Adaptive Methods for Classification of Biological Microarray Data from Multiple Experiments
A Technical Report

Goldschmidt Y.
CS and Applied Math

Sharon E.
CS and Applied Math

Quintana F.J.
Immunology

Cohen I.R.
Immunology

Brandt A.
CS and Applied Math

July 7, 2003

Abstract

We introduce a general classification tool, able to distinguish between two labelled data sets, and which we apply to real biological data as an example. For this, we develop a new normalization procedure integrating data obtained from multiple experiments, then qualitatively identify the candidate genes for differentiating disease states, which are finally used for diagnosing new samples. We demonstrate the algorithm by first training it on real biological data of the immune system obtained from multiple antigen arrays, and then diagnosing the correct disease state of new samples.

1 Introduction

The goal of our work is classification. We introduce a classifier able to distinguish between classes of entities, later used to assign a new unseen entity to the right class. We build this classifier in three stages - a normalization step, a learning step, and a classification step. First, we normalize all the gathered information from all experiments and combine it into one correctly scaled data set. Second, we identify the measurements showing the desirable distinction between the classes. And finally, in the classification step, we use the information found in previous steps to classify new unseen entities.

Typically, we are given a training set of n_e labelled entities, each with a vector of n_d measurements, and asked to assign a label to additional unlearned entities. A set of entities can be anything, e.g. set of tissue samples, blood samples, cells extracts, image segments (shapes) etc, while its vector of measurements describes the different parameters measured in the experiment for that entity, e.g. expression level of genes, cell content, shape moments, sets of correlated measurements etc. We call each measured factor a *determinant*, and assume the same determinants are measured for all entities, otherwise no comparison between entities or determinants can be done.

In this report, we describe the three steps of our algorithm, and show our results on immunological data of healthy samples vs. sample diagnosed with autoimmune diseases, that is, our data measures the binding of IgG and IgM antibodies from a serum sample to a set of antibodies.

This kind of protein data is more noisy than other kinds of microarrays data, and as such cannot be normalized by the simple re-scaling and filtering that are often used on DNA microarrays. Therefore, we first introduce a new normalization procedure. Once the data is properly normalized we aim to learn the determinants that can separate between 2 kinds of populations (e.g. healthy and sick). However, since the number of samples at hand is too small and noisy, simple hypothesis testing are not sufficient. Thus other method should be used. Golub et al. [1] find genes that are highly correlated with the separation of interest, by checking the means and standard deviations of the two separated populations over each gene. Using their method we did not find enough significant antigens, and thus got low prediction rates on our data. A recent work introduced by Quintana et al. [2] uses the coupled two-way clustering algorithm [3] on similar data, predicting Type-I Diabetes and proposing the relevant antigens detected by this disease.

In this work, we introduce a novel adaptive supervised procedure to detect the significant determinant for a disease, successfully classifying Type-I Diabetes, type-II Diabetes, Multiple Sclerosis and Beçket's Disease .

2 Adaptive Normalization of Multiple Experiments

The problem of normalization is a challenging subject for study when analyzing information obtained from multiple experiments. In fact, we found normalization to be a crucial necessity for achieving any useful conclusions especially from biological integrated experimental data. We introduce a novel adaptive iterative procedure, successful in normalizing the data at hand.

Our normalization procedure aims at re-scaling measurements obtained from multiple experiments each may have been done in different conditions or by different protocols, and moreover we wish to eliminate noise in the data. To achieve the correct re-scaling, we integrate into the normalization procedure any known support information in a supervised manner. In other words, all global parameters known to cause a variation in the data over different experiments and within an experiment are integrated simultaneously into the normalization. Some global parameters can be deduced from the data itself, such as the total sum of reactions for an entity, or given from the outside, e.g. experiments temperature, origin of the entities (women vs. men, adults vs. children, nationality etc). For simplicity of presentation, we describe the normalization procedure using only one such parameter, and later generalize. In addition, we need to neutralize the effect of outlier measures, that is, a data point that peaks or troughs just for one or very few entities, and introduces great noise to the data.

Suppose we have n_e entities. First, we divide the entities into non-intersecting populations according to their pre-known labelling. Usually, some of the entities are controls, that is, entities with wildtype or normal phenotype. We denote their size by l, and the others by $\bar{l}$ such that $l + \bar{l} = n_e$.

The normalization procedure finds the correct re-scaling by learning the *normal behavior curve*, that is, the behavior of the control population, and adjusts all other measurements according to that curve.

In the absence of a known a-priory model to describe this behavior in our data, we use the dependence of the data on the list of global parameters, and find the curve by best-fitting a piecewise linear curve, as follows.

2.1 Finding The Normal Behavior Curve

Given a matrix of n_d determinants over l control entities, each entry ij corresponds to the measurement of determinant i over the j'th entity. We normalize each determinant at a time. Denote by $a = (a_1, \ldots, a_l)$ the measurements of the currently analyzed determinant, and by p_j $(1 \leq j \leq l)$ the global parameter corresponding to a_j (e.g. $p_j = \sum_{i=1}^{n_d} a_{ij}$, i.e. the sum of all determinants of entity j).

We regard each pair (a_j, p_j) as a point in a two dimensional *parameter space*, in which p_j is the x-coordinate, and a_j is the y-coordinate . To learn the general behavior of the control population we invoke an adaptive piecewise linear least square (LLS) fitting. For this, we adaptively divide the x-axis into T segments such that there are n data points in each segments, and where n is large enough to avoid over-fitting to the noise (See Sec. 3.2, where we present a scheme for choosing statistically significant segments but in a different context).

Next, we fit a straight line to each segment such that the lines intersect at the borders of the segments, and minimize the sum of squared distances between the data points and the lines. To define such lines we find $T+1$ points, $(x_1, y_1), \ldots, (x_{T+1}, y_{T+1})$, through which the lines go. The x-values of the borders, $x_1, \ldots, x_{T+1}$, are pre-fixed, and we find $y_1, \ldots, y_{T+1}$ by minimizing

$$S = \sum_{j=1}^{l} (a_j - \hat{a}_j)^2, \tag{2.1}$$

where $(\hat{a}_j, p_j)$ are on the fitted lines.

