SVM-Based Analysis of NMR Spectra in Metabolomics: Development of Procedures

Oleg Favorov UNC-Chapel Hill
Jeffrey Macdonald UNC-Chapel Hill
Olcay Kursun University of Central Arkansas

This chapter is intended for the beginning NMR student with a binned metabolomic dataset stored as a matrix of numbers in a data analysis program, such as MS Excel. The reader will be introduce to the multitude of pattern recognition algorithms applied to metabolomic datasets with a specific tutorial on support vetor machines (SVM). A standard practice in metabolomics research is to analyze high-throughput metabolic data using Partial Least Squares (PLS) pattern-recognition approaches. However, in exploratory small-size studies, PLS-based methods work well only when experimental manipulations have strong and widespread effects on metabolic profiles, but not when experimental manipulations have only subtle or complex effects. For such difficult situations, more powerful pattern-recognition techniques are needed. Among such techniques, SVMs are most efficient at finding regularities in limited, but high-dimensional data, which makes them particularly well suited for metabolomics tasks. This paper investigates applicability of SVMs to metabolomics, identifies constraints under which SVMs will have to operate, and lays out a systematic, fully specified protocol for interrogation of metabolic NMR spectral data with a suit of SVM-based analytical procedures. The developed protocol targets two central questions: (1) do NMR spectra of a given set of biosamples exhibit statistically significant dependence on the studied experimental conditions; and (2) which spectral bins vary systematically with those conditions? Unlike PLS methods, the chosen SVM procedures can recognize nonlinear or combinatorial relations in the data, they are much more resistant to overfitting – and thus failing – on small numbers of data samples, and they are much more efficient in identifying the significant spectral bins. With such advantages, this new SVM-based analytical approach should significantly raise the productivity of exploratory metabolomics studies.

Introduction

A high-throughput technique of Nuclear Magnetic Resonance (NMR) spectroscopy offers an effective means of rapid quantification of metabolites present in biological samples, making it one of the methods of choice for obtaining metabolic profiles of biofluids, tissues, or cell cultures. Metabolic profiles are a valuable source of information about physiological processes taking place under different naturally occurring or experimentally induced conditions. Metabolic profiles can be used, in particular, to recognize and distinguish pathophysiological states associated with various diseases, toxic exposures or genetic modifications, thus providing information on disease processes, drug toxicity or gene function (Nicholson et al., 2002; Reo, 2002; Griffin, 2003).

Interpretation of spectral profiles of metabolites is not a simple matter, however. While a particular drug treatment, for example, can change the metabolic profile in a systematic and reproducible way, it is not immediately obvious which of the metabolites – appearing in NMR spectra as spikes – are in fact affected by the treatment and which metabolites simply exhibit normal variability unrelated to the treatment. Practical limitations on labor and expense associated with obtaining and screening each specimen typically greatly constrain the number of samples available for study. Small sample sizes, coupled with efforts to maintain low false-discovery rates, weaken the power of statistical methods to identify metabolites that respond to experimental manipulations. Thus, instead of statistical approaches, metabolomics analysis of NMR spectral data preferentially relies

on Pattern Recognition approaches to interpret metabolic profiles. The most commonly used methods are Principal Component Analysis (PCA) and Partial Least Squares (PLS) Discriminant Analysis, or PLS-DA (Eriksson et al., 1999; Lindon et al., 2001; Holmes and Antii, 2002). PCA and PLS-DA are used to project individual spectra, conceived as points in a high-dimensional space (each dimension corresponding to one of the bins in the spectrum), onto a particular plane bisecting that space. The orientation of the projection plane is chosen so as to reveal, if possible, the clustering of spectra obtained under different experimental conditions in different regions of the projection plane.

PCA and PLS-DA work well when experimental manipulations have strong and widespread effects on metabolic profiles. However, PCA and PLS-DA do not work well when experimental manipulations have only subtle or complex effects on metabolic profiles. Both methods are linear in their design, making them ineffective when relations between experimental conditions and metabolic profiles are fundamentally nonlinear. Even when relations are linear, PLS-DA suffers from a tendency for overfitting when the number of spectral bins exceeds the number of spectra, especially if the differences between compared groups of spectra are minor and confined to relatively small numbers of metabolites. PCA is much less sensitive to the number of samples, but it is limited in the choice of possible planes onto which the samples can be projected. In PCA, if the metabolic profile variability caused by experimental manipulations constitutes only a relatively small fraction of the total profile variability, it will not be exposed in any of the available projection planes.

A pattern recognition approach that is generally recognized for its many advantages over other pattern recognition approaches is the support vector machines, or SVMs (Vapnik, 1995, 1998; Bennett and Campbell, 2000; Scholkopf and Smola, 2002). SVMs are a versatile class of supervised Machine Learning methods that can be trained to learn either linear or nonlinear input-output relations and can perform either classification or regression tasks. Unlike other nonlinear methods, SVMs do not suffer much from the “local minima” problem of getting stuck in a suboptimal solution of the task. SVMs have excellent generalization abilities, and thus a reduced likelihood of overfitting. And of particular significance for metabolomics, SVMs are robust to using large numbers of variables as inputs and can be successfully trained on very limited numbers of training samples (Burges, 1998). Finally, SVMs are very fast, easy to use and have only a few parameters that require optimization.

These important advantages of SVMs make them highly attractive for application to metabolomics problems (Belousov et al., 2002; Thissen et al., 2004). However, a number of technical issues will have to be addressed before SVMs can be used to extract information from metabolic profiles. In this paper we identify these issues, offer and test specific ways to resolve them and describe a program of SVM-based analytical procedures designed to ascertain the effects of experimental manipulations on metabolic profiles. Applied to NMR spectra of specimens obtained under varied experimental conditions, these analytical procedures answer two central questions: (1) do metabolic profiles of the studied biological material – be it urine, or blood, or a specific tissue – reflect in a statistically significant way the different experimental conditions studied; and (2) which metabolites comprising the profiles vary systematically with those conditions? This program of analytical procedures should be especially helpful in those exploratory (limited in sample size) metabolomics studies in which experimental manipulations are found to produce only subtle or complex effects on metabolic profiles.

Methods

Support Vector Machines

A number of software packages are available that implement the SVM theory and automate its application to datasets provided by a user. In this paper we use SVM^{light} version developed by

Thorsten Joachims (1999) and available for downloading at . SVM^{light} is simple to use. It does not require an understanding of the underlying theory, and we provide here only a brief intuitive explanation of the SVM approach using a geometric perspective. To visualize how SVMs work, we use in Figure 1 a graphic example of hypothetical data described by two variables, X_1 and X_2, with data samples divided into two classes. The described concepts, however, are generalizable to larger numbers of input variables and also to regression tasks (i.e., learning a continuous function of the input, rather than its class partitions).

