surface layer proteins from different *Clostridium difficile* clinical isolates. *Microbial Pathogenesis*, 28(6):363–72.
- Chambers, J. M. (1999). *Programming with data: A guide to the S language*, volume 37.
- Chambers, J. M. (2001). *Classes and Methods in the S Language*.
- Chiappini, E., Bonsignori, F., Accetta, G., Boddi, V., Galli, L., Biggeri, A., and De Martino, M. (2012). Interferon-gamma release assays for the diagnosis of *Mycobacterium tuberculosis* infection in children: a literature review. *International journal of immunopathology and pharmacology*, 25(2):335–343.
- Cohn, M. and Langman, R. E. (1990). The Protection: The Unit of Humoral Immunity Selected by Evolution. *Immunological Reviews*, 115(1):7–147.
- Corthier, G., Muller, M. C., Wilkins, T. D., Lysterly, D., and L'Hardion, R. (1991). Protection against experimental pseudomembranous colitis in gnotobiotic mice by use of monoclonal antibodies against *Clostridium difficile* toxin A. *Infection and Immunity*, 59(3):1192–1195.
- Cruz, A. T. and Starke, J. R. (2010). Pediatric Tuberculosis. *Pediatrics in Review*, 31(1):13–26.
- de la Riva, L., Willing, S. E., Tate, E. W., and Fairweather, N. F. (2011). Roles of Cysteine Proteases Cwp84 and Cwp13 in Biogenesis of the Cell Wall of *Clostridium difficile*. *Journal of Bacteriology*, 193(13):3276–3285.
- de Larrea, C. F., de Waard, J. H., Giampietro, F., and Araujo, Z. (2006). The secretory immunoglobulin A response to *Mycobacterium tuberculosis* in a childhood population. *Revista da Sociedade Brasileira de Medicina Tropical*, 39(5):456–461.
- Delacourt, C., Gobin, J., Gaillard, J. L., de Blic, J., Veron, M., and Scheinmann, P. (1993). Value of ELISA using antigen 60 for the diagnosis of tuberculosis in children. *Chest*, 104(2):393–398.

- Drobniewski, F., Caws, M., Gibson, A., and Young, D. (2003). Modern laboratory diagnosis of tuberculosis. *The Lancet Infectious Diseases*, 3(3):141–147.
- Drudy, D., Calabi, E., Kyne, L., Sougioultzis, S., Kelly, E., Fairweather, N., and Kelly, C. P. (2004). Human antibody response to surface layer proteins in *Clostridium difficile* infection. *FEMS Immunology and Medical Microbiology*, 41(3):237–42.
- Dunning, M. J., Smith, M. L., Ritchie, M. E., and Tavaré, S. (2007). Beadarray: R classes and methods for Illumina bead-based data. *Bioinformatics*, 23(16):2183–2184.
- Edwards, D. (2003). Non-linear normalization and background correction in one-channel cDNA microarray studies. *Bioinformatics*, 19(7):825–833.
- Eisen, M. (1999). ScanAlyze.
- Fagan, R. P., Albesa-Jové, D., Qazi, O., Svergun, D. I., Brown, K. A., and Fairweather, N. F. (2009). Structural insights into the molecular organization of the S-layer from *Clostridium difficile*. *Molecular Microbiology*, 71(5):1308–1322.
- Fardin, P., Moretti, S., Biasotti, B., Ricciardi, A., Bonassi, S., and Varesio, L. (2007). Normalization of low-density microarray using external spike-in controls: analysis of macrophage cell lines expression profile. *BMC Genomics*, 8(1):17.
- Fodor, S. P., Read, J. L., Pirrung, M. C., Stryer, L., Lu, A. T., and Solas, D. (1991). Light-directed, spatially addressable parallel chemical synthesis. *Science*, 251(4995):767–773.
- Frank, R. (1992). Spot-synthesis: an easy technique for the positionally addressable, parallel chemical synthesis on a membrane support. *Tetrahedron*, 48(42):9217–9232.
- Frank, R., Heikens, W., Heisterberg-Moutsis, G., and Blöcker, H. (1983). A new general approach for the simultaneous chemical synthesis of large numbers of oligonucleotides: segmental solid supports. *Nucleic Acids Research*, 11(13):4365–4377.

Gallerano, D., Wollmann, E., Lupinek, C., Schlederer, T., Ebner, D., Harwanegg, C., Niespodziana, K., Schmetterer, K., Pickl, W., Puchhammer-Stöckl, E., Sibanda, E., and Valenta, R. (2015). HIV microarray for the mapping and characterization of HIV-specific antibody responses. *Lab on a Chip*, 15(6):1574–89.

Gaseitsiwe, S., Valentini, D., Mahdaviifar, S., Magalhaes, I., Hoft, D. F., Zerweck, J., Schutkowski, M., Andersson, J., Reilly, M., and Maeurer, M. J. (2008). Pattern Recognition in Pulmonary Tuberculosis Defined by High Content Peptide Microarray Chip Analysis Representing 61 Proteins from *M. tuberculosis*. *PLoS ONE*, 3(12):e3840.

Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. (2004). Affy - Analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–315.

Gentleman, R., Carey, V., Huber, W., and Hahne, F. (2013). Genefilter: Methods for Filtering Genes From Microarray Experiments.

Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y. H., and Zhang, J. (2004). Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80.

Geysen, H. M., Meloen, R. H., and Barteling, S. J. (1984). Use of peptide synthesis to probe viral antigens for epitopes to a resolution of a single amino acid. *Proceedings of the National Academy of Sciences of the United States of America*, 81(13):3998–4002.

Geysen, H. M., Rodda, S. J., and Mason, T. J. (1986). A priori delineation of a peptide which mimics a discontinuous antigenic determinant. *Molecular Immunology*, 23(7):709–15.

Ghantaji, S. S., Sail, K., Lairson, D. R., DuPont, H. L., and Garey, K. W. (2010). Economic healthcare costs of *Clostridium difficile* infection: A systematic review. *Journal of Hospital Infection*, 74(4):309–318.

- Glare, E. M., Divjak, M., Bailey, M. J., and Walters, E. H. (2002). beta-Actin and GAPDH housekeeping gene expression in asthmatic airways is variable and not suitable for normalising mRNA levels. *Thorax*, 57(9):765–70.
- Grenier, J., Pinto, L., Nair, D., Steingart, K., Dowdy, D., Ramsay, A., and Pai, M. (2012). Widespread use of serological tests for tuberculosis: data from 22 high-burden countries. *European Respiratory Journal*, 39(2):502–505.
- Gupta, S., Bhatia, R., and Datta, K. K. (1997). Serological diagnosis of childhood tuberculosis by estimation of mycobacterial antigen 60-specific immunoglobulins in the serum. *Tubercle and Lung Disease*, 78(1):21–27.
- Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422.
- Halperin, R. F., Stafford, P., Emery, J. S., Navalkar, K. A., and Johnston, S. A. (2012). GuiTope: an application for mapping random-sequence peptides to protein sequences. *BMC Bioinformatics*, 13(1):1.
- Hecker, M., Lorenz, P., Steinbeck, F., Hong, L., Riemekasten, G., Li, Y., Zettl, U. K., and Thiesen, H.-J. (2012). Computational analysis of high-density peptide microarray data with application from systemic sclerosis to multiple sclerosis. *Autoimmunity Reviews*, 11(3):180–90.
- Hennequin, C., Janoir, C., Barc, M.-C., Collignon, A., and Karjalainen, T. (2003). Identification and characterization of a fibronectin-binding protein from *Clostridium difficile*. *Microbiology*, 149(Pt 10):2779–2787.
- Hopp, T. P. and Woods, K. R. (1981). Prediction of protein antigenic determinants from amino acid sequences. *Proceedings of the National Academy of Sciences of the United States of America*, 78(6):3824–8.
- Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., Bravo, H. C., Davis, S., Gatto, L., Girke, T., Gottardo, R., Hahne, F., Hansen, K. D., Irizarry, R. A., Lawrence, M., Love, M. I., MacDonald, J., Obenchain, V., Oleś, A. K., Pagès, H., Reyes, A., Shannon, P., Smyth, G. K., Tenenbaum, D., Waldron, L., and Morgan, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, 12(2):115–121.

Huber, W., von Heydebreck, A., Sultmann, H., Poustka, A., and Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(Suppl 1):S96–S104.

Hueber, W., Tomooka, B. H., Zhao, X., Kidd, B. a., Drijfhout, J. W., Fries, J. F., van Venrooij, W. J., Metzger, A. L., Genovese, M. C., and Robinson, W. H. (2007). Proteomic analysis of secreted proteins in early rheumatoid arthritis: anti-citrulline autoreactivity is associated with up regulation of proinflammatory cytokines. *Annals of the Rheumatic Diseases*, 66(6):712–9.

Hughes, A. K., Cichacz, Z., Scheck, A., Coons, S. W., Johnston, S. A., and Stafford, P. (2012). Immunosignaturing Can Detect Products from Molecular Markers in Brain Cancer. *PLoS ONE*, 7(7):e40201.

Huygen, K., Ljunqvist, L., Ten Berg, R., and Van Vooren, J. P. (1990). Repertoires of antibodies to culture filtrate antigens in different mouse strains infected with *Mycobacterium bovis* BCG. *Infection and Immunity*, 58(7):2192–2197.

Ihaka, R. and Gentleman, R. (2012). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*.

Imholte, G. C., Sauteraud, R., Korber, B., Bailer, R. T., Turk, E. T., Shen, X., Tomaras, G. D., Mascola, J. R., Koup, R. A., Montefiori, D. C., and Gottardo, R. (2013). A computational framework for the analysis of peptide microarray antibody binding data with application to HIV vaccine profiling. *Journal of Immunological Methods*, 395(1-2):1–13.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264.

