Stockholms universitet

Estimating individual peptide effects from grouped Elispot data

Peter Ström

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2013:5
Matematisk statistik
Oktober 2013

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Matematiska institutionen

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2013:5**
http://www.math.su.se

Estimating individual peptide effects from grouped Elispot data

Peter Ström*

October 2013

Abstract

When there is a need to analyse hundreds of peptides using the Elispot assay, several peptides may be grouped in the same well. This way of analysing peptides is routinely done by lab specialists, but with limitations to the efficiency of this approach. We propose here a method for using the Elispot assay with grouped peptides, involving both an efficient design of the grouping and a statistical method to obtain reliable estimates of the individual peptide effects. The method assigns a distribution to the individual peptide responses and then maximizes the likelihood for the individual peptide effects from the observed group responses using the EM algorithm, which handles the contributions of the individual peptides in a group as unobserved data.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: peter.strom@ki.se. Supervisor: Rolf Sundberg.

Contents

1 Introduction

1.1 Immunology and the Elispot assay

When the human body is infected by some disease-causing organism, the immune system will respond in an attempt to eliminate the intruder. T-cells are a type of white blood cells which are part of the immune system, in particular the adaptive immune system, and help the body to stay free from disease. In the case of infection with Human immunodeficiency virus (HIV), T-cells will recognize peptides from different parts of the virus. These peptides, or short proteins, are the expressions of the viral genes, which are only possible through a host, in this case a human cell. Such peptides, that leads to a reaction of the immune system, are called *antigens*. An antigen is a substance that stimulates the production of antibodies by the body. The purpose of these antibodies is to bind to the antigen so that white blood cells can then disable the antigen and remove it from the body.

After the T-cells recognize a "foreign" peptide, or any antigen, they will respond by producing Interferon-gamma (IFN-γ), a poison-like protein that inhibits viral replication and also activates macrophage cells. Macrophages are large cells that seek up and eat the virus-infected cells. Both T-cells and IFN-γ play important parts in the adaptive immune system. In contrast to the innate immune system, the adaptive immune system needs to be trained. This means that the first time the body is exposed to a specific antigen there may not yet be a specific defense against this antigen. But the T-cells will learn to recognize the antigen and prepare the body, so the next time the body is exposed to the same antigen the T-cells will be ready. It is this property of the adaptive immune system we use in vaccination.

When developing vaccines for HIV it is important to understand the T-cells reaction to the protein expression of the viral gene. By understanding the T-cells reaction to the peptides, or proteins, we can enhance our understanding of what part of the viral gene

should be the direct focus in the development of vaccines.

One tool for measuring T-cell activity in response to a specific peptide is the Enzyme-linked Immunosorbent Spot (Elispot) Assay. The Elispot plate has 96 wells, to each of which is added a blood sample from the patient and also peptides of interest. An automatic reader then measures the number of T-cells that produce IFN-γ on recognition of the antigen by the patient's T-cells. The response appears as spots on the wells that can be counted either by eye using a microscope or by an automatic reader. A picture of an Elispot plate and a well stimulated with an antigen can be seen in Figure 1. The number of spots is a direct measurement of the number of T-cells producing IFN-γ, so the number of spots increases proportionally with strength of the immune response. [6]

Of the 96 wells, 6 are usually used as controls to asses the validity of the assay; 3 for negative controls and 3 for positive controls. The rest of the wells are used to test the peptides of interest. Since the negative controls have no peptide present, they will only generate a small number of spots (or none), and these spots represent of the "noise" present in all wells. The positive control wells are stimulated by a known antigen, to ensure that there is nothing wrong with the assay. For the remaining 90 wells it is common to use triplicates (i.e. to use each peptide in 3 wells) and use the average of the three counts as the estimate of the peptide effect. In this way 30 peptides can be analysed on one plate, with one peptide per well.

To be able to separate positive responding peptides from non-responding peptides we define a positivity criterion. There are many ways of defining a positivity criterion [4], and this topic is still under discussion in the scientific literature. We will use the intuitive and common rule of defining a response as positive if it is at least twice the average of the negative controls.

1.2 Grouping of peptides

Since usually less then 10% of the tested peptides pass the positivity criterion, it is possible to test several peptides in the same well. Grouping peptides into each well of a plate is most commonly done by the matrix pool design [1]. In the matrix pool design each peptide is present in two groups. The idea with this design is that if a group does not pass the positivity criterion neither will any of the peptides in that group, and these peptides can therefore be removed from further analysis. Equivalently, only peptides that belong to groups which both passed the criterion are still potential antigens. Those that remain potentially antigens are hopefully only a small fraction of the initial peptides and can then be tested individually. For example, to create matrix pools (i.e. groups) of 100 peptides, these peptides are arranged in a 10×10 matrix, and then the peptides in each of the 10 columns and each of the 10 rows of the matrix define 20 groups of size 10, and each peptide will be present in two groups.

The matrix pool design has some drawbacks. For efficient use of plates, the number of peptides tested on each plate should be close to a squared number and the number of groups must equal the sum of the number of rows and columns. More importantly, each peptide is only present in two groups, which as we show later can make it difficult to estimate the individual peptide effects. In this project we propose a more flexible design of the groups and a method to estimate the individual peptide effects from the grouped data. We will formulate the problem statistically and use simulations to demonstrate the efficiency of the method. We will also conduct analysis of real grouped data provided by a researcher working with HIV vaccination (prof. Tomas Hanke, University of Oxford).

To our knowledge there are no publication which discusses the possibility of obtaining estimates directly from the grouped data.

2 Materials and methods

In this section we will describe relevant theory for the group of distributions that is called the exponential family, and use this theory to derive a statistical model for grouped Elispot data. We will also describe real grouped Elispot data and describe the details of the simulations we have performed. First we show and motivate the design of the groups that we used in the simulations.

2.1 Proposal for an efficient design

To create the peptide groups according to a Sparse Overlap Design (SOD), we start by forming a list $1, 2, \ldots, n$, where n is the number of peptides that we want to analyse. Since there are 90 wells on one Elispot plate, if we want each peptide to be present 3 times, we divide the plate into 3 blocks of size 30. To make the size of the groups as similar as possible, we divide n by 30 and let the integer part, $\lfloor \frac{n}{30} \rfloor$, be the group size, and then increase the group size by 1 for the first (n modulo 30) groups.

Example: for 200 peptides we get $\frac{200}{30} = 6.67$ *and (200 mod 30) = 20, so for the first block the first 20 groups will consist of 6+1=7 peptides and the last 30-20=10 groups will consist of 6 peptides.*

After creating the groups in the first block as just described, we create the groups in the second block using the same group sizes as in the first block: e.g. create the first group of size 7 by taking the first peptide from each of the first 7 groups in block 1, a second group of size 7 consisting of each of the first peptides in groups 8 to 14 etc. When the first peptide in all 30 groups of block 1 have been allocated, we continue with the second peptide in the first group, and so on until all groups in block 2 have been filled (and all peptides are present in block 2). For block 3 we repeat this process but with the groups in block 2 taking the role of the groups in block 1 previously.

A more formal, or programmatic, way of describing this algorithm is to create a

second list from the first list. We do this by starting with the first element of list 1 and then continue to choose the elements in list 2 by making "jumps" in list 1, and on reaching the end looping back to the beginning of the list. The first (n mod 30) jumps will be of size $\lfloor \frac{n}{30} \rfloor + 1$ and jumps (n mod 30)+1 to 29 will be of size $\lfloor \frac{n}{30} \rfloor$. Then we continue to add elements by alternating between jump size $\lfloor \frac{n}{30} \rfloor + 1$ for (n mod 30)+1 times and jump size $\lfloor \frac{n}{30} \rfloor$ for 29 - (n mod 30) times, until list 2 is of length n (and all peptides are present once). The next step is to create a third list in exactly the same way, but now with list 2 replacing the role of list 1.

Finally, we form the groups in the three blocks by grouping the three lists by the groups sizes described above. This makes $3 \times 30 = 90$ groups of sizes $\lfloor \frac{n}{30} \rfloor$ and $\lfloor \frac{n}{30} \rfloor + 1$. We now let ζ_k be the set of peptides in group k and $g_k(\beta)$ be the sum of all β_i corresponding to the peptides in group k, k=$1, \ldots, 90$.

