Stockholms
universitet

My Adventures with ELISPOT Assays: Design and Analysis of Experiments in Vaccine Development

Rickard Strandberg

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2017:12
Matematisk statistik
December 2017

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Master Thesis **2017:12**
http://www.math.su.se

My Adventures with ELISPOT Assays: Design and Analysis of Experiments in Vaccine Development

Rickard Strandberg[*]

December 2017

Abstract

The ELISPOT assay is a method for testing the human immune system's response to infection, and has become an important component in the development of vaccines for pathogens such as HIV and influenza. Testing the efficacy of these vaccines using ELISPOT requires testing hundreds of peptides associated with a pathogen. But performing such large assays is associated with considerable resource costs and time commitments, and becomes very inefficient when one factors in that only a small percentage of these peptides turn out to give positive responses. For this reason, peptides are often pooled and tested together rather than individually. This reduces the resource requirements, but makes it more difficult to distinguish the individual peptides. If a pool gives a positive response, which peptide(s) in that pool generated the response? The need for smart ways of designing these peptide pools is apparent.

In this thesis we present a generalization of the standard method of pooling, motivated by a newly introduced idea ofpeptide overlapẆe then implement the Expectation Maximization algorithm as a tool to directly estimate individual peptide responses among hundreds using only a single ELISPOT microtiter plate. A new criterion for distinguishing a responding pool from inherent background noise is also introduced and compared to current alternatives. These three components come together to provide an accurate and cost-efficient method for designing and analyzing ELISPOT assays. These tools are made available through a free web application, putting them directly into the hands of immunologists using an easy-to-use interface.

[*] Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: rickardjs@gmail.com. Supervisor: Chun-Biu Li.

Acknowledgments

I would like to give thanks to my supervisor Chun-Biu Li at Stockholm University for his support and advice on this difficult project; and to my external supervisors at Karolinska Institutet, Marie Reilly and Peter Ström, for giving me this opportunity, and for their invaluable help navigating this topic. I also wish to thank the rest of the department of Medical Epidemiology and Biostatistics at Karolinska Institutet for providing a kind and supportive work environment. Lastly, I would like to thank Nathifa Moyo and Nicola Borthwick at the Jenner Institute, University of Oxford, for the opportunity to visit and for answering my layman questions.

Contents

1 Introduction

A powerful feature of the human immune system is its ability to recognize and remember past infections. The T-cell is a type of white blood cell, and an important component of this *adaptive immune system*. Each individual T-cell is programmed to react to the presence of a specific 9-amino acid long protein called a *peptide* [23]. These peptides were at some point in the past expressed by cells infected by some pathogen (such as a type of virus or bacteria). This means that every person has a stable of T-cells which together cover a wide range of peptides associated with various past infections. The idea behind a vaccine is to 'train' the immune system to remember a set of such peptides which are specific to a particular pathogen by creating and maintaining the appropriate T-cells. The effectiveness of a proposed vaccine can thus be evaluated by measuring a vaccinated subject's immune response to the pathogen's associated peptides.

A widely used method for testing if someone's immune system includes T-cells attuned to specific peptides is the Enzyme-Linked ImmunoSPOT (ELISPOT) assay [1], which has for example been used in vaccine trials for HIV [19, 21, 25] and influenza [15]. When a T-cell recognizes its assigned peptide it will secrete a signaling protein called *interferon gamma* (IFN-γ) which inhibits viral replication and stimulates further response from the other parts of the immune system. The ELISPOT assay consists of one or more microtiter plates—small plates with 96 very small 'wells' (compartments) for testing. These wells are first coated with antibodies which have receptors for IFN-γ. Next, small blood samples are placed in each well together with a peptide solution to be tested for each well. If the blood sample includes T-cells attuned to the peptide in a well, each of these T-cells will secrete IFN-γ, which will immediately bind to the receptors closest to the T-cell due to the thin coatings of antibody and blood. These reactions will—after developing the plate—appear as dark spots in the wells, where each spot corresponds to a single T-cell responding to the peptide of that well. By counting the number of spots in a well, one can gauge the person's immune response to that peptide.

An example of wells having formed spots can be found in Figure 1, where 24 out of the 96 wells on an ELISPOT plate can be seen. We can see that the number of spots vary greatly between wells, which indicates different levels of response to different peptides.

A persistent problem since the introduction of the ELISPOT method is determining what constitutes an actual positive immune response. Due to the various stages of sampling and preparation of the assays, a small number of spots usually form even when there is no actual physiological response to any peptide. These spots are called the background, and can vary from no spots at all, up to ten or more. For this reason it is customary to dedicate at least three of the 96 wells on a plate to measuring the background, which is done by leaving them empty of any peptides. These are called the *negative control wells*. Another three wells are typically reserved for positive controls, which are used to make sure the assay even works by intentionally stimulating those wells to form spots. This leaves up to 90 wells per plate to test peptides on. In Figure 1 we can see from the three control wells in the first column (d1, e1, f1) that the background lies somewhere around 20 spots.

In a typical ELISPOT assay several hundred peptides are of interest. In most cases, only a few ($< 10\%$) peptides are expected to respond [18], which makes individual testing of these peptides very inefficient, requiring many plates and large amounts of blood sample. This is especially true if we want test with 2 or 3 replicates for each peptide. Therefore, researchers pool the peptides together, making them share wells with other peptides. This greatly reduces the number of wells and plates needed, but imposes another layer to the problem, namely that if a pool of peptides gives a response, which of the peptides caused the response? Most commonly, this distinction has been made by re-testing the peptides of these positive pools individually [4, 12]. But an alternative to this two-stage analysis has recently been proposed by [21], whereby one uses the EM algorithm to estimate each peptides direct contribution to the pool. Whereas previously the pools were replicated multiple times on a plate (e.g. 30 different pools each used 3 times), [21] proposed using all-unique pools where each peptide is still used multiple times, only in distinctly different pools each time (e.g. 90 different pools used once but where each peptide is still used in 3 of the pools).

This way of 'mixing up' the replicates helps with the identifiability of peptide effects, which would be impossible with identical replicates.

In this thesis we will attempt to create a time and cost efficient way of performing ELISPOT assays. Here we let the relative efficiency be determined by the number of plates (and wells) used for the assay, where fewer would require less time to prepare and perform, and smaller blood samples. The goal is to develop a methodology for designing and analyzing ELISPOT assays which will fit any number of peptides onto a single ELISPOT plate (i.e using 90 peptide pools), and to estimate individual peptide responses directly, without any extra stages of testing. We will introduce the concept of peptide overlap as a measure of how good the 'mixing' of peptide pools is, and find a design optimizing this measure—the *Non-Overlapping Design* (NOD). We will also propose a new criterion for distinguishing an actual response from background noise—the *Bayesian Predictive Criterion* (BPC). This criterion can be used to test the positivity of individual wells, applicable both when testing pools and individual peptides. We will then use this new positivity criterion to augment the estimation procedure described in [21].

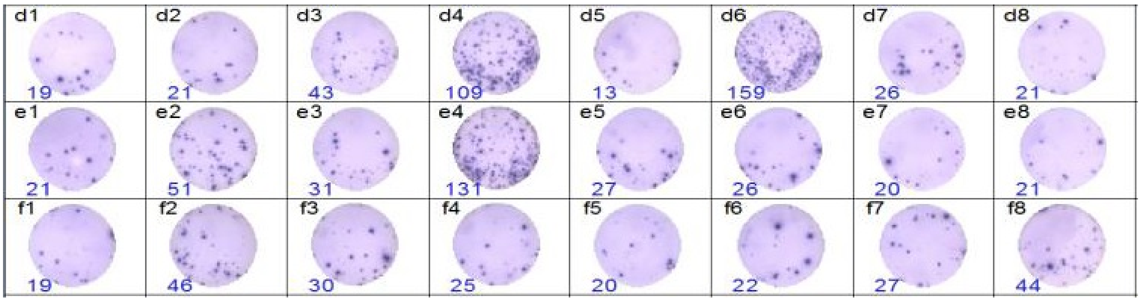


Figure 1: Example wells from an ELISPOT assay from an HIV vaccine trial performed on mice. The code on the top left of each well is the position of that well on the plate, and the number at the bottom is the number of spots counted by software. If we were told that d1, e1 and f1 were negative controls, which of the others should we deem positive?. (Image provided by Nathifa Moyo at the Jenner Institute, University of Oxford.)

2 Materials and Methods

Let us assume that we have an ELISPOT assay where N peptides $p_1, p_2, \ldots, p_N$ are to be tested c times each (i.e. c replicates). If each of the n peptides are tested individually c times, and if $Z_{(k)j}$ is the number of spots generated by the j:th peptide's k:th replicate, then the set of random variables we would observe are

$$\mathbf{Z} = \begin{array}{cccc} Z_{(1)1} & Z_{(2)1} & \cdots & Z_{(c)1} \\ Z_{(1)2} & Z_{(2)2} & \cdots & Z_{(c)2} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ Z_{(1)N} & Z_{(2)N} & \cdots & Z_{(c)N} \end{array} .$$

Since these random variables represent discrete counts, we are inspired make the following distributional assumption:

Assumption 1. *The number of spots generated by a responding peptide p_j follows a Poisson(β_j)-distribution, for some $\beta_j > 0$. These spot counts are independent of those generated by other peptides. Furthermore, the number of background spots in each well is Poisson(β_0)-distributed for some $\beta_0 > 0$ and is independent of the number of spots generated by any peptide.*

The Poisson-distribution is a common choice of distribution for count data, and has also been used for analyzing ELISPOT assays [2, 7, 18]. This distribution has some nice properties which we shall make use of later. The independence of the Z-variables can be motivated by the fact that a given T-cell will respond only to a single particular peptide. Therefore we can assume that different peptides do not compete over the same T-cells. This assumption should be seen as a simplification, as the biological reality is likely to be more complicated.

It will turn out to be notationally convenient to also assume that a non-responding peptide generates spots according to a degenerate *Poisson*(0)-distribution, i.e. if a peptide p_j is non-responding, then $\beta_j = 0$. Note that the distribution for the background is Poisson-distributed just like any of the peptides. It will therefore sometimes be convenient to treat the background as just another peptide—a peptide which is part of all pools, plus the negative control wells, where it is alone.

An assay testing these peptides individually would need $c \cdot N$ wells, which would require multiple plates and increasingly large blood samples as N goes into the hundreds. We would therefore like to fit these cN observations onto a single plate with only 90 available wells. To this end, instead of plating and observing all individual spot count variables, we imagine that we partition the Z-variables into 90 subsets, which are made to be as equal in size as possible, and made sure not to contain multiple replicates of the same peptide. These subsets are what we call the peptide pools.

With the peptide pools established through some partition, it is convenient to change the notation of the spot count variables. For a peptide p_j which is part of pool P_i, let Z_{ij} now denote the number of spots generated by p_j in P_i. This notation change is akin to a mapping given by

$$\{Z_{(1)j}, \ldots, Z_{(c)j}\} \to \{Z_{i_1 j}, \ldots, Z_{i_c j}\},$$

where $i_1, i_2, \ldots, i_c$ are the indices of the pools containing p_j.

Example 1. *Consider a situation with $N = 8$ peptides and $c = 2$ replicates for each peptide. Then one possible partition of the spot count variables $Z_{(k),j}$ $(k = 1, 2;\ j = 1, \ldots, 8)$ into 6 pools is:*

$$\begin{array}{ccc} \left\{Z_{(1)1}, Z_{(1)2}, Z_{(1)3}\right\} & \left\{Z_{(1)4}, Z_{(1)5}, Z_{(1)8}\right\} & \left\{Z_{(1)6}, Z_{(1)7}\right\} \\ \left\{Z_{(2)1}, Z_{(2)5}, Z_{(2)8}\right\} & \left\{Z_{(2)7}, Z_{(2)2}, Z_{(2)3}\right\} & \left\{Z_{(2)4}, Z_{(2)6}\right\} \end{array} .$$

These pools have approximately the same size (either 2 or 3), and do not contain multiple replicates of the same peptide. With the changed notation, the pools can be written as

$$\begin{array}{llllll} P_1 : & \{Z_{11}, Z_{12}, Z_{13}\} & P_2 : & \{Z_{24}, Z_{25}, Z_{28}\} & P_3 : & \{Z_{36}, Z_{37}\} \\ P_4 : & \{Z_{41}, Z_{45}, Z_{48}\} & P_5 : & \{Z_{57}, Z_{52}, Z_{53}\} & P_6 : & \{Z_{64}, Z_{66}\} \end{array} .$$

This makes it easier to identify both which pools a peptide is part of, as well as which peptides are contained in a given pool. □

Once we have partitioned these spot count variables into M (almost) equal-sized pools and perform the assay, we note that what we observe according to Assumption 1 is the sum of the random variables in each pool, plus a background variable (Z_{i0} for the i:th pool). In other words, what we observe in the well containing pool P_i is the random variable

$$Y_i = Z_{i0} + \sum_{j | p_j \in P_i} Z_{ij}.$$

These observed variables $Y_1, Y_2, \ldots, Y_M$ are—due to Assumption 1—also Poisson-distributed, as the following lemma shows:

Lemma 1. *Let $Z_1, Z_2, \ldots, Z_n$ be independent random variables with $Z_j \sim Poisson(\beta_j)$, for $j = 1, 2, \ldots, n$, and let $Y_n = \sum_{j=1}^{n} Z_j$. Then*

$$Y_n \sim Poisson\left(\sum_{j=1}^{n} \beta_j\right),$$

for any $n \geq 1$.