The fundamental SVM design is built on several key insights (Vapnik, 1995; Scholkopf and Smola, 2002). The first insight is to use the “optimally” placed decision hyperplane to separate the sample classes. For example, in Figure 1A the training data samples belonging to two different classes cluster separately in different regions of the input space (i.e., the space defined by input variables X_1 and X_2) and can be easily separated by a line. This line can be used to classify new, test data samples, according to their position relative to this line. In Figure 1A, two among many possible placements of the decision line are shown. While they separate the two groups of training samples equally well, more preferable is the black line. This line is placed so as to maximize the minimal distance between it and the training samples, and it is more preferable because it is less likely to make false classification decisions on future samples.

Note that the placement of the optimal decision hyperplane (the line in Figure 1A) is determined not by all the training samples, but only by the samples closest to the hyperplane (they are indicated in Figure 1A by circles). Such training samples that determine the orientation of the decision hyperplane are called the “support vectors.” Use of the optimal decision hyperplane is the foundation of the SVM superior ability to generalize from training samples to new data.

The second insight is to make classification (or regression) decisions not in the input space (defined by the input variables), but in a “feature” space. This distinction becomes important when the training data samples cannot be separated in the input space by a hyperplane. For example, in Figure 1B the two classes cannot be separated completely by a straight line, but only by a curved line. Unfortunately, finding the optimal curved partition of the input space is much more difficult. Unlike finding optimal linear partitions, finding nonlinear partitions takes much longer time and is quite likely to produce only suboptimal solutions (become trapped in local minima). We can overcome this problem, however, if we would somehow transform the input space into such a new “feature” space, in which the sample classes become linearly separable (see Figure 1B). Then we can use the techniques of linear separation on the transformed data and determine their optimal partition in the feature space.

The third insight is that explicit remapping of the data from the input space to a feature space does not have to be actually done in practice. Evaluating data points in a feature space can be replaced, with exactly the same results, by simply evaluating data points in the original input space using an appropriate kernel function. A very popular kernel function is the Radial Basis Function (RBF). It expresses similarity of two vectors, and , as a function of the Euclidean distance between them, D_{ij}, according to An RBF kernel is shown in Figure 1C. The RBF parameter g controls the width of the kernel.

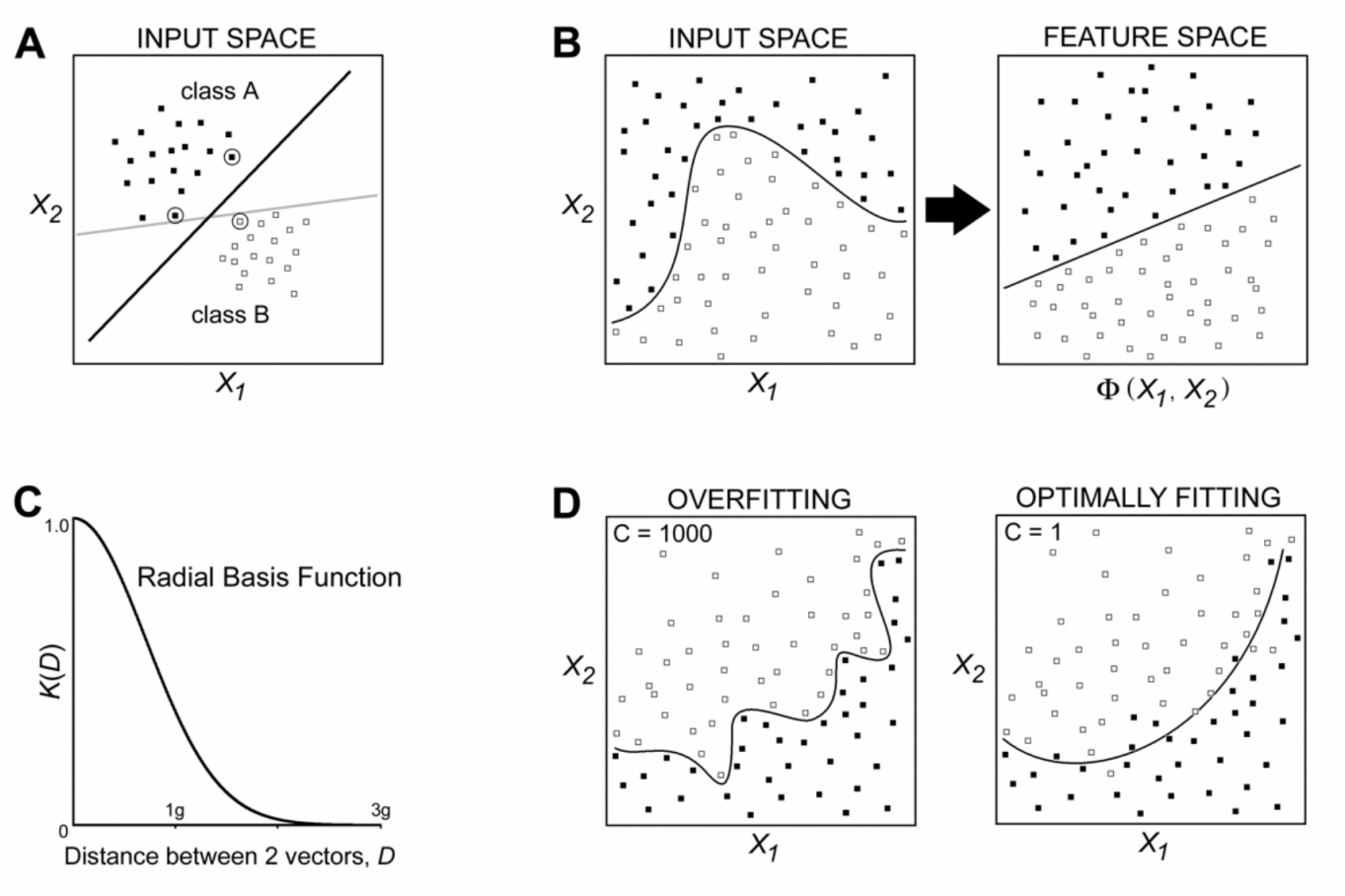

Figure 1. *Key features of SVM design* A: Optimal decision hyperplane. Little black squares are data samples of class A, little open squares are data samples of class B. The optimal boundary between the two classes is shown as the black line. Circled samples are the "support vectors," which determine the orientation of the optimal boundary. B: Transformation of Input space into Feature space. The Feature space has more dimensions than the Input space, but only two are shown in this illustration for display clarity. C: Radial Basis Function. The value of the function is plotted against the distance between two vectors, which is expressed as a fraction of the *g*-parameter. D: Control of the smoothness of the class boundary. The data samples are shown in the input space and are separated into two classes by a curved line. In the left panel, the highly-convoluted boundary is overfitting: it correctly separates all the shown data samples by their classes, but is likely to be less accurate on new data samples than the smoother boundary in the right panel.