To be able to efficiently distinguish between the effects of different peptides it is important that the peptides in the groups overlap as little as possible, in the sense that any two groups should have as few peptides in common as possible. This design guarantees minimal overlap of the peptides in any adjacent blocks, which means that any pair of groups in block 1 and 2 or block 2 and 3 will have as few peptides in common as possible. The design does not, however, guarantee minimal overlap between block 1 and 3, so there may be room for improvement.

2.2 Assumptions and definitions

We will assume that for a given individual, the observed response to a specific peptide has a Poisson distribution, with mean value equal to the underlying "true" response [3]. Later we will relax this restriction and evaluate the method when we see larger variation with the same peptide than expected for a Poisson variable. We will also assume that different peptides within a well generate spots independently of each other, so that the expected number of spots from each peptide in a pool add up to the expected number of

spots in the pool. This will not be the case if the number of spots in a well is too high due to limitations of the area where spots form, or limitations in ability of the reader to discriminate between dense spots.

As mentioned earlier, the noise is estimated as the average of the 3 negative control wells, and the positivity criterion defined as twice the estimated noise. Any estimated peptide effect less than this value will be classified as a "non-response". Those that were estimated higher than the positivity criterion are regarded as "responders". We will also define responses to be low (up to 40 spots), medium (41 to 100 spots) or high (above 100 spots).

2.3 Exponential families and the Poisson distribution

2.3.1 Definition and notation

We will formulate the theory in terms of the exponential family as the Poisson distribution is a member of the exponential family. The Expectation-Maximization (EM) algorithm [2][7], that we use for estimation takes a simple form when the complete data likelihood is based on a distribution from the exponential family. Although many elegant statistical results have been derived for exponential families [7] we will mainly focus on the expected value of a sufficient statistic as a function of only distribution parameters, and vice versa. We will then generalize to the case of incomplete data and show how to get a simple expression for the EM algorithm.

Definition (Exponential families)**.** *If a parametric model for* $\boldsymbol{y}$ *can be written in the form*

$$f(\boldsymbol{y};\theta) = a(\theta)h(\boldsymbol{y})e^{\theta^T \boldsymbol{t}(\boldsymbol{y})},$$

for some statistic $\boldsymbol{t}$*, then the distribution is an exponential family with canonical parameter vector* $\theta = (\theta_1, \ldots, \theta_k)$ *belonging to the k-dimensional parameter space* Θ *and canonical statistic vector* $\boldsymbol{t}(\boldsymbol{y}) = (t_1(\boldsymbol{y}), \ldots, t_k(\boldsymbol{y}))$*.*

If $\mathbf{y}$=$(y_1, \ldots, y_n)$ is a *sample* of independent observations y_i from an exponential family we get

$$f(\mathbf{y};\theta) = \prod_{i=1}^{n} f(y_i;\theta) = \prod_{i=1}^{n} a(\theta)h(y_i)e^{\theta^T \mathbf{t}(y_i)} = a(\theta)^n \prod h(y_i)e^{\theta^T \sum \mathbf{t}(y_i)}.$$

This is also an exponential family since $a(\theta)^n$ is a function only of θ and not of the data, $\prod h(y_i)$ is a function only of the data and not of θ, and $\theta^T \sum \mathbf{t}(y_i)$ is the inner product of the canonical parameter vector θ and the canonical statistic $\sum \mathbf{t}(y_i)$. In this case it should be noted that the canonical statistic is a sum over the canonical statistic for each individual observation.

The part of the exponential family model that only depends on the parameters, a(θ), can be interpreted as the normalizing constant which makes the density integrate to 1 (or the probability mass function sum to 1).[7] If we denote $\frac{1}{a(\theta)}$ by C(θ) we get the following expression

$$C(\theta) = \sum h(\mathbf{y})e^{\theta^T \mathbf{t}(\mathbf{y})}$$

in the discrete case (substitute the sum for an integral in the continuous case).

In vector notation we can express the expected value of the canonical statistic in terms of the gradient vector of the function log $C(\theta)$,

$$\mu_t(\theta) = E_\theta(\mathbf{t}) = \mathrm{D}\ \log\ C(\theta),$$

and in matrix notation we can express the variance in terms of the Hessian matrix,

$$\mathrm{Var}_\theta(\mathbf{t}) = \mathrm{D}^2 \log\ C(\theta).$$

2.3.2 Log-likelihood in canonical parametrization - MLE and I(θ)

To get the Maximum Likelihood Estimate (MLE) we solve the likelihood equations (i.e. set the score function equal to zero). The log-likelihood has the following form in canonical parametrization:

$$\log \mathrm{L}(\theta) = \theta^T \mathbf{t} - \log C(\theta) + constant.$$

Taking partial derivatives of the log-likelihood function:

$$\frac{\partial \log L(\theta)}{\partial \theta_j} = t_j - \frac{\partial \log C(\theta)}{\partial \theta_j} = t_j - E_\theta(t_j).$$

In vector notation this is the score function

$$U(\theta) = \mathbf{t} - \mu_t(\theta).$$

Finally, setting the score function equal to zero and solving for θ gives the MLE:

$$\hat{\theta}(\mathbf{t}) = \mu_t^{-1}(\theta).$$

To get the Fisher information we take the expected value of minus the second partial derivative of the log-likelihood.

$$E\left(-\frac{\partial^2 \log L(\theta)}{\partial \theta_i \partial \theta_j}\right) = E\left(\frac{\partial^2 \log C(\theta)}{\partial \theta_i \partial \theta_j}\right) = E\left(\mathrm{Cov}(t_i, t_j)\right) = \mathrm{Cov}(t_i, t_j)$$

In matrix notation we therefore get the Fisher information as

$$I(\theta) = \mathrm{Var}_\theta(\mathbf{t}).$$

2.3.3 The Poisson distribution

The Poisson distribution can be written in the form that defines an exponential family

$$f(x;\beta) = \frac{\beta^x}{x!}e^{-\beta} = e^{-e^\theta}\frac{1}{x!}e^{\theta x},$$

where the canonical parameter $\theta = \log\beta$ and $C(\theta) = e^{e^\theta}$. This gives $E_\theta(\mathbf{t}) = E_\theta(X) =$ D $\log C(\theta) = e^\theta$ and the MLE $\hat{\theta} = \log y$ or, equivalently, $\hat{\beta} = y$.

For a sample $\mathbf{x} = (x_1, \ldots, x_n)$ we get

$$f(\mathbf{x};\theta) = e^{-ne^\theta}\prod\frac{1}{x_i!}e^{\theta\sum x_i},$$

and the canonical statistic is the sum of the sample. Since for a sample $C(\theta) = e^{ne^\theta}$, we get the MLE as an average of the observed values, $\hat{\theta} = \log\frac{\sum x_i}{n}$ or, equivalently, $\hat{\beta} = \frac{\sum x_i}{n}$.

Conditioning on the sum of independent Poisson variables results in a multinomial variable. Thus for independent Poisson variables $\mathbf{x} = (x_1, \ldots, x_n)$, each with a unique parameter β_i, conditional on their sum $y = \sum x_i$, we can express the probability mass function as multinomial:

$$\begin{aligned} f(\mathbf{x}|y;\beta) = \frac{f(\mathbf{x};\beta)}{f(y;\beta)} &= \\ &= \frac{\prod e^{-\beta_i}\frac{\beta_i^{x_i}}{x_i!}}{e^{\sum\beta_i}\frac{(\sum\beta_i)^y}{y!}} = \\ &= \frac{y!}{\prod x_i!}\frac{\prod\beta_i^{x_i}}{\prod(\sum\beta_i)^{x_i}} = \\ &= \frac{y!}{\prod x_i!}\prod\left(\frac{\beta_i}{\sum\beta_i}\right)^{x_i}. \end{aligned}$$

2.4 Incomplete data

2.4.1 Implications of incomplete data for exponential families

When we cannot observe the complete data, x, but only the incomplete data, $y = y(x)$, we need to modify the theory of exponential families described above. For our application the main distinctions are that the likelihood equations changes from $t - E_\theta(t) = 0$ to $E_\theta(t|y) - E_\theta(t) = 0$, and that the Fisher information changes from $I(\theta) = \mathrm{Var}_\theta(t)$ to $I(\theta) = \mathrm{Var}_\theta(E_\theta(t|y))$.[7]