Proof. See Appendix A. □

If $\beta = (\beta_0, \beta_1, \ldots, \beta_N)^T$ is the parameter vector for respective Poisson rates of the background and the peptides, Lemma 1 implies that $Y_i \sim Poisson(\beta_0 + \sum_{j|p_j \in P_i} \beta_j)$. Furthermore, due to the partitioning and independence of the Z-variables, the Y-variables are also independent.

2.1 The Design Matrix

In order to keep track of which peptides are part of which pools, we introduce the following design matrix:

Definition 1. *Given some set of N peptides $\mathbf{p} = \{p_1, p_2, \ldots, p_N\}$ arranged into M peptide pools $\mathbf{P} = \{P_1, P_2, \ldots, P_M\}$, the* ELISPOT design matrix $\mathbf{X}$ *is defined as the $M \times (N+1)$-matrix*

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1N} \\ 1 & x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ 1 & x_{(M-1)1} & x_{(M-1)2} & \cdots & x_{(M-1)N} \\ 1 & x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix},$$

where

$$x_{ij} = \begin{cases} 1 & \textit{if } p_j \in P_i \\ 0 & \textit{otherwise} \end{cases},$$

for $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$, where the first column of ones represents the presence of background in all pools.

Once we have both the parameter vector $\beta = (\beta_0, \beta_1, \ldots, \beta_N)^T$ and a design matrix $\mathbf{X}$, we can rather compactly write out the expected values for the vector $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_M)^T$ of pooled spot counts as

$$E[\mathbf{Y}] = \mathbf{X}\beta. \tag{1}$$

Example 2. *For the pools in Example 1 and with 3 extra negative controls, the corresponding design matrix would be*

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

With 8 peptides, the 9 parameter vector is $\beta = (\beta_0, \beta_1, \ldots, \beta_8)^T$, and the expected spot counts for the 6 pools are

$$\mathbf{X}\beta = \begin{bmatrix} \beta_0 + \beta_1 + \beta_2 + \beta_3 \\ \beta_0 + \beta_4 + \beta_5 + \beta_8 \\ \beta_0 + \beta_6 + \beta_7 \\ \beta_0 + \beta_1 + \beta_5 + \beta_8 \\ \beta_0 + \beta_2 + \beta_3 + \beta_7 \\ \beta_0 + \beta_4 + \beta_6 \\ \beta_0 \\ \beta_0 \\ \beta_0 \end{bmatrix}$$

□

2.2 The EM Algorithm for Pooled ELISPOT Assays

Now that we have set up a parametric framework for a pooled ELISPOT assay, we set out to estimate the parameter vector $\beta = (\beta_0, \beta_1, \ldots, \beta_n)^T$. These estimates should then tell us which peptides give a response, as well as the magnitude of that response. The estimation procedure we shall use is the one described in [21]. The key to estimating β lies in viewing the pooled assay as an incomplete data problem, where instead of observing all $Z - variables$, we only observe the Y-variables.

The Expectation-Maximization (EM) algorithm [6, 22] is a method which can be applied to the missing data problem described above. We note that the *complete* likelihood, i.e. the likelihood if all Z_{ij} were observed, is under our distributional assumptions a set of independent Poisson variables with rates $\beta_j, j = 0, \ldots, n$ corresponding to the background and each of the n peptides. (Remember that we treat the background just like any other peptide, with its own Poisson rate β_0.) In such a likelihood, the maximum likelihood estimates of the β_j:s are simply the sample means:

$$\hat{\beta}_j = \frac{\sum_{i|p_j \in P_i} z_{ij}}{c_j}, \quad \text{where } c_j = |\{P_i; p_j \in P_i\}|.$$

This will constitute the Maximization step of the EM algorithm.

The Expectation step now consists of calculating the expected likelihood of the complete data $\mathbf{Z}$ *given* the observed data $\mathbf{Y}$. First we note that the distribution of a set of independent Poisson variables given their sum is the Multinomial distribution given by Lemma 2:

Lemma 2. *Let $Z_1, Z_2, \ldots, Z_n$ be independent random variables with $Z_j \sim Poisson(\beta_j)$, for $j = 1, 2, \ldots, n$, and let $Y_n = \sum_{j=1}^n Z_j$. Then*

$$(Z_1, Z_2, \ldots, Z_n)|(Y_n = y) \sim Multinomial\left(y, \left(\frac{\beta_1}{\sum_{j=1}^n \beta_j}, \frac{\beta_2}{\sum_{j=1}^n \beta_j}, \ldots, \frac{\beta_n}{\sum_{j=1}^n \beta_j}\right)\right),$$

for any $n \geq 1$.

Proof. See Appendix A. □

Since the Z_{ij} are partitioned into pools, and the Y_i are independent, the conditional likelihood will consist of 90 such independent multinomial distributions—one for each of the 90 pools. (Note that since the negative controls only contain the background, they are fully observed even in this conditional likelihood.) Given some interim parameter values $\beta^{(0)}$ the expectation of $Z_{ij}|\mathbf{y}, \beta^{(0)}$ is

$$E[Z_{ij}|\mathbf{y}, \beta^{(0)}] = y_i q_{ij},$$

where

$$q_{ij} = \frac{\beta_j^{(0)}}{\sum_{k|p_k \in P_i} \beta_k^{(0)}},$$

in accordance with Lemma 2. This will be the Expectation step.

Combining the Expectation and Maximization steps, we get a function for iterative estimates of β. The estimate of β_j at step $t+1$ of this iteration is given by

$$\beta_j^{(t+1)} = \frac{\sum_{i|p_j \in P_i} E[Z_{ij}|y, \beta^{(t)}]}{c_j} = \frac{\sum_{i|p_j \in P_i} y_i \frac{\beta_j^{(t)}}{\sum_{k|p_k \in P_i} \beta_k^{(t)}}}{c_j}. \tag{2}$$

This update step can be vectorized for easier execution in e.g. **R** by using a design matrix $\mathbf{X}$ in addition to the vector $\mathbf{c} = (c_0, c_1, \ldots, c_n)$. It then updates the entire parameter vector $\beta^{(t)}$ by

$$\beta^{(t+1)} \leftarrow X^T \left[\frac{y}{X\beta^{(t)}}\right] * \left[\frac{\beta^{(t)}}{\mathbf{c}}\right],$$

where '*' and '/' denote element-wise multiplication and division respectively (similar to matrix addition is performed).

This iteration proceeds from some starting values $\beta^{(0)}$ until convergence at some step T, whereby we let our final estimates be $\hat{\beta} = \beta^{(T)}$. The entire procedure can be summarized in Algorithm 1:

Algorithm 1: Estimate peptide effects using EM algorithm.

```
Input: Observed spot counts $y$, ELISPOT design $X$, starting values $\beta^{(0)}$.
Output: EM estimates of peptide effects $\hat{\beta}$.
$t \leftarrow 0$
$\beta^{(t)} \leftarrow$ starting values for $\beta$
while not converged do
    $\beta^{(t+1)} = \mathbf{X}^T \left[\frac{\mathbf{y}}{\mathbf{X}\beta^{(t)}}\right] * \left[\frac{\beta^{(t)}}{\mathbf{c}}\right]$
    $t \leftarrow t + 1$
end
$\hat{\beta} \leftarrow \beta^{(t)}$
```

Example 3. *Let us study what happens to the EM algorithm when there is no way to distinguish the peptides. For simplicity we limit ourselves to 3 pools with the same 3 peptides in them, and 3 negative controls. The design matrix X is then*

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

Due to all three peptides being part of all three pools, the update equation (2) for their respective parameters simplifies to

$$\beta_j^{(t+1)} = \frac{\sum_{i=1}^{3} y_i \frac{\beta_j^{(t)}}{\sum_{k=0}^{3}=\beta_k^{(t)}}}{3} = \left[\frac{\frac{1}{3}\sum_{i=1}^{3} y_i}{\sum_{k=0}^{3} \beta_k^{(t)}}\right] \beta_j^{(t)},$$

for $j = 1, 2, 3$. In this case the three estimates are updated by scaling the old estimate with the same factor for all three parameters. This means that the final estimates of these parameters will be proportional to the vector of starting parameters, i.e. $(\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3) \propto (\beta_1^{(0)}, \beta_2^{(0)}, \beta_3^{(0)})$. This means that the estimates are completely determined by the starting values. Since the peptide parameters are unidentifiable*, the EM algorithm does not work. The background parameter β_0, however,* is *identifiable. It's update step is given by*

$$\beta_0^{(t+1)} == \frac{\left[\frac{\sum_{i=1}^{3} y_i}{\sum_{k=0}^{3} \beta_k^{(t)}}\right] \beta_0^{(t)} + \sum_{i=4}^{6} y_i}{6},$$

and will in this case converge to $\sum_{i=4}^{6} y_i/3$. □

While Example 3 is an extreme case, it serves to highlight the importance of mixing up the peptide pool compositions in order to have as many identifiable parameters as possible. The necessity of having negative controls also becomes apparent, since β_0 is inevitably part of every pool, and must therefore be 'tested individually' in order to be identifiable. In the next section we will define and study a particular type of design which provides a way of mixing pools.

2.3 The Non-Overlapping Design

As highlighted by Example 3, in order to obtain better and more identifiable estimates, we need to find a good assay design which mixes up the pools. Assuming we have no prior information about any of the peptides we want to test, we will in this thesis propose the following four core ideas for a good pooled design:

1. **Use one plate with 90 wells for peptide analysis**. This restriction puts an upper limit on the number of pools needed and the time it takes to develop and analyze the assay.

2. **The sizes of all 90 pools are as equal as possible**. The sizes of the pools determine how easy it is to distinguish the individual peptides in case of a response, with smaller pools being easier. If we assume no prior knowledge of which peptides might be positive it makes sense to not favor any peptides by placing some of them in considerably smaller pools. If we knew exactly which peptides were positive and negative, we would put the positive in pools of size 1 (i.e. alone) to get better estimates of their rates; and all the negatives in a single large pool, knowing their rates to be 0.

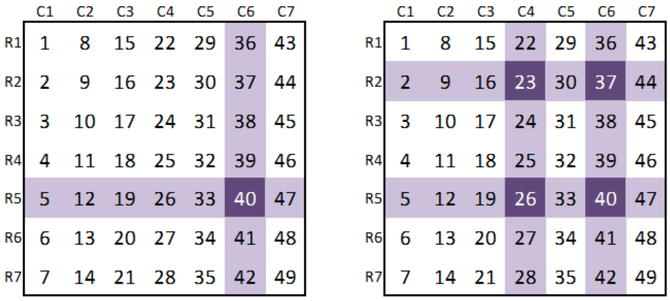
3. **Each peptide is in the same number of pools**, where we focus on using three replicates for each peptide for this thesis. If we do not assume any prior knowledge about the peptides, we should not favor any by using some peptides more than others. If we knew exactly which peptides were positive or negative, we would omit the negatives completely (knowing their rates to be 0), and use the positive peptides as many times as possible to better estimate their rates.

4. **No two peptides share a pool more than once**, but possibly zero times. This comes from the idea that the more times two peptides share pools, the harder they can be to distinguish. If two peptides, p_1 and p_2 are in exactly the same set of pools in which all tests are positive, it is impossible to determine if it is p_1, p_2 or both which are positive, as shown in Example 3. Therefore we set this upper limit of how many times any two peptides can be present in different pools.

With these four ideas stated and motivated, we an begin the process of actually finding such a design. It is not a journey taken lightly, but will result in two methods for explicitly finding one. We first do a quick review of one of the current ways of designing pools.

2.3.1 ELISPOT Matrix Designs

One strategy for pooling peptides which has been utilized in the past is the so-called *matrix design* [25, 26]. For such a design, a squared number $N = m^2$ of peptides are arranged in an $m \times m$ matrix, and the peptides in each of the m rows and m columns of this matrix are used to create $2m$ pools for testing. In Figure 2 we see an example of a matrix design for $n = 49 = 7^2$ peptides.



	C1	C2	C3	C4	C5	C6	C7
R1	1	8	15	22	29	36	43
R2	2	9	16	23	30	37	44
R3	3	10	17	24	31	38	45
R4	4	11	18	25	32	39	46
R5	5	12	19	26	33	40	47
R6	6	13	20	27	34	41	48
R7	7	14	21	28	35	42	49

	C1	C2	C3	C4	C5	C6	C7
R1	1	8	15	22	29	36	43
R2	2	9	16	23	30	37	44
R3	3	10	17	24	31	38	45
R4	4	11	18	25	32	39	46
R5	5	12	19	26	33	40	47
R6	6	13	20	27	34	41	48
R7	7	14	21	28	35	42	49

Figure 2: Example of a matrix design for $N = 49$ peptides. The $M = 2m = 14$ pools are the rows and columns of the matrix. The potentially positive peptides are the ones at the intersection of positive row and column pools (highlighted in purple).