RBF kernel has been found to be very effective in a wide range of SVM applications. In principle, for problems of a particular nature, there might be a special kernel that will be most effective in separating different sample classes there. However, finding such an optimal problem-specific kernel usually is not practical, and use of a known, "general-purpose" kernel (such as RBF) will still provide a reasonably successful solution. RBF is generally the first kernel type to try; if it fails, other common kernels provided by software packages (in particular, polynomial and sigmoid kernels) can be tried next.

The fourth insight concerns the danger of SVM overfitting on the training data. As illustrated in Figure 1D, two sample classes might have partially overlapping distributions in the input space. Using kernels, we will be able to achieve 100% separation of samples belonging to the two classes by fitting a highly convoluted boundary to them (see the left panel in Figure 1D). But this boundary will be mistaken, being misled by noise in the data. A much less convoluted boundary (as shown in the left panel in Figure 1D) will be, objectively, more accurate, reflecting the true interface between the two class distributions. Thus, by setting limits on the degree of acceptable complexity of the SVM-drawn boundaries, we might be able to improve the SVM performance on future, test data samples, despite doing worse on the training data. An SVM parameter that controls the complexity of class partitions is known as "penalty error," or parameter *C*. As *C* decreases in value,

the boundaries become smoother. As a rule, when fewer numbers of data samples are available for training an SVM, the attempted class partitions should be more constrained in their complexity by reducing the value of *C*-parameter. Parameter *C* enhances SVM ability to generalize successfully from training samples to new data.

In conclusion, in order to use an SVM on a particular dataset, only three basic parameters have to be specified: (1) *C*-parameter; (2) the choice of the kernel (RBF is recommended first); and (3) a kernel-specific parameter (e.g., *g*-parameter for RBF, or degree of the polynomial for polynomial kernel). The optimal values of these parameters are problem-specific and are determined empirically by trial-and-error procedures (which will be described in Results).

Demonstration Dataset of "NMR" Spectra

The aim of this paper is to describe how SVMs can be effectively used in NMR spectral studies of metabolic profiles. This aim will be best served by using an artificial dataset of NMR spectra for demonstration, rather than a real, experimentally generated dataset, because the knowledge of its hidden information content will be helpful in explaining challenges and opportunities associated with SVM use. A follow-up paper (manuscript in preparation) will demonstrate a successful application of the analytical techniques developed in this paper to detection of metabolic signs of colon tumors in mouse urine.

Since metabolic NMR spectra are typically divided into approximately 200 regions, or bins, for peak integration, and each bin is mean-centered and scaled to unit variance, we created random artificial spectra to be strings of 200 bins, the values of which were drawn at random from a normal distribution with mean value of zero and unit variance. Only 40 such random spectra were created to be subjected to our SVM-based analysis, so as to approximate the analytical challenges posed by comparably small datasets obtained in typical exploratory metabolomics studies. Next, since the spectra are analyzed in metabolomics studies for hidden information they might contain about some biological variable of interest (e.g., a disease state, or drug treatment, etc.), we defined our variable of interest – calling it a "score" – as a sum of the first seven bins (the first bins were chosen simply for narrative convenience, since all bins are random and independent). Specifically, for each spectrum, its score was computed from values of its first seven bins as:

$$score = bin\#1 - bin\#2 + bin\#3 - bin\#4 + bin\#5 - bin\#6 + bin\#7 + H.$$

H is conceived as an additional factor (a "hidden" bin) determining the score that is not reflected by the measured metabolic profiles. Its value was also generated at random for each spectrum from a normal distribution ($\mu=0$, $\sigma=1$).

We chose a continuously varying score, rather than a binary, class-defining one, because regression is a more general problem, and classification can be viewed as a special case of regression. The choice of the number of significant bins was determined by a general consideration that in many metabolomics studies the differences between metabolic profiles obtained under varying conditions of experimental or clinical interest will be quite subtle, arising from relatively small numbers of metabolites, most of which will be contributing only modestly to the overall distinction between profiles. In our dataset, each significant bin contributes only 1/8 to the overall score. Trying to find such scarce and only modestly significant bins will be challenging, when given only a limited dataset of spectra to analyze. To find 7 significant bins among 200 – each bin explaining only 12.5% of the score – by analyzing only 40 spectra is close to the limits of capabilities of our SVM-based method (see Results).

Results

Overall Design of the Data Analysis Program

The program of SVM-based analyses of NMR spectra is developed in this paper to answer two central questions that might be addressed to NMR spectra of a given set of biosamples obtained under particular varied conditions of interest:

1. Do such spectra carry any information about the studied conditions?
2. If they do, which spectral bins carry this information?

In the program, the studied conditions of interest are expressed quantitatively by a "score," and each biosample in the dataset is tagged by the score of the condition under which it was obtained. Our basic approach to determining the presence of condition-specific information in spectra involves: (1) training an SVM on the spectra and condition scores of a subset of biosamples; and (2) measuring how well can the trained SVM predict the condition scores of new biosamples from their spectra.

The data analysis program comprises a number of procedures that together perform the following series of analytical steps.

STEP 1: Find optimal SVM parameters and measure the accuracy with which the optimized SVM predicts the scores of test data samples.

STEP 2: Permute the scores among all the biosamples in the dataset, optimize the SVM on this dataset, and measure its accuracy in predicting the permuted scores. Repeat such permutations at least 20 times. The fraction of times when the SVM predictive accuracy on permuted scores matched or exceeded the accuracy on the true scores gives the estimate of the statistical probability (the *p*-value) that the measured SVM accuracy on the true scores was simply due to a chance. If this probability is less than the acceptable threshold for statistical significance (e.g., □ = 0.05), the analysis concludes that the spectra do carry information about the studied conditions.

STEP 3: If STEP 2 analysis concludes that the spectra carry some score-predicting information, test individual spectral bins for their involvement, by measuring the effect of removal of each bin on the SVM score-predicting performance. This procedure selects a set of potentially significant bins; i.e., the best estimate of which bins carry the score-related information. Positive outcomes of STEP 2 and STEP 3 analyses can be used as a justification for collecting an additional set of biosamples for validation.

STEP 4: When a new, "validation" dataset of spectra becomes available, test – using the score-permutation approach of STEP 2 – whether the score-predicting accuracy of the potentially significant bins is statistically significant; i.e., whether this selected subset of all bins is truly significant.