2.4.2 The EM algorithm for exponential families

The EM algorithm takes a very simple form when the complete data are from an exponential family. Solving the likelihood equations for incomplete data, the MLE must satisfy

$$\theta = \mu_t^{-1}(E_\theta(t|y)). \tag{1}$$

We can find a solution to (1) by choosing a starting value, θ^0, and iteratively alternating the following two steps until convergence:

1. Calculate $E_{\theta^0}(t|y)$.
2. Insert $E_{\theta^0}(t|y)$ in (1) to get a new value, θ^1.

The (n+1)th step thus has the form $\theta^{n+1} = E(E_{\theta^n}(t|y))^{-1}$, and we continue updating this formula until $\theta^{n+1} \approx \theta^n$. However, this algorithm is not guaranteed to find a global maximum of the likelihood, L(θ|y), but if it converges, it will at least find a local maximum. The algorithm can be thought of as climbing the hill(s) of the likelihood function. If the starting value, θ^0, corresponds to a point on a slope of the likelihood, the algorithm will climb (sometimes very slowly) the likelihood function and stop when it reaches a maximum, regardless whether this is the global maximum or some unwanted local maximum of the likelihood.[7]

2.5 Statistical model

To derive a model, we assume we want to estimate the individual effects of n peptides using 90 wells on a single plate. By allowing each peptide to be present 3 times we divide the 90 wells into 3 blocks of 30 wells each. We then allocate the n peptides over 30 wells in 3 different ways (blocks). This means a total of $3 \times n$ peptides on the plate and approximately $\frac{n}{30}$ peptides per group.

From the assumptions above, peptide i in block j generates spots according to the distribution $X_{ij} \sim \text{Po}(\beta_i)$ for $i = 1, 2, ..., n$ and $j = 1, 2, 3$. If we could observe all 3n values of x_{ij}, the likelihood would be

$$L(\beta|x_{11}, x_{12}, ..., x_{n3}) = \prod_{i=1}^{n} \prod_{j=1}^{3} e^{-\beta_i} \frac{{\beta_i}^{x_{ij}}}{x_{ij}!}, \tag{2}$$

leading to the maximum likelihood estimates of the peptide effects

$$\hat{\beta}_i = \frac{\sum_{j=1}^{3} x_{ij}}{3} = \frac{x_{i1} + x_{i2} + x_{i3}}{3}. \tag{3}$$

However, the x_{ij} are unobserved and we only observe the group values denoted y_k for $k = 1, 2, ..., 90$. An explicit formula that expresses y_k as a sum of x_{ij} is not possible because the grouping of peptides (using SOD or some other design) differs between applications. Since the sum of independent Poisson variables is Poisson, the groups generate spots according to the distribution $Y_k \sim \text{Po}(\text{g}_k(\beta))$ for $k = 1, 2, ..., 90$, where $\text{g}_k(\beta)$ is the sum of all β_i corresponding to the peptides in group k. We now aim to maximize the group likelihood which, due to assumed independence between unobserved values of the x_{ij}, is

$$L(\beta|y) = \prod_{k=1}^{90} e^{-\text{g}_k(\beta)} \frac{\text{g}_k(\beta)^{y_k}}{y_k!}.$$

This likelihood has a maximum value at $\text{g}_k(\beta) = y_k$ for all k, if these parameter values

exists. There need not exist parameter values that satisfy this equality since there are 3 unobserved values from each peptide. Because $g_k(\beta)$ is a function of $\beta_i,\ i = 1, 2, ..., n$, this system of equations will be under-determined whenever n is larger than k (i.e. more peptides than observations), and we cannot expect a unique solution. Any value of β that satisfies $g_k(\beta) = y_k$ for all k will be a global maximum of the unconstrained likelihood, but since β is a vector of Poisson parameters, these need to be non-negative to make sense (i.e. β must be constrained to be non-negative). We can impose this constraint by considering this as an incomplete data problem and using the Expectation-Maximization (EM) algorithm [2] to find the maximum for the likelihood.

The EM algorithm works by calculating the expected value (E-step) of a sufficient statistic for the "complete data" variables (i.e. the X_{ij}), $t_i = \sum_{j=1}^{3} X_{ij}$, by an initial guess for the parameter vector, β^0, and conditioning on the observed data (i.e. the group values y_k). Since these 90 group values are sums of independent Poisson variables, conditioning on these sums gives 90 multinomial variables, each with parameters $\zeta_k(\beta)$, where $\zeta_k(\beta)$ is the set of parameters that belong to the peptides in group k. After the first E-step we maximize (M-step) the "complete data" likelihood (2), using the formula (3), by replacing each sufficient statistic t_i with the corresponding expected value calculated in the previous step, $\mathrm{E}[t_i|y, \beta^0]$. Combining these two steps gives the expression

$$\beta_i^1 = \frac{\sum_{k|\beta_i \in \zeta_k(\beta)} y_k \frac{\beta_i^0}{g_k(\beta^0)}}{3},$$

where each term in the sum is an expected value for a peptide response considered as an outcome from a multinomial distribution. The division by $g_k(\beta^0)$ is to normalize the multinomial probabilities so that they sum to 1.

The algorithm continues by updating the estimate by replacing β^0 with β^1 to get β^2 and so on, until $\beta^n \approx \beta^{n-1}$. The EM algorithm is guaranteed to increase the likelihood $L(\beta|y)$ for each step [2], but we cannot be sure that it "climbs the right hill" if the

likelihood is multimodal. It would be intractable to derive conditions for when there is only one mode, or when we have found the right mode, so we rely on simulations to investigate how well the estimates correspond to the true values in realistic settings.

2.6 Fisher confidence intervals for grouped Elispot data

Here we will derive expressions that may be used to provide measures of uncertainty and confidence intervals. Using the expression for Fisher information for incomplete data and the structure of the data from our Elispot application, we can do the following calculations for the canonical parameters. Since we only get a contribution to the covariance when two peptides belong to the same group, we introduce G_{ijk} to indicate when peptide i and j belong to group k (i.e $\mathrm{G}_{ijk} = 1$ if $\beta_i \cup \beta_j \in \zeta_k$ and 0 otherwise).

$$
\begin{aligned}
I_\theta(\theta)_{i,j} &= \mathrm{Cov}_\theta(E_\theta(t_i|\mathbf{y}), E_\theta(t_j|\mathbf{y})) = \\
&= \mathrm{Cov}_\theta(E_\theta(\sum_k(\mathrm{G}_{ijk}\mathrm{X}_i|y_k)), E_\theta(\sum_k(\mathrm{G}_{ijk}\mathrm{X}_j|y_k))) = \\
&= \mathrm{Cov}_\theta(\sum_k(\mathrm{G}_{ijk}E_\theta(\mathrm{X}_i|y_k)), \sum_k(\mathrm{G}_{ijk}E_\theta(\mathrm{X}_j|y_k))) = \\
&= \sum_k \mathrm{G}_{ijk}\mathrm{Cov}_\theta(E_\theta(\mathrm{X}_i|y_k)), E_\theta(\mathrm{X}_j|y_k)) = \\
&= \sum_k \mathrm{G}_{ijk}\frac{\beta_i\beta_j}{g_k(\beta)^2}\mathrm{Var}_\beta(\mathrm{Y}_k) = \\
&= \beta_i\beta_j \sum_k \left(\frac{\mathrm{G}_{ijk}}{g_k(\beta)}\right)
\end{aligned}
$$

We can reparametrize to get the information for β, which is a more interpretable parametrization:

$$I_\beta(\beta) = J^T I_\theta(\theta(\beta))J,$$

where J is the Jacobian matrix of $\theta_i(\beta) = \log\beta_i$. Since J is an $n \times n$ diagonal matrix with elements $\frac{1}{\beta_i}$ for $i = 1, ..., n$, we get the following expression:

$$I_\beta(\beta)_{i,j} = \frac{1}{\beta_i}\frac{1}{\beta_j} I_\theta(\theta(\beta))_{i,j} = \sum_k \left(\frac{\mathrm{G}_{ijk}}{g_k(\beta)}\right).$$

If peptide i and j never occur in the same group, then the corresponding element in the information matrix will be zero, and if they occur together only in group k, the corresponding element will be $\frac{1}{g_k(\beta)}$. Since each peptide occur in 3 groups, the diagonal elements of the information matrix are $\frac{1}{g_{k_1}(\beta)} + \frac{1}{g_{k_2}(\beta)} + \frac{1}{g_{k_3}(\beta)}$, where $g_{k_1}(\beta), g_{k_2}(\beta)$ and $g_{k_3}(\beta)$ are the sum of the parameters in the groups to which peptide i belong, respectively.

