Supervised classification for biomarker discovery with PamChip® data

Rik de Wijn, Alexandre Maurel and Faris Naji
(www.pamgene.com)

Summary
Supervised classification is the main tool for data analysis in biomarker discovery. Pamgene has implemented a tool based on partial least squares discriminant analysis (PLS-DA) in BionavigatoR data analysis package, resulting in PamIndex plots that have become a standard way of reporting the results of PamChip® based biomarker studies. Here we discuss some general concepts of supervised class prediction, and its implementation in BionavigatoR.

Introduction
PamChips®-microarray based protein tyrosine kinase (PTK), serine threonine kinase, and nuclear hormone receptor (NHR) assays are frequently applied in biomarker discovery studies. Measured, using cell lines, xenografts, or clinical sample material [1-3]. The kinase assays measure enzymatic phosphorylation activity on a large panel of peptides that represent kinase substrate proteins. In the NHR assay nuclear receptor binding to peptides derived from co-regulator proteins is measured.

A distinctive feature of the PamChip® assays is that these allow determination of in vitro compound effects on the measured activity. For instance, the kinase activity present in a particular lysate can be measured with and without a kinase inhibitor present. This results in compound effect profiles or inhibition profiles that can also be used in biomarker discovery. Hence, PamChip® assays allow for biomarker research, at the level of cellular signal transduction and including the effect of drugs such as kinase inhibitors on their direct targets [4].

To assist researchers in analyzing its assays, PamGene has developed the data analysis package BionavigatoR, comprehensive and flexible software that allows basic to highly sophisticated data analysis for PamChip® measurements. BionavigatoR features several data analysis tools for biomarker research. Of these methods, supervised classification merits some extra consideration.

Supervised class prediction
The main data analysis approach in biomarker studies is learning by example: supervised classification uses example samples with known status or class to predict the status of new samples. (Figure 1).

As an example consider the work on non small cell lung cancer (NSCLC) by Hilhorst et al. [3]. Here measurements were performed on samples from a group of patients that underwent neo-adjuvant treatment with the EGFR kinase inhibitor Erlotinib. From these patients the response to this treatment was known prior to the measurements, i.e. patients were assigned to the

\[ y = \beta_0 + \sum \beta_i S_i \]

Step 1

Step 2

Figure 1. The two main steps in supervised class prediction. In step 1 a set of example samples with known status (e.g. patients that are responders or non-responders) are used to train a model that discriminates between the two groups. As a second step this model can then be used to predict the status of new patients.
respond or non-respond class based on the assessment of a pathologist. From this set of samples Erlotinib inhibition profiles were obtained using the PamChip® PTK assay, and a model for discriminating responders from non-responders was created. This model could then be used to accurately predict treatment response in an independent test set of example samples.

PamIndex Tool

We have found that a method called partial least squares discriminant analysis (PLS-DA) [5] works well with various types of PamChip® data. The standard supervised classification tool in BionavigatoR uses this method. With this tool a classification model can be created and applied for prediction (Figure 1).

With a two group (binary) classification problem, using such a model for prediction results in a classification score that we have called PamIndex. When a sample has PamIndex > 0 it is assigned to one group (e.g. the non-responders) and to the other when it has PamIndex < 0. When the PamIndex is further away from the decision boundary at PamIndex = 0, it is less likely to belong to the opposite group.

PamIndex plots like Figure 2 have become a standard way of presenting PamChip® based biomarker results.

A simple classification problem in 2 dimensions

In order to provide some idea on how our classification method might work we will use a simple example of a classification problem. Consider some hypothetical microarrays containing two distinct peptides only: peptide A and peptide B. Hence, for each sample two signals are measured: the signal obtained with peptide A and that with peptide B. Of course, microarrays tend to carry a much larger number of spots but the advantage of our 2-spot arrays is that the data can be easily visualized in a 2 dimensional graph. For instance, in Figure 3A a scatter plot is made of measurements on peptide A versus that on peptide B for 10 responder samples (red dots) and 10 non-responder samples (blue dots). On inspection it can be seen that the measurements appear to separate the responders from the non-responders. Responders have, on average a higher signal for peptide A and a lower signal for peptide B. In Figure 3B these 20 measurements are used to train a simple linear classifier, a green line indicates the resulting classification boundary.