STEP 5: Repeat STEP 3 procedure, using both the original and validation datasets together, to refine the selection of the potentially significant bins by providing more samples for training.

The following sections explain the reasons for the choice and the design of analytical procedures used in STEPS 1-5 and describe their practical implementation.

Training and Testing an SVM

Problem of irrelevant bins.

A major obstacle to training SVMs successfully on metabolomics NMR spectra is that a majority of bins in such spectra are not sensitive to the studied conditions of interest. Unfortunately, an SVM's learning/predictive performance is degraded when many among its inputs do not provide any task-relevant information. To illustrate, we trained an SVM on the demonstration dataset of "NMR" spectra, using a progressively larger number of bins, from #1 to #200. The SVM was trained on all

40 samples using the objectively optimal *C*- and *g*-parameters (determined on additional 200 samples that were generated just to illustrate this problem – these samples will not be used in the main analysis). After training, the SVM was tested on additional 100 samples, also generated solely for this demonstration.

The accuracy of the SVM predictions of the scores of the test samples was measured by computing the *coefficient of determination* (i.e., Pearson's correlation coefficient squared, r^2; it measures how much variability in one variable can be accounted for by the variability in the other variable) between the scores of the test samples and the SVM's predictions of those scores. The SVM score-predicting performance is plotted in Figure 2 as a function of the number of bins used as inputs. As the plot shows, when up to the first seven bins were used as inputs to the SVM, its performance grew quickly with addition of each new bin. This is understandable, of course, because these seven bins are the ones that are used to compute the score. Together, the first seven bins give the SVM its highest predictive accuracy. Training the SVM on those seven bins in combination with some irrelevant bins, however, reduces the SVM predictive accuracy. The larger the number of irrelevant bins included among the SVM inputs, the worse its predictive accuracy becomes.

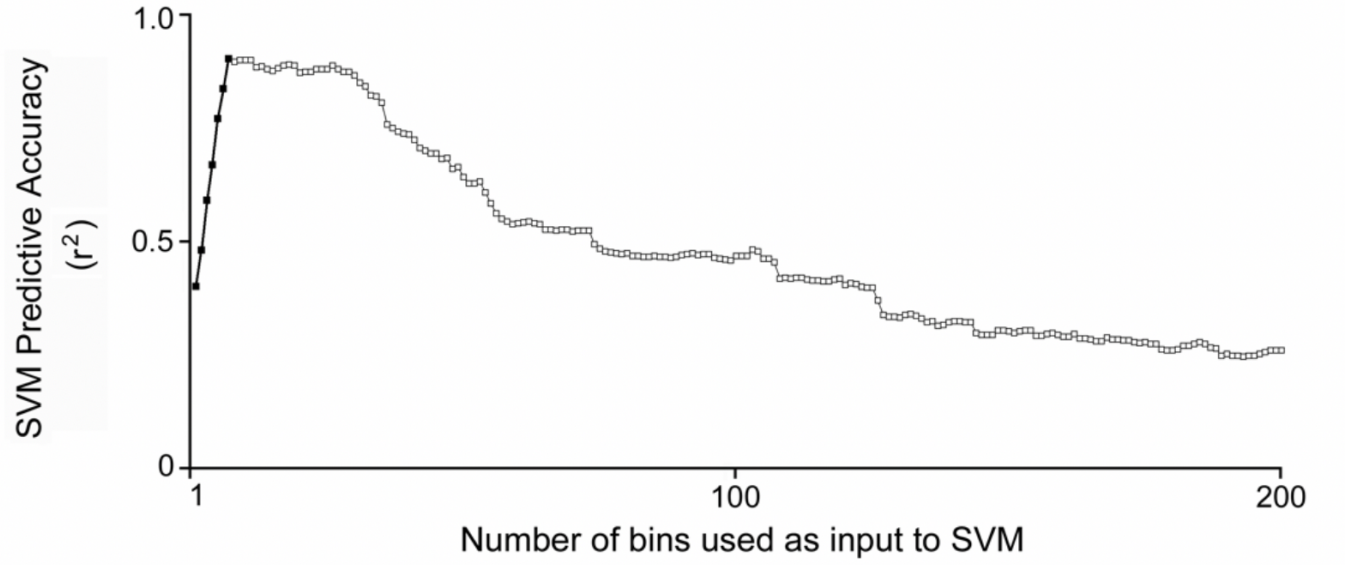

Figure 2. *SVM score-predictive accuracy as a function of the number of unrelated inputs.* SVM predictive accuracy is measured on a set of 100 test samples as correlation r^2 between the sample scores and SVM predictions of those scores. Bins are added to the SVM input set in the order of their position in the spectrum. The score-significant bins (i.e., bins 1-7) are shown in black, score-unrelated bins are white. Note that the SVM predictive accuracy gradually declines with inclusion of progressively larger numbers of score-unrelated bins.

Thus, to conclude, if an SVM is trained indiscriminately on all the bins in the NMR spectra, it will fail to learn – or, at a minimum, it will greatly underestimate – the relationship between the spectra and the score. Therefore, we have to have a strategy for reducing the number of irrelevant bins among the inputs to an SVM.

In this paper we adopt a most obvious strategy of selecting bins by their correlation with the score. Unless the relationship between the spectra and the score is heavily nonlinear, we might expect the significant bins to have higher correlation than irrelevant bins. But this is true only on average. For example, in our demonstration dataset of "NMR" spectra the expected r^2 for significant bins is 0.125, for irrelevant bins it is zero. However, when determined on our limited dataset of 40 samples, correlations of individual bins exhibit prominent deviations from their true correlations. This is shown in Figure 3, which plots correlations of all 200 bins with the score. Note that the first 7 bins (i.e., the significant bins, shown in black) do not stand out and, in fact, some of them are

quite small. Thus, in a typical exploratory metabolic study, in which significant bins in NMR spectra will have only minor association with the score and the number of samples is small, the significant bins will not be clearly distinguished from insignificant ones by their correlations.

Although we cannot reliably distinguish between significant and irrelevant bins by their correlation with the score, we can at least discard bins with the lowest correlations (as the least likely to include significant bins), thus raising an SVM's predictive performance. More specifically, we can try to discard progressively greater fractions of the bins, starting with those having the lowest correlation, until the SVM performance reaches its peak. However, before we use this strategy, we need to resolve some technical issues.