Further, we can argue that three observations from a Poisson variable with mean parameter of 10 or more could motivate the normality assumption, in which case we get a 95% confidence interval,

$$\left[\hat{\beta}_i - 1.96\sqrt{I^{ii}}, \hat{\beta}_i + 1.96\sqrt{I^{ii}}\right],$$

where I^{ii} is the i'th diagonal term of the inverted information matrix $I(\hat{\beta})^{-1}$.

2.7 Simulations of grouped Elispot data

2.7.1 Assumptions

To illustrate the method we simulated data similar to what is observed in a real situation. The parameters were therefore either set to zero, corresponding to peptides that resulted in no immune response, or drawn from a Gamma distribution. We used two Gamma distributions, $\Gamma(\text{shape} = 6, \text{scale} = 5)$ for mostly mild responses and $\Gamma(\text{shape} = 6, \text{scale} = 10)$ for moderate responses, see Figure 2. We created a vector, β, of length n to represent the parameter values by letting 96% of the values be zero and 4% come from one of the two Gamma distributions, and then permuted the vector. To vary the scenarios we also

used 8% of non-zero elements.

Next we generated 3 vectors of length n, each with values drawn from Poisson variables with mean values corresponding to the parameter vector β. In vector notation the 3 vectors of individual peptide responses were generated according to

$$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \sim \text{Po}(\beta).$$

The values in these vectors are the "complete data" which we suppose unknown in conducting the EM analysis. When estimating the values in β we only observed the 90 group values $y_k,\ k = 1, 2, ..., 90$, where the first 30 group values (block 1) were sums of disjoint groups of values in $\mathbf{x}_1$, likewise for block 2 and 3. In notational form the y vector was created as

$$\begin{aligned} y_k &= g_k(\mathbf{x}_1),\ k = 1, ..., 30 \\ y_k &= g_k(\mathbf{x}_2),\ k = 31, ..., 60 \\ y_k &= g_k(\mathbf{x}_3),\ k = 61, ..., 90, \end{aligned}$$

where g_k (i.e. the grouping function) was decided by the SOD design.

The final step in generating values was to add "noise" to each well (i.e. every group value y_k) and to 3 additional wells (i.e. the negative controls). The noise was simulated from a Poisson(5) variable: a mean of 5 for the noise is reasonable in HIV applications where 250.000 cells are added to each well [5]. The y vector is now of length 93, with the first 90 values representing the sum of spots from each peptide in a group plus the additional noise, and the last 3 values representing only noise. The β vector was also extended with an extra parameter for the noise.

Although we derived our methods under the assumption that the data are from Poisson variables, we illustrate the performance of the method also when there is overdis-

persion (i.e. extra variation) in the responses from each peptide. For this purpose we followed the example in previous published work [4] [3] to not only generate Poisson data but also negative binomial data with mild ($\varphi = 2$) and moderate ($\varphi = 10$) dispersion parameter.

2.7.2 Different methods of estimation

We have compared two different ways of estimating the parameter vector β, both of which numerically maximize the likelihood $L(\beta|y)$: the constrained MLE from the EM algorithm described above and the unconstrained MLE from the *nlm* function in R, which maximizes the likelihood with a Newton-Raphson algorithm. In contrast to the constrained MLE, which considers the individual peptide effects to be Poisson, this latter way of estimating β does not impose any restriction on the estimated vector $\hat{\beta}$. Since we assumed all peptides were equally likely to elicit responses, we let the initial guess for β be $\beta^0 = (1, 1, ..., 1)^T$, and used these starting values for both the EM method and the *nlm* function. The R-code for all simulation steps, and estimation, can be found in the appendix.

The comparison of the methods was done by calculating the sensitivity (i.e. the number of true estimated positive responses divided by the number of actual true positive responses) and specificity (i.e. the number of true estimated negative responses divided by the actual number of negative responses) under different scenarios. For each scenario 50 simulations were made, and for each simulation the sensitivity and specificity were calculated for both the constrained and the unconstrained MLE.

2.7.3 Comparing the number of blocks

Our choice of 3 blocks agrees with the convention of using triplicates in Elispot assays, but it is not obvious that an assay should be divided into 3 blocks: we could just as well use more or fewer blocks. For investigating the optimal number of blocks, we simulated

30 data sets for each of 6 scenarios. In each simulation we used either Poisson variables with low or with medium mean values to generate data for each of 200 peptides (of which 13 were positively responding peptides). We used the SOD design and estimated the peptide effects by the EM algorithm.

2.8 Real grouped Elispot data

We have access to grouped data for 6 subjects for which we also have verification in terms of individually assayed peptides. The groups were designed by a program called DeconvoluteThis!, which tries to minimize the overlap of peptides between groups, but based on simulations instead of an analytic approach. The grouped data for the 6 subjects consists of 199 peptides in 80 groups, each peptide being present 4 times. Each of the 6 subjects assays had between 2 and 4 wells used for negative controls (to estimate noise level of the assay). We then compared the constrained MLEs to individually assayed peptide estimates (for the same subjects) provided by prof. Tomas Hanke, University of Oxford.

The EM algorithm had no problem to converge and by using several different starting values we found just one maximum of the likelihood for each of the 6 subjects.

3 Results

3.1 Simulation study

3.1.1 Quality of estimates

In Figure 3 and 4 we see an illustration of the characteristics of the estimates from both the maximization of the unconstrained likelihood using the *nlm* function and from the maximization of the constrained likelihood using the EM algorithm. The three rows of plots in a figure each correspond to only one simulated plate, with $200 \times 0.04 = 8$, $400 \times 0.04 = 16$ and $400 \times 0.08 = 32$ responding peptides respectively. To the right

we see the estimated group values (i.e. $g_k(\hat{\beta})$) plotted against the true group values y_k for $k = 1, 2, ..., 90$. If the dots and stars closely follow the straight line with slope of 1, it indicates that both methods find a global maximum (or close to) since $g_k(\hat{\beta}) \approx y_k$. To the left we see each of the two estimates of β plotted against the true values, in the sense that the true values are the estimates that would have been obtained if all individual responses x_{ij} were observed. These plots reveal if the methods of estimating the β vector have been successful. Values close to the line with slope 1 are correctly identified and if all estimated values from one of the methods are close to the line we know that that method has found the correct MLE. The vertical and horizontal lines are twice the estimated noise (i.e. the positivity criterion).

We also averaged the result from several simulations to investigate the ability of the two methods to correctly discriminate between positive and negative responses. Table 1 shows the mean of the 50 estimations of the sensitivity and specificity under several scenarios (in per cent). Here we see systematic trends in the ability to separate positive and negative responses as we increase the variation in the responses of the same peptide (i.e. increase dispersion) or increase the number of responding peptides. We can also notice a systematic difference in this ability between 8% positive responding peptides of 200 tested peptides and 4% of 400 (i.e. the same number of positive responding peptides).

3.1.2 Analysis of the number of blocks

We simulated data with 2, 3 and 4 blocks using a SOD design and the EM algorithm for estimates. From Table 2 we see a large increase in sensitivity and specificity going from 2 to 3 blocks, but practically no difference between 3 and 4 blocks.