The matrix design has two interesting features. The first is that each peptide is used in multiple pools. The second feature is that no two peptides share a pool more than once. This is used when analyzing the assay. After determining which pools gave a positive response, any peptide which was at the intersection of a positive row pool and column pool is considered a possible responding peptide, and is brought forward for individual testing. The fact that no two peptides share a pool more than once assures that only one peptide sits at the intersection of a given row and column pair, as can be seen on the left side of Figure 2. The situation complicates when there are multiple positive rows and columns, as can be seen on the right side of Figure 2. Here we cannot for instance determine if only peptides 23 and 40 are positive, or if all four peptides (23, 26, 37, 40) are. This uncertainty is what necessitates an extra stage of individual testing for these suspected positives. Our reason for using the EM method is to hopefully eliminate this extra step, saving time and resources.

While the matrix design has nice properties, it is very restrictive, both with regards to the relation between the number of pools and peptides, as well as only providing designs for using peptides in two pools. If we want to use all 90 wells available on a plate for a matrix design—which is the first design idea—we would be required to find $N = (90/2)^2 = 2025$ peptides to analyze, or $N = (90/3)^2 = 900$ if we were to introduce a third dimension equivalent to the rows and columns to make pools of. Needless to say, we want the number of peptides to be determined by the researchers, irrespective of design limitations. Therefore, we want to accommodate any number of peptides N. We do this by first replacing the $m \times m$ matrix with a similar $m \times s$ matrix for some $s \leq m$ determined by the number of peptides N through $s = N/m$. To allow for N which are not divisible by m, we will allow the last column of the new matrix to be incomplete. This leads to the following definition:

Definition 2. *In an ELISPOT assay with N peptides and M pools where each peptide is used in c pools, we define the* poolsize s *to be*

$$s := \lceil \frac{cN}{M} \rceil,$$

i.e. the total number of instances of all peptides divided by the number of pools, rounded up.

With this definition we can keep working with the idea of each pool having a size s. The only consequence is that some pools will have $s-1$ peptides instead.

Example 4. *Suppose we had $N = 9$ peptides we wanted to use $c = 3$ times and $M = 12$ pools total. By Definition 2 we would have the poolsize 3, since $s = \lceil 27/12 \rceil = 3$. The $m \times s$ matrix used in matrix design would then be*

$$\begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & \_ \\ 3 & 7 & \_ \\ 4 & 8 & \_ \end{bmatrix}.$$

Notably, the matrix is not square, and the last column only contains one peptide. Either of these mean that the matrix design will not work as-is. We could create the 4 row pools, which would be of poolsize 3. But we could only make 3 column pools, and they would be of either size 4 or 1. This matrix method needs to be generalized in order to accommodate situations like this. □

One way of looking at the type of design we seek is as a generalization of the matrix design. It already fulfills ideas 2 and 4, as well as idea 3 except for 2 pseudo-replicates instead of 3. In section 2.3.7 we will derive an explicit design fulfilling all four ideas, using a method similar to the matrix design by using the more generalized $m \times s$ matrix described above.

2.3.2 Overlap and Non-Overlapping Designs

Arguably the most difficult of the four design ideas to fulfill is the last one. In order to easier understand this idea, we define the 'overlap' between two peptides to be the number of times they share a pool together, *not counting the first pool in which they do.* More formally:

Definition 3. *For two peptides p_1 and p_2, let $C_{p_1,p_2} := \{P_i, i = 1, 2, \dots, M | \{p_1, p_2\} \subseteq P_i\}$ be the set of pools where p_1 and p_2 appear together. We define the* overlap *$v(p_1, p_2)$ between p_1 and p_2 to be*

$$v(p_1, p_2) := max\left(0, |C_{p_1,p_2}| - 1\right),$$

i.e. the number of pools beyond the first (if any) where the two peptides appear together. We then define the overlap for a particular peptide p_k to be

$$v_{p_k} := \sum_{j \neq k} v(p_k, p_j).$$

Let D be an ELISPOT design with N peptides $p_1, p_2, \dots, p_N$. The overlap V_D of the design D is defined as

$$V_D := \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} v(p_i, p_j).$$

This definition of overlap gives a word to describe what we are trying to do with our design—namely to *eliminate overlap*, or at least reduce it as much as possible. This definition also gives us a way to quantify how far from idea 4 a particular design is. A design D which has no overlap (i.e. $V_D = 0$) deserves it's own definition.

Definition 4. *A design D with N peptides and M pools is called a Non-Overlapping Design (NOD) if*

i) It has poolsize s (i.e. s or $s-1$ peptides in each pool).

ii) Each peptide appears in exactly $c \geq 2$ pools.

iii) $V_D = 0$.

Such a design may be referred to as a (N, M, c)-NOD.

It is clear from this definition that a $(N, 90, 3)$-NOD is a design which embodies the four design ideas.

Example 5. *An example of a NOD is the matrix design D_{matrix} in Figure 2. For the* j*:th peptide in the matrix, the row and column pools it belongs to share no other peptide. This means that p_j shares a pool with each other peptide at most once, and thus $v_{p_j} = 0$. Since this is true for all peptides in this design, $V_{D_{matrix}} = 0$, and we conclude that this particular matrix design is a $(49, 7, 2)$-NOD.* □

To investigate how to find an explicit $(N, 90, 3)$-NOD, we will first look at another type of design used in more general experiment design. Afterwards, we will study a very particular combinatorial problem, which will have striking similarities to our problem of finding a NOD.

2.3.3 Balanced Incomplete Block Designs

A well-studied type of experiment design is the *Balanced Incomplete Block Design* (BIBD) [10, 20]. It is a design which has some similarities to our NODs, which makes the BIBD a natural starting point for finding them.

Definition 5. *A* block design *is a design consisting of non-empty subsets*

$$P_1, P_2, \dots, P_M$$

of a non-empty domain

$$\{p_1, p_2, \dots, p_N\}.$$

These subsets are called blocks. A block design is called simple *if no two blocks are identical, and called* regular *if*

i) each block has the same number of elements $s \geq 2$, called the blocksize*, and*

ii) each element appears in the same number $c > 0$ of blocks.

In the ELISPOT setting, we use the terms pool and poolsize instead of block and blocksize. It seems the NOD bears a resemblance to a simple and regular block design, where in particular the regular attribute corresponds to ideas 2 and 3. The simple attribute is however too weak (in the sense of idea 4) for our purposes. A more restrictive class of designs is given by the following:

Definition 6. *A Balanced Incomplete Block Design (BIBD) is a simple and regular block design with n elements and blocksize s, where each pair of elements p_i and p_j is contained in exactly $\lambda > 0$ blocks. Such a design is also referred to as a (N, s, λ)-design.*

The reason the BIBD is called *balanced* is that each element is paired with every other element exactly λ times. It is called *incomplete* because each block does not contain all elements ($s < N$).

Example 6. *An example of a $(7, 3, 1)$-design—a BIBD with 7 elements p_j and blocks of size 3 where each pair of elements appear exactly once—consists of the following 7 blocks:*

$$\begin{array}{cccc} \{p_1, p_2, p_3\} & \{p_1, p_4, p_5\} & \{p_1, p_6, p_7\} & \{p_2, p_4, p_6\} \\ \{p_2, p_5, p_7\} & \{p_3, p_4, p_7\} & \{p_3, p_5, p_6\} & \end{array}$$

We can see that each element is in exactly 3 blocks, and that each pair is in exactly 1 block. □

Example 7. *The matrix design in Figure 2 is* not *a BIBD. While it might appear to be a $(49, 7, 1)$-design, it is not since each peptide pair does not appear* exactly *once. For instance, peptides 1 and 8 never appear together.* □

A necessary—and for us very relevant—condition for a BIBD is given by Fisher's inequality. It states that $M \geq N$ is a necessary condition for a BIBD to exist (see [20] for proof). For our purposes, this is devastating. We wish to restrict ourselves to $M = 90$ pools, but still test hundreds of peptides. Fisher's inequality thus tells us that we cannot find a BIBD which suits our needs. The strong restriction stems from the *balanced* part, i.e. that each element is paired with every other element an exact number of times. We need to look for something without this restriction. Before we do, we borrow a property from a certain type of BIBD, which we will find applicable in the next section.

Definition 7. *A BIBD is* resolvable *if the blocks are such that they can be partitioned into* c *disjoint classes where each element is in exactly one block per class. Such a partition is called a* resolution, *and the classes are called* parallel classes.

Example 8. *An example of resolvable BIBD is the following* $(9, 3, 1)$*-design—a BIBD with 9 elements* p_j *and blocks of size 3 where each pair of elements appear exactly once. It consists of the following 12 blocks:*

$$\begin{array}{c|c|c|c}
\{p_1, p_2, p_3\} & \{p_1, p_4, p_7\} & \{p_1, p_5, p_9\} & \{p_1, p_6, p_8\} \\
\{p_4, p_5, p_6\} & \{p_2, p_5, p_8\} & \{p_2, p_6, p_7\} & \{p_2, p_4, p_9\} \\
\{p_7, p_8, p_9\} & \{p_3, p_6, p_9\} & \{p_3, p_4, p_8\} & \{p_3, p_5, p_7\}
\end{array}$$

If we partition these blocks by column, we get four classes where every peptide appears exactly once in each class. This design is therefore resolvable into 4 classes. □

Example 9. *While not a BIBD, the matrix design in Figure 2 is resolvable. The row pools and the column pools each make up a class, meaning that the matrix design is resolvable into 2 classes.*
□

With the BIBD being too restrictive for application to the ELISPOT assay, we continue the search for a NOD, and find it in an unexpected application.

2.3.4 The Social Golfer Problem

The Social Golfer Problem (SGP) is a combinatorial problem which has attracted some interest in the field of computer science [8, 14, 24]. First posed in [11], the SGP asks:

> *A group of 32 social golfers get together each week to play in 8 smaller groups of 4. Every week, they make new groups such that everyone has new people to play with. How many weeks can they play before any two golfers have to play together for a second time?*

Since then, the problem has been generalized to m groups of s golfers, and has been changed to looking for a solution for a specific number of weeks w instead [24]. An instance of the SGP can then be specified by the triple (m, s, w).

(The solution to the original $(8, 4, w)$-SGP turns out to be $w = 10$. An explicit solution has been found [24] for $w = 10$, and pigeon-hole principle can be used to show that a solution does not exist for $w > 10$.)

We can see from the generalized description of the SGP that it bears resemblance to a resolvable $(ms, s, 1)$-BIBD (with w parallel classes), only without the requirement that each golfer plays together exactly once. Indeed, there is no guarantee that golfers 1 and 2 will play together at all for a given solution. For the SGP, the parallel classes are the weeks, since each golfer plays exactly once per week.

If we look at the ELISPOT assay with N peptides and M pools, where we wish to use each peptide in 3 pools, we can look at it as a $(M/3, s, 3)$-SGP, where the golfer group size s corresponds to the

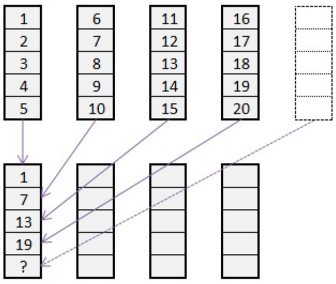


Figure 3: An illustration of the proof in Proposition 1 when $s = 5$ and $m = 4$.

poolsize s. For the NOD assay we do not require the resolvable property, but limiting ourselves to this subtype of NOD might make visualizing the solution easier. In practice, this means that we partition our $M = 90$ pools into $c = 3$ equal size sections (i.e. classes) with $m = M/c = 30$ pools in each, where each section contains each peptide exactly once. Each class then corresponds to a distinct partition of the N peptides. We can then try to create our NOD one class at a time by each time partitioning the N peptides, making sure each new class does not introduce any overlap with previous classes. This sequential construction of a NOD can be considered easier to visualize, and the NODs found in the rest of this thesis are all resolvable as a result of this (with the exception of Example 10 below).

2.3.5 A Necessary Condition for Resolvable NODs

Now that we have limited ourselves to searching for a resolvable NOD, we can turn to finding explicit solutions. First, we make the following statement about the relationship between the class size m and poolsize s which is necessary for a resolvable NOD to exist:

Proposition 1. *In an ELISPOT assay with N peptides and poolsize s, a necessary condition for a resolvable NOD to exist is $s \leq m$, where m is the number of pools per class.*

Proof. To see this, let us assume that $s > m$. We can create the first class arbitrarily, resulting in m pools with s peptides in each. In order to create the first pool in the second class without any overlap, we must pick s peptides such that no two were part of the same pool in the first class. This is equivalent to picking at most one peptide from each of the m pools from the first class. But since $m < s$, we will run out of pools to pick from before we get to s peptides. Thus, we cannot create a pool of size s for the second class without creating overlap with the first. Therefore, no resolvable NOD can exist when $s > m$. An illustration is provided in Figure 3 for the case $s = 5$ and $m = 4$. □

Proposition 1 tells us that the maximum number of peptides for which we can hope to find a *resolvable* NOD which fits a single plate and uses each peptide 3 times is one which has poolsize $s = \frac{90}{3} = 30$, which is $N = 30^2 = 900$. The question still remains whether or not such a design exists even when $N \leq 900$.