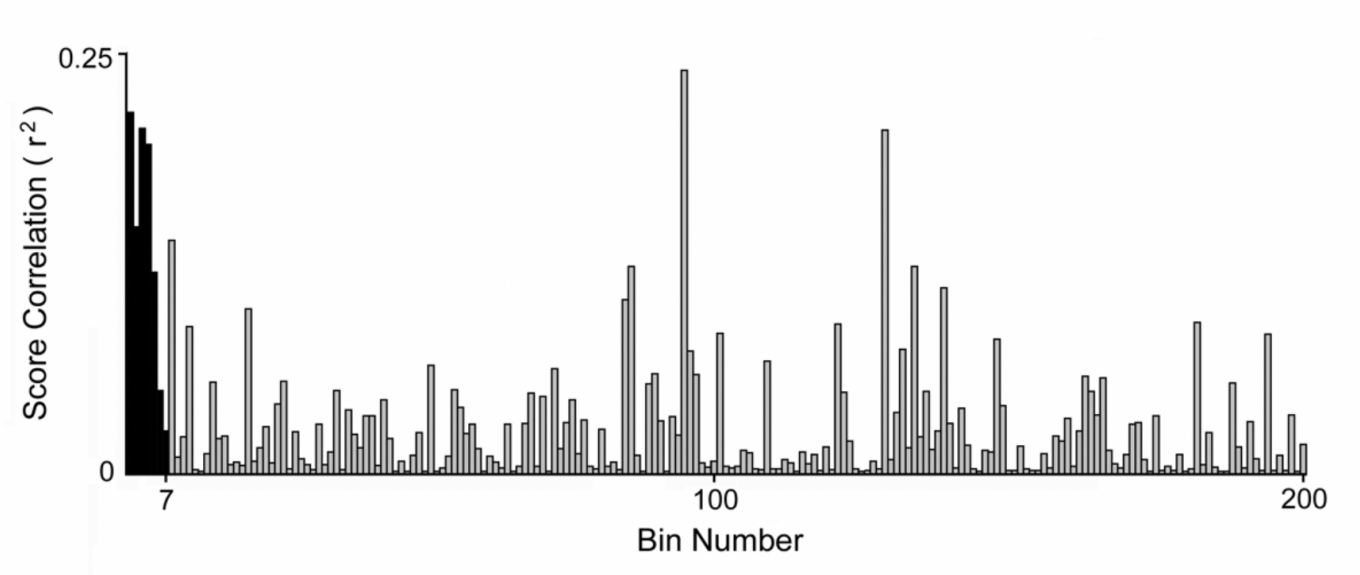

Figure 3. *Correlation between spectral bins and the sample score*. Plotted is the correlation r^2 computed for each spectral bin across all 40 samples of the demonstration dataset of "NMR" spectra.

Problems associated with training and testing SVMs on small-size datasets.

First, we need to decide how to use the limited available number of samples to both train and test an SVM. We cannot simply split the dataset evenly into training and testing subsets, because that will leave us with too few samples to do well on either task. Instead, a commonly used effective method is "leave-one-out." In it, one sample is set aside and the SVM is trained on all the other samples in the dataset. The trained SVM is then tested on the left-out sample. The score predicted by the SVM is compared with the true score of that sample. This training/testing process is repeated, with each sample in the dataset set aside and tested on in its turn. In this way, the ability of the SVM to learn the relationship between the spectra and their scores is tested repeatedly, the same number of times as the number of spectra in the entire dataset.

Two technical problems still remain. To demonstrate them, lets measure SVM performance using leave-one-out method on a randomly permuted dataset. That is, the scores are randomly shuffled among samples, destroying any relationship between bins and the score. Therefore we would expect SVM predictive accuracy to be approximately zero regardless of the number of bins used. The actual result is plotted in Figure 4 (gray curve). Here the bins were added to the SVM input set in the descending order of their correlation to the score (the permuted one, of course), which was computed across all 40 available samples. Surprisingly, SVM predictive accuracy is much better than the expected zero when first 100-120 bins are used, and it is significantly below zero when all bins are used. This indicates that we are not subjecting the trained SVM to a properly independent test.

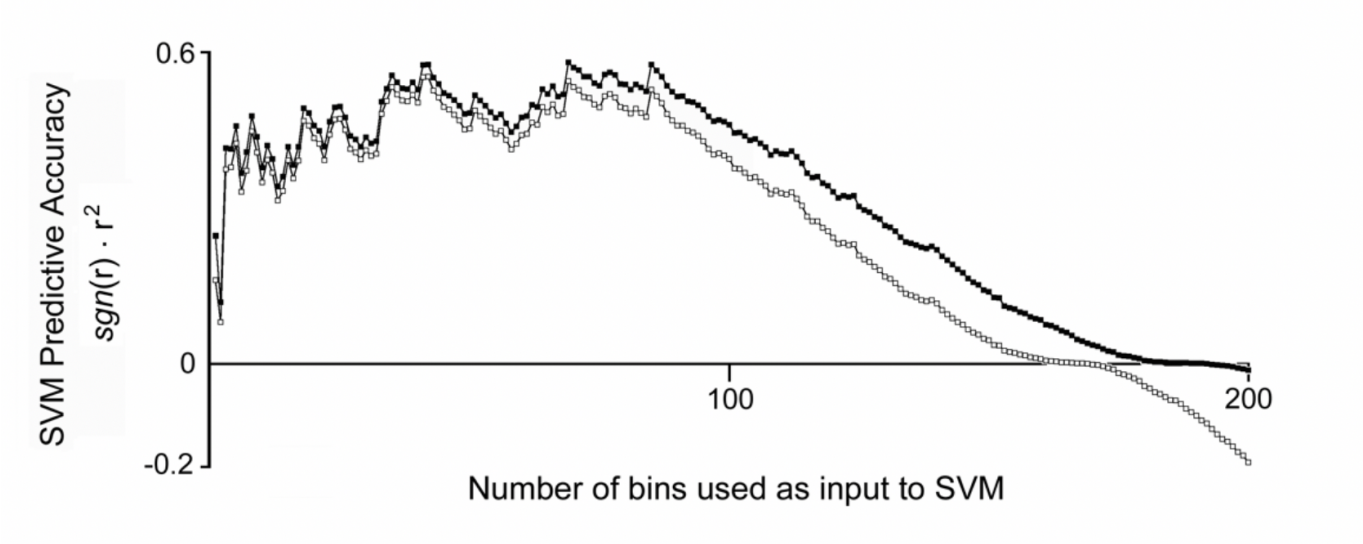

Figure 4. *SVM score-predictive accuracy on the demonstration dataset of 40 "NMR" spectra that was modified by randomly shuffling the scores among the samples.* The number of bins providing input to the SVM varies from 1 to 200 along the abscissa. The bins are added to the SVM input set in the descending order of their correlation with the score, computed across all 40 samples (see Figure 3). The SVM predictive accuracy is measured as correlation r^2 between the sample scores and SVM predictions of those scores, preserving the sign of the correlation. SVM performance is estimated using the leave-one-out procedure (see text). Gray curve: SVM is trained and tested on the permuted sample scores. Black curve: SVM is trained and tested on the "normalized" scores; i.e., adjusted at each repeat of the leave-one-out procedure to have the average score of 39 SVM training samples be equal zero.