		Low responders			Medium responders		
		Poisson	Disp. 2	Disp. 10	Poisson	Disp. 2	Disp. 10
200 peptides	4%	**94.8 / 99.3**	**95.7 / 99.9**	**91.2 / 99.4**	**99.8 / 100**	**100 / 99.8**	**99.2 / 99.2**
		50.6 / 98.3	49.6 / 98.8	52.6 / 96.9	77.0 / 92.9	74.6 / 92.3	74.7 / 91.2
	8%	**93.9 / 99.2**	**94.7 / 99.0**	**81.1 / 97.9**	**99.6 / 99.1**	**98.5 / 98.2**	**95.0 / 97.3**
		60.7 / 95.0	62.2 / 94.1	53.4 / 93.1	80.8 / 83.3	86.4 / 81.0	82.9 / 81.5
400 peptides	4%	**86.9 / 99.0**	**81.6 / 99.0**	**67.3 / 98.3**	**97.2 / 98.7**	**94.2 / 98.0**	**87.2 / 97.5**
		26.7 / 98.6	24.1 / 99.1	22.3 / 98.5	61.3 / 93.0	64.9 / 91.1	56.1 / 91.4
	8%	**58.6 / 95.7**	**60.3 / 95.8**	**50.3 / 94.7**	**75.0 / 90.5**	**76.1 / 89.9**	**66.5 / 82.4**
		32.2 / 96.4	32.1 / 96.2	33.8 / 94.7	65.2 / 83.5	70.5 / 81.9	64.0 / 82.4

Table 1: The values are *sensitivity / specificity* under different scenarios (e.g. 200 peptides of which 4% are mostly low responders with a within variation that follows a Poisson distribution) based on 50 simulations each, in per cent. For each scenario two estimates of the sensitivity and specificity are presented, one based on the constrained MLEs (bold) and the other based on the unconstrained MLEs. Disp. 2 and 10 means a within variation that is 2 and 10 times that of a Poisson variable, respectively.

	Low responders	**Medium responders**
2 block	*78.4/97.1*	*87.1/96.0*
3 block	*97.4/99.9*	*99.2/99.7*
4 block	*96.4/100*	*99.7/99.9*

Table 2: The values are *sensitivity / specificity* for constrained MLEs based on 30 simulations in each setting, with 13 responding peptides among 200 tested peptides. The low and medium responding peptides where generated from the densities shown in Figure 3 in Appendix.

3.2 Analysis of real grouped Elispot data

The real grouped Elispot data (described above) were analyzed for 6 subjects and compared to estimates of individually assayed peptides for the same subjects. The response from the individually assayed peptides described above can be found in Figure 5 in Appendix. We compared the ranks of these responses to the ranked constrained MLEs for each subject assigning rank 1 to the highest values, and compared the rank of constrained MLEs for the top 5 individually-assayed peptides for each of the 6 subjects.

The results of this comparison are presented in Table 3.

	ID					
Rank	403	404	406	409	410	411
1	7	1	14	6	1	1
2	63	79	27	5	3	3
3	28	18	4	12	15	4
4	4	181	2	4	56	48
5	105	57	13	31	74	25

Table 3: Comparing rank of individually assayed peptide responses and rank of grouped assayed constrained MLEs for 6 subjects (assigning rank 1 to highest values). The numbers under subject ID are the rank of constrained MLEs corresponding to the top 5 ranks of individually assayed peptides for each subject.

4 Discussion

It is clear that the number of responding peptides strongly affects the ability to obtain reliable estimates, but other factors also influence the performance. It can be seen from Table 1 that increasing the number of non-responding peptides while maintaining the number of responding peptides (from row 8% of 200 to row 4% of 400) will lower the sensitivity and specificity. We also see a lower sensitivity and specificity when simulating low responses compared to medium responses, and when increasing the variance of the responses from each peptide.

From our simulations, we conclude that the constrained MLEs are more reliable than the estimates obtained from maximizing an unconstrained Poisson likelihood. The constrained MLEs are also better in separating positive and negative peptides. The constrained MLE maintain acceptable sensitivity and specificity as long as the number of responding peptides is not too high. From Table 1 the constrained MLE shows a sensitivity around 90% and specificity close to 100% for 16 responding peptides out of 200 or 400 tested peptides. For $400 \times 0.08 = 32$ responding peptides the sensitivity of the EM was just above 50% for low responding peptides, and a bit higher for medium

responding peptides.

Besides distinguishing between positive and negative responses the EM algorithm provides good estimates of the true value of the response, as long as the number of responding peptides is not too high. This is illustrated for one simulation for each of 6 scenarios with Poisson responses in Figures 3 and 4 (overdispersion in Figure 6 to 9 in Appendix). These figures illustrate the correlation of constrained and unconstrained MLEs with the true value of the peptide effects. Each figure show three simulations, each with a unique β. The 3 plots to the left in each figure show estimated peptide effects plotted against the true value of the peptide effects. The 3 plots to the right in each figure show the estimated group value (i.e. sum of estimated peptide effects in a group) plotted against the true value. Maximizing the constrained likelihood generally leads to sensible group estimates in the sense that they more or less follows the line with slope 1. This need not be the case when maximizing the unconstrained likelihood as can be seen in Figure 3a. For the individual peptide effects, the MLEs from the constrained likelihood are often close to the true values while maximization of the unconstrained likelihood can result in estimates that are too low. These latter estimates cannot be adjusted upwards as that will also increase the number of false positive estimates (and therefore lower the specificity).

We have proposed a way to obtain a measure of uncertainty for the constrained MLEs, which is based on an approximate quadratic form of the log-likelihood function. This approximation may not be valid due to the constraint on the parameter space. We have not, theoretically or empirically, validated the reliability of the expressions for confidence intervals we derived. This is merely thought of as a starting point or inspiration for further research.

When analysing the most efficient number of blocks, we saw a large improvement on increasing from 2 to 3 blocks but practically no increase from 3 to 4 blocks. Taking into account the potential extra uncertainty in a real setting of grouping more peptides

together (as needs to be done if 4 blocks are used instead of 3 blocks) and the increased risk of losing the additivity property of the peptide effects because of the limited area for spots to form in a well, we recommend the use of 3 blocks.

Although we use a HIV application for illustration, the method proposed here is quite general. The R-code can be used to simulate data from other applications. Even though we made an effort to represent true data by also imposing extra Poisson variation, it would be useful to evaluate the method in a laboratory setting using real Elispot plates with the proposed design and follow-up verifications of the estimates.

It is unclear why the analysis of the real data for the 6 subjects performed rather poorly. Contributing factors to this may be the lack of an efficient design of the groups (this was not SOD), that each peptide was tested individually only once for the 6 subjects, that too few cells (specimen) were added in the grouped assay or that the grouped and individually assayed peptides were not assayed at the same time. Another explanation may be that there simply are too many sources of uncertainty in a real lab situation to be able to realistically model the assay, and to obtain reliable estimates of the individual peptide effects.

In summary, we believe that our method of analyzing a large number of peptides can be useful and sufficiently accurate for practical purposes in some situations. The R-code can be modified to represent a wide range of scenarios, and to analyse Elispot plates for any application. An important advantage of the method is the potential for more cost-effective laboratory assays. If 200 or 300 peptides instead of 30 can be analysed on one plate, while maintaining an acceptable precision in the estimates, this could lead to dramatic reductions in the number of plates required.

5 Conclusion

We have demonstrated an efficient way of using Elispot plates when there is a need to test hundreds of peptides for each patient. We present both an efficient design and a method to obtaining reliable estimates. The basic idea is to design the groups in a way that minimizes the overlap of peptides in any pair of groups, and to regard each peptide within a group as an unobserved event. By using Poisson distributions for the number of spots generated by the individual peptides we can use the EM algorithm to maximize the likelihood with the constraint that all estimated peptide effects must be non-negative.

The estimated peptide effects were shown to have a much better sensitivity and specificity compared to maximization of the unconstrained likelihood, for a number of scenarios considered. The accuracy of the estimates was also much higher when using the EM algorithm to estimate the individual peptide effects. R-code is supplied in the appendix and can easily be modified to evaluate and apply the method in other applications of the Elispot assay.

6 Supplementary figures

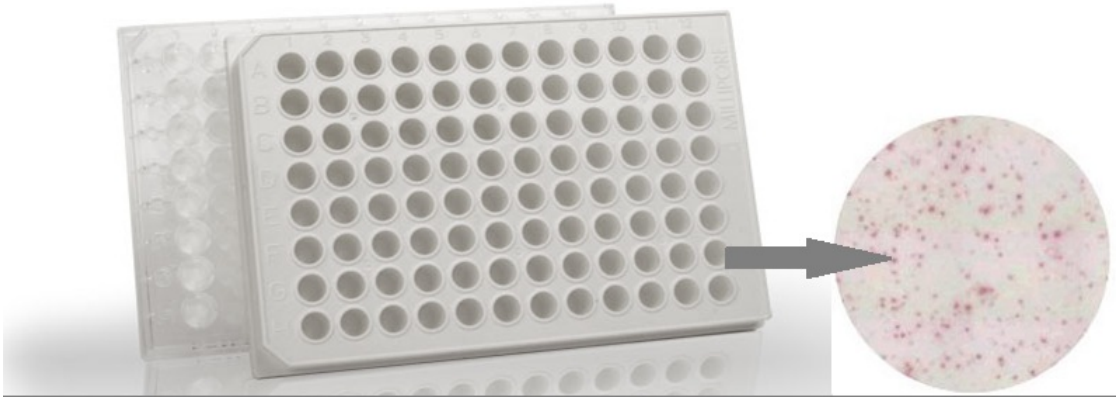

Figure 1: An Elispot plate and how a well could look like after being challenged with an antigen. The response of interest is the number of spots, which usually is counted by an automatic reader.