2.2 Iterative Reduction of Outliers' Effect

To eliminate the effect of outliers on the normalization, we assign a weight to each data point, determined by the distance of the point to the fitted curve, and run the above procedure in iterations but altering Eq. (2.1) to consider weights,

$$S = \sum_{j=1}^{l} w_j (a_j - \hat{a}_j)^2. \tag{2.2}$$

For each iteration the weights are defined by the previous iteration, until the piecewise linear curve does not change much between iterations. We define the weight as follows (although other weighting methods can be used),

$$w_j = \begin{cases} 1 & \text{if} \quad d_j < \text{threshold} \\ 0 & \text{otherwise ,} \end{cases} \tag{2.3}$$

where d_j is the distance of a_j from the fitted curve, divided by the standard deviation of the distances of points in the a_j's segment to their line. Convergence is usually achieved within 2 to 10 iterations, depending on the threshold in Eq. (2.3). Typically, we take a threshold of 2.5.

In case lines of successive segments appear as one continues line, there is no point in defining multiple segments, and a linear regression model suffices. This is often the case with DNA microarrays. However, in view of our results, this is not the case with antigen data, where the above local adaptive estimation is indeed necessary.

2.3 Normalizing Data Points

The normalized value for each measurement, is its scaled distance from the normal behavior curve. For every measurement a_k that did not participate in determining the fitted lines (i.e. sample outside the control sub-population), we calculate p_k, and find the segment τ it falls in, and the corresponding normal-behavior value $\hat{a}_j = a(p_k)$. Then we define its normalized value $\tilde{a}_k$ as follows:

$$\begin{aligned} d_k &= a_k - \hat{a}_k, \\ \sigma_\tau &= (\sum_{\substack{j \in C \\ p_j \in \tau}} {d_j}^2)^{\frac{1}{2}}, \\ \tilde{a}_k &= \frac{d_k}{\sigma_\tau}. \end{aligned} \tag{2.4}$$

That is, the normalized value is the difference between the original value (a_k) and the corresponding fitted-curve value ($\hat{a}_k$), divided by the L_2-average of the distances of control entities (denoted by C) from the line in its segment τ.

In case an entity, not of the controls class, has fallen outside the segments borders, we would extend the line in the closest segment to the needed range. Or else, we can define the borders by all the population in the first place.

After this basic normalization has been applied, we standardize each determinant to obtain a similar range of measure for all determinants. Namely, for each determinant measurement, we subtract the mean of (normalized) measurements of that determinant, and divide by the standard deviation of these measurements, to obtain a distribution with zero mean and standard deviation of 1. This last calculation is very important for later classification, since similar range of measurements for all determinants ensure, for example, that one highly varying determinant will not dominate the procedure.

2.4 Larger Number of Global Parameters

For integrating any number of global parameters into the normalization procedure, we build a multidimensional parameter space, that is, an axis for each parameter. Then, we partition the domain in that space, that contain all measured parameters, into sub-domains (e.g., triangles in a two-parameter case) and best-fit a surface (e.g., continuous surface, linear over each triangle) to the data set.

3 Identifying Significant Determinants

Many determinants may have nothing to do with the classes we aim to distinguish, therefore we filter out only the ones we are interested in, namely, those who show a significant difference between the classes we wish to separate. For this we propose a test that outperforms simpler hypothesis tests (t-test, ranksum etc.). Our test is related to the χ^2-test, which is non-parametric and assumes nothing on the distribution of the classes, but in addition, our test is suited also for having low number of samples.

Again, we concentrate on the values of one determinant at a time, and assume there are n_e entities, and l entities out of n_e for a specific label L. We estimate the probability *to obtain the given distribution of determinant-values for entities coming from this label, provided that the distribution density generating these values is identical to the distribution of this determinant for the whole population.* Low probability means that the determinant values for this label are distributed differently than the determinant values for the whole population. The lower this probability is the more the value of this determinant will be important in our classifier to determine whether a specific entity is coming from that specific label.

3.1 Estimating Distributions Assuming Multinomial Probabilities Over Bins

Given a partition of the determinant-value axis into m disjoint bins $\{b_i\}_{i=1}^m$, we denote the total number of entities falling in bin b_i by n_i, and define $n = \sum_{i=1}^m n_i$, (i.e. $n = n_e$). For a specific label L, we denote the number of labelled entities falling in bin b_i by l_i, so $l = \sum_{i=1}^m l_i$. We consider each entity as a random variable with a probability $p_i = n_i/n$ to fall in bin b_i.

According to the multinomial distribution, we have that

$$P(\underline{l}) = P(l_1, l_2, ..., l_m) = \frac{l!}{l_1! l_2! \cdots l_m!} p_1^{l_1} \cdots p_m^{l_m}. \tag{3.1}$$

This coefficient in Eq. (3.1) is expensive to compute, instead we calculate $P(\underline{l})$ by the following approximation.

Let $\{x_k\}_{k=1}^l$ be the determinant values coming from label L. We associate with every such value a vector $X^{(k)}$ whose length is the number of bins m, defined by

$$X_i^{(k)} = \begin{cases} 1 & \text{if} \quad x_k \in b_i \\ 0 & \text{if} \quad x_k \notin b_i \ . \end{cases} \tag{3.2}$$

Assuming the labelled data to be distributed like the entire population we have that each $X_i^{(k)}$ has probability p_i to be one. Hence $X^{(k)}$ are i.i.d (identically, independently distributed) random variables, each being an m-vector, with $E(X^{(k)}) = \underline{\mu} = (p_1, p_2, ..., p_m)^T$, and the covariance matrix

$$\Sigma = \begin{pmatrix} p_1(1-p_1) & -p_1p_2 & \cdots & -p_1p_m \\ -p_2p_1 & p_2(1-p_2) & \cdots & -p_2p_m \\ \vdots & \vdots & & \vdots \\ -p_mp_1 & -p_mp_2 & \ldots & p_m(1-p_m) \end{pmatrix}. \tag{3.3}$$