In fact, there are two problems. The first one is that when an SVM cannot learn (i.e., when there is no predictive relationship between input channels and the desired output), it will learn to output the average score of all training samples, regardless of the input vector. Thus, if the average score across all (n) samples is M_{all}, and we set aside sample i for testing during the leave-one-out procedure, then the SVM will learn to output the new average of the training samples $M_{all\text{-}i} = (n*M_{all} - Score_i)/(n - 1)$. At the end of the leave-one-out procedure, we will measure the SVM accuracy across all n samples by correlating $Score_i$ with $M_{all\text{-}i} = constant - Score_i/(n - 1)$, giving us a negative correlation.

To remedy this artifact of the leave-one-out procedure, we have to mean-center the scores so that every time we leave out one sample, the scores are adjusted so that the mean of (n – 1) samples used to train the SVM is zero. Using this correction, we generate a new performance curve, shown in Figure 4 in black. Now the performance does not go below zero, when all or most of the bins are used.

However, we still have erroneously good performance when the first 40-80 bins are used. The reason is that the first bins have the highest correlation with the scores across the available 40 samples (although objectively their correlation is zero). Thus they can predict to some degree the scores of the left-out samples (although they would, of course, fail on newly generated samples). To correct this problem, each time we train an SVM we should select for its inputs the bins that have the highest correlations computed on all the samples except the one that is left out. Then we will get the expected flat no-performance, as shown in Figure 5 by the gray curve.

With these two corrections, we now are ready to return to our demonstration dataset of "NMR" spectra and train an SVM on the highest bins (properly determined) to predict the score (properly mean-centered). The resulting SVM performance is plotted in Figure 5 in black. The curve shows that when we used progressively more bins (up to 8), the SVM performance grew progressively. Using more than 8 highest-correlated bins was detrimental, and progressively more so when more bins were used.

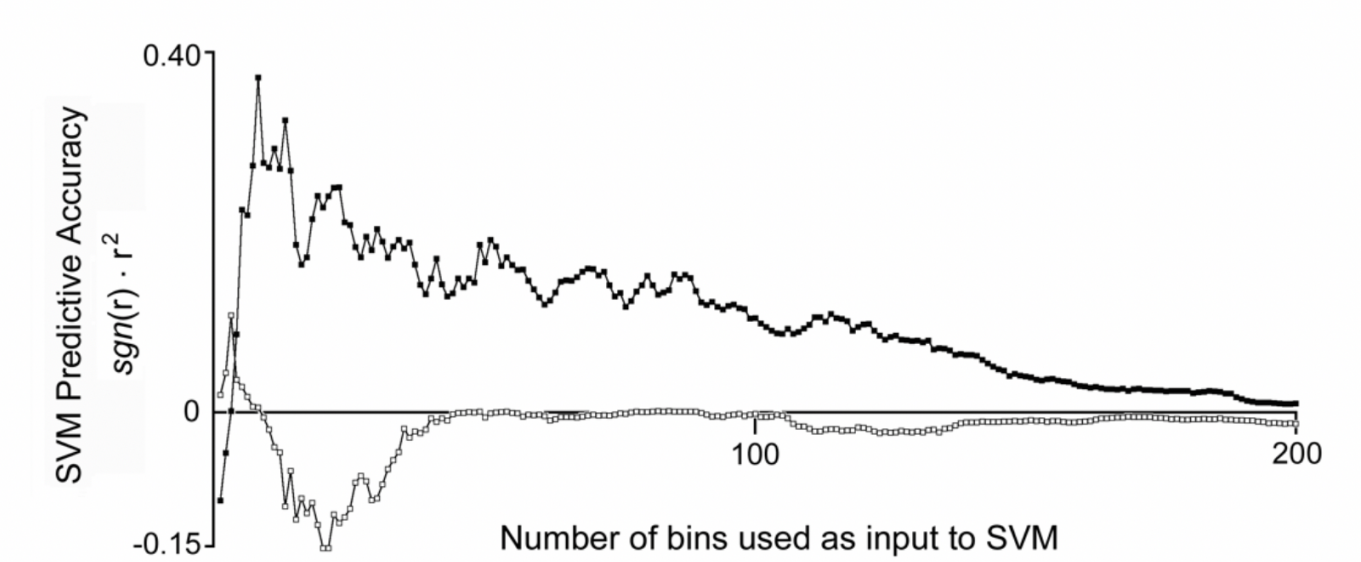

Figure 5. *SVM predictive accuracy estimated using the leave-one-out procedure and two sample-bias corrections.* The plot has the same format as in Figure 4. The two sample-bias corrections are done at each repeat of the leave-one-out procedure and involve: (1) the sample scores are normalized to have their mean on all SVM training samples be equal zero; and (2) bins are chosen for inclusion in the SVM input set based on their correlation with the score that is computed only on SVM training samples. Black curve: SVM is trained and tested on the original demonstration dataset of 40 "NMR" spectra. Gray curve: the scores are randomly permuted among the 40 data samples, thereby destroying any predictive relationship between the spectra and the scores assigned to them. Note that the estimated SVM predictive accuracy on this dataset is, appropriately, around zero.

STEPS 1-2: Establishing the Presence of Score-Related Information in NMR Spectra

Score-permutation test of the statistical significance of SVM predictive accuracy.

How statistically significant is the SVM performance in Figure 5? Can we conclude from this plot that the "NMR" spectra of the demonstration dataset carry information about the score? The peak SVM performance is above zero, but it is quite low (r^2 = 0.35). However, we should expect the performance to be partially compromised, because the top 8 bins used to generate this performance most likely include among them some irrelevant (and therefore performance-degrading) bins.

To estimate the statistical significance of the SVM performance of r^2 = 0.35, we can use a common permutation approach, which involves random shuffling of the scores among the 40 samples. We will test the *null hypothesis* that **the bins do not contain any information about the score and the above-zero performance of the SVM is simply due to chance**. Our *alternative hypothesis* will be that the above-zero accuracy of the SVM is not accidental. By randomly shuffling the scores among the samples we make sure that the bins do not have any information about the score. Then, according to the null hypothesis, the SVM accuracy on the original scores should fall within the range of accuracies achieved on the shuffled scores.

As a part of this approach, we will do leave-8-out version of the training/testing procedure, repeated 20 times (each time setting aside a different subset of 8 samples), thus using 8x20 = 160 samples in testing, rather than 40 in the original leave-one-out. This should improve the accuracy of the test.