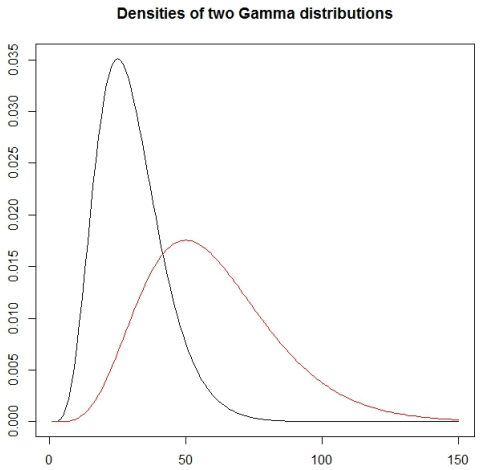

Figure 2: This graph shows the densities of the two gamma distributions, $\Gamma(\text{shape} = 6, \text{scale} = 5)$ and $\Gamma(\text{shape} = 6, \text{scale} = 10)$, that were used to represent the positive effect peptides. The former has mean 30 and variance 150 and the latter has mean 60 and variance 600.

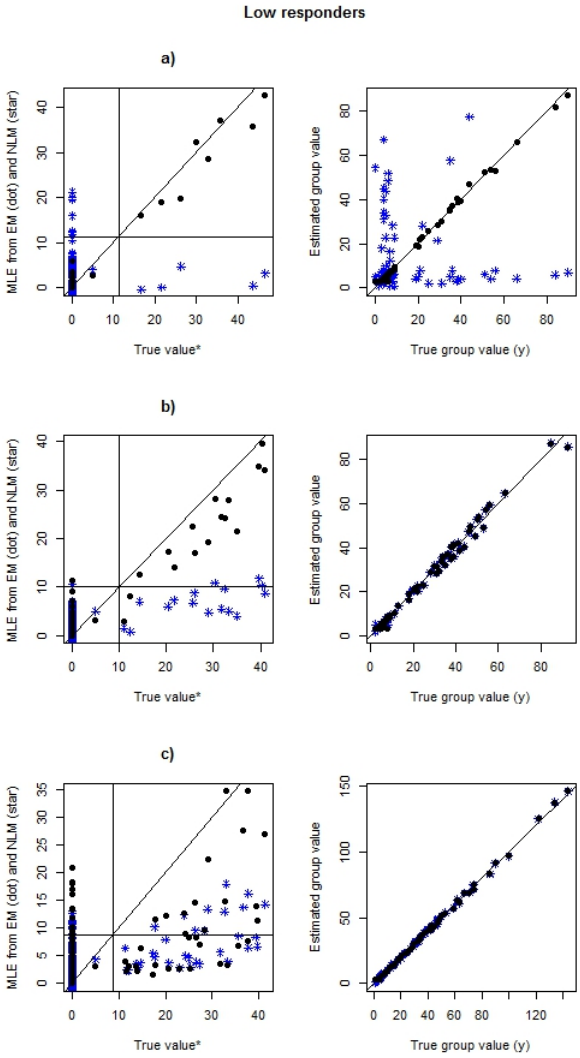

Figure 3: Three simulations of low responding peptides that each generate Poisson data. Peptides grouped into 90 groups and 3 blocks using SOD. Vertical and horizontal line is twice the estimated noise (i.e. the positivity criterion). A line with slope 1 is also provided.
a) 200 peptides of which 4% are responding peptides.
b) 400 peptides of which 4% are responding peptides.
c) 400 peptides of which 8% are responding peptides.
∗"True value" denotes the estimate that would have been obtained if all individual responses had been observed.

.

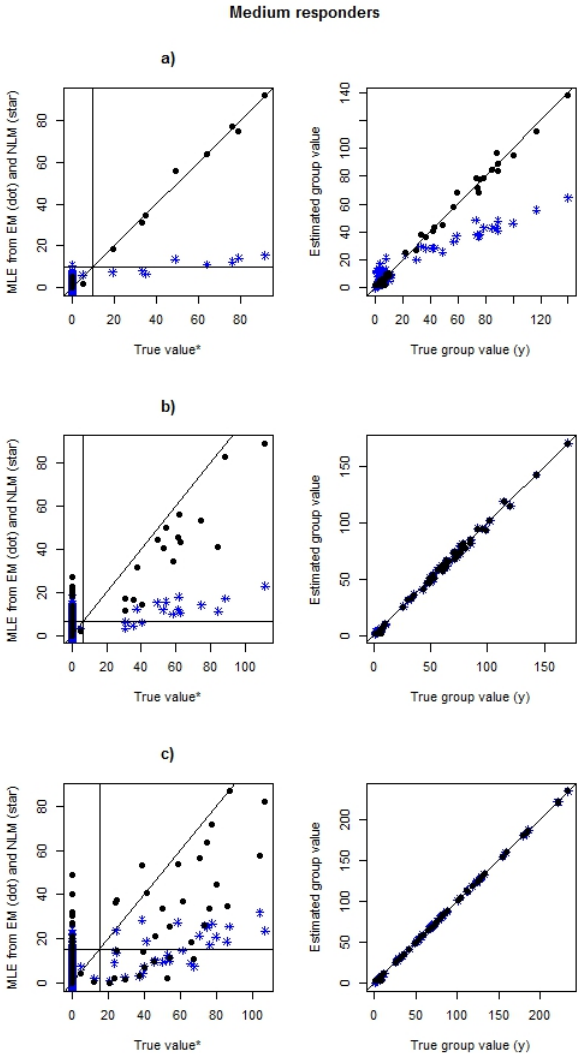

Figure 4: Three simulations of medium responding peptides that each generate Poisson data. Peptides grouped into 90 groups and 3 blocks using SOD. Vertical and horizontal line is twice the estimated noise (i.e. the positivity criterion). A line with slope 1 is also provided.
a) 200 peptides of which 4% are responding peptides.
b) 400 peptides of which 4% are responding peptides.
c) 400 peptides of which 8% are responding peptides.
∗"True value" denotes the estimate that would have been obtained if all individual responses had been observed.

.

References

[1] Anthony DD and Lehmann PV (2003). T-cell epitope mapping using the ELISPOT approach. *Methods 29: 260-269.*

[2] Dempster AP, Laird NM, Rubin DB (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Stat. Society, B, 39, 1-38.*

[3] Dittrich M, Lehmann PV (2012). Statistical analysis of ELISPOT assays. *Methods Mol Biol. 792:173-83.*

[4] Moodie Z, Huang Y, Gu L, Hural J, Self SG (2006). Statistical positivity criteria for the analysis of ELISpot assay data in HIV-1 vaccine trials., *J Immunol Methods. Aug 31;315(1-2):121-32.*

[5] Moodie Z, Price L, Gouttefangeas C, Mander A, Janetzki S, Löwer M, Welters MJ, Ottensmeier C, van der Burg SH, Britten CM (2010). Response definition criteria for ELISPOT assays revisited. *Cancer Immunol Immunother. Oct;59(10):1489-501.*

[6] Smith JG, Liu X, Kaufhold RM, Clair J, Caulfield MJ (2001). Development and Validation of a Gamma Interferon ELISPOT Assay for Quantitation of Cellular Immune Responses to Varicella-Zoster Virus. *Clin Diagn Lab Immunol. 8(5): 871–879.*

[7] Sundberg R (2012). Statistical Modeling by Exponential Families. Lecture Notes. Dep. Math. Stockholm University.