Since $\underline{l} = (l_1, l_2, ..., l_m)^T = \sum_{k=1}^{l} X^{(k)}$, for large l/m we can use the *Central Limit Theorem* to approximate $P(l_1, l_2, ..., l_m)$ by a multivariate normal distribution

$$P(l_1, l_2, ..., l_m) \approx \frac{1}{(2\pi)^{m/2}\sqrt{det\Sigma}} \exp\left(-\frac{1}{2}\left(\underline{l}/l - \underline{\mu}\right)^T \Sigma^{-1} \left(\underline{l}/l - \underline{\mu}\right)\right). \tag{3.4}$$

Since actually $det(\Sigma) = 0$ (the sum of every one of its columns is zero), we have to understand the value of the approximating expression in Eq. (3.4) by taking a limit in Eq. (3.4). This will yield

$$P(l_1, l_2, \ldots l_m) = C \exp\left(-\frac{1}{2}\sum_{i=1}^{m}\frac{1}{p_i}\left(\frac{l_i}{l} - p_i\right)^2\right), \tag{3.5}$$

where, if $l \gg 1$, $C \approx \prod_{i=1}^{m} \frac{1}{l\sqrt{2\pi p_i}}$ to ensure that $\sum_{l_1, l_2, \ldots, l_m} P(l_1, l_2, ..., l_m) = 1$.

In particular, choosing the bins $\{b_i\}_{i=1}^{m}$ so that in each bin we have the same number of entities $n_i \equiv \tilde{n} = n/m$, coming from the entire population, we have that $p_i = 1/m$ and hence, if $l \gg m$,

$$P(l_1, l_2, ..., l_m) \approx C \exp\left(-\frac{m}{2}\sum_{i=1}^{m}\left(\frac{l_i}{l} - \frac{1}{m}\right)^2\right). \tag{3.6}$$

Low P means that L is distributed differently from the entire population, thus we choose as the important determinants those with lowest values of P.

The statistical rules for choosing the number of entities in a bin $\tilde{n}$ are described in Sec. 3.2 below. An adjustment to Eq. (3.6) is needed in case l is large in comparison to $n_e - l$, the size of the rest of the population, and dominates the probability p_i to fall in a bin. This adjustment is described in Sec. 3.3.

3.2 Building statistically significant and correctly scaled Bins

Prior to comparing the distribution of label L and the entire population, we need to generate an appropriate binning over which to compare.

Given a determinant's values over the entities of the entire population, we first generate a partition of the value axis into m disjoint bins, so that we get a *similar* number of entities in each bin, $n_i \approx \tilde{n} = n/m$. We refer to this partitions of the value axis as an "equi-n_i binning". In order to get a meaningful calculation, $\tilde{n}$ should be statistically significant, i.e. considerable larger than its predictable statistical error. To this end, we first calculate the statistical variation in the number of entities in a bin (Sec. 3.2.1), and then merge bins that are not significant enough with respect to a specific label (Sec. 3.2.2).

3.2.1 The statistical variation in a bin

Concentrating on n_i, the number of entities from the population falling in bin b_i, we recall that the binomial variable $X_i^{(k)}$ equals 1 whenever the associated entity x_k falls in bin b_i and zero otherwise. We formally denote the unknown probability with which $X_i^{(k)} = 1$ by p_i.

Since $P(n_i = k) = \binom{n_e}{k}(p_i)^k(1-p_i)^{n_e-k}, \quad E(n_i) = n_e p_i, \quad$ and $V(n_i) = n_e p_i(1-p_i)$, the standard deviation of entities falling in a bin satisfies

$$V^{\frac{1}{2}}(n_i) \leq \sqrt{E(n_i)}. \tag{3.7}$$

The practical conclusion from this is that we cannot trust n_i that is not significantly larger than $\sqrt{n_i}$. We can start trusting statistically only $n_i \geq 8$. This is the main different to a χ^2 test, which usually uses $n_i = 5$.

The equi-n_i binning is generating the *correct scaling* of the determinant value axis. That is, each bin is defining a unit on that axis. When doing so we can reliably consider $\frac{n_i}{n}$ to represent the probability of the determinant value to fall in bin b_i, and hence we can reliably represent the distribution of the determinant values over the bins. We consider the distribution to be uniform within each bin.

3.2.2 Generating correctly-scaled bins for different labels

Given no previous knowledge about the distribution of the entities coming from L we use the equi-n_i binning generated by the entire population as the basic units of the determinant value axis. We assume uniform probability density value over those bins, and approximate the probability density of the entities coming from any L, using the bins of the entire population as the point of reference. We do so for each label at a time by looking at the number of entities of that label in each bin while merging bins that are not "informative enough" for that label, as follows.

Given the equi-n_i units, for any specific label L we generate a new binning $\{b_j^{(L)}\}_{j=1}^{m_L}$. For each $j \in \{1, 2, ..., m_L\}$, $b_j^{(L)}$ is a union of several consecutive bins of the entire population (equi-n_i bins), such that $b_i^{(L)} \cap b_j^{(L)}$ is empty for $i \neq j$, and such that all of the original bins take part in the new binning. We denote the number of entities from label L in each bin $b_j^{(L)}$ by $l_j^{(L)}$.

There is one reason for wanting to enlarge bins (enlarging each $l_j^{(L)}$ while decreasing μ_L) and one for keeping the bins not too large. In each bin $b_j^{(L)}$ we must have $l_j^{(L)}$ large compared with its expected error $\sqrt{l_j^{(L)}}$, and at least $l_j^{(L)} \geq 8$ (as above in Sec. 3.2.1). On the other hand, the shorter the bins, the better resolved probability density we get.

To optimize between these two reasons for each L, we define a greedy "bin-merging" process that merges neighboring bins of $\{b_i\}_{i=1}^{m}$ into $\{b_j^{(L)}\}_{j=1}^{m_L}$. We will first define a criterion for merging two consecutive bins. The process then sequentially visits the bins, for each bin judging whether it should be merged to the next bin in line, and keep merging the resulting bin to its next until this is prohibited by the criterion. Then, moving to the next bin considers its merging to its next in line, and so on until the last bin is met.