Before we proceed with our score-permutation test, we have to select an unbiased SVM parameter-optimization procedure. That is, our goal is to compare the best SVM predictive accuracy on the

original scores with – equally important – the best SVM accuracy on each of the score permutations. (To remind, SVM performance depends in particular on its parameters C and g, as well as the number of input bins.)

To elaborate, we have to optimize SVM parameters for the original dataset; otherwise the SVM performance will be poor and indistinguishable from random performance. But if we optimize SVM parameters for the original dataset, then we have to optimize SVM parameters for the permuted datasets equally well. Unfortunately, since these permuted datasets have no predictive relationship between the bins and the score, there are no objectively optimal SVM parameter settings – only pseudo-optimized specifically to that set of data samples. Consequently, from one permutation to the next, the "optimal" SVM parameter values will vary greatly, which makes them harder to find.

Procedure for optimizing SVM parameters.

Thus, we have to adopt a uniform parameter-search procedure that will find optimal (or near-optimal) SVM parameters for the original dataset, and will also have equally good chances of finding the optimal parameters for each score permutation. An exhaustive parameter search is not practical, since it would be very time-consuming, given that here we have a very large three-dimensional parameter space: it is defined by the number of input bins and SVM parameters C and g, whose optimal values can vary by several orders of magnitude. Instead, we should adopt a reduced search protocol that is guided only by general considerations and execute it exactly the same on every dataset.

With these considerations in mind and based on our general experience with SVMs, we selected the following parameter-optimization procedure. Given a particular dataset (original or permuted), we start by using the default value of parameter C (C =1) and using as SVM inputs the 30 bins most highly correlated with the score (our heuristic here is that the number of input SVM channels should be large, but less than the number of training samples). We train/test an SVM (using our leave-8-out procedure) with a wide range of g-parameter settings, certain to include the range of possible optimal g values. Based on our general experience with RBF-kernel SVMs, we try 30 g -parameter values, starting at 1.5 and successively reducing the g-value by 25%, ending up with the last value of 0.0004. The g-value that gives the best SVM performance is chosen as "provisionally optimal." Next we optimize C-parameter by using the provisionally optimal g-value and 30 most correlated bins. Now we train/test the SVM with a wide range of C-parameter values, starting at clearly too-high value of 128 (this will force the SVM to overfit on the training set) and gradually reducing the C-value each time by 50% in 15 steps, ending up with the last value of 0.0001 (clearly under-fitting). The C-value that gives the best SVM performance is again chosen as "provisionally optimal." Next we optimize the number of input bins, trying all values between 15 (with fewer input bins, an SVM can produce spuriously high performances) and 30. The number of bins that gives the best SVM performance is taken as "finally optimal."

Generally, the C- and g-values optimal for 30 input bins are not optimal for the finally-optimal number of bins. To find the finally-optimal C- and g-values, we repeat (exactly as before) the 30-step search of g-parameter (from 1.5 to 0.0004), followed by 15-step search of C-parameter (from 128 to 0.0001).

Application of the score-permutation test to the demonstration dataset of "NMR" spectra.

To perform the permutation test of the statistical significance of the SVM predictive accuracy, we first use our uniform parameter-optimization procedure on the original dataset and determine that according to it the optimal SVM parameters are: C = 16, g = 0.0015, number of input bins = 20. With these parameter settings, the SVM achieves test predictive accuracy of r^2 = 0.23. Next, we randomly shuffle the scores among the 40 samples, generating a new, "randomly-permuted" dataset. We use our uniform parameter-optimization procedure on this dataset, determine its optimal SVM parameter settings and measure the SVM test predictive accuracy. Such generation of

randomly-permuted datasets and measurement of SVM performance on those datasets is repeated 50 times. These measurements are plotted in Figure 6A.

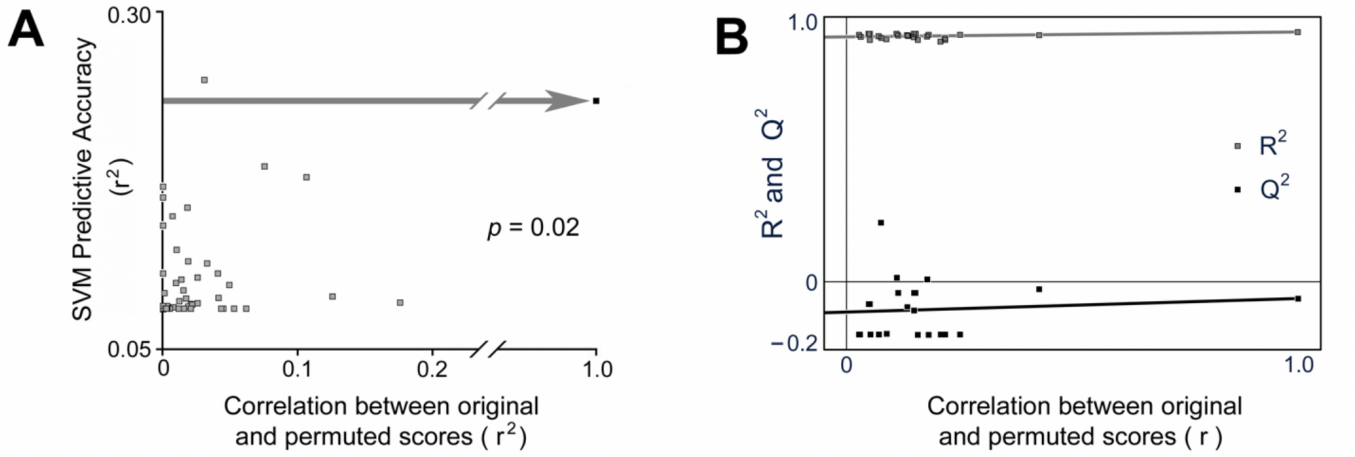

Figure 6. *Establishing the presence of score-related information in the "NMR" spectra of the demonstration dataset.* A: score-permutation test of the statistical significance of SVM predictive accuracy. The plot shows SVM predictive accuracy on the original demonstration dataset (black point, with an arrow pointing at it) and on 50 other datasets, each one generated by randomly permuting the scores among the samples in the original dataset (gray points). The permuted datasets are plotted along the abscissa according to correlation r^2 of their scores with the scores of the original dataset. Note a clearly superior predictive performance of the SVM on the original scores as compared to the permuted scores, which demonstrates that the studied spectra do carry information about the scores. B: score-permutation test of the statistical significance of PLS predictive accuracy. The outcome of the PLS analysis is shown in the standard Umetrics format. R^2 (gray points) describes how well the derived PLS model fits the training data, and is the proportion of the sum of squares explained by the model. Q^2 (black points) describes the predictive ability of the PLS model and is the cross-validated R^2. Note that Q^2 on the true scores (rightmost point, at r=1.0) falls in the middle of the range of Q^2 values obtained on permuted scores (points with r<0.5). PLS, thus, fails to detect any sign of score-related information in the studied "NMR" spectra.