Jank, T., Giesemann, T., and Aktories, K. (2007). Rho-glucosylating *Clostridium difficile* toxins A and B: new insights into structure and function. *Glycobiology*, 17(4):15R–22.

Jin, L., Fendly, B. M., and Wells, J. A. (1992). High resolution functional analysis of antibody-antigen interactions. *Journal of Molecular Biology*, 226(3):851–865.

Johnson, W. E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127.

Kairdolf, B. a., Smith, A. M., Stokes, T. H., Wang, M. D., Young, A. N., and Nie, S. (2013). Semiconductor quantum dots for bioimaging and biodiagnostic applications. *Annual Review of Analytical Chemistry*, 6:143–162.

Kamboj, M., Khosa, P., Kaltsas, A., Babady, N. E., Son, C., and Sepkowitz, K. A. (2011). Relapse versus reinfection: Surveillance of clostridium difficile infection. *Clinical Infectious Diseases*, 53(10):1003–1006.

Karasavvas, N., Billings, E., Rao, M., Williams, C., Zolla-Pazner, S., Bailer, R. T., Koup, R. A., Madnote, S., Arworn, D., Shen, X., Tomaras, G. D., Currier, J. R., Jiang, M., Magaret, C., Andrews, C., Gottardo, R., Gilbert, P., Cardozo, T. J., Rerks-Ngarm, S., Nitayaphan, S., Pitisuttithum, P., Kaewkungwal, J., Paris, R., Greene, K., Gao, H., Gurunathan, S., Tartaglia, J., Sinangil, F., Korber, B. T., Montefiori, D. C., Mascola, J. R., Robb, M. L., Haynes, B. F., Ngauy, V., Michael, N. L., Kim, J. H., and de Souza, M. S. (2012). The Thai Phase III HIV Type 1 Vaccine trial (RV144) regimen induces antibodies that target conserved regions within the V2 loop of gp120. *AIDS Research and Human Retroviruses*, 28(11):1444–57.

Katz, C., Levy-Beladev, L., Rotem-Bamberger, S., Rito, T., Rüdiger, S. G. D., and Friedler, A. (2011). Studying protein–protein interactions using peptide arrays. *Chemical Society Reviews*, 40(5):2131.

Kauffmann, A., Gentleman, R., and Huber, W. (2009). arrayQualityMetrics - A bioconductor package for quality assessment of microarray data. *Bioinformatics*, 25(3):415–416.

Kirby, J. M., Ahern, H., Roberts, A. K., Kumar, V., Freeman, Z., Acharya, K. R., and Shone, C. C. (2009). Cwp84, a surface-associated cysteine protease, plays a role in the maturation of the surface layer of *Clostridium difficile*. *The Journal of Biological Chemistry*, 284(50):34666–34673.

Kringelum, J. V., Nielsen, M., Padkjær, S. B., and Lund, O. (2013). Structural analysis of B-cell epitopes in antibody:protein complexes. *Molecular Immunology*, 53(1-2):24–34.

Kunnath-Velayudhan, S., Salamon, H., Wang, H.-Y., Davidow, A. L., Molina, D. M., Huynh, V. T., Cirillo, D. M., Michel, G., Talbot, E. A., Perkins, M. D., Felgner, P. L., Liang, X., and Gennaro, M. L. (2010). Dynamic antibody responses to the Mycobacterium tuberculosis proteome. *Proceedings of the National Academy of Sciences*, 107(33):14703–14708.

Kyne, L., Warny, M., Qamar, A., and Kelly, C. P. (2000). Asymptomatic carriage of Clostridium difficile and serum levels of {IgG} antibody against toxin A. *The New England Journal of Medicine*, 342(6):390–397.

Kyne, L., Warny, M., Qamar, A., and Kelly, C. P. (2001). Association between antibody response to toxin A and protection against recurrent Clostridium difficile diarrhoea. *Lancet*, 357(9251):189–93.

Lagatie, O., Van Loy, T., Tritsmans, L., and Stuyver, L. J. (2014). Antibodies reacting with JCPyV_VP2 _167-15mer as a novel serological marker for JC polyomavirus infection. *Virology Journal*, 11:174.

Lam, K. S., Salmon, S. E., Hersh, E. M., Hruby, V. J., Kazmierski, W. M., and Knapp, R. J. (1991). A new type of synthetic peptide library for identifying ligand-binding activity. *Nature*, 354(6348):82–4.

Leav, B. A., Blair, B., Leney, M., Knauber, M., Reilly, C., Lowy, I., Gerding, D. N., Kelly, C. P., Katchar, K., Baxter, R., Ambrosino, D., and Molrine, D. (2010). Serum anti-toxin B antibody correlates with protection from recurrent Clostridium difficile infection (CDI). *Vaccine*, 28(4):965–969.

Leek, J. T., Scharpf, R. B., Bravo, H. C., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K., and Irizarry, R. A. (2010). Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–9.

Leung, D. Y., Kelly, C. P., Boguniewicz, M., Pothoulakis, C., LaMont, J. T., and Flores, A. (1991). Treatment with intravenously administered gamma globulin of chronic relapsing colitis induced by Clostridium difficile toxin. *The Journal of Pediatrics*, 118(4 Pt 1):633–637.

Lewis, S. J. and Heaton, K. W. (1997). Stool form scale as a useful guide to intestinal transit time. *Scandinavian Journal of Gastroenterology*, 32(9):920–924.

Lin, J., Bruni, F. M., Fu, Z., Maloney, J., Bardina, L., Boner, A. L., Gimenez, G., and Sampson, H. A. (2012). A bioinformatics approach to identify patients with symptomatic peanut allergy using peptide microarray immunoassay. *Journal of Allergy and Clinical Immunology*, 129(5):1321–1328.e5.

List, C., Qi, W., Maag, E., Gottstein, B., Müller, N., and Felger, I. (2010). Serodiagnosis of Echinococcus spp. infection: Explorative selection of diagnostic antigens by peptide microarray. *PLoS Neglected Tropical Diseases*, 4(8):e771.

Lowy, I., Molrine, D. D. C. D., Leav, B. B. A., Blair, B. M., Baxter, R., Gerding, D. N., Nichol, G., Thomas, W. D., Leney, M., Sloan, S., Hay, C. A., and Ambrosino, D. M. (2010). Treatment with monoclonal antibodies against Clostridium difficile toxins. *The New England Journal of Medicine*, 362(3):197–205.

Lyashchenko, K. P., Pollock, J. M., Colangeli, R., and Gennaro, M. L. (1998). Diversity of Antigen Recognition by Serum Antibodies in Experimental Bovine Tuberculosis. *Infection and Immunity*, 66(11):5344.

Maksimov, P., Zerweck, J., Dubey, J. P., Pantchev, N., Frey, C. F., Maksimov, A., Reimer, U., Schutkowski, M., Hosseini, M., Ziller, M., Conraths, F. J., and Schares, G. (2013). Serotyping of Toxoplasma gondii in Cats (Felis domesticus) Reveals Predominance of Type II Infections in Germany. *PLoS ONE*, 8(11):e80213.

Maksimov, P., Zerweck, J., Maksimov, A., Hotop, A., Groß, U., Pleyer, U., Spekker, K., Däubener, W., Werdermann, S., Niederstrasser, O., Petri, E., Mertens, M., Ulrich, R. G., Conraths, F. J., and Schares, G. (2012). Peptide Microarray Analysis of In Silico-Predicted Epitopes for Serological Diagnosis of Toxoplasma gondii Infection in Humans. *Clinical and Vaccine Immunology*, 19(6):865–874.

Marais, B. J., Hesselting, A. C., Gie, R. P., Schaaf, H. S., and Beyers, N. (2006). The burden of childhood tuberculosis and the accuracy of community-based surveillance data. *International Journal of Tuberculosis and Lung Disease*, 10(3):259–263.

Mayrose, I., Shlomi, T., Rubinstein, N. D., Gershoni, J. M., Ruppin, E., Sharan,

- R., and Pupko, T. (2007). Epitope mapping using combinatorial phage-display libraries: a graph-based algorithm. *Nucleic Acids Research*, 35(1):69–78.
- McCafferty, J., Griffiths, A. D., Winter, G., and Chiswell, D. J. (1990). Phage antibodies: filamentous phage displaying antibody variable domains. *Nature*, 348(6301):552–4.
- McDonald, L. C., Killgore, G. E., Thompson, A., Owens, R. C., Kazakova, S. V., Sambol, S. P., Johnson, S., and Gerding, D. N. (2005). An epidemic, toxin gene-variant strain of *Clostridium difficile*. *The New England Journal of Medicine*, 353(23):2433–2441.
- McFarland, L. V., Surawicz, C. M., Rubin, M., Fekety, R., Elmer, G. W., and Greenberg, R. N. (1999). Recurrent *Clostridium difficile* disease: epidemiology and clinical characteristics. *Infection Control and Hospital Epidemiology: The Official Journal of the Society of Hospital Epidemiologists of America*, 20(1):43–50.
- McNerney, R. and Daley, P. (2011). Towards a point-of-care test for active tuberculosis: obstacles and opportunities. *Nature Reviews Microbiology*, 9(3):204–213.
- Merrifield, R. B. (1963). Solid Phase Peptide Synthesis. I. The Synthesis of a Tetrapeptide. *Journal of the American Chemical Society*, 85(14):2149–2154.
- Min, F., Zhang, Y., Huang, R., Li, W., Wu, Y., Pan, J., Zhao, W., and Liu, X. (2011). Serum antibody responses to 10 *Mycobacterium tuberculosis* proteins, purified protein derivative, and old tuberculin in natural and experimental tuberculosis in rhesus monkeys. *Clinical and vaccine immunology : CVI*, 18(12):2154–60.
- Monot, M., Boursaux-Eude, C., Thibonnier, M., Vallenet, D., Moszer, I., Medigue, C., Martin-Verstraete, I., and Dupuy, B. (2011). Reannotation of the genome sequence of *Clostridium difficile* strain 630. *Journal of Medical Microbiology*, 60(8):1193–1199.
- Na, X., Kim, H., Moyer, M. P., Pothoulakis, C., and LaMont, J. T. (2008). gp96 is a human colonocyte plasma membrane binding protein for *Clostridium difficile* toxin A. *Infection and Immunity*, 76(7):2862–2871.