In Figure 3C two new measurements on samples with unknown class are represented by black dots. One turns out to be on the “red side” of the classification boundary and will be classified as a responder, the other one is on the “blue” side and will be classified a non-responder. A prediction score may be obtained as the distance of the sample to the classification boundary (e.g. the dashed line in Figure 3C). Samples classified in one of both classes with a high score are less likely to really have originated in the opposite class than samples with a lower score.

Figure 3D illustrates the application of a more sophisticated (or possibly just more flexible) classification method. Clearly, this method is better at separating the samples used for training and it may seem a good idea to use this classifier rather than the inflexible classifier of Figs 3B and 3C. It remains to be seen, however, if this classifier also performs better on new data.
More on supervised classification

Put simply, the workings of a classification method can be thought of as finding classification boundaries in the sample measurement space. In Figure 3 this has been done for a 2 dimensional sample space (i.e. measurements on two peptides per sample) but the concept is readily extended to N dimensions, where N can in principle be anything from 4 for the famous Iris flower classification [6] up to 30,000 for genome wide expression arrays. For humans the task of visualizing or grasping the problems becomes extensively more difficult when the number of dimensions exceeds 3, but mathematically computing is in principle as straightforward for N dimensions as for 2.

Supervised classification may be contrasted to unsupervised classification or clustering. The
The goal of the latter is to partition samples into groups without any prior knowledge of the status of the samples. Although it is sometimes thought that this "blind" nature of clustering makes it a better method than supervised classification, clustering is not really suitable for classification problems [7]. Although clustering may provide useful support for biomarker data, supervised methods are the methods of choice for classification problems.

At PamGene we have selected PLS-DA as a classification method. From experience we have learned that this method often works better or at least as well as other methods. In addition, our PLS-DA implementation has been optimized for the current data characteristics. However, PLS-DA is just one of an enormous amounts of methods and algorithms available for supervised classification. An enormous amount of methods and algorithms for supervised classification are available. An overview of methods often used with micro-array data can be found in a paper by Slawski et al [8].

Usually these methods implicitly involve some way of determining which variables (dimensions, spots) are discriminative in the classification problem at hand, and which are not. For high dimensional data explicit variable selection is often applied. Here, some rule is used to select the only the most informative variables for processing by the classification method. Moreover, several of the classical classification methods (e.g. linear or quadratic discriminant analysis) will not work when the number of variables exceeds the number of spots, so here explicit variable selection can become a necessity.

Classification methods often go by intriguing names like Neural Networks, Random Forest, Support Vector Machines, Elastic Net, to name just a few. Combined with a magic sense of artificial intelligence these can be the ingredients of heated discussions on the question of the best classification method. However, it appears that with this large amount of different methods that have become available over the last-so-much years, the pattern recognition community has failed to provide a method – apart from empirical testing - to decide which (type of) classifier is the best for a particular problem. Interestingly, in a study by the MAQC consortium on the performance of predictive models it was found that the proficiency of the team of data analysts was a more important indicator of model performance than the method/model used [9].

Testing a classifier: performance and validation

Once the application of a classification method has resulted in a classifier the all important question is: how good is it? In this section we will first consider some metrics for classifier performance and in the next section some methods to estimate the value of these metrics. The latter is often referred to as classifier validation. For a more extensive discussion see [10].