More specifically, Consider two sequential bins b_j, b_{j+1}, with l_i, l_j entities (from label L) respectively. The probability *density* function in b_j will be defined to be

$$p_j = \frac{l_j/l}{\kappa_j}, \tag{3.8}$$

where κ_j denotes the number of equi-n_i bins contained in b_j. Similar probability p_{j+1} is defined for b_{j+1}. We do not merge the two neighboring bins b_j and b_{j+1} only if the difference between p_j and p_{j+1} is significant compared with the estimated standard deviation of

$$p_{[j,j+1]} = \frac{\left(l_j + l_{j+1}\right)/l}{\kappa_j + \kappa_{j+1}}, \tag{3.9}$$

which is the probability density function estimated in their unified bin $b_{[j,j+1]} = b_j \cup b_{j+1}$.

The standard deviation in the number of entities from label L falling in bins b_j, b_{j+1} and $b_{[j,j+1]}$ is (by Eq. (3.7)) $\sqrt{l_j}$, $\sqrt{l_{j+1}}$ and $\sqrt{\left(l_j + l_{j+1}\right)}$, respectively. Thus, our criteria for *not merging* the bins b_j and b_{j+1} into $b_{[j,j+1]}$ is first that l_j and l_{j+1} are large enough to be statistically meaningful, and second that

$$\left|p_j - p_{j+1}\right| > V^{\frac{1}{2}}\left(p_{[j,j+1]}\right), \tag{3.10}$$

meaning that the difference between the values of the probability density function in the two separate bins is bigger than the expected error in this value in their unified bin. Explicitly, the *non-merging* criteria in Eq. (3.10) above can be re-written as follows

$$\left|\frac{l_j}{\kappa_j} - \frac{l_{j+1}}{\kappa_{j+1}}\right| > \frac{\sqrt{l_j + l_{j+1}}}{\kappa_j + \kappa_{j+1}}. \tag{3.11}$$

We keep going over the bins until no two consecutive bins are merged any more. At last, we end up with a new binning we denote by $\{b_j^{(L)}\}_{j=1}^{m_L}$.

Now all is left is to superpose the new binning of each label over the original equi-n_i binning of the entire population, and to compute the probability in Eq. (3.6). These are explained next.

3.2.3 Comparing the labelled and the entire-population data distributions over correctly-scaled bins

For each $i \in \{1, 2, ..., m\}$ and $j_i \in \{1, 2, ..., m_L\}$ such that $b_i \subseteq b_{i_j}^{(L)}$ we define the "induced probability density function" of L over the equi-n_i bins by

$$\tilde{p}_i^{(L)} = p_{j_i}^{(L)} = \frac{l_{j_i}^{(L)}/l}{\kappa_{j_i}^{(L)}}, \tag{3.12}$$

and the "induced entity counting" by

$$\tilde{l}_i^{(L)} = l_{j_i}^{(L)}/\kappa_{j_i}^{(L)}. \tag{3.13}$$

According to the construction of the correctly-scaled binning $\{b_j^{(L)}\}_{j=1}^{m_L}$ of label L, the statistics in each bin is enough to reliably estimate the probability density function over these bins by the uniform values $\{p_j^{(L)}\}_{j=1}^{m_L}$. We can therefore trust the induced entity counting from Eq. (3.13), i.e. $\{\tilde{l}_i^{(L)}\}_{i=1}^{m}$, to replace $\{l_i\}_{i=1}^{m}$ in Eq. (3.6).

Lastly, since the estimated statistical deviation in $\tilde{l}_i^{(L)}$ is simply $\sqrt{l_{j_i}^{(L)}/\kappa_{j_i}^{(L)}}$, then the statistical deviation of $\sum_{i=1}^{m}\left(\frac{\tilde{l}_i^{(L)}}{l}-\frac{1}{m}\right)^2$ in Eq. (3.6) can be roughly estimated by the sum of deviations $\sum_{i=1}^{m}\left(\frac{\sqrt{l_{j_i}^{(L)}/\kappa_{j_i}^{(L)}}}{l}\right)^2 \leq \frac{1}{l}$, where the bound $\frac{1}{l}$ is achieved in the case where $\kappa_{j_i}^{(L)} \equiv 1$, and gets tighter when $\{\kappa_{j_i}^{(L)}\}_{i=1}^{m}$ are larger. And so we can work less by looking for large values of

$$l\sum_{i=1}^{m}\left(\frac{\tilde{l}_i^{(L)}}{l}-\frac{1}{m}\right)^2. \tag{3.14}$$

instead of using Eq. (3.6).

3.3 Adjustment to Large Labelled Population

In most cases in real life, there is only a small amount of entities of the population we are required to distinguish (the one labelled by L), that is $l << n_e$. However, in experiments l may be as large as the rest of the population, that is $l \approx n_e - l$. In that case p_i (the probability to fall in a bin according to the entire population, see Sec. 3.1), is greatly affected by the measures of the entities labelled by L, rather than by the rest of the population. Consequently, Eq. (3.6) is greatly biased. An adjustment to Eq. (3.6) substitutes p_i in Eq. (3.5) by the distribution of the rest of the population in each bin, rather than by $1/m$ which is the distribution of the entire population. In other words we compare entities of label L to the rest of the population, which we denote by $\bar{L}$. More specifically, recall that l_i denotes the number of L labelled entities falling in bin b_i. Similarly, denote by $\bar{l}_i$ the number of entities labelled by $\bar{L}$ in bin b_i, $\bar{l} = \sum_{i=1}^{m}\bar{l}_i$. Next, instead of using $1/m$ to describe the distribution of the entire population in a bin in Eq. (3.6), we use $p_i = \bar{l}_i/\bar{l}$. Thus,

$$P(l_1, l_2, ..., l_m) \sim \exp\left(-\frac{m}{2}\sum_{i=1}^{m}\left(\frac{l_i}{l}-\frac{\bar{l}_i}{\bar{l}}\right)^2\right). \tag{3.15}$$

Now we can again generate equi-entities bins as described in Sec. 3.2, and generate the correctly scaled bins for each label, that is, also for label $\bar{L}$. Note, that the general equi-entities binning is generated by the entire population as before.