Nahtman, T., Jernberg, A., Mahdavifar, S., Zerweck, J., Schutkowski, M., Maeurer, M., and Reilly, M. (2007). Validation of peptide epitope microarray experiments and extraction of quality data. *Journal of Immunological Methods*, 328(1-2):1–13.

Navalkar, K. A., Johnston, S. A., Woodbury, N., Galgiani, J. N., Magee, D. M., Chicacz, Z., and Stafford, P. (2014). Application of Immunosignatures for Diagnosis of Valley Fever. *Clinical and Vaccine Immunology*, 21(8):1169–1177.

Ngo, Y., Advani, R., Valentini, D., Gaseitsiwe, S., Mahdavifar, S., Maeurer, M., and Reilly, M. (2009). Identification and testing of control peptides for antigen microarrays. *Journal of Immunological Methods*, 343(2):68–78.

Nicol, M. P., Workman, L., Isaacs, W., Munro, J., Black, F., Eley, B., Boehme, C. C., Zemanay, W., and Zar, H. J. (2011). Accuracy of the Xpert MTB/RIF test for the diagnosis of pulmonary tuberculosis in children admitted to hospital in Cape Town, South Africa: a descriptive study. *The Lancet infectious diseases*, 11(11):819–824.

Nobrega, A., Grandien, A., Haury, M., Hecker, L., Malanchère, E., and Coutinho, A. (1998). Functional diversity and clonal frequencies of reactivity in the available antibody repertoire. *European Journal of Immunology*, 28(4):1204–1215.

Nusrat, A., von Eichel-Streiber, C., Turner, J. R., Verkade, P., Madara, J. L., and Parkos, C. A. (2001). Clostridium difficile toxins disrupt epithelial barrier function by altering membrane microdomain localization of tight junction proteins. *Infection and Immunity*, 69(3):1329–1336.

Olson, M. M., Shanholtzer, C. J., Lee, J. T. J., and Gerding, D. N. (1994). Ten years of prospective Clostridium difficile-associated disease surveillance and treatment at the Minneapolis VA Medical Center, 1982-1991. *Infection Control and Hospital Epidemiology: The Official Journal of the Society of Hospital Epidemiologists of America*, 15(6):371–381.

Parman, C., Halling, C., and Gentleman, R. (2005). affyQCReport: QC Report Generation for affyBatch objects. *R package version*.

Pechine, S. (2005). Immunological properties of surface proteins of Clostridium difficile. *Journal of Medical Microbiology*, 54(2):193–196.

Péchiné, S., Janoir, C., and Collignon, A. (2005). Variability of *Clostridium difficile* surface proteins and specific serum antibody response in patients with *Clostridium difficile*-associated disease. *Journal of Clinical Microbiology*, 43(10):5018–5025.

Perelle, S., Gibert, M., Bourlioux, P., Corthier, G., and Popoff, M. R. (1997). Production of a complete binary toxin (actin-specific {ADP-ribosyltransferase}) by *Clostridium difficile* {CD196}. *Infection and Immunity*, 65(4):1402–1407.

Perez-Gordo, M., Lin, J., Bardina, L., Pastor-Vargas, C., Cases, B., Vivanco, F., Cuesta-Herranz, J., and Sampson, H. A. (2011). Epitope mapping of Atlantic salmon major allergen by peptide microarray immunoassay. *International Archives of Allergy and Immunology*, 157(1):31–40.

Perkins, M. D. and Kritski, A. L. (2002). Diagnostic testing in the control of tuberculosis. *Bulletin of the World Health Organization*, 80(6):512.

Popoff, M. R., Rubin, E. J., Gill, D. M., and Boquet, P. (1988). Actin-specific ADP-ribosyltransferase produced by a *Clostridium difficile* strain. *Infection and Immunity*, 56(9):2299–2306.

Press, H., Press, H., Smyth, G. K., Smyth, G. K., Yang, Y. H., Yang, Y. H., Speed, T., Speed, T., Office, P., and Office, P. (2002). Statistical Issues in cDNA Microarray Data Analysis. *Methods in Molecular Biology*, 224:1–26.

Price, J. V., Jarrell, J. a., Furman, D., Kattah, N. H., Newell, E., Dekker, C. L., Davis, M. M., and Utz, P. J. (2013). Characterization of Influenza Vaccine Immunogenicity Using Influenza Antigen Microarrays. *PLoS ONE*, 8(5):e64555.

Quackenbush, J. (2002). Microarray data normalization and transformation. *Nature Genetics*, 32 Suppl:496–501.

R Development Core Team (2014). R: A Language and Environment for Statistical Computing.

Raghava, G. P. S. and Barton, G. J. (2006). Quantification of the variation in percentage identity for protein sequence alignments. *BMC Bioinformatics*, 7(1):415.

- Reilly, M. and Valentini, D. (2009). Visualisation and pre-processing of peptide microarray data. *Methods in Molecular Biology*, 570:373–389.
- Reineke, U., Volkmer-Engert, R., and Schneider-Mergener, J. (2001). Applications of peptide arrays prepared by the SPOT-technology. *Current Opinion in Biotechnology*, 12(1):59–64.
- Renard, B. Y., Lower, M., Kuhne, Y., Reimer, U., Rothermel, A. A., Tureci, O., Castle, J. C., Sahin, U., Löwer, M., Kühne, Y., and Türeci, Ö. (2011). rapmad: Robust Analysis of Peptide Microarray Data. *BMC Bioinformatics*, 12(1):324.
- Restrepo, L. (2013). Feasibility of an early Alzheimer's disease immunosignature diagnostic test. *Journal of Neuroimmunology*, 254(1-2):154–160.
- Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., and Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47–e47.
- Ritchie, M. E., Silver, J., Oshlack, A., Holmes, M., Diyagama, D., Holloway, A., and Smyth, G. K. (2007). A comparison of background correction methods for two-colour microarrays. *Bioinformatics*, 23(20):2700–2707.
- Rocke, D. M., Durbin, B., Wilson, M., and Kahn, H. D. (2003). Modeling uncertainty in the measurement of low-level analytes in environmental analysis. *Ecotoxicology and Environmental Safety*, 56(1):78–92.
- Rosen, E. U. (1990). The diagnostic value of an enzyme-linked immune sorbent assay using adsorbed mycobacterial sonicates in children. *Tubercle*, 71(2):127–130.
- Rosen, O. and Anglister, J. (2009). Epitope mapping of antibody-antigen complexes by nuclear magnetic resonance spectroscopy. *Epitope Mapping Protocols*, 524:37–57.
- Rubinstein, N. D., Mayrose, I., Martz, E., and Pupko, T. (2009). Epitopia: a web-server for predicting B-cell epitopes. *BMC Bioinformatics*, 10(1):287.
- Rupnik, M., Avesani, V., Janc, M., Von Eichel-Streiber, C., and Delmée, M. (1998). A novel toxinotyping scheme and correlation of toxinotypes with serogroups of *Clostridium difficile* isolates. *Journal of Clinical Microbiology*, 36(8):2240–2247.

Rupnik, M., Brazier, J. S., Duerden, B. I., Grabnar, M., and Stubbs, S. L. J. (2001). Comparison of toxinotyping and PCR ribotyping of *Clostridium difficile* strains and description of novel toxinotypes. *Microbiology*, 147(2):439–447.

Salcedo, J., Keates, S., Pothoulakis, C., Warny, M., Castagliuolo, I., LaMont, J. T., and Kelly, C. P. (1997). Intravenous immunoglobulin therapy for severe *Clostridium difficile* colitis. *Gut*, 41(3):366–370.

Sandberg, M., Eriksson, L., Jonsson, J., Sjöström, M., and Wold, S. (1998). New chemical descriptors relevant for the design of biologically active peptides. A multivariate characterization of 87 amino acids. *Journal of Medicinal Chemistry*, 41(14):2481–2491.

Saul, F. A. and Alzari, P. M. (1996). Crystallographic Studies of Antigen–Antibody Interactions. *Epitope Mapping Protocols*, 66:11–24.

Schutkowski, M., Zerweck, J., Masch, A., and Wenschuh, H. (2009). Antibody signatures defined by high-content peptide microarray analysis. *Nature Methods*, 6(3).

Sebahia, M., Wren, B. W., Mullany, P., Fairweather, N. F., Minton, N., Stabler, R., Thomson, N. R., Roberts, A. P., Cerdeño-Tárraga, A. M., Wang, H., Holden, M. T. G., Wright, A., Churcher, C., Quail, M. A., Baker, S., Bason, N., Brooks, K., Chillingworth, T., Cronin, A., Davis, P., Dowd, L., Fraser, A., Feltwell, T., Hance, Z., Holroyd, S., Jagels, K., Moule, S., Mungall, K., Price, C., Rabbinowitsch, E., Sharp, S., Simmonds, M., Stevens, K., Unwin, L., Whithead, S., Dupuy, B., Dougan, G., Barrell, B., and Parkhill, J. (2006). The multidrug-resistant human pathogen *Clostridium difficile* has a highly mobile, mosaic genome. *Nature Genetics*, 38(7):779–786.