The basic metric of classifier performance is the miss classification rate (MCR), defined as the fraction of tested samples that is miss classified. Obviously, a lower MCR is considered to be better. However, consider a classification problem with two classes called positive and negative (i.e. these might reflect the diagnosis or a prediction of treatment response). In practice, it may well be the case that if a positive sample is wrongly classified as a negative (i.e. a false negative) this has much more severe consequences than a false positive (i.e. a negative that is wrongly classified as a positive). For these circumstances “asymmetric” performance parameters may be considered. The positive predictive value (PPV) and negative predictive value (NPV) are defined as the fraction correct in each of the respective classes:

\[
PPV = \frac{true \, positives}{true \, positives + false \, positives};
\]

\[
NPV = \frac{true \, negatives}{true \, negatives + false \, negatives}
\]

Where true positives, false positives etc. refers to the number of true positives encountered. Also, the fraction of positives classified correctly (true positive rate, TPR, or sensitivity) and the fraction of negatives classified correctly (true negative rate, TNR, or specificity) is often used:

\[
TPR = \frac{true \, positives}{true \, positives + false \, negatives};
\]

\[
TNR = \frac{true \, negatives}{true \, negatives + false \, positives}
\]

Asymmetric performance parameters are often inter-connected, in the sense that e.g. a better PPV will result in a lower NPV. Receiver operator curves (ROC curves) may be used to evaluate this connection, and possibly adjust the performance of a classifier to a desired setting.
Classifier validation

The goal of classifier testing or classifier validation is to assess the performance of a classifier. More precisely, the goal is to estimate what performance metrics (MCR, PPV, NPV, etc.) the classifier would yield when applied to new samples. Two methods to do this will be discussed here: test set validation and the method most often applied in the practice of PamChip® based biomarker research: cross validation.

Test set validation uses a set of example samples (i.e. samples of which the class is known a priori). This test set should be not involved in any way in the creation of the classifier, i.e. it should be completely independent from the classifier. If this is the case the performance of the classifier on the test set provides a realistic estimate of the performance of the classifier with new data.

The disadvantage of using a separate test set is that it should be relatively large to in order to obtain a performance estimate with reasonable precision. More often than not, however, the total number of example samples that is available for both training and test set is rather limited, and one would want to use as many examples as possible in training the classifier because this usually results in the best possible performance. In other words, leaving out a substantial number of example samples out of the training set, for use in the test set, may results in an overly conservative estimate of the performance of a classifier that is trained using all available example samples.

An important advantage of using a separate training and test set is that the class membership of the test set examples may be blinded to the data analyst until after class predictions have been submitted. In this way it can be objectively guaranteed that an unbiased performance estimate has been obtained.

Cross validation is an approach that is particularly useful when only a limited number of samples are available. It simulates the application of a classifier to new samples using the training set examples in a stepwise manner.

In k-fold cross validation the examples are randomly subdivided in k groups, the folds (e.g. 10-fold cross validation with 50 samples would start by randomly assigning 5 samples to each fold). On each of the k steps of the cross validation process a classifier is trained using the samples available in all folds minus one. The samples in this remaining fold are then used to test the classifier. During the k steps of the procedure each fold gets the role of test set once. The performance of the classifier is then estimated form the aggregated performance of the classifier on these successive steps. A special case of k-fold cross validation occurs when the number of folds is equal to the number of samples, this is called leave one out cross validation, and is illustrated in Figure 4.

A k-fold cross validation may be repeated several times with a different distribution of samples over the folds. This will provide some idea of the precision of the error estimate obtained. In the limit of LOOCV all distributions of samples over the folds are equivalent and this information is not available.

In conclusion, if there are plenty of example samples available or if there is a special need for objectively guaranteeing the quality of the performance estimate, use a separate (blinded) training set for classifier validation. If this is not the case cross validation is probably the best option. For somewhat larger samples sets (> 20 - 30) consider using repeated 10 or 20 fold cross validation because it provides at least some idea on the precision of the estimated error rate. For smaller sample sets LOOCV is the only viable option.