Example 10. *Note the emphasis in Proposition 1 on a* resolvable *NOD. The proposition does not necessarily hold for a NOD which is not resolvable. The following (6, 4, 2)-NOD is not resolvable and violates the proposition:*

$$\begin{array}{llll} P_1: & 1 & 2 & 3 \\ P_2: & 1 & 4 & 5 \\ P_3: & 2 & 4 & 6 \\ P_4: & 3 & 5 & 6 \end{array}$$

□

Next, we will present two methods for creating resolvable NODs—one using a random search algorithm, and the other creating the NOD directly in a similar fashion as the matrix designs do.

2.3.6 A Random Search Solution

Taking inspiration from the SGP solutions in [8, 24], we can create a random search algorithm for finding a resolvable NOD (viewed as an instance of the SGP).

For a resolvable design D, let $D[k, p_i \leftrightarrow p_j]$ be the resolvable design identical to D except in class k, where the positions of peptides p_i and p_j have been swapped. Starting from some initial (resolvable) design, the random search algorithm to transform this design into a resolvable NOD proceeds as follows:

```
Algorithm 2: Create a Resolvable NOD.
  Input: An initial (resolvable) design with 3 classes.
  Output: A resolvable NOD with 3 classes.
  $D \leftarrow$ initial design
  while $V(D) > 0$ do
      choose a $p_i$ for which $v_{p_i} > 0$
      find a $k$ and $p_j$ minimizing $V(D[k, p_i \leftrightarrow p_j])$
      if $V(D[k, p_i \leftrightarrow p_j]) \leq V(D)$ then
          $D \leftarrow D[k, p_i \leftrightarrow p_j]$
      end
  end
```

This algorithm is very simplified compared to the algorithm described in [8]. In Algorithm 2, we only consider one overlapping peptide at each iteration step. This saves us a considerable amount of time by not having to look for optimal swaps among all overlapping peptides. However, this will potentially require many more steps, since we might overlook better swaps. Additionally, both the 'tabu' [24] component (list of previously made swaps, to avoid repeating them back and forth) and the forced stopping time have been skipped, as we are confident that a solution will be found eventually. In practice, this algorithm has been tested to find a solution in single-plate ELISPOT settings for up to 900 peptides (which is the maximum covered by Proposition 1), and saves a very significant amount of execution time compared to the more thorough algorithms in [8, 24].

2.3.7 A Heuristic Solution

The next proposition requires some work to prove, but will explicitly provide a resolvable NOD with 3 classes. It can be seen as complementary to Proposition 1.

Proposition 2. *In an ELISPOT assay with N peptides and $M = cm$ pools of size s such that $s \leq m$, there exists a resolvable NOD with $c = 3$ parallel classes.*

Proof. We will prove this by explicitly constructing the 3 classes one-by-one. For illustration we will provide figures showing these schemes for some (small) assays. We begin by arranging the peptides in an $m \times s$ matrix, filling it in column-by-column as

$$\begin{bmatrix} 0 & m & 2m & \cdots & (s-1)m \\ 1 & m+1 & 2m+1 & \cdots & (s-1)m+1 \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ m-1 & 2m-1 & 3m-1 & \cdots & sm-1 \end{bmatrix},$$

where we have used the numbers $0, 1, \ldots, sm-1$ to represent the N peptides. If $N < sm$, the last $sm - N$ positions in the last column of the matrix will be empty.

The first class is created by taking the rows of this matrix, as in the old matrix design. This creates m pools of poolsize s as required. Since each row takes at most one peptide from the last column, they contain at most one missing element, and so there are no pools with less than $s-1$ peptides. An illustration of this for $(N, m, s) = (28, 6, 5)$ can be seen in Figure C1.

If we let $[a\ ,\ b]$ for $a = 0, 1, ..., m-1$ and $b = 0, 1, ..., s-1$ denote the element on the a:th row and b:th column of the matrix, then $P_{ij}^{(1)}$—the j:th element of the i:th pool $P_i^{(1)}$ in the first class—is

$$P_{ij}^{(1)} = [i\ ,\ j]\,,$$

for $i = 0, 1, \ldots, m-1$ and $j = 0, 1, \ldots, s-1$. Here we number the rows and columns from 0, which will make it easier to write the other two classes using modulo operations.

The second class is made by taking the m diagonals of the matrix. If we imagine a copy of the matrix directly underneath it, we take the i:th diagonal to be the diagonal starting from peptide i, and proceeding downwards and to the right until we reach the s:th peptide of that pool at the last column of the matrix. This creates m pools of poolsize s, and since each pool takes at most one peptide from the last column, they contain at most one missing element, and therefore have at least $s-1$ peptides, as required. An illustration of this for $(N, m, s) = (28, 6, 5)$ can be seen in Figure C2. Using matrix coordinates, $P_{ij}^{(2)}$ becomes

$$P_{ij}^{(2)} = [(i+j) \bmod m\ ,\ j]\,.$$

Since $m \geq s$, the fact that the diagonals move downwards one row at a time guarantees that each pool of the second class take at most one peptide from each row of the matrix. Therefore, there is no overlap between the first and second class.

For the third class, we need to make the distinction between odd and even poolsizes. We start with the cases where the poolsize is odd. To visualize the construction, we imagine four copies of the matrix arranged in a 2×2 configuration. We then create the $i : th$ pool of the third class by starting with peptide i, and taking the next peptide to be the one positioned one row down and two columns across. We proceed like this until the i:th pool has poolsize s. An illustration for $(N, m, s) = (28, 6, 5)$ can be seen in Figure C3. In matrix coordinates, $P_{ij}^{(3)}$ can be expressed as

$$P_{ij}^{(3)} = [(i+j) \bmod m\ ,\ 2j \bmod s]\,,$$

or alternatively

$$P_{ij}^{(3)} = \begin{cases} [(i+j) \bmod m\ ,\ 2j] & ,\quad j = 0, 1, \ldots, (s-1)/2, \\ [(i+j) \bmod m\ ,\ 2j - s] & ,\quad j = (s+1)/2, \ldots, s-1. \end{cases}$$

Since s is assumed to be odd, this scheme guarantees that each pool contains exactly one peptide from each column of the matrix by first traversing all even numbered columns $(0, 2, \ldots, s-1)$, followed by all odd numbered $(1, 3, \ldots, s-2)$. Therefore, each pool will have at most one missing element, which means the poolsize s will be correct.

What remains is to show that there is no overlap between the second and third class. When s is odd, we note that a given peptide $P_{ij}^{(3)}$ in the third class is on the same diagonal as

$$[(i+j-2j) \bmod m\ ,\ 2j-2j] = [(i-j) \bmod m\ ,\ 0] = P^{(2)}_{(i-j) \bmod m, 0}.$$

We then conclude that for any given pool $P_i^{(3)}$ in the third class,

$$P_{ij}^{(3)} \in P^{(2)}_{(i-j) \bmod m}, \quad j = 0, 1, \ldots, s-1.$$

Since $s \leq m$, no two peptides in $P_i^{(3)}$ shared a pool in the second class. Since this is true for any $i = 0, 1, \ldots, m-1$, we conclude that there is no overlap between the second and third class.

When the poolsize is even, the above scheme will not traverse all s columns. But with a simple adjustment we can guarantee that it will. When we reach the end of the matrix at column $s-2$, we add one step down and one step across. This extra move causes us to traverse the columns just as in the odd case. An illustration for $(N, m, s) = (40, 7, 6)$ can be seen in Figure C3. In coordinate notation $P_{ij}^{(3)}$ becomes

$$P_{ij}^{(3)} = \begin{cases} [(i+j) \bmod m \ , \ 2j] & , \quad j = 0, 1, \ldots, s/2-1, \\ [(i+j+1) \bmod m \ , \ 2j-s+1] & , \quad j = s/2, s/2+1, \ldots, s-1. \end{cases}$$

If we assume for a moment that $m > s$, each pool in class 3 takes at most one peptide from each row, meaning there is no overlap between the first and this third class. The strict inequality comes from skipping one row in the scheme, thus requiring at least $s+1$ rows to guarantee no overlap with the first class. A similar argument as for odd poolsizes shows that there is no overlap with the second class.

The only case left is when s is even and $s = m$, where the above construction would create overlap with the first class. When we have a square matrix the temptation is to use the columns of the matrix, as in the old matrix designs. The issue with that is that all the missing elements are in the last column. If we want to use the columns, we therefore need to rearrange the missing elements of the matrix. Remembering that the scheme for the third class guarantees that each pool contains at most one peptide from each row, column, and diagonal, we can use that scheme to place the missing elements. If we have k missing elements ($k < s$) in the matrix, we place them in positions

$$\begin{cases} [(1+j) \bmod m \ , \ 2j] & , \quad j = 0, 1, \ldots, s/2-1, \\ [(1+j+1) \bmod m \ , \ 2j-s+1] & , \quad j = s/2, s/2+1, \ldots, k-1, \end{cases}$$

meaning we put all missing elements in what would have been $P_1^{(3)}$ if $s < m$. This placement then guarantees that each row, column, and diagonal contains at most one empty element, and we can take the columns as the third class instead. Since the first and second class by design takes at most one peptide from each column, there is no overlap in this design. We have now found a resolvable NOD with 3 classes for every case where $s \leq m$, which concludes the proof. □

Propositions 1 and 2 together tell us that a resolvable NOD exists in 3 classes if and only if $s \leq m$. When proving Proposition 2 for $s = m$, we actually stumbled upon the following corollary:

Corollary 1. *In an ELISPOT assay with n peptides and $M = cm$ pools of odd poolsize s such that $N = m^2 = s^2$, there exists a resolvable NOD with $c = 4$ parallel classes.*

Proof. When there are no missing elements and the poolsize is odd, the four classes are given by the rows, columns, diagonals, and the "one step down, two across" pools. From the reasoning in the proof for Proposition 2, there is no overlap between these four classes. □

2.4 A New Positivity Criterion

We now shift our focus from the design aspects of ELISPOT assays, and turn to statistical analysis. Prior to the EM method for directly estimating the peptide response rates, the identification of responding peptides came down to identifying which individual wells generated a positive response. This would be the case whenever one tested peptides individually, or when testing peptide pools (as in matrix designs, for instance). Even in the advent of the EM alternative, this form of analysis can still be useful. At the time of this writing there is no consensus or 'gold standard' [17] for testing the positivity of ELISPOT wells.

The positivity criteria currently in use consist of rules of thumb such as *'at least 50 spots'*, or *'two times the background'*; and statistical tests such as the t-test or permutation based tests [18][17]. The first type of criteria has a questionable arbitrariness and has little appeal to a statistician, and the statistical tests listed all rely on either identical replicates of each well, or large sample approximations. Identical replication means that the same peptide composition (either a pool or an individual peptide) must be used in multiple wells for the tests to work. While identical replicates are a good idea in general, in the use of pooling designs in particular—which is the focus of this thesis—one would prefer to mix up the pools instead of doing identical replicates. For instance, when one mixes the pools using a NOD, each peptide is still used multiple times, only not together with the same peptide partners more than once. In a sense this is replication, but it is not *identical* replication.

2.4.1 The Bayesian Predictive ELISPOT Criterion

Motivated by mixed pool designs [12, 16, 21]—especially the theory presented in this thesis up to this point—we wish to introduce a new positivity criterion which does not rely on this identical replication. We will call it the Bayesian Predictive ELISPOT Criterion (BPEC). It has been made to combine the nice properties of the Poisson-distribution assumed in Assumption 1 together with Bayes' theorem to create an exact test statistic.

The main idea behind the BPEC is to use the negative control wells to predict the spot counts for the peptide stimulated wells, which under the loosely formulated null hypotheses H_0^i : *the i:th well is negative*, will have some known distribution. If our observed spot count for any given well (containing one or more peptides) is considered extreme for this distribution, we reject the null hypothesis for that well. With the parametric framework we have set up, an alternative way to state the null hypothesis for the i:th well is

$$H_0^i : \sum_{j|p_j \in P_i} \beta_j = 0,$$

which is tested against the one-sided alternative

$$H_a^i : \sum_{j|p_j \in P_i} \beta_j > 0.$$

Note in particular that the alternative hypothesis does not tell us which of the parameters are positive, and that the one-sidedness stems from Assumption 1, where $\beta_j \geq 0$ for all $j = 0, 1, \ldots n$.