Severance, E. G., Lin, J., Sampson, H. A., Gimenez, G., Dickerson, F. B., Halling, M., Gressitt, K., Haile, L., Stallings, C. R., Origoni, A. E., Dupont, D., and Yolken, R. H. (2011). Dietary antigens, epitope recognition, and immune complex formation in recent onset psychosis and long-term schizophrenia. *Schizophrenia research*, 126(1-3):43–50.

Shreffler, W. G., Beyer, K., Chu, T. H. T., Burks, A. W., and Sampson, H. A. (2004). Microarray immunoassay: Association of clinical history, in vitro IgE function,

and heterogeneity of allergenic peanut epitopes. *Journal of Allergy and Clinical Immunology*, 113(4):776–782.

Shreffler, W. G., Lencer, D. a., Bardina, L., and Sampson, H. a. (2005). IgE and IgG4 epitope mapping by microarray immunoassay reveals the diversity of immune response to the peanut allergen, Ara h 2. *Journal of Allergy and Clinical Immunology*, 116(4):893–899.

Silver, J. D., Ritchie, M. E., and Smyth, G. K. (2009). Microarray background correction: maximum likelihood estimation for the normal–exponential convolution. *Biostatistics*, 10(2):352–363.

Smith, G. P. (1985). Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface. *Science*, 228(4705):1315–1317.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.

Stabler, R. A., He, M., Dawson, L., Martin, M., Valiente, E., Corton, C., Lawley, T. D., Sebahia, M., Quail, M. A., Rose, G., Gerding, D. N., Gibert, M., Popoff, M. R., Parkhill, J., Dougan, G., and Wren, B. W. (2009). Comparative genome and phenotypic analysis of *Clostridium difficile* 027 strains provides insight into the evolution of a hypervirulent bacterium. *Genome Biology*, 10(9):R102.

Stafford, P., Cichacz, Z., Woodbury, N. W., and Johnston, S. A. (2014). Immunosignature system for diagnosis of cancer. *Proceedings of the National Academy of Sciences of the United States of America*, 111(30):E3072–80.

Stafford, P., Halperin, R., Legutki, J. B., Magee, D. M., Galgiani, J., and Johnston, S. a. (2012). Physical Characterization of the ”Immunosignaturing Effect”. *Molecular & Cellular Proteomics*, 11(4):M111.011593–M111.011593.

Steingart, K. R., Dendukuri, N., Henry, M., Schiller, I., Nahid, P., Hopewell, P. C., Ramsay, A., Pai, M., and Laal, S. (2009). Performance of purified antigens for serodiagnosis of pulmonary tuberculosis: A meta-analysis. *Clinical and Vaccine Immunology*, 16(2):260–276.

Steingart, K. R., Flores, L. L., Dendukuri, N., Schiller, I., Laal, S., Ramsay, A., Hopewell, P. C., and Pai, M. (2011). Commercial Serological tests for the diagnosis

of active pulmonary and extrapulmonary tuberculosis: An updated systematic review and Meta-Analysis. *PLoS Medicine*, 8(8):e1001062.

Steingart, K. R., Henry, M., Laal, S., Hopewell, P. C., Ramsay, A., Menzies, D., Cunningham, J., Weldingh, K., and Pai, M. (2007a). A systematic review of commercial serological antibody detection tests for the diagnosis of extrapulmonary tuberculosis. *Thorax*, 62(10):911–918.

Steingart, K. R., Henry, M., Laal, S., Hopewell, P. C., Ramsay, A., Menzies, D., Cunningham, J., Weldingh, K., and Pai, M. (2007b). Commercial serological antibody detection tests for the diagnosis of pulmonary tuberculosis: A systematic review. *PLoS Medicine*, 4(6):1041–1060.

Stephenson, K. E., Neubauer, G. H., Reimer, U., Pawlowski, N., Knaute, T., Zerweck, J., Korber, B. T., and Barouch, D. H. (2015). Quantification of the epitope diversity of HIV-1-specific binding antibodies by peptide microarrays for global HIV-1 vaccine development. *Journal of Immunological Methods*, 416:105–123.

Stubbe, H., Berdoz, J., Kraehenbuhl, J. P., and Corthésy, B. (2000). Polymeric IgA is superior to monomeric IgA and IgG carrying the same variable domain in preventing *Clostridium difficile* toxin A damaging of T84 monolayers. *Journal of Immunology*, 164(4):1952–1960.

Sweredoski, M. J. and Baldi, P. (2008). PEPITO: improved discontinuous B-cell epitope prediction using multiple distance thresholds and half sphere exposure. *Bioinformatics*, 24(12):1459–1460.

Sykes, K. F., Legutki, J. B., and Stafford, P. (2013). Immunosignaturing: A critical review. *Trends in Biotechnology*, 31(1):45–51.

Tasteyre, A., Barc, M. C., Collignon, A., Boureau, H., and Karjalainen, T. (2001). Role of FliC and FliD flagellar proteins of *Clostridium difficile* in adherence and gut colonization. *Infection and Immunity*, 69(12):7937–7940.

Thiele, A., Stangl, G., and Schutkowski, M. (2011). Deciphering Enzyme Function Using Peptide Arrays. *Molecular Biotechnology*, 49(3):283–305.

Tonegawa, S. (1983). Somatic generation of antibody diversity. *Nature*, 302(5909):575–581.

Tucker, K. D. and Wilkins, T. D. (1991). Toxin A of *Clostridium difficile* binds to the human carbohydrate antigens I, X, and Y. *Infection and Immunity*, 59(1):73–78.

Viscidi, R., Laughon, B. E., Yolken, R., Bo-Linn, P., Moench, T., Ryder, R. W., and Bartlett, J. G. (1983). Serum Antibody Response to Toxins A and B of *Clostridium difficile*. *The Journal of Infectious Diseases*, 148(1):93–100.

Waligora, A. J., Hennequin, C., Mullany, P., Bourlioux, P., Collignon, A., and Karjalainen, T. (2001). Characterization of a cell surface protein of *Clostridium difficile* with adhesive properties. *Infection and Immunity*, 69(4):2144–2153.

Warny, M., Pepin, J., Fang, A., Killgore, G., Thompson, A., Brazier, J., Frost, E., and McDonald, L. C. (2005). Toxin production by an emerging strain of *Clostridium difficile* associated with outbreaks of severe disease in North America and Europe. *Lancet*, 366(9491):1079–1084.

Wickham, H. (2011). The Split-Apply-Combine Strategy for Data Analysis. *Journal of Statistical Software*, 40(1):1–29.

Wilcox, M. H., Fawley, W. N., Wigglesworth, N., Parnell, P., Verity, P., and Freeman, J. (2003). Comparison of the effect of detergent versus hypochlorite cleaning on environmental contamination and incidence of *Clostridium difficile* infection. *Journal of Hospital Infection*, 54(2):109–114.

Wilson, D. S., Keefe, A. D., and Szostak, J. W. (2001). The use of mRNA display to select high-affinity protein-binding peptides. *Proceedings of the National Academy of Sciences of the United States of America*, 98(7):3750–5.

World Health Organization (2014). Global tuberculosis report 2014. Technical report.

World Health Organization (WHO) (2015). *World Health Statistics 2015*. World Health Organization, Geneva.

- Yang, Y., Stafford, P., and Kim, Y. (2011). Segmentation and intensity estimation for microarray images with saturated pixels. *BMC Bioinformatics*, 12:462.
- Yang, Y. H., Buckley, M. J., and Speed, T. P. (2001). Analysis of cDNA microarray images. *Briefings in Bioinformatics*, 2(4):341–349.
- Yang, Y. H., Yang Y.H., P. A., and S., D. (2009). marray: Exploratory analysis for two-color spotted microarray data. R package version 1.32.0.
- Yin, W., Chen, T., Zhou, S. X., and Chakraborty, A. (2005). Background correction for cDNA microarray images using the TV+L1 model. *Bioinformatics*, 21(10):2410–2416.
- Yu, X., Xu, D., and Cheng, Q. (2006). Label-free detection methods for protein microarrays. *Proteomics*, 6(20):5493–503.
- Zheng, Y. J., Wang, R. H., Lin, Y. Z., and Daniel, T. M. (1994). Clinical evaluation of the diagnostic value of measuring IgG antibody to 3 mycobacterial antigen preparations in the capillary blood of children with tuberculosis and control subjects. *Tubercle and Lung Disease*, 75(5):366–370.