To detect significantly low values of $P(l_1, l_2, ...l_m)$ in Eq. (3.15) we use the same reasoning as with the original equation and can simply check whether

$$\sum_{i=1}^{m}\left(\frac{\tilde{l}_i^{(L)}}{l}-\frac{\tilde{\bar{l}}_i^{(\bar{L})}}{\bar{l}}\right)^2 > \frac{l+\bar{l}+2\sqrt{l\cdot\bar{l}}}{l\cdot\bar{l}}. \tag{3.16}$$

This since the statistical deviation in $\tilde{l}_i^{(L)}$ is simply $\sqrt{l_{j_i}^{(L)}/\kappa_{j_i}^{(L)}}$, and similarly the statistical deviation in $\tilde{\bar{l}}_i^{(\bar{L})}$ is $\sqrt{\bar{l}_{j_i}^{(\bar{L})}/\kappa_{j_i}^{(\bar{L})}}$. Meaning that the statistical deviation in $\sum_{i=1}^{m}\left(\frac{\tilde{l}_i^{(L)}}{l}-\frac{\tilde{\bar{l}}_i^{(\bar{L})}}{\bar{l}}\right)^2$ can be roughly estimated by the sum of deviations $\sum_{i=1}^{m}\left(\frac{\sqrt{l_{j_i}^{(L)}/\kappa_{j_i}^{(L)}}}{l}+\frac{\sqrt{l_{j_i}^{(\bar{L})}/\kappa_{j_i}^{(\bar{L})}}}{\bar{l}}\right)^2 \leq \frac{l+\bar{l}+2\sqrt{l\cdot\bar{l}}}{l\cdot\bar{l}}$, where the bound is approached for $\kappa_{j_i}^{(L)} \equiv \kappa_{j_i}^{(\bar{L})} \equiv 1$.

Notice that for a large enough sampling $\sqrt{l\cdot\bar{l}}$ is negligible compared to $l+\bar{l}$ and can be omitted, and so, we rewrite Eq. (3.16) so that important determinants are those having the largest values of

$$\frac{l\cdot\bar{l}}{l+\bar{l}}\sum_{i=1}^{m}\left(\frac{\tilde{l}_i^{(L)}}{l}-\frac{\tilde{\bar{l}}_i^{(\bar{L})}}{\bar{l}}\right)^2. \tag{3.17}$$

4 Diagnosis: Assigning Labels to Unlearned Entities

The significant determinants found in Sec. 3 define a profile for each disease class. For diagnosing new unseen entities, we compare their normalized determinant composition to the diseases profiles, that is, look at the distinguishing determinants only. There are many methods for doing the prediction by the relevant determinants, such as decision trees - constructing the tree based on the significant determinants, clustering - assign the new samples to the class of the cluster it fallen in, or clustering trees - using an hierarchical clustering output as a decision tree based on the underlying features of the clusters. Here we chose to use a Bayesian classifier, which gives good prediction rates on out data.

First, we normalize the new entity by adjusting it to the normal behavior curve already used to normalized the training set (see Sec. 2.3), then use the Bayesian classifier to produce a probability for the new entity to come from each of the classes previously learned.

4.1 Bayesian Classifier over Important Determinants

We denote by the vector x_k the set of K normalized measurements of *significant* determinants of *labelled* entity k. That is, $x_k = (x_{k_1}, x_{k_2}, ..., x_{k_K})^T$. For each new and *unlabelled* entity $x = (x_1, x_2, ..., x_K)^T$ and each label L we aim to compute $P(x \in L | x = (x_1, x_2, ..., x_K)^T)$, that is the probability that the entity x is coming from the data characterized by the already observed samples label by L (We would denote this in short by $P(L|x)$). We use the Bayes formula, by which

$$P(L|x) = \frac{P(x|L)P(L)}{P(x|L)P(L)+P(x|\bar{L})P(\bar{L})}, \tag{4.1}$$

where $\bar{L}$ labels the part of the population complementary to L.

The priors $P(L)$ and $P(\bar{L})$ are usually given, or can be easily learned from the labelled data. The probability $P(x|L)$ can be described as the probability to observe the set of determinants measurements of x, when x is assumed to belong to label L, and similarly for $P(\bar{L})$. By comparing the distribution functions of measurements originated from L and of measurements originated from $\bar{L}$ we can answer this question for x. In fact, it shows in Eq. (4.1) that in order to estimate $P(L|x)$ we only need to estimate the probability ratio

$$R(x|L) = \frac{P(x|L)}{P(x|\bar{L})}. \tag{4.2}$$

The distribution of a label over the entire space is hard to estimate, but the ratio in Eq. (4.2) can be easily estimated in the vicinity of x. We do it as follows.

4.1.1 Estimating the Bayes Ratio

We build a K-dimensional space, where K is the number of the significant determinants, and each axis represents a different distinguishing determinant. We call that space the *Determinant Space*, and place all samples in the space according to their normalized determinants values, along with the new unlabelled entity x.

To estimate the distribution functions in the vicinity of x, we consider for each label, the n closest points to x (measured by their Euclidian distance to x), and define the *neighborhood* of x to be the smallest *ball* containing the closest points to x, with x at its center (see fig. 1). Its radius is the distance between x and the farthest point in the neighborhood. The smaller the radius of the neighborhood the bigger the chance that entity x came from the same distribution. To obtain a statistically significant estimation of the distribution function in the neighborhood of x, we demand the neighborhood to contain enough entities, so that the predictable statistical error in the estimation of the distribution will be small (for the same reasoning as in Sec. 3.2.1). A neighborhood of $n \geq 8$ may usually suffice. On the other hand, we also limit the maximal number of points defining the neighborhood, since we prefer as small neighborhoods as possible, where the distribution estimation is more accurate.