7 Appendix

7.1 R-code

```
#Simulated data

#Decide number of peptides and estimated percent responders

pep<- #assign no. peptides

resp<- #assign % responding peptides


#Vector of peptide names

pn<-1:pep


#Vector of expected values for peptides.

#m<-sample(c(rep(0,pep-floor(0.01*resp*pep)),rgamma(floor(0.01*resp*pep),6,0.2)))

m<-sample(c(rep(0,pep-floor(0.01*resp*pep)),rgamma(floor(0.01*resp*pep),6,0.1)))

noise<-5


#Simulate values.

#r1<-rpois(pep,m)

#r2<-rpois(pep,m)

#r3<-rpois(pep,m)


#Negative binomial with phi=2 and phi=10

phi<-2

k = (m+0001)/(phi - 1)

r1 = rnbinom(pep, mu = m+0.0001, size = k)

r2 = rnbinom(pep, mu = m+0.0001, size = k)

r3 = rnbinom(pep, mu = m+0.0001, size = k)
```

```
#3 blocks
blocks<-3
wells<-90/blocks
pepmax<-ceiling(pep/wells)

d1<-c(1:pep,rep(NA,pepmax*wells-pep))
d2<-as.vector(matrix(d1, nrow=wells, byrow=T))
d3<-as.vector(matrix(d2, nrow=wells, byrow=T))
d4<-as.vector(matrix(d3, nrow=wells, byrow=T))
d<-c(d2,d3,d4)

pool<-rep(1:90,each=pepmax)
x<-table(pool,d)
p<-t(x)/colSums(x)
p<-t(p)

mean<-(r1+r2+r3)/3

#Add noise
xx<-matrix(rep(0,3*pep),nrow=3)
xnoise<-rep(1,93)
x<-rbind(x,xx)
x<-cbind(x,xnoise)
p<-t(x)/colSums(x)
p<-t(p)
```

```
#Sum to group values
y<-c(x[1:30,1:pep]%*%r1,x[31:60,1:pep]%*%r2,x[61:90,1:pep]%*%r3)
rnoise<-rpois(93,noise)
es.noise<-mean(rnoise[91:93])
y<-rnoise+c(y,rep(0,3))

#Marie's algorithm
beta_0=rep(1,pep+1)
cond=1
 while (cond > .001) {
 mu=as.vector(x%*%beta_0)
 beta=(t(p)%*%(y/mu))*beta_0
 cond=sum(abs(beta-beta_0))
 beta_0=beta
 }

#NLM
fkn<-function(beta){
t(x%*%beta-y*log(x%*%beta))%*%rep(1,length(y))
}
as<-nlm(fkn,rep(1,pep+1))

#Compare estimates with responders.
positive<-length(mean[mean>=2*es.noise])
negative<-length(mean[mean<=2*es.noise])
tp.em<-length(beta[1:pep][beta[1:pep]>=2*es.noise & mean>=2*es.noise])
sens.em<-tp.em/positive
```

```
tp.nlm<-length(as$es[1:pep][as$es[1:pep]>=2*es.noise & mean>=2*es.noise])
sens.nlm<-tp.nlm/positive
tn.em<-length(beta[1:pep][beta[1:pep]<=2*es.noise & mean<=2*es.noise])
spec.em<-tn.em/negative
tn.nlm<-length(as$es[1:pep][as$es[1:pep]<=2*es.noise & mean<=2*es.noise])
spec.nlm<-tn.nlm/negative

#Real data
#Read design matrix
MP80<-read.table("Z://...//MP80.txt",header=F)
MP<-as.matrix(MP80)
MP<-as.numeric(MP)

#Read grouped data
y403<-read.table("Z://...//y403.txt",header=F)
y403<-as.matrix(y403)
y403<-as.numeric(y403)
y403<-c(y403,1,0,0,1)

#Create an 83*200 matrix indicating which peptides are present in each group
pool<-rep(1:80,10)

w<-table(pool,MP)
v<-matrix(c(w,rep(1,80)),nrow=80)
z0<-matrix(0,nrow=4,ncol=199)
z<-matrix(c(z0,rep(1,4)),nrow=4)
v403<-rbind(v,z)
```

```
x<-v403
p=t(x)/colSums(x)
p<-t(p)

y<-y403

#Obtain the estimates
beta_0=rep(1,200)
cond=1
 while (cond > .0001) {
 mu=as.vector(x%*%beta_0)
 beta=(t(p)%*%(y/mu))*beta_0
 cond=sum(abs(beta-beta_0))
 beta_0=beta
 }
```

7.2 Figures

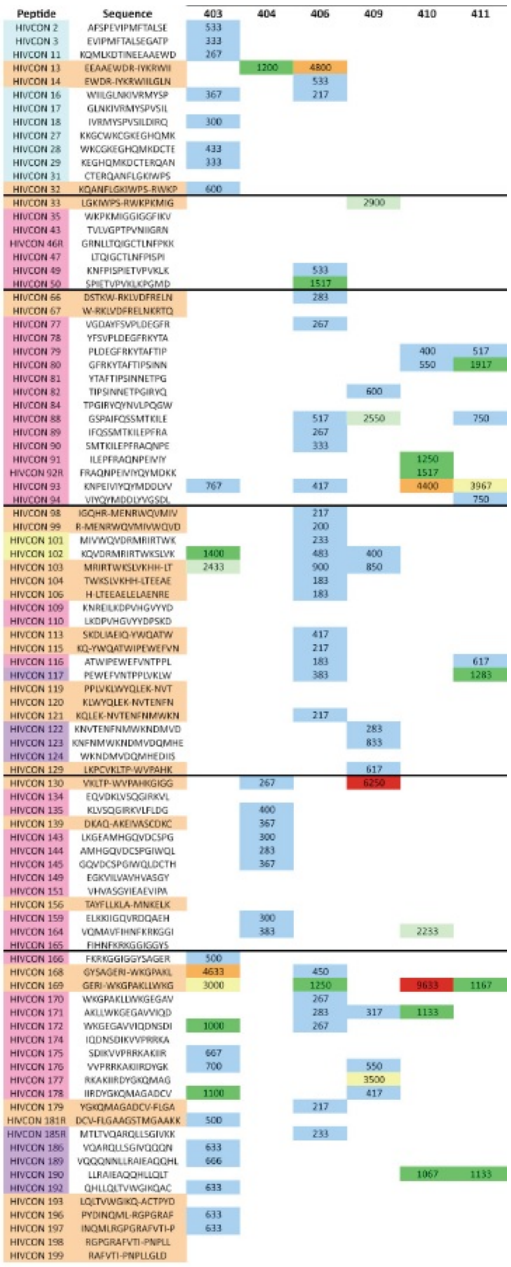

Figure 5: Estimates of 199 peptides for 6 subjects assayed using individual peptides. Any number between 1 and 199 not shown means that peptide was estimated a non-responder for all 6 subjects. Numbers in each box show spots per million cultured cells (higher number means stronger response). Subject ID is shown in the top row. This figure is a part of an image provided by prof. Tomas Hanke, University of Oxford.

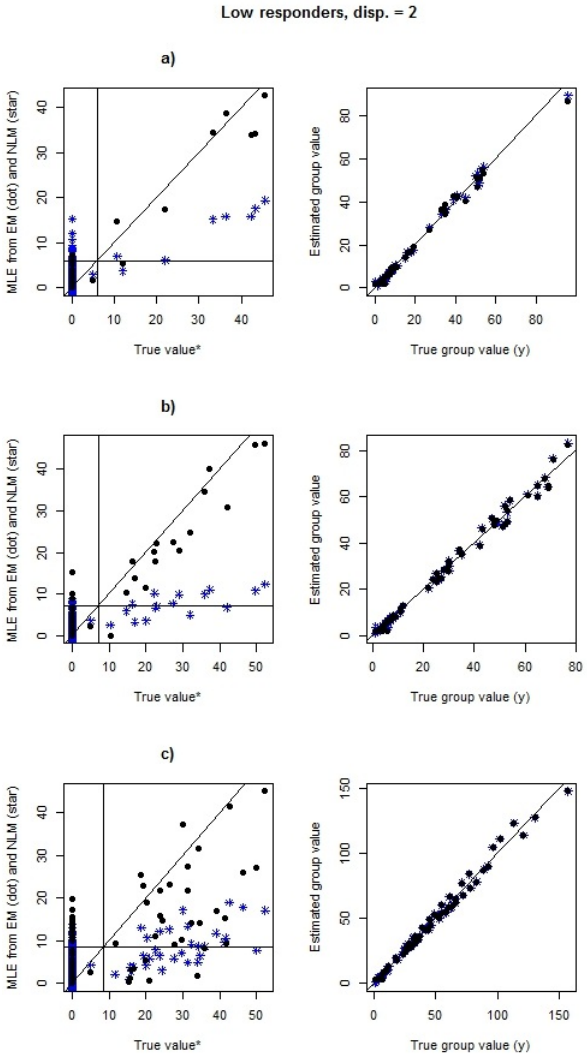

Figure 6: Three simulations of low responding peptides that each generate data with 2 times the variance of a Poisson variable. Peptides grouped into 90 groups and 3 blocks using SOD. Vertical and horizontal line is twice the estimated noise (i.e. the positivity criterion). A line with slope 1 is also provided.
a) 200 peptides of which 4% are responding peptides.
b) 400 peptides of which 4% are responding peptides.
c) 400 peptides of which 8% are responding peptides.
∗"True value" denotes the estimate that would have been obtained if all individual responses had been observed.