Appendix 1: PMPA Source Code

accessor-methods.R

```
#' Foreground Intensity Accessor
#'  
#' Extracts the matrix of foreground intensities (fMedian)  
#' from a MultiSet object created by \link{readArrays}  
#'  
#' @param x MultiSet object  
#' @return matrix of foreground intensities  
#' @exportMethod fg  
#' @docType methods  
#' @rdname fg-methods  
setGeneric(  
  name = "fg",  
  def = function(x) standardGeneric("fg")  
)  
#' @rdname fg-methods  
#' @aliases fg  
setMethod(  
  f = "fg",  
  signature = "MultiSet",  
  definition = function(x){  
    assayDataElement(x, "fMedian")  
  }  
)  
#-----  
#' Background Intensity Accessor  
#'  
#' Extracts the matrix of background intensities (bg)  
#' from a MultiSet object created by \link{readArrays}  
#'  
#' @param x MultiSet object  
#' @return matrix of background intensities  
#' @exportMethod bg  
#' @docType methods  
#' @rdname bg-methods  
setGeneric(  
  name = "bg",  
  def = function(x) standardGeneric("bg")  
)  
#' @rdname bg-methods  
#' @aliases bg  
setMethod(  
  f = "bg",  
  signature = "MultiSet",  
  definition = function(x){  
    assayDataElement(x, "bg")  
  }  
)
```

```

#-----
#' Median Background Intensity Accessor
#'
#' Extracts the matrix of median background intensities (bMedian)
#' from a MultiSet object created by \link{readArrays}
#'
#' @param x MultiSet object
#' @return matrix of median background intensities
#' @exportMethod bmedian
#' @docType methods
#' @rdname bmedian-methods
setGeneric(
  name = "bmedian",
  def = function(x) standardGeneric("bmedian")
)
#' @rdname bmedian-methods
#' @aliases bmedian
setMethod(
  f = "bmedian",
  signature = "MultiSet",
  definition = function(x){
    assayDataElement(x, "bMedian")
  }
)
#-----
#' Mean Background Intensity Accessor
#'
#' Extracts the matrix of mean background intensities (bMean)
#' from a MultiSet object created by \link{readArrays}
#'
#' @param x MultiSet object
#' @return matrix of mean background intensities
#' @exportMethod bmean
#' @docType methods
#' @rdname bmean-methods
setGeneric(
  name = "bmean",
  def = function(x) standardGeneric("bmean")
)
#' @rdname bmean-methods
#' @aliases bmean
setMethod(
  f = "bmean",
  signature = "MultiSet",
  definition = function(x){
    assayDataElement(x, "bMean")
  }
)
#-----
#' Flags Accessor
#'

```

```

#' Extracts the matrix of flagged values
#' from a MultiSet object created by \link{readArrays}
#'
#' @param x MultiSet object
#' @return matrix of flagged values
#' @exportMethod flags
#' @docType methods
#' @rdname flags-methods
setGeneric(
  name = "flags",
  def = function(x) standardGeneric("flags")
)
#' @rdname flags-methods
#' @aliases flags
setMethod(
  f = "flags",
  signature = "MultiSet",
  definition = function(x){
    assayDataElement(x, "flags")
  }
)

)
#-----
#' Diameter Accessor
#'
#' Extracts the matrix of feature diameter values
#' from a MultiSet object created by \link{readArrays}
#'
#' @param x MultiSet object
#' @return matrix of feature diameters
#' @exportMethod dia
#' @docType methods
#' @rdname dia-methods
setGeneric(
  name = "dia",
  def = function(x) standardGeneric("dia")
)
#' @rdname dia-methods
#' @aliases dia
setMethod(
  f = "dia",
  signature = "MultiSet",
  definition = function(x){
    assayDataElement(x, "dia")
  }
)
)

```


arrayBGcorr-methods.R

```
#' Local background feature correction algorithms
#'
#' Implements local background correction for peptide microarray data
#'
#' @param x MultiSet object with fMedian and
#' bMedian matrices in the assayData slot
#' @param method Character string specifying background correction method.
#' Valid methods are 'none', 'subtract', 'edwards', 'ratio' or 'normexp'.
#' Defaults to 'none' if no method is specified.
#' @param offset numeric value added to raw signal intensity
#' before background correction is implemented
#' @param transform Expression to transform raw data. Defaults to log2
#' @return MultiSet object with transformed and
#' background corrected foreground signal in the fMedian matrix
#'
#' @import limma
#' @exportMethod arrayBGcorr
#' @docType methods
#' @rdname arrayBGcorr-methods
setGeneric(
  name = "arrayBGcorr",
  def = function(x, ...) standardGeneric("arrayBGcorr")
)

#' @rdname arrayBGcorr-methods
#' @aliases arrayBGcorr
setMethod(
  f = "arrayBGcorr",
  signature = "MultiSet",
  definition = function(x, method = "none", offset = 1, transform = "log2", ...){

    if (offset < 0){
      stop("Offset value must be positive")
    }

    if (is.function(transform)){
      transformFunc <- transform
    } else if (transform == "none"){
      transformFunc <- function(y) identity(y)
    } else {
      transformExpression <- parse(text = paste(transform, "(y)", sep = ""))
      transformFunc <- function (y){
        eval(transformExpression)
      }
    }
  }
}
```

```

if (method == "none"){
  assayDataElement(x, "fMedian") <-
    transformFunc(assayDataElement(x, "fMedian"))
  return (x)

} else if (method == "subtract"){
  assayDataElement(x, "fMedian") <-
    (assayDataElement(x, "fMedian") + offset) - assayDataElement(x, "bMedian")
  assayDataElement(x, "fMedian")[assayDataElement(x, "fMedian") <= 0] <- offset
  assayDataElement(x, "fMedian") <- transformFunc(assayDataElement(x, "fMedian"))
  return(x)

} else if (method == "ratio"){
  assayDataElement(x, "fMedian") <- transformFunc(assayDataElement(x, "fMedian"))
  - transformFunc(assayDataElement(x, "bMedian"))
  return(x)

} else if (method == "edwards"){
  assayDataElement(x, "fMedian") <- backgroundCorrect.matrix(E = assayDataElement(x,
    "fMedian"), Eb = assayDataElement(x, "bMedian"), method = "edwards", ...)
  assayDataElement(x, "fMedian") <- transformFunc(assayDataElement(x, "fMedian"))
  return(x)

} else if (method == "normexp"){
  assayDataElement(x, "fMedian") <-
    backgroundCorrect.matrix(E = assayDataElement(x, "fMedian"),
      Eb = assayDataElement(x, "bMedian"),
      method = "normexp", offset = offset, ...)
  assayDataElement(x, "fMedian") <-
    transformFunc(assayDataElement(x, "fMedian"))
  return(x)

} else {

  stop("Method must be either 'none', 'subtract', 'ratio', 'edwards' or 'normexp'")
}
}
)

```

arrayNorm-methods.R

```
#' Normalisation for peptide microarray data
#'
#' @param x Multiset object with fMedian matrix in the assayData slot
#' @param method Character string specifying normalisation method.
#' Valid methods are 'none', 'scale', 'quantile' or 'LM'.
#' Defaults to 'none' if no method is specified.
#' @param ... additional arguments passed to scaleNorm or lmNorm
#' @return Multiset object with normalised signal in the fMedian matrix
#'
#' @importFrom preprocessCore normalize.quantiles
#' @exportMethod arrayNorm
#' @docType methods
#' @rdname arrayNorm-methods
setGeneric(
  name = "arrayNorm",
  def = function(x, ...) standardGeneric("arrayNorm")
)

#' @rdname arrayNorm-methods
#' @aliases arrayNorm
setMethod(
  f = "arrayNorm",
  signature = "Multiset",
  definition = function(x, method = "none", ...){

    if (method == "none"){

      return(x)

    } else if(method == "scale"){

      x <- scaleNorm(x, ...)
      return(x)

    } else if(method == "quantile"){

      assayDataElement(x, "fMedian") <- normalize.quantiles(fg(x))
      return(x)

    } else if (method == "LM"){

      assayDataElement(x, "fMedian") <- lmNorm(x, ...)
      return (x)

    }

    stop("Method must be either 'scale', 'quantile', 'LM' or 'none'")
  }
}
```

```

    }
)

#' @rdname arrayNorm-methods
#' @aliases arrayNorm
setMethod(
  f = "arrayNorm",
  signature = "ExpressionSet",
  definition = function(x, method = "none", skip.array = NULL, ...){

    fnames <- featureNames(x)

    if (method == "none"){

      return(x)

    } else if(method == "scale"){

      exprs(x) <- scaleNorm(exprs(x))
      return(x)

    } else if(method == "scaleGMM"){

      exprs(x) <- scaleNormGMM(exprs(x))
      return(x)

    } else if(method == "quantile"){

      if(is.null(skip.array)){
        exprs(x) <- normalize.quantiles(exprs(x))
        featureNames(x) <- fnames
        return(x)
      }
    }

    stop("Method must be either 'scale', 'quantile', or 'none'")

  }
)

```

scaleNorm-methods.R

```
#' Scale Normalisation
#'
#'
#' @param x MultiSet object with fMedian matrices
#' in the assayData slot
#' @param offset numeric value added to raw signal intensity
#' before background correction is implemented
#' @return MultiSet object with normalised
#' foreground signal in the fMedian matrix
#'
#' @exportMethod scaleNorm
#' @docType methods
#' @rdname scaleNorm-methods
setGeneric(
  name = "scaleNorm",
  def = function(x, ...) standardGeneric("scaleNorm")
)

#' @rdname scaleNorm-methods
#' @aliases scaleNorm
setMethod(
  f = "scaleNorm",
  signature = "MultiSet",
  definition = function(x, controlID = NULL){

    if (is.null(controlID)){
      y <- apply(fg(x), 2, median, na.rm = TRUE)
      y <- y - exp(mean(log(abs(y))))
      z <- sweep(fg(x), 2, y)
      assayDataElement(x, "fMedian") <- z

      return(x)

    } else {
      y <- fg(x)[fData(x)$ID %in% controlID, ]
      y <- apply(y, 2, median, na.rm = TRUE)
      y <- y - exp(mean(log(abs(y))))
      z <- sweep(fg(x), 2, y)
      assayDataElement(x, "fMedian") <- z

      return(x)
    }
  }
)

#' @rdname scaleNorm-methods
#' @aliases scaleNorm
setMethod(
```

```

f = "scaleNorm",
signature = "ExpressionSet",
definition = function(x, controlID = NULL){

  if (is.null(controlID)){
    y <- apply(exprs(x), 2, median, na.rm = TRUE)
    y <- y - exp(mean(log(abs(y))))
    z <- sweep(exprs(x), 2, y)
    exprs(x) <- z

    return(x)

  } else {
    y <- exprs(x)[featureNames(x) %in% controlID, ]
    y <- apply(y, 2, median, na.rm = TRUE)
    y <- y - exp(mean(log(abs(y))))
    z <- sweep(exprs(x), 2, y)
    exprs(x) <- z

    return(x)
  }
}
)