If we adopt a Bayesian approach, and make the distributional assumptions of spot counts as in Assumption 1, we can find that this predictive distribution only depends on the observed negative controls. If we let $\beta_0 > 0$ be some unknown spot count rate which we consider as a random variable, and a priori assume it to have a $Gamma(a, b)$ distribution for some $a > 0, b > 0$. If we then make the assumption that the spot counts for the negative controls, $X_1, X_2, \ldots, X_n$, are iid $Poisson(\beta_0)$-distributed given the value of β_0, Bayes' theorem can give us the *posterior predictive distribution* under $H_0^1, H_0^2, \ldots, H_0^M$ respectively for each peptide stimulated well $Y_1, Y_2, \ldots, Y_M$, given $X_1, X_2, \ldots, X_n$. The following proposition gives us this distribution in closed form:

Proposition 3. *Let $X_1, X_2, \ldots, X_n, Y$ be independent, identically distributed random variables having a Poisson(β_0) distribution, given some $\beta_0 > 0$. If we assume a priori that $\beta_0 \sim Gamma(a, b)$ for some specified $a, b > 0$, then the posterior distribution of β_0, and the posterior predictive distribution of Y given $\mathbf{X} = \mathbf{x}$ is given by*

i) $\beta_0|\mathbf{x} \sim Gamma(a + \sum_i^n x_i, b + n)$,

ii) $Y|\mathbf{x} \sim NegBin(a + \sum_i^n x_i, 1/(b + n + 1))$.

Proof: See Appendix B.

Since in practice we are interested in testing each peptide stimulated well (individually), and since all those wells have the same distribution under their respective null hypothesis, we can use the appropriate quantile of the $NegBin\left(a + \sum_{i=1}^{n} x_i, 1/(b+n+1)\right)$-distribution to create a 'detection limit'—a spot count value above which any well is considered positive. This gives us an intuitive and easy-to-use interpretation, which is also shared by the current ad hoc criteria.

2.4.2 The BPEC with Identical Replicates

We now have a criterion that by design works without identical replicates; but what statistic would we use if we did have identical replicates? One possibility is to use the sum of these replicates, which also has a posterior predictive distribution which is negative binomial, as the following proposition shows:

Proposition 4. *Let $Y_1, Y_2, \ldots, Y_k$ be iid random variables having a $NegBin(r, p)$ distribution. Then $Y_1 + Y_2 + \ldots + Y_k \sim NegBin(kr, p)$.*

Proof. An easy way to prove this is to use interpretation of a $NegBin(r, p)$ random variable as the sum of r iid *Geometric*(p) random variables. In our present situation this means that

$$\begin{aligned} Y_1 &= X_1 + X_2 + \ldots + X_r \\ Y_2 &= X_{r+1} + X_{r+2} + \ldots + X_{2r} \\ &\vdots \\ Y_k &= X_{(k-1)r+1} + X_{(k-1)r+2} + \ldots + X_{kr} \end{aligned}$$

where $X_1, X_2, \ldots, X_{kr}$ are iid, having *Geometric*(p) distributions. It then follows that the sum $Y_1 + Y_2 + \ldots + Y_k$ is the sum of kr such random variables, meaning it has a $NegBin(kr, p)$ distribution, as was to be shown. □

So in the case of k identical replicates, we can use the sum of these replicates, and compare it to the appropriate quantile $q_{1-\alpha}$ of this sum's posterior predictive distribution, for some significance level α. Equivalently, one can compare the mean of these replicates to $q_{1-\alpha}/k$.

2.4.3 Choice of Prior

By introducing Bayesian methodology we also introduce the matter of choosing a prior distribution for β_0. As seen in Proposition 2, using the *Gamma* distribution family results in a simple closed form for the posterior predictive distribution, and it gives a decent amount of flexibility in prior choice. What remains then is to choose the *Gamma* distribution's two parameters a and b.

If we are timid about bringing prior information about the background into our criterion, we can create a vague or uninformative prior by choosing small values for a and b. However, using a vague prior like this could be a missed opportunity, since we might have at least some sense about what the negative controls could look like. For example, is the average background more likely to be less than 100 spots, or more than 100 spots? An uninformative prior does not utilize even such basic intuition.

To study the effect our choice of prior distribution for β_0 has on the detection limit, we will compare four priors which are applicable to ELISPOT assays. *Prior 1* is a *Gamma*$(0.001, 0.001)$ distribution, and can be considered a vague prior. This is seen by looking at the mean of the posterior distribution, $E[\beta_0|\mathbf{x}] = \frac{a+\sum x_i}{b+n}$, which is approximately the sample mean of $\mathbf{x}$ when a and b are small. *Prior 2, Prior 3, Prior 4* are *Gamma*-distributions with parameters $(2, 0.25), (1, 0.125), (2, 0.125)$ respectively. These four priors can be seen together in Figure 4. One can note in particular that

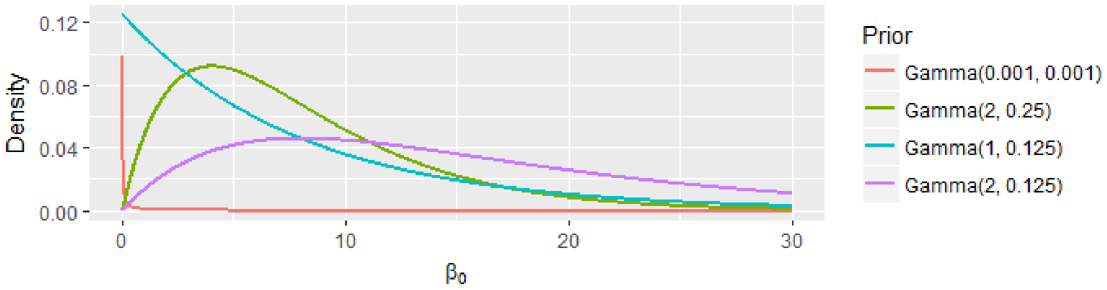


Figure 4: Illustration of the four Gamma-distributed priors used in this thesis. Notably they assign different probabilities to high and low values of β_0. The first prior is considered *uninformative*.

they give different probabilities both for very low ($\beta_0 < 2$) and very high ($\beta_0 > 20$) background rates.

2.4.4 Multiple Testing

In the application to ELISPOT assays using the proposed criterion, each peptide stimulated well has its own null hypothesis, which means up to 90 similar tests at once on a single plate. Because of this, the question of multiple test correction inevitably comes up. We therefore wish for our criterion to allow this type of correction without giving up too much statistical power.

The type of multiple test correction we will consider in this thesis is the Bonferroni correction [9]. Instead of testing the M null hypotheses at significance level α, the Bonferroni correction rejects each null hypothesis at significance level $\alpha^* = \alpha/M$ instead. This should put the *overall* significance level closer to α. With $M = 90$ tests and standard significance level $\alpha = 0.05$, the Bonferroni corrected significance level is $\alpha^* \approx 0.0006$.

Other types of correction exist—such as Holm-Bonferroni [13], and Benjamini-Hochberg [3] corrections—and should still be compatible with the proposed criterion.

2.4.5 Alternative Criteria

In order to assess the viability of the BPEC, we will compare the performance of out criterion to three previously used tests/criteria—either in ELISPOT analysis, or in wider applications. Assuming only one replicate and three negative controls, these criteria are:

- **An ad hoc criterion**. These typically come as a combination of two statements of the form:
  - *At least N spots*. Usually relative to the concentration of the blood sample, e.g. '50 spots per million cells.'
  - *At least K times the background*, where the background is estimated as the mean of the negatives controls. Typically K is 2 or 3.

  For these comparisons we will use the criterion *at least 50 spots per million cells and 2 times the background* (whichever is the highest).

- **The student t-test.** More specifically, the two-sample t-test with pooled variance. With one peptide stimulated well y being one of the samples, and three controls $\mathbf{x} = x_1, x_2, x_3$ being the other, the test statistic becomes

$$T = \frac{y - \bar{\mathbf{x}}}{\sqrt{\frac{0+\sum_{i=1}^{n}(x_i-\bar{\mathbf{x}})^2}{1+3-2}\left[\frac{1}{1}+\frac{1}{3}\right]}} = \frac{y - \bar{\mathbf{x}}}{\sqrt{\frac{2}{3}\sum_{i=1}^{3}(x_i - \bar{\mathbf{x}})^2}}.$$

Under the null hypothesis, T is approximately t-distributed with 2 degrees of freedom. This statistic can be inverted to 'detection limit' form

$$y \geq \bar{\mathbf{x}} + t_{1-\alpha}(2)\sqrt{\frac{2}{3}\sum_{i=1}^{3}(x_i - \bar{\mathbf{x}})^2}$$

at risk level α, where $t_{1-\alpha}(2)$ is the $(1-\alpha)$-quantile of the $t(2)$-distribution.

- **The binomial test**. This test is an exact test which can be used more broadly, but the motivation for it becomes simple under Assumption 1. Using Lemma 2 again—with X_1, X_2, X_3, Y being iid *Poisson*-variables under the null hypothesis—the test statistic $B = Y|\left(Y + \sum_{i=1}^{3} X_i\right)$ is *Binomial* $\left(y + \sum_{i=1}^{3} x_i, \frac{1}{4}\right)$-distributed under the null hypothesis. Note that this test cannot be inverted to yield a detection limit in the way that the other criteria can, since this binomial distribution depends on Y.

To our knowledge, these are all the criteria currently used by clinicians, except for the one described in [17] using permutation methods. This permutation based criterion has been excluded in this thesis because it explicitly requires identical replicates.

2.5 The Filtered EM Method for ELISPOT

The EM estimation procedure described in Algorithm 1 was employed when we need to estimate a couple of hundred parameters. Trying to attribute the observed spot counts correctly to each parameter is not a small task, especially with relatively few observations. The large number of parameters also results in a large design matrix $\mathbf{X}$, which requires more time to perform the necessary matrix operations in the algorithm. This added execution time is particularly compounded when generating parametric bootstrap confidence intervals for the parameters, which is the method used in this thesis.

To help speed up the EM estimation procedure, we can try to eliminate some of the β-parameters before the actual estimation. We propose to do this by first using the new BPC to test and determine which pools are negative. Let $Q = \{Q_1, Q_2, \ldots, Q_k\}$ be the subset of pools $\{P_1, P_2, \ldots, P_M\}$ with negative test results when subjected to the BPEC (or some other positivity criterion). Any peptide p_j found in $\cup_{i=1}^{k} Q_i$ is eliminated, or equivalently its rate β_j is fixed to 0. This also means that the pools $Q_1, \ldots, Q_k$ are assumed to contain only the background (expectation β_0). Note that this way of 'filtering out' peptides is in principle the same as the one used in the first stage of the old matrix designs (before re-testing individual peptides). The remaining parameter vector β^* of size $n \leq N$ is then used together with the $93 \times n$ submatrix $\mathbf{X}^*$ of the design matrix $\mathbf{X}$ consisting of the columns corresponding to β^*, to perform the EM estimation as in Algorithm 1. This new procedure—which we will call the *Filtered EM Method for ELISPOT* (FEMME)—is the last contribution of this thesis to the design and analysis of ELISPOT assays.

2.6 Web Application

All the tools presented in this thesis have been made available through a user-friendly web application written using the Shiny package in **R**. It is available at:

https://rickardstrandberg.shinyapps.io/elispotapp/

The features include creating a NOD in three classes from user input of number of peptides and wells; estimation of peptide response rates from data uploaded by the user using either the FEMME or the standard EM algorithm in [21]; and detection of positive wells using either the BPEC or a user-specified ad hoc criterion. Example files are also available to demonstrate these features.

3 Results

3.1 Simulating ELISPOT Assays

In order to benchmark the methods presented in this thesis, we were given access to data from an ELISPOT assay performed by Nathifa Moyo, at the Jenner Institute, University of Oxford. The assay tested an HIV vaccine using splenocytes from mice. A total of 203 peptides were tested individually with 2 replicates for each, requiring 5 plates. Out of these peptides, 22 were considered positive by Moyo N. (according to an ad hoc criterion). Using the BPEC with prior 1 and Bonferroni correction (see Section 2.4.3–2.4.4), we found 23 positive peptides. The average background across the 5 plates was ≈ 20 spots. Anecdotal observations and private communications with Moyo N. indicate that both the background and number of positive peptides was unusually high—but still reasonable—compared to most other ELISPOT assays.

We use these positive peptide rates in order to simulate other ELISPOT assays. We do this by first (roughly) fitting a density to these observed rates, as can be seen in Figure 5. When simulating an assay, we randomly assign a number of the peptides as positive (e.g. 23 out of 203), and sample a rate from this density for each of these peptides. The rest are considered to have a rate of 0, and we can create the parameter vector β by choosing an appropriate background rate β_0 (e.g. 20). Depending on the design matrix $\mathbf{X}$ used, we calculate the total rate of each well through (1), and create the simulated spot counts by sampling from the corresponding Poisson-distribution.

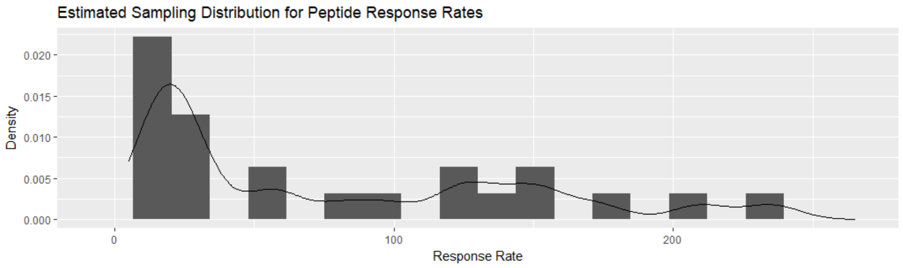


Figure 5: Estimated sampling density of (positive) peptide response rates. Based on 23 positive peptides estimated from individual assays performed on mice.