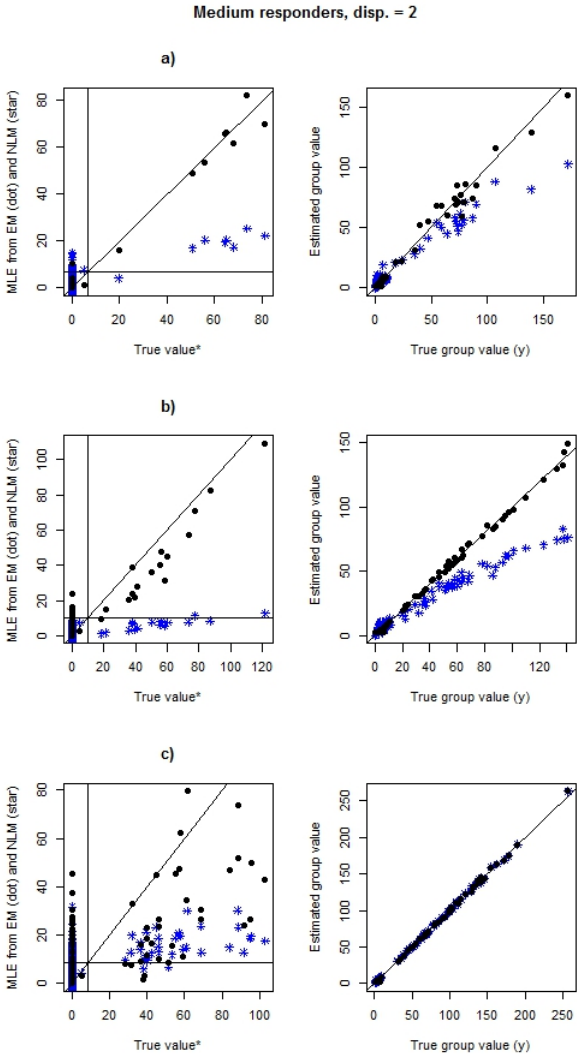

Figure 7: Three simulations of medium responding peptides that each generate data with 2 times the variance of a Poisson variable. Peptides grouped into 90 groups and 3 blocks using SOD. Vertical and horizontal line is twice the estimated noise (i.e. the positivity criterion). A line with slope 1 is also provided.
a) 200 peptides of which 4% are responding peptides.
b) 400 peptides of which 4% are responding peptides.
c) 400 peptides of which 8% are responding peptides.
∗"True value" denotes the estimate that would have been obtained if all individual responses had been observed.

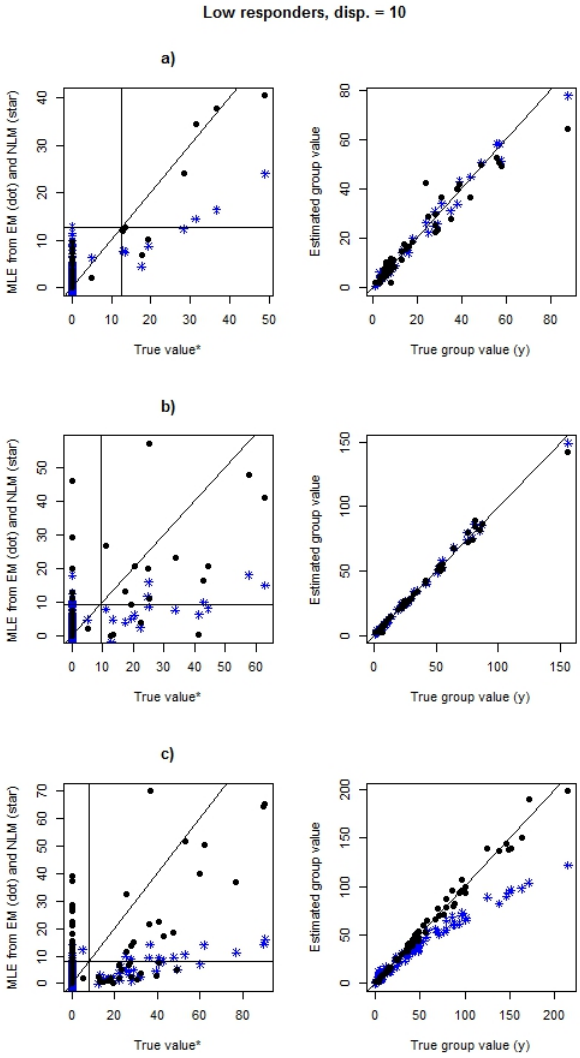

Figure 8: Three simulations of low responding peptides that each generate data with 10 times the variance of a Poisson variable. Peptides grouped into 90 groups and 3 blocks using SOD. Vertical and horizontal line is twice the estimated noise (i.e. the positivity criterion). A line with slope 1 is also provided.
a) 200 peptides of which 4% are responding peptides.
b) 400 peptides of which 4% are responding peptides.
c) 400 peptides of which 8% are responding peptides.
∗"True value" denotes the estimate that would have been obtained if all individual responses had been observed.

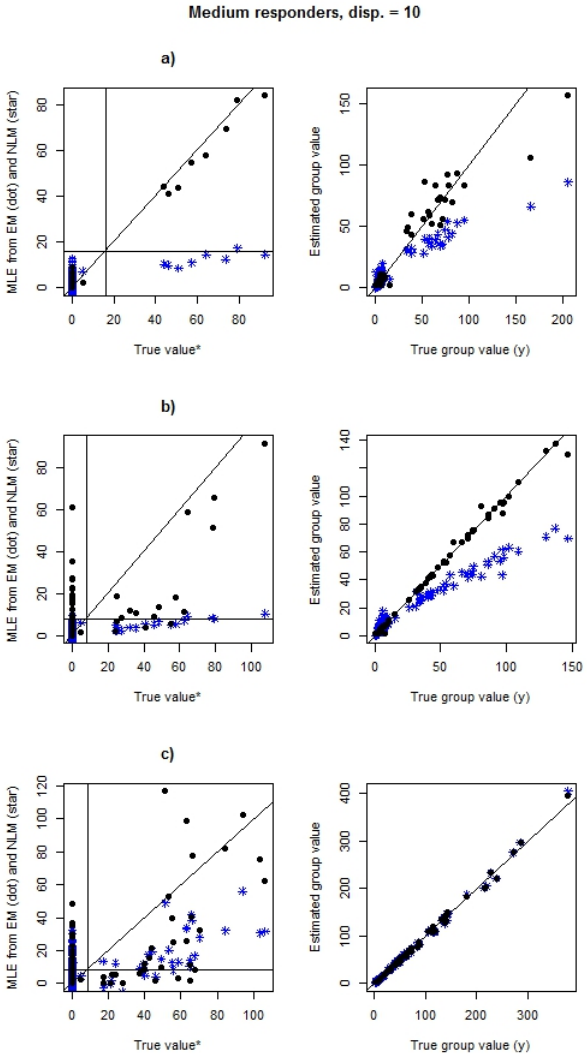

Figure 9: Three simulations of medium responding peptides that each generate data with 10 times the variance of a Poisson variable. Peptides grouped into 90 groups and 3 blocks using SOD. Vertical and horizontal line is twice the estimated noise (i.e. the positivity criterion). A line with slope 1 is also provided.
a) 200 peptides of which 4% are responding peptides.
b) 400 peptides of which 4% are responding peptides.
c) 400 peptides of which 8% are responding peptides.
∗"True value" denotes the estimate that would have been obtained if all individual responses had been observed.