```



```

    )
  )
}
normmat[is.na(normmat)] <- 0
normdata <- fg(x) - normmat
return (normdata)

} else{

# Global LM normalisation using subarrays
arraylayout <- as.data.frame(getArrayLayout(x))
modeldata <- data.frame(Intensity = as.numeric(fg(x)),
                        Sample = factor(rep(sampleID, each = nrow(x)),
                                       levels = unique(sampleID)),
                        Subarray = factor(rep(arraylayout$subarray,
                                             times = ncol(x)))
)
fit <- lm(Intensity ~ Sample:Subarray, data = modeldata)
norm <- matrix(coef(fit)[-1],
              nrow = max(arraylayout$subarray),
              byrow = TRUE
)
normmat <- NULL
for (i in 1: max(arraylayout$subarray)){
  normmat <- rbind(normmat,
                  matrix(rep(norm[i, ],
                              times = sum(arraylayout$subarray == i)),
                          ncol = ncol(x), byrow = TRUE)
)
}
normmat[is.na(normmat)] <- 0
normdata <- fg(x) - normmat
return (normdata)
}

} else {

if(!is.null(controlID)){
# Control probe LM normalisation without subarray factor
ctrldata <- x[featureID %in% controlID, ]
if(nrow(ctrldata) == 0){
  stop("Control IDs not found in data to be normalised")
}
modeldata <- data.frame(Intensity = as.numeric(fg(ctrldata)),
                        Sample = factor(rep(sampleID, each = nrow(ctrldata)),
                                       levels = unique(sampleID))
)
fit <- lm(Intensity ~ Sample, data = modeldata)
norm <- coef(fit)[-1]
normmat <- matrix(rep(c(0, norm), times = nrow(x)),
                  ncol = ncol(x),

```



```

        byrow = TRUE
      )
    normdata <- fg(x) - normmat
    return (normdata)

  } else {

    # Global LM normalisation without subarray factor
    modeldata <- data.frame(Intensity = as.numeric(fg(x)),
                           Sample = factor(rep(sampleID, each = nrow(x)),
                                           levels = unique(sampleID))
    )
    fit <- lm(Intensity ~ Sample, data = modeldata)
    norm <- coef(fit)[-1]
    normmat <- matrix(rep(c(0, norm), times = nrow(x)),
                     ncol = ncol(x),
                     byrow = TRUE
    )
    normdata <- fg(x) - normmat
    return (normdata)
  }
}
}
}

```

arraySummary-methods.R

```
#' Summarises intra-array replicates with flagged values excluded
#'
#' @param x Multiset object
#' @param method c("Median", "Mean")
#' @return ExpressionSet with summarised intensity values in exprs slot
#' @exportMethod arraySummary
#' @docType methods arraySummary
#' @rdname arraySummary-methods
setGeneric(
  name = "arraySummary",
  def = function(x, ...) standardGeneric("arraySummary")
)

#' @rdname arraySummary-methods
#' @aliases arraySummary
setMethod(
  f = "arraySummary",
  signature = "Multiset",
  definition = function(x, method, cv.threshold){

    if (!is.null(fData(x)$ID)){
      ID <- fData(x)$ID
      ID <- factor(ID, levels = unique(ID))

    } else {
      stop("Peptide IDs not defined")

    }

    y <- assayDataElement(x, "fMedian")
    flags <- assayDataElement(x, "flags")
    y[flags < -99] <- NA

    if (method == "mean"){
      arraySumm <- .meanID(y, ID)

    } else if (method == "mean.closest"){

      if(abs(max(table(fData(x)$Subarray)) - min(table(fData(x)$Subarray))) != 0){
        stop("Subarray lengths must be equal to use this method.")
      }

      ID.subarray <- fData(x)$ID[fData(x)$Subarray == 1]
      ID.subarray <- factor(ID.subarray, levels = unique(ID.subarray))

      z <- array(y, dim = c(table(fData(x)$Subarray)[1], 3, dim(x)[2]))
      z <- aperm(z, c(1,3,2))
      z.mean.pair <- apply(z, c(1,2), .meanClosestPair, cv.threshold)
```

```

    arraySumm <- .meanID(z.mean.pair, ID.subarray)
  }

  obj <- new("ExpressionSet")
  assayData(obj) <- assayDataNew(exprs = arraySumm)
  phenoData(obj) <- phenoData(x)
  fData(obj) <- fData(x)[!duplicated(fData(x)$ID), ]
  featureNames(obj) <- fData(obj)$ID
  fData(obj) <- fData(obj)[ ,!(names(fData(obj)) %in%
    c("ID", "Block", "Column", "Row")), drop = FALSE]
  experimentData(obj) <- experimentData(x)
  protocolData(obj) <- protocolData(x)

  return(obj)
}
)

#' Calculate the mean of intra-array replicates
#' (INTERNAL FUNCTION)
#' @keywords internal
.meanID <- function(x, ID){
  csum <- rowsum(x, group = ID, reorder = TRUE, na.rm = TRUE)
  n <- tabulate(ID)
  cmean <- csum/n
  return (cmean)
}

#' Calculate the CV of intra-array replicates
#' (INTERNAL FUNCTION)
#' @keywords internal
.cvID <- function(x, ID){
  csum <- rowsum(x, group = ID, reorder = TRUE, na.rm = TRUE)
  csumsq <- rowsum(x^2, group = ID, reorder = TRUE, na.rm = TRUE)
  n <- tabulate(ID)
  cmean <- csum/n
  cvar <- ((n * csumsq) - csum^2)/n^2
  cv <- sqrt(cvar) / cmean
  return(cv)
}

#' Mean of closest pair of subarray replicates if CV over threshold value
#' (INTERNAL FUNCTION)
#' For triple subarray microarrays only
#' @keywords internal
.meanClosestPair <- function(x, cv.threshold ){
  x.mean <- mean(x, na.rm = TRUE)
  x.sd <- sd(x, na.rm = TRUE)
  x.cv <- x.sd/x.mean
  x.cv[is.na(x.cv)] <- 0
}

```

```
if(x.cv >= cv.threshold){
  x.means <- c(mean(c(x[1], x[2]), na.rm = TRUE),
              mean(c(x[1], x[3]), na.rm = TRUE),
              mean(c(x[2], x[3]), na.rm = TRUE)
            )
  y.mean <- x.means[which.min(c(abs(x[1]-x[2]), abs(x[1]-x[3]), abs(x[2]-x[3])))]
  return(y.mean)
} else{
  return (x.mean)
}
}
```

getArrayLayout-methods.R

```
#' Array Layout Accessor
#'
#' Gets microarray layout information matrix
#' Used to obtain the layout information for \link{plotImage}
#'
#' @param x Multiset object with block, column and row information in the fData slot
#' @return layout matrix
#' @exportMethod getArrayLayout
#' @docType methods
#' @rdname getArrayLayout-methods
setGeneric(
  name = "getArrayLayout",
  def = function(x, ...) standardGeneric("getArrayLayout")
)

#' @rdname getArrayLayout-methods
#' @aliases getArrayLayout
setMethod(
  f = "getArrayLayout",
  signature = "eSet",
  definition = function(x, ncols = 4, subarrays = 3){

    if(!all(c("Block", "Row", "Column") %in% names(fData(x))))
      stop("Feature data does not contain Block, Row and/or Column indexes")

    block.row <- ((fData(x)$Block - 1) %% ncols) + 1
    block.col <- ((fData(x)$Block - 1) %% ncols) + 1
    spot.row <- fData(x)$Row
    spot.col <- fData(x)$Column
    subarray <- ((block.row - 1) %% (max(block.row)/subarrays)) + 1

    y <- cbind(block.row, block.col, spot.row, spot.col, subarray)

    return (y)
  }
)
```