3.2 FEMME Compared to EM Estimation

In Figure 6 we have compared the convergence of the FEMME to that of the regular EM method from [21] (Algorithm 1). In this peculiar figure we generated three single-plate assays with 203 peptides (23 positive) and $\beta_0 = 20$, using as design the NOD from the proof of Proposition 2. Each plate was estimated using both the regular EM procedure in Algorithm 1 and the new FEMME. The estimations were carried out 100 times, with random starting values each time. In Figure 6 we see the 'standard deviations' of the 100 estimates of each parameter, where any variation is due to the choice of starting values. This standard deviation can therefore be seen as the estimates' sensitivity to the starting values, where zero indicates that the estimate is independent of the starting values. We can see that the FEMME produces more consistent estimates (all standard deviations are are zero) than the regular EM method, whose estimates varies by several spots for some of the parameters.

Also notable is the relative speed of the FEMME. Compared to the regular EM estimation in [21]—which took 29 minutes to perform 3 simulations with 100 sets of starting values each—the FEMME executed these simulations in 4.2 minutes, which is approximately 15% as long. For instance, this means that a significant amount of time can be saved when producing parametric bootstrap samples for these estimates (here we can approximate the execution time for 300 bootstrap samples with the time recorded for these simulations).

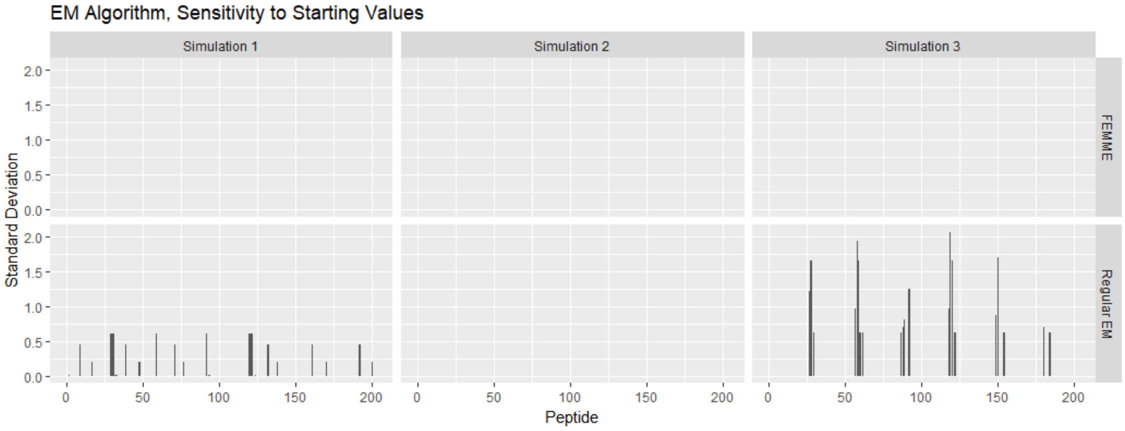


Figure 6: Plot of the standard deviations of parameter estimates when their starting values are varied. This can be seen as a measure of each method's sensitivity to the choice of starting values.

3.3 The BPEC versus Alternatives

In Table 1 we can see the effects that the choice of prior and Bonferroni correction (assuming 90 tests) has on the detection limit for the BPEC (see section 2.4.1), for various levels of estimated background $\hat{\beta}_0$ (estimated from 3 negative controls). We see that we get a slight difference between our four priors, with the ones favoring higher background rates giving higher detection limits. A much greater effect can be seen when using Bonferroni correction, with significant increase to the detection limit (from 2 to 6; from 29 to 39; etc.). With 90 tests, the significance level is adjusted from 5% to 0.06%, so the difference should not come as a surprise.

	No Correction				Bonferroni Correction (90 tests)			
$\hat{\beta}_0$	Prior 1	Prior 2	Prior 3	Prior 4	Prior 1	Prior 2	Prior 3	Prior 4
0	0	2	2	2	0	6	5	6
1	3	4	4	4	7	9	8	9
5	10	10	10	10	16	16	16	17
10	16	16	16	17	24	24	24	25
20	29	28	28	29	39	38	39	39

Table 1: Summary of the effect that prior selection and Bonferroni correction has on the detection limit of the BPEC for different levels of background $\hat{\beta}_0$. We can see a slight change caused by prior choice, and a much larger effect when introducing Bonferroni correction.

In Figure 7 we have compared the statistical power of the various positivity criteria mentioned in this thesis, for increasing (true) peptide response rates β_1. For each value of β_1 we simulated 10000 sets of three negative controls with background β_0 and one peptide stimulate well with rate $\beta_0 + \beta_1$, and applied each criterion. What we see in Figure 7 is the fraction of these 10000 tests which turn out positive. We can see that the BPEC is better at identifying peptides with low response rates. The alternative criterion which comes the closest is the binomial test.

Focusing more on the BPEC, we see that the differences between Prior 1 and Prior 2 is also quite pronounced considering that the difference between their detection limits is only a single spot (Table 1). We can also see that the Bonferroni corrected BPEC (here using prior 1) is close to the ad hoc criterion. Since Bonferroni corrected tests are considered conservative, the same can therefore in a sense be said about the ad hoc criterion.

Particularly notable in Figure 7 is the 'power' at $\beta_1 = 0$, which is the case where there is no actual response (the null hypothesis is true). The value at this point then represents the rate of Type I errors (false positives) for each criterion. The values are too difficult to read from the figure, but they are:

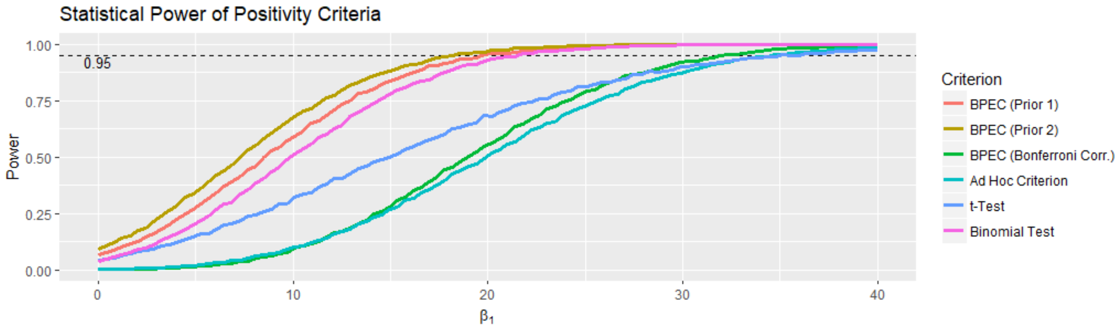


Figure 7: The estimated powers of various positivity criteria, plotted as functions of a hypothetical peptide response rate β_1. Estimated using the background rate $\beta_0 = 20$.

	BPEC (Prior 1)	BPEC (Prior 2)	BPEC (Bf. Corr.)	Ad Hoc Criterion	Student t-Test	Binomial Test
Type I	0.065100	0.089000	0.000700	0.001500	0.043217	0.036300

We see that the non-corrected BPEC, and the binomial test, have error rates relatively close to the significance level $\alpha = 0.05$. Note that the values are not exactly α since we use discrete distributions for these criteria. In this case greater power also means more false positives. On the other hand, the Bonferroni corrected BPEC and the ad hoc criterion have a remarkably low type I error rate, but significantly less power for low response rates. The t-test seems to fall somewhere in-between the rest of the criteria, not excelling in either having low Type I error rate, or high power for low responses.

3.4 Design Comparison

To compare the differences between ELISPOT designs in terms of being able to correctly identify positive (and negative) peptides, we simulate a large number of assays for each of the following three designs:

D_{NOD}: NOD created according to the proof of Proposition 2.

D_{rand}: Design with peptides assigned to pools randomly, with a new design generated for each sample. The overlap for each generated design was $30 \leq V(D_{rand}) \leq 74$.

D_{rNOD}: NOD created by using Algorithm 2, starting from a random design. This NOD is significantly different from D_{NOD}, which is completely deterministic. We wish to see if this more 'randomized' version of the NOD will give different results compared to D_{NOD}.

For each design, 200 ELISPOT assays were simulated. The FEMME was used to estimate the peptide rates and used to perform parametric bootstrap tests for each estimate (sample size 500) for different significance levels α ranging from 0 to 1. We determined a peptide to be positive for a given α if the lower α-quantile of the samples was greater than one spot. Using these tests, we calculated the average *sensitivity* (true positive rate) and *specificity* (true negative rate) over all 200 assays for each α. Since the data is simulated, we know the exact values of both for each assay.

We start by looking at the average (over the 200 simulated assays) sensitivities and specificities for significance level $\alpha = 0.05$. In Table 2 we have compared the differences between the three designs with respect to these two values, where we have also performed t-tests for testing these differences. Beginning with the sensitivities, there is a statistically significant difference between D_{rNOD} and the other two designs, which are in turn not significantly different. This difference is reversed when we look at the specificities, with D_{rNOD}'s being significantly lower than those of the others. This time, we also see a difference between D_{NOD} and D_{rand}, in favor of D_{rand}.

Sensitivity

D_{NOD}	D_{rand}	D_{rNOD}	diff.	p-value
0.7979	0.8011		-0.0032	0.6813
0.7979		0.8584	-0.0605	1.11e-14
	0.8011	0.8584	-0.0573	2.565e-15

Specificity

D_{NOD}	D_{rand}	D_{rNOD}	diff.	p-value
0.9860	0.9907		-0.0047	1.095e-06
0.9860		0.9780	0.0080	1.399e-11
	0.9907	0.9780	0.0127	2.2e-16

Table 2: A comparison of the average sensitivities and specificities for the different designs for $\alpha = 0.05$.

In the simulated setting, where we have 23 true positives and 180 true negatives, a sensitivity difference of $1/23 = 0.043$ represents a difference of one more true positive; and a specificity difference of $1/180 = 0.0056$ means one more true negative. From Table 2 we can then see that D_{rNOD} on average finds 1-1.5 more true positives than the other two, but at the expense of eliminating 1-2 fewer negative peptides on average.

Comparing the sensitivities and the specificities separately does not give an accurate picture, since the two are closely correlated. If we control our tests to have a high true positive rate (sens.), the false positive rate (1-spec.) also increases. In Figure 8 we have created a Receiver Operating Characteristic (ROC) curve, which plots for each design the sensitivity against 1 minus the specificity. Here we control these rates using the significance level α of our parametric bootstrap tests. We have also marked the sensitivity/specificity for $\alpha = 0.05$ from Table 2. We can see that the curve for D_{NOD} falls entirely below that of D_{rand}, which indicates that randomized designs outperform the NOD from Proposition 2. It is more difficult to compare D_{rNOD} to D_{rand}, however, since their curves do not span all the same specificity values. But if we were to demand a sensitivity of at least 0.85, we can based on Figure 8 say that D_{rNOD} outperforms even D_{rand}.

We can notice that the curves in Figure 8 do not go from $(0, 0)$ to $(1, 1)$ as one might expect from a ROC curve. This is due to the nature of the bootstrap samples giving us limited control over the curves. For instance, if all the entire bootstrap sample of a given parameter falls above 1 spot, that parameter will be considered positive for all α, and the sensitivity will in that case never be 0 for any $0 \leq \alpha \leq 1$.

As a last comparison between the three designs, we use the mice data from the Jenner Institute (unpublished results)—where 203 peptides were estimated *individually*—to simulate three single-plate *pooled* assays, one for each of the three designs. We then used each of these assays to re-estimate the peptide rates, and compared them to the estimates from the original (individual) assay. The results can be found in Figure 9. Here, the light gray bars indicate the positions of the 23 positive peptides (as determined by the individual assays); the dark gray bars their value; and the colored dots the FEMME estimates for each of the three designs. While the particular estimates vary between designs, we can see that the peptides with the largest responses in the individual assays also receive the largest estimates from the FEMME. We can also see that all three designs seem to produce at least some false positives. These appear to be different for each design, which seems to indicate that they are a result of that particular design not being able to distinguish them as negative. Based on Figure 9 alone, the differences between the three designs are difficult to distinguish, but overall each of them produces good estimates.