Handle with care

Supervised class prediction is a powerful tool to discriminate between groups using multi-

![Figure 4. With leave-one-out-cross-validation the application of the model on new samples is simulated using the available set of training samples. A model is created based on the set of example samples with one sample left out to take up the role of a new sample (compare to Figure 1). Prediction on this sample can be verified. This is repeated for all samples.](image-url)
Supervised class prediction component. As of version 5.1 PamGene’s data analysis software package Bionavigator features partial least squares discriminant analysis (PLS-DA) method, undeservedly, somewhat of a bad validation results. Misuse has given the unwillingly add optimistic bias to cross validation. Hence, in order to obtain a realistic unbiased performance estimate from classifier validation (either using a separate test set, or by applying cross validation) care must be taken to make sure that the test samples are completely independent of the training examples. A common mistake is e.g. to select a subset of discriminative peptides on the entire set of example samples and then perform a cross validation using the discriminative peptides only. Hence, the successive training and test sets are not completely independent and the performance will be substantially overestimated [7]. Selecting discriminative peptides is part of constructing a classifier and the proper way to cross validate such a classifier is to repeat the selection process on each iteration on the training set only. Similar considerations apply to actions like selecting an optimal classifier from a panel of classifiers, or finding optimal classifier settings [11]. The main disadvantage of cross validation is that there are many ways by which data analyst can willingly or unwillingly add optimistic bias to cross validation results. Misuse has given the method, undeservedly, somewhat of a bad name.

More on the PLS-DA implementation in Bionavigator

As of version 5.1 PamGene’s data analysis software package Bionavigator features partial least squares discriminant analysis (PLS-DA) supervised class prediction component. In combination with example data the component can be used to train and cross validate PLS-DA models. Trained models can be stored and used for class prediction with new data (or to evaluate performance with a test set of examples).

The component is somewhat specialized for class prediction with two classes (binary class prediction), but will work for multiclass problems as well. For binary class prediction problems, the PamIndex representation of prediction results is produced. When training a PLS-DA classifier partial least squares regression is used to fit the multiple peptide signals measured on the example set to a one dimensional response variable $y$, where $y = 1$ for the positive class and $y = -1$ for the negative class. This results in a set of coefficients $\beta$, including a coefficient for each peptide signal used during training plus an offset value. This $\beta$ in fact constitutes the predictive model; it can be used to generate a prediction for a new sample that is measured with the same set of peptides:

$$
PamIndex = \beta_0 + \beta_1 S_1 + \beta_2 S_2 + \ldots + \beta_n S_n \tag{3}
$$

Here, $\beta_1$, $\beta_2$, ..., $\beta_n$ are the individual coefficients for the signals $S_1$, $S_2$, ... $S_n$, measured on the n peptides used, $\beta_0$ represents an offset value. Hence, the set coefficients $\beta$ is used to convert the measured multiple peptide signals to a single score: PamIndex. A sample is assigned to the positive class when $PamIndex > 0$, and to the negative class when $PamIndex < 0$. Note that a PamIndex can be obtained by either cross validation (i.e. the predictions obtained when the corresponding sample is in the test set) or by applying the classification model to new samples. When the real classes of the samples are known the model performance may

![Figure 5 Value of model coefficients (y-axis) of a trained model (black line) and that of the successive models generated during a leave one out cross validation (gray lines). On the x-axis the 79 peptides in the model represented, sorted according to their value in the trained model. It can be seen that all gray lines follow the trained model quite well, which indicates a](image)
be visualized using PamIndex plots (Figure 2). The set of model coefficients $\beta$ contains information on which peptides are relatively important in assigning samples to classes: those peptides with the largest absolute coefficient value. Hence, those peptides that are the most positive or negative have the largest weight in discriminating between the classes, peptides with a near zero coefficient value have little weight. After running the component coefficient values are available in Bionavigator.

A special plot is available that can be used to compare the coefficient values in the trained model to that generated in the successive generations of a cross validation. From this it can be judges how stable the model was on the different training sets used. An example of such a plot is shown in Figure 5.

References
4. See www.pamgene.com for more information, application notes etc.