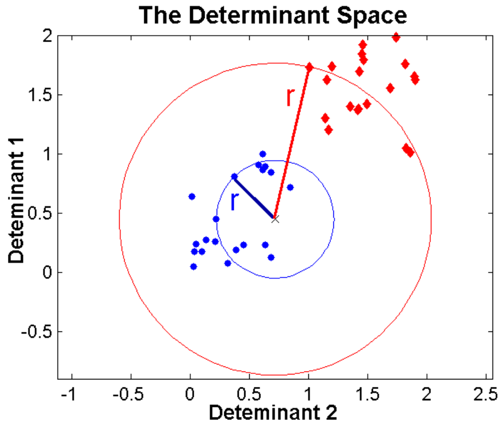

Figure 1: A determinant space for two important determinants. Two types of cases are presented: red dots represents one class of entities, and blue the other. the x point represents the entity to diagnose. Colored circles demonstrate the neighborhoods consisting of 10 entities each.

Denote the neighborhood size of label L by n_L and of $\bar{L}$ by $n_{\bar{L}}$. (usually $n_L = n_{\bar{L}}$, but if there is more than one point with the maximal distance in the neighborhood, we take it also, and redefine the neighborhood size accordingly, so now there may be a case where $n_L \neq n_{\bar{L}}$.)

We estimate the required ratio by

$$R(x|L) = \frac{P(x|L)}{P(x|\bar{L})} = \frac{\frac{n_L/|L|}{(r_L)^{d_L}}}{\frac{n_{\bar{L}}/|\bar{L}|}{(r_{\bar{L}})^{d_{\bar{L}}}}} \quad , \tag{4.3}$$

where r is the radius of the neighborhood, and d is the embedded dimension of the neighborhood, so that r^d is simply proportional to the volume of the neighborhood (i.e. volume of a ball with radius r in d dimensions). d inside the neighborhood may be smaller than K, therefore we calculate d by neighborhood principal component analysis [4], taking it to be the number of components with eigenvalues which are not much smaller then the maximal eigenvalue.

To estimate $P(L|x)$ all is left is to plug $R(x|L)$ in the Bayes formula in Eq. (4.1).

4.2 Classification over an Ordinal Space

The problem of outliers may be relevant also in the classification process described above. Outliers rise as a problem usually when having only a small number of entities to learn from for one of the labels (e.g. we usually have only a small number of sick samples in hand), and the majority of them must be always taken for the neighborhood of the point to predict. Consider the case that one of the entities contains an outlier in one of the determinants found to be important and therefore used in the classification. The outlier may introduce a bias, resulting in many false positives or negatives predicted by the Bayes formula.

Next, we propose one optional procedure to overcome the effect of outliers in the classification process as well as correct any misscaling of determinants.

For any determinant, we substitute each of its values by their *ordinal value*, that is, the ordinal location (rank) of each value among all the determinant's values.

Now we invoke the Bayesian classification procedure described in Sec. 4.1 using the ordinal values instead of the real values. For each new and *unlabelled* entity, we also denote its determinants values by their ordinal location among the other entities. By using ordinal values, we change the placement of the points on the parameter space, so that each point would have a fix scaling compared to other points, and different determinants will have similar scaling. Outliers may occur at the edges of the order, but would not enlarge the neighborhoods as much as when using real values. Indeed, this adjustment improves the results for small learning sets.

5 results

We analyze 4 different data sets from 4 different ELISA (Enzyme-Linked Immunosorbent Assay)[5] experiments, studying 4 kinds of autoimmune diseases - type-I diabetes, type-II diabetes, Beçket's Disease , and Multiple Sclerosis, each with a set of healthy samples used as a control set, except for Beçket's Disease where the control are some healthy and some sick with a disease with similar phenotype to Beçket's Disease [6]. For each sample 114 measurements are taken, 57 measures of binding to IgG+IgM antigens, and 57 of binding only to IgM antigens, which we normalize separately. Table 1 shows the number of samples in each experiment along with the disease codes and colors that are used in our analysis. Fig. 3 displays all measurements before and after the normalization colored according to the experiment they were taken from, and Fig. 4 shows an example of a normalization curve. Notice that each experiment has it's own scaling and therefore different experiments are comparable only after normalization. For the purpose of testing our algorithm, we take one