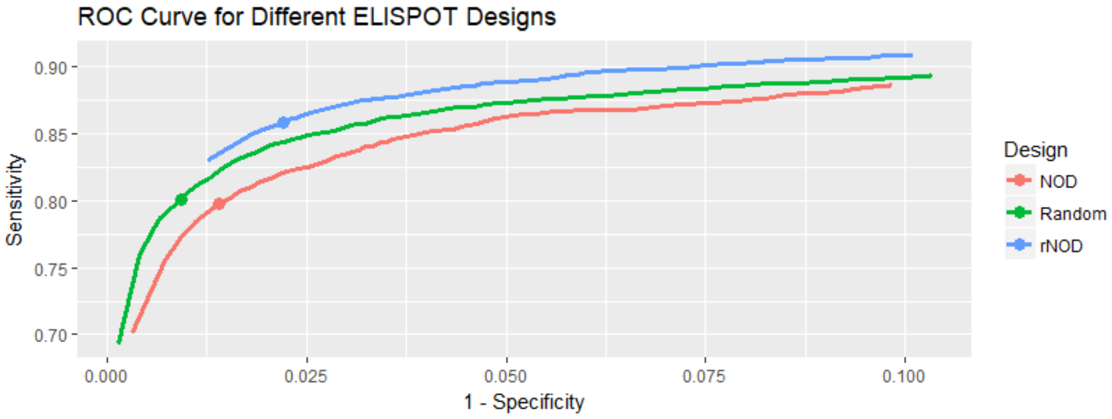


Figure 8: ROC curve plotting the true negative rate (specificity) against the true positive rate (sensitivity) for the three designs. The curve is controlled by the significance level α of the parametric bootstrap intervals. The points indicate $\alpha = 0.05$.

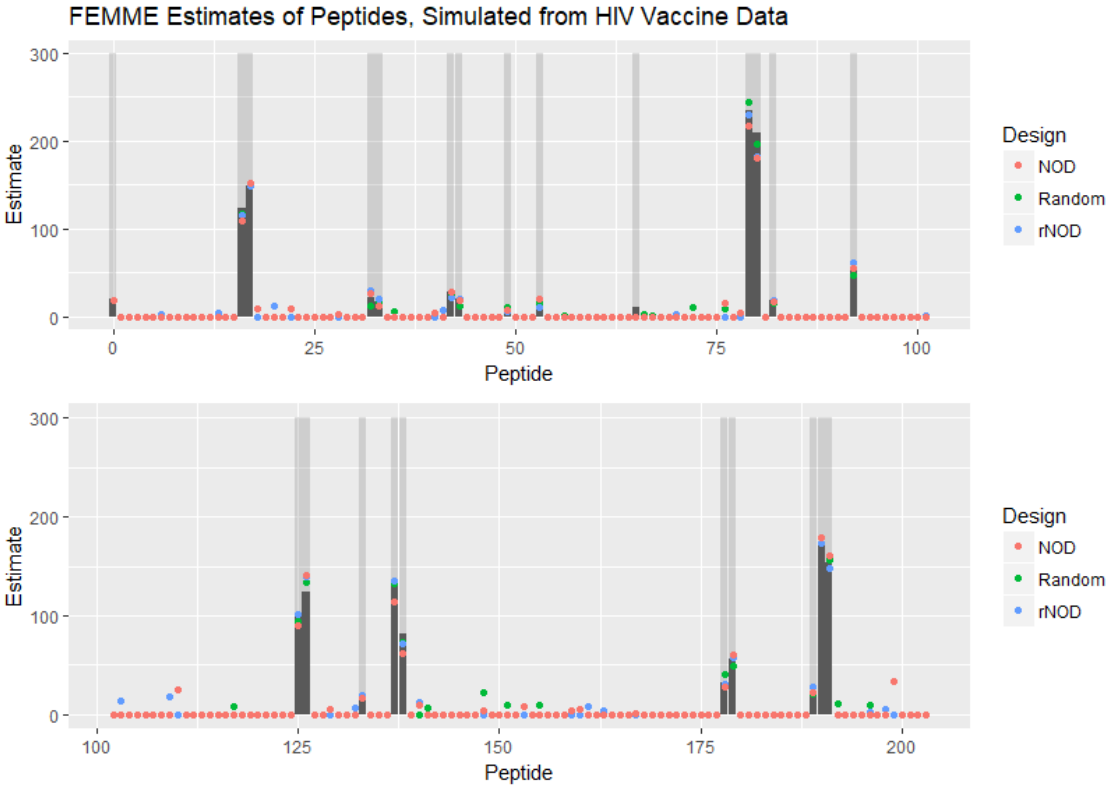


Figure 9: Peptide estimates using the different designs. The light gray bars indicate the true responses; the dark gray bars represent the true response rates; and the colored dots are the estimates for the different designs.

4 Discussion

In this thesis we have presented three new contributions to pooled ELISPOT design and analysis: the NOD to design pooled assays to more easily distinguish responding peptides from non-responding ones; the BPEC to separate wells containing responding peptides from just background; and the FEMME to estimate peptide response rates directly from the pooled assay. These can be used separately based on their own merits, but together they form a synergy that allows for considerable savings in both time and resources. In this thesis we showed that an assay testing 203 peptides individually on five plates could be reduced to a single plate, with the peptide rates still estimated with good agreement to the ones from the five plate assay.

We found the BPEC to have high statistical power compared to the established ELISPOT criteria (ad hoc and t-test), especially when the underlying response rates are low. We also saw that the Bonferroni-corrected BPEC for 90 tests was remarkably similar to the ad hoc criterion *at least 50 spots, and three times the background.* We can interpret this to mean that the ad hoc criterion is a conservative criterion, since the Bonferroni correction is known to make tests conservative.

By taking a cue from the old matrix designs and adding the step of filtering out the peptides which are members of negative pools (e.g. by the BPEC criterion), the FEMME was able to build on the previous EM estimation method of [21], managing to significantly speed up the estimation and produce more reliable estimates, specifically by reducing the number of parameters. It is clear that the criterion used for the filtration will influence the final estimates. Choosing the criterion and significance level alone makes for countless variations on the FEMME. It would therefore be of interest to look more closely at which variants could result in even better estimates.

The linchpin of both the BPEC and the FEMME is Assumption 1, which assumes both Poisson-distributed and independent spot counts. The first, while computationally convenient, as it has been suggested that some overdispersion relative to the Poisson distribution might exist [7]. Modifying the BPEC and the FEMME to work under the more general negative binomial distribution instead of a Poisson could be an avenue for potential improvement. The consequence of using the BPEC with Poisson distributions in that case is that we underestimate the right tail of the posterior predictive distribution, and therefore set the detection limit too low. If used on its own, this potentially means more false positives; and if used as part of the FEMME, it means filtering out fewer negative peptides before the estimation.

We saw in Figure 8 that not all NODs are equal in terms of performance. The fact that the random search NOD created from Algorithm 2 outperformed the completely random design, but the NOD from Proposition 2 did not seems to indicate that there is more to finding a good ELISPOT design than just eliminating overlap (as we have defined it). In this thesis we used the concept of overlap to quantify a type of 'mixing' of the pools that would help the EM algorithm find positive peptides.

Eliminating overlap does seem to improve the performance, however, which we can surmise if we look at the random search NODs as a subtype of random designs. It would appear from our results that this subtype does perform better than random designs in general. So we hypothesize that no overlap, together with some vague concept of 'randomness' could better characterize a good ELISPOT design.

For now, we can recommend the random search NOD as the best design we have found so far. But in order to make such designs accessible to clinicians and researchers, we would need to improve the speed of Algorithm 2, as the current iteration of it is quite slow. Alternatively, one can search for a new measure of a good ELISPOT design other than overlap, and try to optimize over that measure. This measure would ideally include overlap and randomness as aspects which would affect this measure.

One restriction in the two methods we have presented for finding a NOD is that it only finds *resolvable* designs, which according to Proposition 1 will limit us to 900 peptides per plate. As seen in Example 10, non-resolvable NODs can exist even though $m < s$. While finding such designs might be a worthwhile pursuit, those might be stretching the limit of pooled designs in general, as the poolsize for a $(N, 30, 3)$-NOD would be $s > 30$. Besides, in most applications [5, 15, 19] 900 peptides will be more than enough.

In conclusion, we have taken several steps towards efficiently designing and analyzing ELISPOT assays. Reducing the assay to a single plate and performing everything in a single step presents a tremendous opportunity for clinicians to save time and resources. We have also identified multiple potential areas where these methods can be improved further.

References

[1] D. D. Anthony and P. V. Lehmann. T-cell epitope mapping using the {ELISPOT} approach. *Methods*, 29(3):260 – 269, 2003. T Cell Epitope Mapping.

[2] T. Beißbarth, J. A. Tye-Din, G. K. Smyth, T. P. Speed, and R. P. Anderson. A systematic approach for comprehensive t-cell epitope discovery using peptide libraries. *Bioinformatics*, 21(suppl1):i29, 2005.

[3] Y. Benjamini and Y. Hochberg. On the adaptive control of the false discovery rate in multiple testing with independent statistics. *Journal of Educational and Behavioral Statistics*, 25(1):60–83, 2000.

[4] T. Berger, J. W. Mandell, and P. Subrahmanya. Maximally efficient two-stage screening. *Biometrics*, 56(3):833–840, 2000.

[5] N. I. Cardona, D. M. Moncada, and J. E. Gómez-Marin. A rational approach to select immunogenic peptides that induce ifn-γ response against toxoplasma gondii in human leukocytes. *Immunobiology*, 220(12):1337 – 1342, 2015.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. 1976.

[7] M. Dittrich and P. V. Lehmann. *Statistical Analysis of ELISPOT Assays*, pages 173–183. Humana Press, Totowa, NJ, 2012.

[8] I. Dotú and P. Van Hentenryck. Scheduling social tournaments locally. *AI Communications*, 20(3):151 – 162, 2007.

[9] O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

[10] J. L. Gross. *Combinatorial Methods with Computer Applications*, pages 541–579. Chapman & Hall/CRC,. Chapman & Hall/CRC,, Boca Raton, FL :, 2007.

[11] W. Harvey. CSPLib problem 010: Social golfers problem. http://www.csplib.org/Problems/prob010, 1999.

[12] A. L. Heffernan, L. L. Aylward, L.-M. L. Toms, P. D. Sly, M. Macleod, and J. F. Mueller. Pooled biological specimens for human biomonitoring of environmental chemicals: Opportunities and limitations. *Journal of Exposure Science & Environmental Epidemiology*, 24(3):225 – 232, 2014.

[13] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[14] F. Lardeux, E. Monfroy, B. Crawford, and R. Soto. Set constraint model and automated encoding into SAT: application to the social golfer problem. *Annals of Operations Research*, 235(1):423–452, 2015.

[15] L. Y.-H. Lee, C. Simmons, M. D. de Jong, N. V. V. Chau, R. Schumacher, Y. C. Peng, A. J. McMichael, J. J. Farrar, G. L. Smith, A. R. Townsend, et al. Memory t cells established by seasonal human influenza a infection cross-react with avian influenza a (h5n1) in healthy individuals. *The Journal of clinical investigation*, 118(10):3478–3490, 2008.

[16] E. M. Mitchell, R. H. Lyles, and E. F. Schisterman. Positing, fitting, and selecting regression models for pooled biomarker data. *Statistics in Medicine*, 34(17):2544–2558, 2015. sim.6496.

[17] Z. Moodie, Y. Huang, L. Gu, J. Hural, and S. G. Self. Statistical positivity criteria for the analysis of {ELISpot} assay data in hiv-1 vaccine trials. *Journal of Immunological Methods*, 315(1–2):121 – 132, 2006.

[18] Z. Moodie, L. Price, C. Gouttefangeas, A. Mander, S. Janetzki, M. Löwer, M. J. P. Welters, C. Ottensmeier, S. H. van der Burg, and C. M. Britten. Response definition criteria for elispot assays revisited. *Cancer Immunology, Immunotherapy*, 59(10):1489–1501, 2010.

[19] A. Shete, P. Suryawanshi, C. Chavan, A. Kulkarni, S. Godbole, M. Ghate, and M. Thakar. Development of ifn-γ secretory {ELISPOT} based assay for screening of {ADCC} responses. *Journal of Immunological Methods*, 441:49 – 55, 2017.

[20] D. R. Stinson. *Introduction to Balanced Incomplete Block Designs*, pages 1–21. Springer New York, New York, NY, 2004.

[21] P. Ström, N. Støer, N. Borthwick, T. Dong, T. Hanke, and M. Reilly. A statistical approach to determining responses to individual peptides from pooled-peptide {ELISpot} data. *Journal of Immunological Methods*, 435:43 – 49, 2016.

[22] R. Sundberg. An iterative method for solution of the likelihood equations for incomplete data from exponential families. *Communications in Statistics - Simulation and Computation*, 5(1):55–64, 1976.

[23] A. R. M. Townsend, J. Rothbard, F. M. Gotch, G. Bahadur, D. Wraith, and A. J. McMichael. The epitopes of influenza nucleoprotein recognized by cytotoxic T lymphocytes can be defined with short synthetic peptides. *Cell*, 44(6):959–968, 1986.

[24] M. Triska and N. Musliu. An effective greedy heuristic for the social golfer problem. *Annals of Operations Research*, 194(1):413–425, 2012.

[25] D. J. Westreich, M. G. Hudgens, S. A. Fiscus, and C. D. Pilcher. Optimizing screening for acute human immunodeficiency virus infection with pooled nucleic acid amplification tests. *Journal of Clinical Microbiology*, 46(5):1785–1792, 2008.