readArrays.R

```
#' Read peptide microarray data from GPR files
#'
#' \code{readArrays} is a function used to read
#' in peptide microarray data from Genepix GPR files
#' (in Axon ATF format). It produces a Bioconductor
#' Multiset object with microarray signal intensity
#' data (foreground intensity, background intensity
#' and other measures for assessing feature quality)
#' in the assayData slot. A minimal annotation set
#' is created by recording sample data
#' (sample unique identifier and file name)
#' in the phenoData slot, and feature data
#' (feature names and ID, and layout information)
#' in the featureData slot. The scan date and time
#' for the GPR file (if recorded in the GPR header)
#' is written to the annotated data frame in the
#' protocolData slot. Only signal intensity data from
#' a single wavelength (single colour data) is imported.
#'
#' @param files a data frame with 3 columns:
#' sampleName - unique identifier for the sample,
#' fileName - GPR file name and extension,
#' path - full path or URL to the directory holding the GPR file
#' @param wavelength integer value for the scan wavelength
#' (typically 635 for Cy5 and 532 for Cy3)
#' @return an object of class Multiset
#' @import plyr
#' @export
readArrays <- function(samplename = NULL, filename = NULL,
                       path = NULL, wavelength = NULL) {
  if(is.null(samplename)){
    stop("At least one sample name must be specified.")
  }

  if(is.null(filename)){
    stop("At least one GPR input file must be specified.")
  }

  if(is.null(path)){
    path <- getwd()
  }

  if(is.null(wavelength)){
    stop("wavelength must be specified.")
  }

  filePath <- file.path(path, filename)

  gprHeader <- list()
```

```

for (i in 1:length(filePath)){
  gprHeader[[i]] <- readArrayHeader(filePath[i], wavelength)
}

gprHeader <- rbind.fill(gprHeader)
rownames(gprHeader) <- samplename

dataHeader <- c(sprintf("F%i Median", wavelength),
                sprintf("F%i Mean", wavelength),
                sprintf("B%i", wavelength),
                sprintf("B%i Median", wavelength),
                sprintf("B%i Mean", wavelength)
)

colHeaders <- list()
for (i in 1:length(filePath)){
  colHeaders[[i]] <-
    read.table(filePath[i],
               skip = gprHeader$HeaderLines[i],
               nrows = 1,
               stringsAsFactors = FALSE,
               sep = "\t"
    )

  colHeaders[[i]] <- colHeaders[[i]] %in%
    c("Block", "Column", "Row", "Name", "ID",
      "Dia.", dataHeader, "F Pixels", "Flags")
}

colClasses <- list()
ncols <- sapply(colHeaders, length)
for (i in 1:length(filePath)){
  colClasses[[i]] <- rep("NULL", ncols[i])
  colClasses[[i]][colHeaders[[i]]] <- NA
}

gpr <- list()
for (i in 1:length(filePath)){

  cat("Reading GPR file:", filePath[i], "\n")
  gpr[[i]] <- read.table (file = filePath[i],
                        skip = gprHeader$HeaderLines[i],
                        header = TRUE,
                        stringsAsFactors = FALSE,
                        colClasses = colClasses[[i]],
                        sep = "\t"
  )
}

cat("Reading", length(filePath), "array files completed")

```

```

feature.id <- lapply(gpr, function(x) {
  sprintf("%s_%i_%i_%i", x$ID, x$Block, x$Column, x$Row)
})

if(length(unique(feature.id)) == 1){
  feature.id <- feature.id[[1]]

} else {
  stop("Feature data for imported GPR files is not identical.")
}

obj <- new("MultiSet")
assayData(obj) <-
  assayDataNew(fMedian = sapply(gpr, function(x) x[,7]),
               fMean = sapply(gpr, function(x) x[,8]),
               bg = sapply(gpr, function(x) x[,9]),
               bMedian = sapply(gpr, function(x) x[,10]),
               bMean = sapply(gpr, function(x) x[,11]),
               fPixels = sapply(gpr, function(x) x$F.Pixels),
               dia = sapply(gpr, function(x) x$Dia.),
               flags = sapply(gpr, function(x) x$Flags)
               )
sampleNames(assayData(obj)) <- samplename
featureNames(assayData(obj)) <- feature.id

pData(obj) <- data.frame (row.names = samplename,
                          fileName = filename
                          )

fData(obj) <- data.frame(row.names = feature.id,
                        ID = gpr[[1]]$ID,
                        Block = gpr[[1]]$Block,
                        Column = gpr[[1]]$Column,
                        Row = gpr[[1]]$Row,
                        Name = gpr[[1]]$Name,
                        stringsAsFactors = FALSE
                        )
protocolData(obj) <- AnnotatedDataFrame(data = gprHeader)
experimentData(obj) <- new("MIAME")

return(obj)
}

```



```
wl.col <- which(header[[wl.row]] == wavelength)
wl.data <- sapply(header, function(y) y[wl.col])
wl.data <- wl.data[!is.na(wl.data)]

header.data <- sapply(header, function(y) y[2])
header.data[!is.na(sapply(header, function(y) y[wl.col]))] <- wl.data
names(header.data) = sapply(header, function(y) y[1])
header.data <- data.frame(lapply(header.data, type.convert),
                          stringsAsFactors=FALSE
                          )
return(header.data)
}
}
```

arrayQApplot.R

```
## Peptide Microarray Quality Assessment Plots
##
## @param x Multiset object
## @param arr integer value indicate array to plot - corresponds to
## column of Multiset data object
## @param transform Expression to transform data. Defaults to log2
## @return plot on current graphics device
## @export
arrayQApplot <- function(x, arr, titletext, transform = "log2"){
  par(oma = c(0, 0, 3, 0))
  layout(matrix(c(1:6,2,3,7,8),2,5, byrow = T),
    widths = c(1.4, 0.8, 0.8, 1)
  )
  plotsubarrayBlocks(x, arr,
    outcex = 0.6,
    transform = transform
  )
  plotImage(x, arr,
    slot = "fg",
    lowcol = "white",
    highcol = "black",
    titletext = "FG Image",
    transform = transform
  )
  plotImage(x, arr,
    slot = "bg",
    lowcol = "white",
    highcol = "black",
    titletext = "BG Image"
  )
  plotsubarrayScatter(x, arr,
    c(1,2),
    cex = 0.6,
    transform = transform
  )
  plotsubarrayScatter(x, arr,
    c(2,3),
    cex = 0.6,
    transform = transform
  )
  plotsubarrayDensity(x, arr,
    transform = transform
  )
  plotsubarrayScatter(x, arr,
    c(1,3),
    cex = 0.6,
    transform = transform
  )
  plotsubarrayClosestValues(x, arr,
```

```
        cex = 0.6,  
        transform = transform  
    )  
    mtext(titletext, outer = TRUE)  
}
```

plotImage-methods.R

```
#' Image plot of microarray data
#'#' Creates image plot representing the microarray
#'#' foreground or background signal intensity
#'#' organised by the spatial location of the spots on the array.
#'#'#' @param x MultiSet object with fMedian and/or
#'#' bMedian matrices in the assayData slot
#'#' and layout data (block, column and row) in the fData slot.
#'#' @param lowcol Colour associated with low signal intensities
#'#' @param highcol Colour associated with high signal intensities
#'#' @param ncols number of colour shades used
#'#' @return MultiSet object with transformed and
#'#' background corrected foreground signal in the fMedian matrix
#'#'#' @export
#'#' @docType methods
#'#' @rdname plotImage-methods
setGeneric(
  name = "plotImage",
  def = function(x, ...) standardGeneric("plotImage")
)

#'#' @rdname plotImage-methods
#'#' @aliases plotImage
setMethod(
  f = "plotImage",
  signature = "MultiSet",
  definition = function(x, arr = 1, slot = "bg", lowcol, highcol,
                        ncols = 123, titletext, transform = "none", ...){
    layout <- getArrayLayout(x)

    if (is.function(transform)){
      transformFunc <- transform

    } else if (transform == "none"){
      transformFunc <- function(y) identity(y)

    } else {
      transformExpression <- parse(text = paste0(transform, "(y)"))
      transformFunc <- function (y) eval(transformExpression)
    }

    if (slot == "fg"){
      y = transformFunc(fg(x)[ ,arr])

    } else if (slot == "bg"){
      y = transformFunc(bg(x)[ ,arr])
    }
  }
)
```

```

} else if (slot %in% assayDataElementNames(x)){
  transformExpressionSlot <-
    parse(text = paste0("transformFunc(assayDataElement(x,'" , slot, "')[ ,arr]"))
  y <- eval(transformExpressionSlot)

} else {
  stop("Only valid assayData slots can be plotted as a microarray image.")
}

blockrows <- max(layout[, 'block.row'])
blockcols <- max(layout[, 'block.col'])
spotrows <- max(layout[, 'spot.row'])
spotcols <- max(layout[, 'spot.col'])
features <- blockrows * blockcols * spotrows * spotcols

low <- col2rgb(lowcol)/255
high <- col2rgb(highcol)/255
col <- rgb(seq(low[1], high[1], len = ncols),
           seq(low[2], high[2], len = ncols),
           seq(low[3], high[3], len = ncols)
)

nr <- spotrows * blockrows
nc <- spotcols * blockcols
row.ind <- (layout[, 'block.row'] - 1) * spotrows + layout[, 'spot.row']
col.ind <- (layout[, 'block.col'] - 1) * spotcols + layout[, 'spot.col']
ord <- order(row.ind, col.ind)

z <- matrix(y[ord], nrow = nr, ncol = nc, byrow = TRUE)
z <- t(z)[, nr:1]

image(1:nc, 1:nr, z, axes = F, xlab = "", ylab = "", col = col, ...)
box(lwd = 1)
abline(v = ((1:blockcols - 1) * spotcols + 0.5), lwd = 1)
abline(h = ((1:blockrows - 1) * spotrows + 0.5), lwd = 1)
title(titletext)
}
)