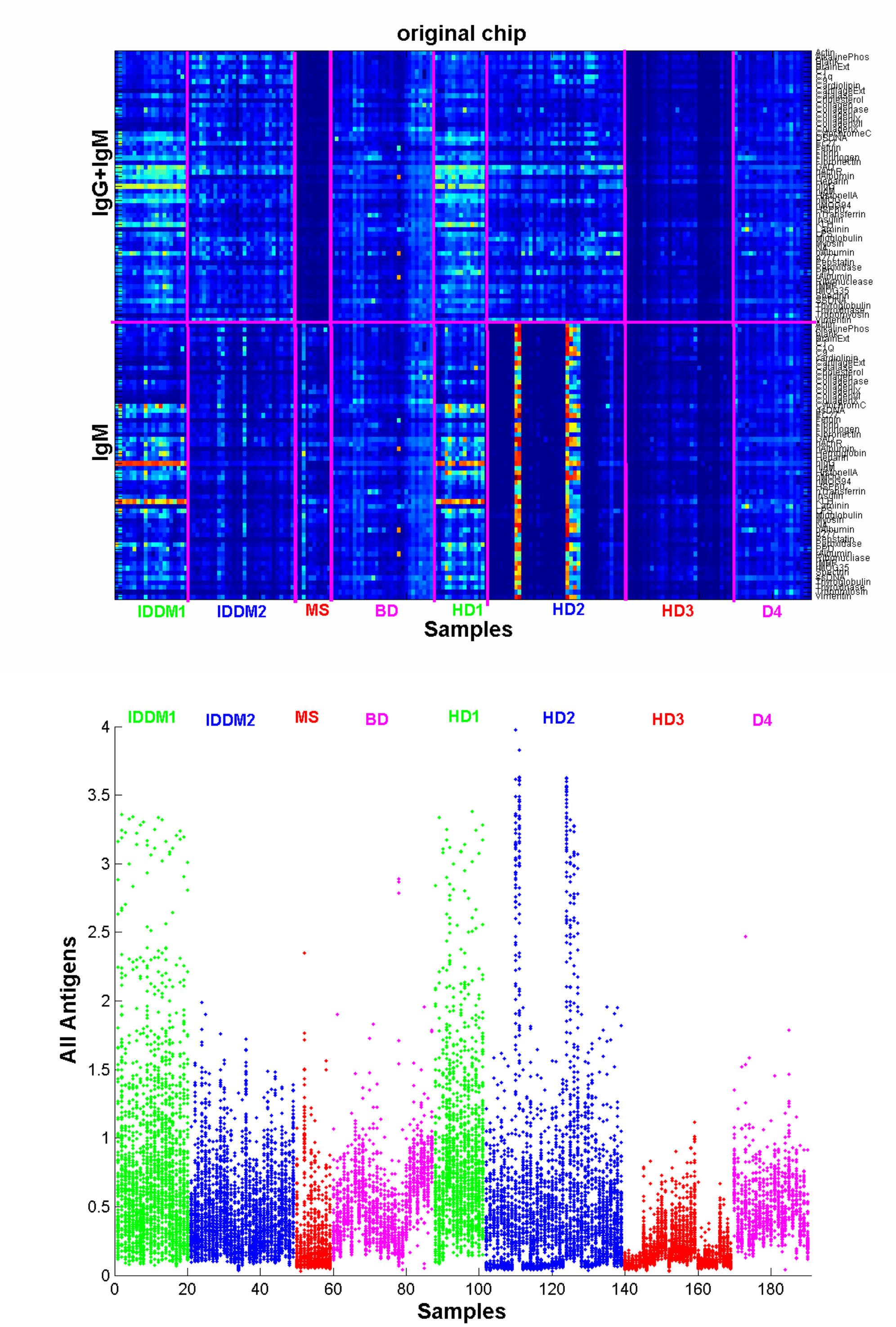

Figure 2: original un-normalized data, up: chip view, down:dots view - For each sample, each antigen measurement is displayed as a dot in that sample's column. Different experiments are marked by a different color, defined in Table 1.

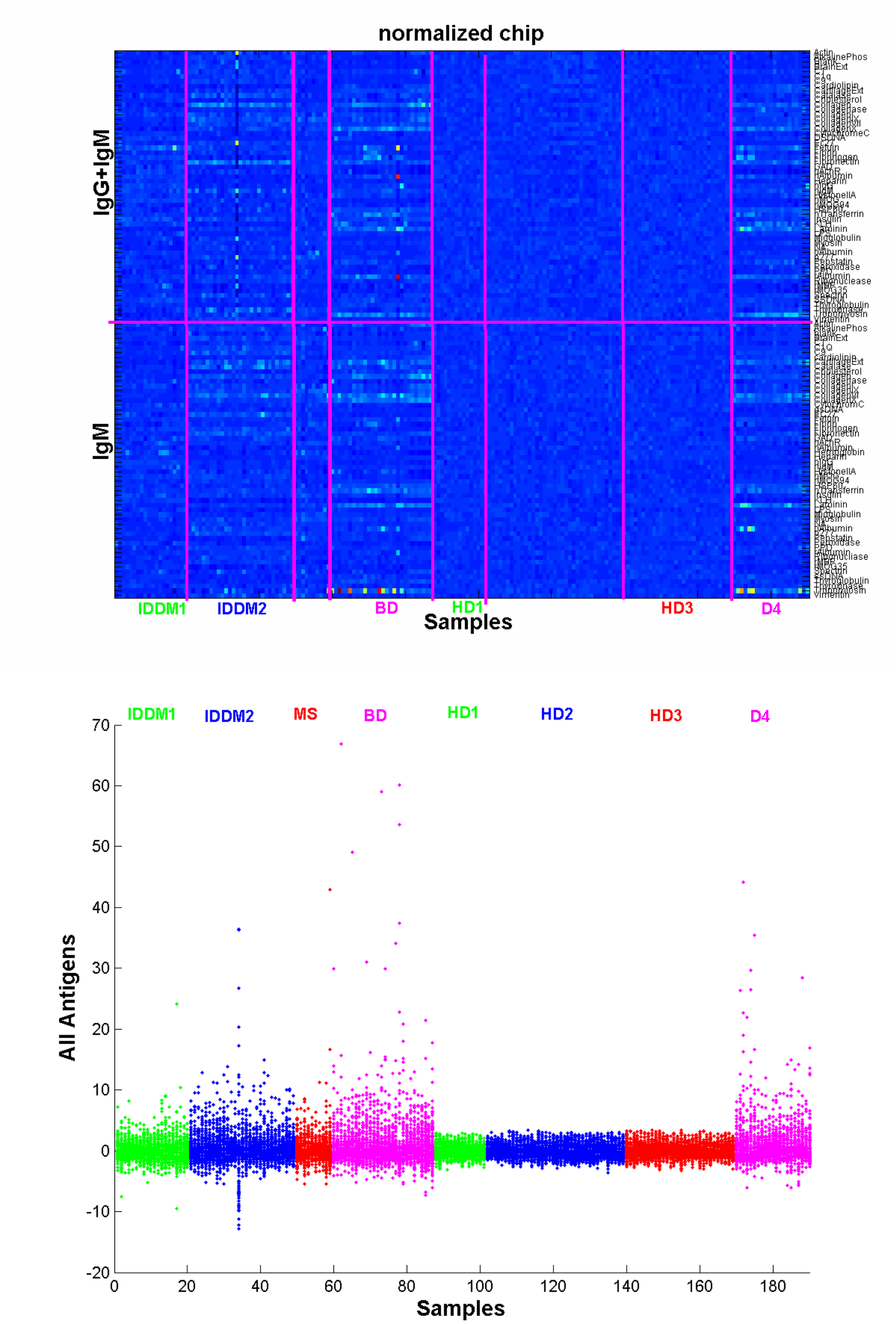

Figure 3: normalized data, up: chip view, down: dots view - For each sample, each antigen measurement is displayed as a dot in that sample's column. Different experiments are marked by a different color, defined in Table 1.

#	disease	code	color	number of Samples
1	Type-I Insulin Dependent Diabetes Mellitus	IDDM1	green	20
	Healthy donors	HD1		14
2	Type-II Insulin Dependent Diabetes Mellitus	IDDM2	blue	29
	Healthy donors	HD2		38
3	Multiple Sclerosis	MS	red	10
	Healthy donors	HD3		30
4	Beçket's Disease	BD	magenta	28
	Healthy donors + BD-like phenotype	D4		21

Table 1: Each disease has been given a code and a color. The first column refers to the number of the experiment the sample was analyzed in.