[26] C. P. Woodbury, J. F. Fitzloff, and S. S. Vincent. Sample multiplexing for greater throughput in hplc and related methods. *Analytical Chemistry*, 67(5):885–890, 1995. PMID: 7762825.

A Proof of Lemma 1 and Lemma 2

Let $Z_1, Z_2, \ldots, Z_n$ be independent random variables with $Z_j \sim Poisson(\beta_j)$, for $j = 1, 2, \ldots, n$, and let $Y_n = \sum_{j=1}^{n} Z_j$. Then

i) $Y_n \sim Poisson\left(\sum_{j=1}^{n} \beta_j\right)$, *for any* $n >= 1$.

ii) $(Z_1, Z_2, \ldots, Z_n)|(Y_n = y) \sim Multinomial\left(y, \left(\frac{\beta_1}{\sum_{j=1}^{n}\beta_j}, \frac{\beta_2}{\sum_{j=1}^{n}\beta_j}, \ldots, \frac{\beta_n}{\sum_{j=1}^{n}\beta_j}\right)\right)$, *for* $k = 1, 2, \ldots, n$.

Proof. We prove *i)* using induction. First we note that $Y_1 = Z_1 \sim Poisson(\beta_1)$. Next, we assume that $Y_p \sim Poisson\left(\sum_{j=1}^{p} \beta_j\right)$ for some $p > 1$, and let $\gamma := \sum_{j=1}^{p} \beta_j$. We then have for Y_{p+1} the probability mass function

$$\begin{aligned}
P(Y_p + Z_{p+1} = y) &= \sum_{k=0}^{y} P(Y_p + Z_{p+1} = y | Y_p = k) P(Y_p = k) \\
&= \sum_{k=0}^{y} P(Z_{p+1} = y - k) P(Z_{p+1} = k) \\
&= \sum_{k=0}^{y} e^{-(\gamma + \beta_{p+1})} \frac{\gamma^k \beta_{p+1}^{y-k}}{k!(y-k)!} \\
&= e^{-(\gamma + \beta_{p+1})} \frac{1}{y!} \sum_{k=0}^{y} \binom{y}{k} \gamma^k \beta_{p+1}^{y-k} \\
&= e^{-(\gamma + \beta_{p+1})} \frac{(\gamma + \beta_{p+1})^y}{y!},
\end{aligned}$$

which is the probability mass function of a $Poisson\left(\sum_{j=1}^{p+1} \beta_j\right)$ distributed random variable. It then follows by induction that $Y_n \sim Poisson\left(\sum_{j=1}^{n} \beta_j\right)$ for all $n \geq 1$.

To prove *ii)*, we use *i)* and get for any $k = 1, 2, \ldots, n$:

$$\begin{aligned}
P(Z_1 = z_1, Z_2 = z_2, \ldots, Z_n = z_n | Y_n = y) &= \frac{P(Z_1 = z_1, Z_2 = z_2, \ldots, Z_n = z_n)}{P(Y_n = y)} \\
&= \frac{e^{-\sum_{k=1}^{n} \beta_k} \frac{\prod_{k=1}^{n} \beta_k^{z_k}}{\prod_{k=1}^{n} z_k!}}{e^{-\sum_{j=1}^{n} \beta_j} \frac{\left(\sum_{j=1}^{n} \beta_j\right)^y}{y!}} \\
&= \frac{y!}{\prod_{k=1}^{n} z_k!} \prod_{k=1}^{n} \left(\frac{\beta_k}{\sum_{j=1}^{n} \beta_j}\right)^{z_k} \\
&= \binom{y}{z_1, z_2, \ldots, z_n} \prod_{k=1}^{n} \left(\frac{\beta_k}{\sum_{j=1}^{n} \beta_j}\right)^{z_k}
\end{aligned}$$

,

which is the probability mass function of a $Multinomial\left(y, \left(\frac{\beta_1}{\sum_{j=1}^{n}\beta_j}, \frac{\beta_2}{\sum_{j=1}^{n}\beta_j}, \ldots, \frac{\beta_n}{\sum_{j=1}^{n}\beta_j}\right)\right)$ random variable, as was to be shown. □

B Proof for Proposition 3

Let $X_1, X_2, ..., X_n, Y$ be independent, identically distributed random variables having a Poisson(β_0) distribution, given some $\beta_0 > 0$. If we assume a priori that $\beta_0 \sim Gamma(a, b)$ for some specified $a, b > 0$, then the posterior distribution of β_0, and the posterior predictive distribution of Y given $\mathbf{X} = \mathbf{x}$ is given by

i) $\beta_0|\mathbf{x} \sim Gamma(a + \sum_i^n x_i, b + n)$,

ii) $Y|\mathbf{x} \sim NegBin(a + \sum_i^n x_i, 1/(b + n + 1))$.

Proof. Starting with *i*), let $f(\beta_0)$ and $f(\beta_0|\mathbf{x})$ be the probability density functions (PDF) of β_0 and $\beta_0|\mathbf{x}$ respectively, i.e. the prior and posterior PDF of β_0; and let $p(\mathbf{x}|\lambda)$ be the probability mass function of $\mathbf{X}|\lambda$, i.e. the likelihood of $\mathbf{X}$ given β_0. Then by Bayes theorem,

$$\begin{aligned}
f(\beta_0|\mathbf{x}) &\propto p(\mathbf{x}|\beta_0)f(\beta_0) \\
&= e^{-n\beta_0}\frac{\lambda^{\sum_i x_i}}{\prod_i x_i!} \cdot \frac{b^a}{\Gamma(a)}\beta_0^{a-1}e^{-b\beta_0} \\
&\propto e^{-n\beta_0}\beta_0^{\sum_i x_i} \cdot \beta_0^{a-1}e^{-b\beta_0} \\
&= \beta_0^{a-1+\sum_i x_i}e^{-(b+n)\beta_0} \\
&\propto \left[\frac{(b+n)^{a+\sum_i^n x_i}}{\Gamma(a+\sum_i x_i)}\right]\beta_0^{a-1+\sum_i x_i}e^{-(b+n)\beta_0},
\end{aligned}$$

which is the PDF of the $Gamma(a + \sum_{i=1} x_i, b + n)$ distribution. We therefore conclude that $\beta_0|\mathbf{x} \sim Gamma(a + \sum_{i=1}^n x_i, b + n)$, as was to be shown.

For *ii*), let $p(y|\mathbf{x})$ be the posterior predictive probability mass function of $Y|\mathbf{x}$. To simplify the following derivations we introduce the notation $A := a + \sum_i^n x_i$ and $B := b + n$. Using *i*), and the fact that a PDF integrated over its entire support equals 1, we get

$$\begin{aligned}
p(y|\mathbf{x}) &= \int_0^\infty p(y|\beta_0, \mathbf{x})f(\beta_0|\mathbf{x})\mathrm{d}\beta_0 \\
&= \int_0^\infty p(y|\beta_0)f(\beta_0|\mathbf{x})\mathrm{d}\beta_0 \\
&= \int_0^\infty e^{-\beta_0}\frac{\beta_0^y}{y!} \cdot \frac{B^A}{\Gamma(A)}\lambda^{A-1}e^{-B\beta_0}\mathrm{d}\beta_0 \\
&= \frac{B^A}{\Gamma(A)\Gamma(y+1)}\int_0^\infty \beta_0^{y+A-1}e^{-(B+1)\beta_0}\mathrm{d}\beta_0 \\
&= \frac{B^A}{\Gamma(A)\Gamma(y+1)}\frac{\Gamma(y+A)}{(B+1)^{y+A}}\int_0^\infty \frac{(B+1)^{y+A}}{\Gamma(y+A)}\beta_0^{y+A-1}e^{-(B+1)\beta_0}\mathrm{d}\beta_0 \\
&= \frac{\Gamma(y+A)}{\Gamma(A)\Gamma(y+1)}\frac{B^A}{(B+1)^{y+A}} \times 1 \\
&= \binom{y+A-1}{y}\left(1 - \frac{1}{B+1}\right)^A\left(\frac{1}{B+1}\right)^y.
\end{aligned}$$

This is the probability mass function for a negative binomial distribution with size $r = A$ and probability $p = 1/(B + 1)$. Returning to the original notation, we can conclude that $Y|\mathbf{x} \sim NegBin\left(a + \sum_{i=1}^n x_i, 1/(b + n + 1)\right)$, as was to be shown. □

C Illustrations for Proposition 2

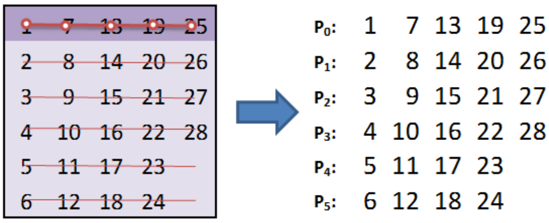


Figure C1: The first class consists of the rows of the matrix. This is an illustration for $N = 28$ peptides, $m = 6$ pools per class, and poolsize $s = 5$.

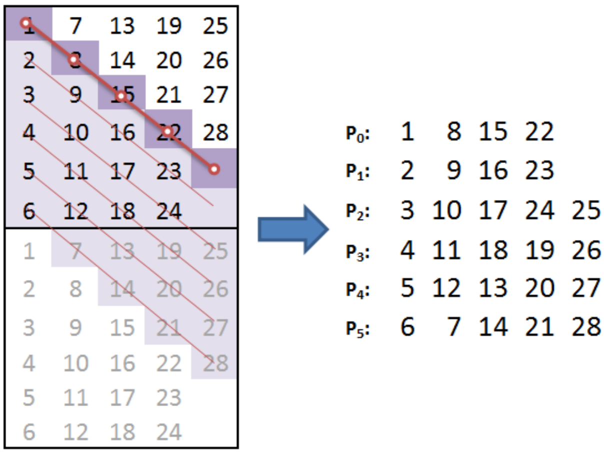


Figure C2: The second class consists of the diagonals of the matrix. This is an illustration for $N = 28$ peptides, $m = 6$ pools per class, and poolsize $s = 5$.

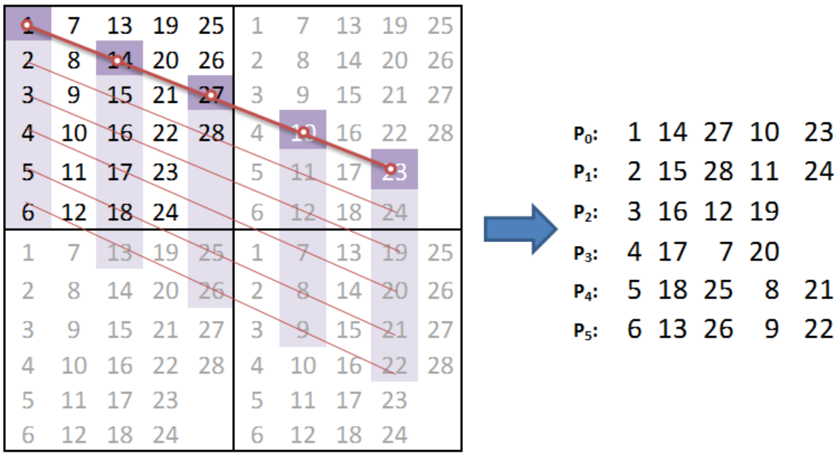
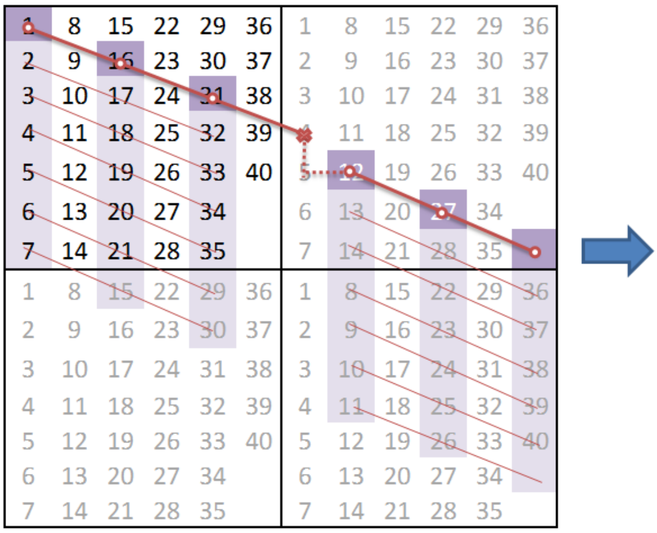


Figure C3: When the poolsize is odd, the third class consists of the diagonals created by going down one row and 2 columns across. This is an illustration for $N = 28$ peptides, $m = 6$ pools per class, and poolsize $s = 5$.



P_0:	1	16	31	12	27	
P_1:	2	17	32	13	28	36
P_2:	3	18	33	14	22	37
P_3:	4	19	34	8	23	38
P_4:	5	20	35	9	24	39
P_5:	6	21	29	10	25	40
P_6:	7	15	30	11	26	

Figure C4: When the poolsize is even, the third class is the same as for odd poolsizes, except for one extra step down and across when reaching the last column of the matrix. This is an illustration for $N = 40$ peptides, $m = 7$ pools per class, and poolsize $s = 6$.