```

plotSubarrayBlocks-methods.R

```
#' Boxplots of array blocks (print tip groups) from peptide microarray data
#'
#' @param x MultiSet object with fMedian and/or bMedian matrices in assayData slot
#' @param arr Index indicating which array should be plotted
#' @param transform function to apply to transform the raw data
#' @return plot on current graphics device
#'
#' @export
#' @docType methods
#' @rdname plotSubarrayBlocks-methods
setGeneric(
  name = "plotSubarrayBlocks",
  def = function(x, ...) standardGeneric("plotSubarrayBlocks")
)

#' @rdname plotSubarrayBlocks-methods
#' @aliases plotSubarrayBlocks
setMethod(
  f = "plotSubarrayBlocks",
  signature = "MultiSet",
  definition = function(x, arr, transform = "log2", ...){
    if (is.function(transform)){
      transformFunc <- transform

    } else if (transform == "none"){
      transformFunc <- function(y) identity(y)

    } else {
      transformExpression <- parse(text = paste(transform, "(y)", sep = ""))
      transformFunc <- function (y){
        eval(transformExpression)
      }
    }

    arraydata <- transformFunc(fg(x[,arr]))
    plotdata <- data.frame(block = fData(x)$Block,
                          subarray = fData(x)$Subarray,
                          arraydata
                          )

    boxplot(arraydata ~ block, data = plotdata,
            col = rep(2:4, each = max(fData(x)$Block)/max(fData(x)$Subarray)),
            las = 1,
            pch = 20,
            xlab = "Block",
            ylab = "Signal Intensity", ...
            )
  }
)
```

plotSubarrayScatter-methods.R

```
## Scatter plots of subarrays from peptide microarray data
##
## @param x Multiset object with fMedian matrix in the assayData slot
## @param arr Index indicating which array should be plotted
## @param subarray Vector of length = 2 indicating
## which subarrays should be plotted
## @param transform function to apply to transform the raw data
## @return plot on current graphics device
##
## @export
## @docType methods
## @rdname plotSubarrayScatter-methods
setGeneric(
  name = "plotSubarrayScatter",
  def = function(x, ...) standardGeneric("plotSubarrayScatter")
)

## @rdname plotSubarrayScatter-methods
## @aliases plotSubarrayScatter
setMethod(
  f = "plotSubarrayScatter",
  signature = "Multiset",
  definition = function(x, arr, subarray = c(1,2), flagval = -100,
                        transform = "log2", ...){
    if (is.function(transform)){
      transformFunc <- transform

    } else if (transform == "none"){
      transformFunc <- function(y) identity(y)

    } else {
      transformExpression <- parse(text = paste(transform, "(y)", sep = ""))
      transformFunc <- function (y){
        eval(transformExpression)
      }
    }

    arraydata <- transformFunc(fg(x[, arr]))
    minval <- min(arraydata)
    maxval <- max(arraydata)

    flagdata <- flags(x[, arr])
    flagdata <- data.frame(SA.x = flagdata[fData(x)$Subarray == subarray[1]],
                          SA.y = flagdata[fData(x)$Subarray == subarray[2]]
                          )

    plotdata <- data.frame(SA.x = arraydata[fData(x)$Subarray == subarray[1]],
                          SA.y = arraydata[fData(x)$Subarray == subarray[2]]
                          )
  }
}
```



```

plot(SA.y ~ SA.x, data = plotdata,
     las = 1,
     pch = 20,
     xlim = c(minval, maxval),
     ylim = c(minval, maxval),
     xlab = paste("SA", subarray[1]),
     ylab = paste("SA", subarray[2]),
     ...
)
points(SA.y ~ SA.x,
       data = plotdata[apply(flagdata, 1, function(x) any(x == flagval)), ],
       pch = 20,
       col = "red",
       ...
)

lmfit <- lm(SA.y ~ SA.x, data = plotdata,)
abline(lmfit, col = "blue")
abline(0,1, col = "red")

lgnd1 <- bquote(R^2== .(round(summary(lmfit)$adj.r.squared, 3)))
lgnd2 <- bquote(beta== .(round(coef(summary(lmfit))[2,1], 3)))
legend("topleft",
      c(as.expression(lgnd1),as.expression(lgnd2)),
      bty = "n",
      cex = 0.9
)
}
)

```

plotSubarrayClosestValues-methods.R

```
#' Scatter plot of closest two values from subarrays
#' of peptide microarray data
#'
#' @param x Multiset object with fMedian matrix in the assayData slot
#' @param arr Index indicating which array should be plotted
#' @param transform function to apply to transform the raw data
#' @return plot on current graphics device
#'
#' @export
#' @docType methods
#' @rdname plotSubarrayClosestValues-methods
setGeneric(
  name = "plotSubarrayClosestValues",
  def = function(x, ...) standardGeneric("plotSubarrayClosestValues")
)

#' @rdname plotSubarrayClosestValues-methods
#' @aliases plotSubarrayClosestValues
setMethod(
  f = "plotSubarrayClosestValues",
  signature = "Multiset",
  definition = function(x, arr, transform = "log2", ...){
    if (is.function(transform)){
      transformFunc <- transform

    } else if (transform == "none"){
      transformFunc <- function(y) identity(y)

    } else {
      transformExpression <- parse(text = paste0(transform, "(y)"))
      transformFunc <- function (y) eval(transformExpression)
    }

    arraydata <- transformFunc(fg(x[,arr]))
    minval <- min(arraydata)
    maxval <- max(arraydata)

    plotdata <-
      list(P1_2 = data.frame(SA.x = arraydata[fData(x)$Subarray == 1, ],
                            SA.y = arraydata[fData(x)$Subarray == 2, ]
      ),
          P1_3 = data.frame(SA.x = arraydata[fData(x)$Subarray == 1, ],
                            SA.y = arraydata[fData(x)$Subarray == 3, ]
      ),
          P2_3 = data.frame(SA.x = arraydata[fData(x)$Subarray == 2, ],
                            SA.y = arraydata[fData(x)$Subarray == 3, ]
      )
  }
)
```

```

plotdata.abs.diff <-
  cbind(abs(plotdata$P1_2[,1]-plotdata$P1_2[,2]),
        abs(plotdata$P1_3[,1]-plotdata$P1_2[,2]),
        abs(plotdata$P2_3[,1]-plotdata$P2_3[,2])
  )
plotdata.closest <- apply(plotdata.abs.diff, 1, which.min)

a <- list()
for(i in 1: length(plotdata.closest)){
  a[[i]] <- plotdata[[plotdata.closest[i]]][i,]
}

plotdata.final <- do.call("rbind", a)
plot(SA.y ~ SA.x, data = plotdata.final,
     las = 1,
     pch = 20,
     xlim = c(minval, maxval),
     ylim = c(minval, maxval),
     xlab = "",
     ylab = "",
     ...
)

lmfit <- lm(SA.y ~ SA.x, data = plotdata.final,)
lmfit.1 <- lm(SA.y-SA.x ~ SA.x, data = plotdata.final,)
p <- (1 - pt(coef(summary(lmfit))[2,1]/coef(summary(lmfit))[2,2],
            lmfit$df.residual))*2
p1 <- (1 - pt(abs(coef(summary(lmfit))[2,1] - 1)/coef(summary(lmfit))[2,2],
            lmfit$df.residual))*2

abline(lmfit, col = "blue")
abline(0,1, col = "red")

betaval <- round(coef(summary(lmfit))[2,1], 3)
lci <- round(confint(lmfit)[2,1], 3)
uci <- round(confint(lmfit)[2,2], 3)
lgnd1 <- bquote(R^2== .(round(summary(lmfit)$adj.r.squared, 3)))
lgnd2 <- bquote(beta== .(betaval) * (.lci)-.(uci))

legend("topleft",
      c(as.expression(lgnd1),
        as.expression(lgnd2)),
      bty = "n",
      cex = 0.9
)

summary.stats <-
  data.frame(R2 = round(summary(lmfit)$adj.r.squared, 3),
            Beta = betaval,
            LCI = lci,
            UCI = uci
  )

```

```

    )
    return(summary.stats)
}
)

```

plotSubarrayDensity-methods.R

```

#' Density plots of subarrays from peptide microarray data
#'
#' @param x Multiset object with fMedian and/or
#' bMedian matrices in the assayData slot
#' @param arr Index indicating which array should be plotted
#' @param subarray vector of length = 2 indicating
#' which subarrays should be plotted
#' @param transform function to apply to transform the raw data
#' @return plot on current graphics device
#'
#' @export
#' @docType methods
#' @rdname plotSubarrayDensity-methods
setGeneric(
  name = "plotSubarrayDensity",
  def = function(x, ...) standardGeneric("plotSubarrayDensity")
)

#' @rdname plotSubarrayDensity-methods
#' @aliases plotSubarrayDensity
setMethod(
  f = "plotSubarrayDensity",
  signature = "Multiset",
  definition = function(x, arr, transform = "log2", ...){
    if (is.function(transform)){
      transformFunc <- transform

    } else if (transform == "none"){
      transformFunc <- function(y) identity(y)

    } else {
      transformExpression <- parse(text = paste(transform, "(y)", sep = ""))
      transformFunc <- function (y){
        eval(transformExpression)
      }
    }
  }

  arraydata <- transformFunc(fg(x[,arr]))

  plotdata <- data.frame(SA1 = arraydata[fData(x)$Subarray == 1],
                        SA2 = arraydata[fData(x)$Subarray == 2],
                        SA3 = arraydata[fData(x)$Subarray == 3])

```

```

    )
plotdata <- apply(plotdata, 2, density)
range.x <- lapply(plotdata, function(y) range(y$x))
range.y <- lapply(plotdata, function(y) range(y$y))

plot(0,
     type = "n",
     las = 1,
     xlim = range(range.x),
     ylim = c(0, range(range.y)[2]),
     xlab = "Signal Intensity",
     ylab = "Density",
     ...
)
lines(plotdata$SA1, col = "red")
lines(plotdata$SA2, col = "green")
lines(plotdata$SA3, col = "blue")








legend("topright",
      inset = 0,
      c("SA1", "SA2", "SA3"),
      bty = "n",
      fill = c(2:4),
      cex = 0.5
)
}
)

```

Appendix 2: Bristol Stool Chart

The Bristol Stool chart was developed by Lewis and Heaton (1997) at the University of Bristol as an objective means of classifying stool consistency. The Bristol stool type is a useful surrogate marker of the colon transit time.

Bristol Stool Chart

Type 1		Separate hard lumps, like nuts (hard to pass)
Type 2		Sausage-shaped but lumpy
Type 3		Like a sausage but with cracks on its surface
Type 4		Like a sausage or snake, smooth and soft
Type 5		Soft blobs with clear-cut edges (passed easily)
Type 6		Fluffy pieces with ragged edges, a mushy stool
Type 7		Watery, no solid pieces. Entirely Liquid