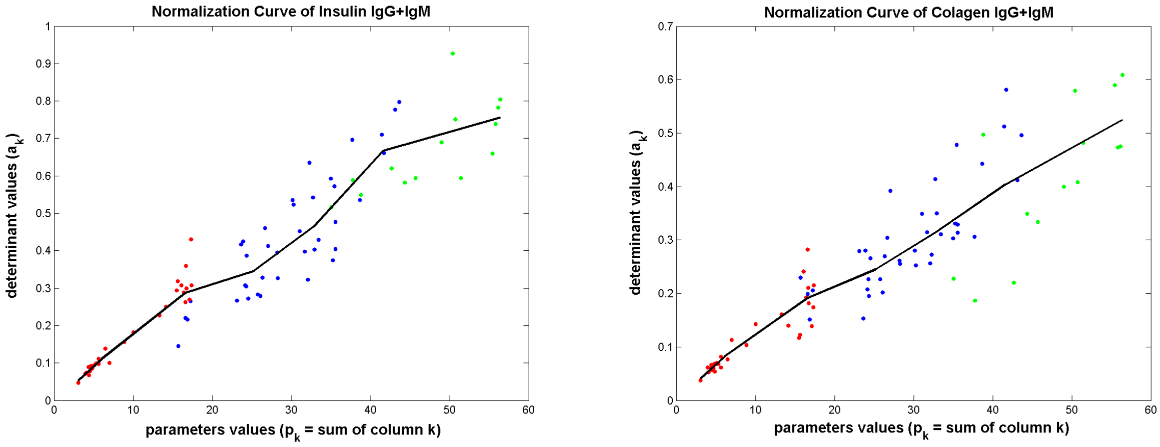

Figure 4: Examples for normalization curves, using 12 in a segment. Different experiments are marked by a different color, defined in Table 1.

disease at a time along with the 3 sets of healthy samples, and predict a classification for samples in a jackknife procedure (i.e. leave one out cross-validation), predicting one sample at a time while training by the rest, over an ordinal space, thus detecting for each disease the determinants that rise as important, and give the best prediction rates. Tables 2 and 3 summarizes our results.

Disease	Accuracy	(FN) Sensitivity	(FP) Selectivity
IDDM1	97%	(0) 100%	(3/82) 96.34%
IDDM2	97.3%	(2/29) 93.1%	(1/82) 98.78%
MS	96.7%	(3/10) 70%	(0) 100%
BD	100%	(0) 100%	(0) 100%

Table 2: Each disease was compared to the 82 healthy donor's samples of 3 experiments. We calculate the total accuracy of classigying the right disease state for all samples (healthy/sick), and the number and percent of False Negatives (FN) and False Positives (FP) predictions.

As for the control samples of the Beçket's Disease , we can classify which ones are indeed healthy and which show a resemblance to Beçket's Disease samples using them as a test set for our procedure. For this we normalize D4 and BD according to HD1-3 sets, then train the algorithm to distinguish between BD and HD1-3. Finally, we use the most distinguishing antigens, learned already by the jackknife procedure described above (see the Beçket's Disease column in Table 3), to classify each of the D4 samples by our Bayesian classifier over an ordinal space. We suspect 6 out of the 21 samples of the D4 set to resemble healthy samples, and the others show a resemblance to the Beçket's Disease samples.

IDDM1	IDDM2	MS	BD
CartilageExt G+M (4.80)	Collagen G+M (7.46)	CollagenIX M (4.48)	CollagenX G+M (7.22)
hIgG M (3.92)	Fibronectin G+M (5.53)	CollagenIX G+M (4.48)	hTransferrin M (5.98)
Insulin G+M (3.69)	Fibronectin M (4.99)	Cardiolipin G+M (3.51)	Laminin M (5.96)
Fetuin G+M (3.49)	Cholesterol G+M (4.89)	CollagenX G+M (3.11)	CollagenX M (5.89)
Laminin G+M (3.08)	Ec27 M (4.06)		Tropomyosin G+M (5.89)
CollagenX G+M (2.93)	KLH M (3.82)		Collagen G+M (5.82)
hMOG G+M (2.84)	Catalase G+M (3.72)		Laminin G+M (5.33)
Cardiolipin G+M (2.70)	Thyroxinase G+M (3.68)		Tropomyosin M (5.11)
Cardiolipin M (2.44)	Thyroglobulin G+M (3.63)		hIgG M (5.02)
Tropomyosin G+M (2.30)	CollagenIX G+M (3.52)		Collagen M (4.75)
	Collagen M (3.38)		
	Cardiolipin G+M (3.22)		
	rMOG35 M (3.18)		
	rMBP G+M (3.11)		
	CartilageExt M (3.05)		
	Pepstatin M (3.00)		
	hTransferrin M (2.94)		
	Fibrinogen G+M (2.89)		
	p277 M (2.83)		
	Fibrinogen M (2.82)		
	rMOG35 G+M (2.80)		
	C1q M (2.79)		
	CollagenVII M (2.79)		
	LPS M (2.62)		
	DSDNA G+M (2.59)		

Table 3: Antigens with highest significance value (in parenthesis) and used for the class prediction.

References

[1] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[2] Francisco J. Quintana, Gad Getz, Guy Hed, Eytan Domany, and Irun R. Cohen. Cluster analysis of human autoantibody reactivities in health and in type 1 diabetes mellitus: A bio-informatic approach to immune complexity. Submitted to European Journal of Immunology, 2002.

[3] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci. USA*, 94:12079–12084, 2000.

[4] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, New York, 2 edition, 2001.

[5] Janeway and Travers. *Immuno Biology*, chapter 2, pages 10–12. Garland, 1996.

[6] F. Mor, A. Weinberger, and I. Cohen. Identification of alpha-tropomyosin as a target self-antigen in beçket's syndrom. *Eur. J. Immunol.*, 32:356–365, 2002.