According to the plot in Figure 6A, the test performance of SVM on the original dataset was exceeded by only one out of 50 permuted datasets, indicating that the probability that our null hypothesis is true (i.e., that the above-zero SVM performance on the original dataset is accidental) is around 2%. This finding of p = 0.02 allows us to state with 98% confidence that the spectra do carry significant information about the score.

Would we be able to reach the same conclusion if we used the conventional metabolomics approach of PLS analysis? Applying the Umetrics software package to the same original dataset of 40 samples, we built and validated a PLS model for predicting sample scores. Figure 6B is the standard plot generated by the software, showing the PLS model performance on the training data (R^2) and the test data (Q^2) using the true scores, as compared to using permuted scores. The predictive performance of PLS on the test data is clearly at a chance level, which would lead us to conclude – erroneously – that these spectra carry no information about the score.

STEP 3: Identification of Bins Carrying Score-Related Information

Bin elimination procedure.

The successful SVM performance in Figure 6A on the original dataset was achieved using 20 bins as inputs to the SVM. It should be pointed out, however, that the entire set of bins that served as SVM inputs is greater than 20. To remind, the SVM was trained and tested using a leave-8-out procedure with 20 repetitions. It is true that in each repetition 20 bins most highly correlated with the score were used as SVM inputs. However, those correlations were computed only across 32 data samples used in that repetition to train the SVM. Since in each repetition a partially different subset of samples was used to train the SVM, each bin's correlation with the score varied to some

degree across the repetitions. As a result, some bins ended up being used as SVM inputs in all 20 repetitions, others were never used, and yet others were used, but only in some of the repetitions. Thus, across all 20 leave-8-out repetitions, 69 bins were used at least once as SVM inputs, and thereby contributed to its statistically significant score-predicting performance.

Included among the 69 bins are seven truly significant bins (i.e., the bins from which the score was computed in the first place). The other 62 bins are, objectively, insignificant bins and it would be highly desirable to discard at least some of them, so as to narrow down – as much as possible – the list of the potentially significant bins. On what grounds would we be able to discard some bins as insignificant?

An obvious approach (called a “wrapper” approach; Kohavi and John, 1997) would be to measure how important is each bin for the success of the SVM predictive performance. A particularly robust and reliable procedure that implements this approach is backward sequential input variable elimination (Bishop, 1995). To follow this procedure, we should start with the full set of 69 SVM-used bins. For each of these 69 bins, we train/test a separate SVM, whose inputs include all but this bin. Our intent here is to measure the relative importance of each bin by how much damage its absence does to SVM performance. The bin, in which absence the SVM achieved the highest test performance, is identified as the “least significant.”

In the next step, this least significant bin is removed from the list and the same procedure of training/testing of SVMs is repeated on – now – 68 bins, identifying the next “least significant” bin, as the one in which absence the SVM achieved the highest among 68 performances. This bin is removed next from the list, reducing it to 67, and so on. This gradual “pruning” of the list continues until we are left with just one remaining bin.

In this process we generate a new list: i.e., the order in which the bins were removed, reflecting the relative importance of those bins to SVM test performance.

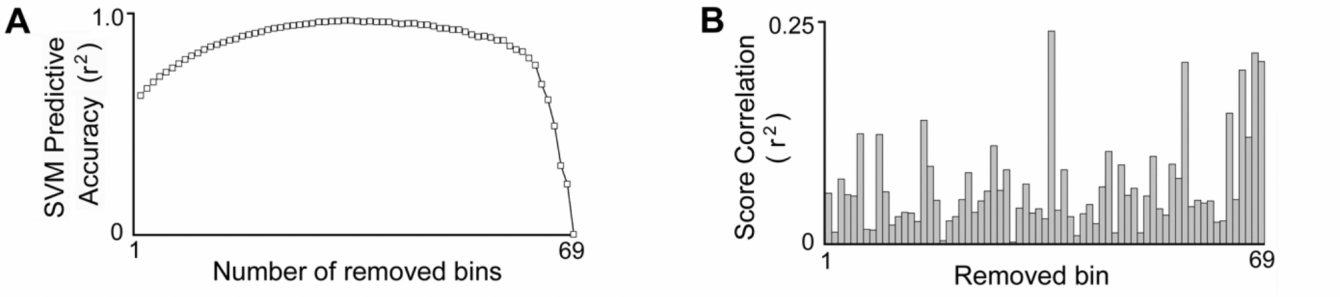

Figure 7. SVM predictive accuracy on the demonstration dataset is plotted after removal of each successive bin from the SVM input set. The input bin set initially comprises those 69 bins that were used at least once as SVM inputs in the score-permutation test (see the main text). The bins are removed from the set one by one, selected according to the backward sequential elimination procedure explained in the main text. Note that with each successive bin removal, SVM predictive performance at first improves, but towards the end falls precipitously. Throughout this process, the same SVM C- and g -parameters are used as those found to be optimal during the score-permutation procedure (i.e., C=16, g=0.0015). B: correlation r^2 of each of the bins removed in A with the score. Note that the order in which the bins are removed by the backward sequential elimination procedure is not tied to their correlation with the score.

Figure 7A plots the results of this bin-elimination procedure. According to this plot, in the first 20-30 pruning steps a removal of each successive bin resulted in an improvement of the SVM performance. This is not surprising, since these bins were insignificant ones and, as we have already seen in Figure 2, the presence of irrelevant bins degrades SVM performance. In the middle of the pruning process (30-50 steps into the process), a bin removal does not have a significant effect on SVM performance. Finally, towards the end of the pruning process the effect of a bin

removal becomes negative and progressively more pronounced with fewer remaining bins.

To show that this bin-elimination procedure does not simply remove bins in the order of their correlation with the score, the score correlations of the removed bins are plotted in Figure 7B.

The Figure 7A plot does not show any great discontinuities in SVM performance that could be clearly indicative of when the process finished removing the insignificant bins and started to remove the significant ones. To identify significant bins, it is helpful to re-plot Figure 7A in terms of the difference between SVM performance in step i-1 and step i (i.e., $\Delta r^2 = r^2_{i-1} - r^2_i$). This difference expresses the contribution of the bin removed in step i to the SVM performance (i.e., in step i-1 this bin is available to the SVM, but in step i it is not).

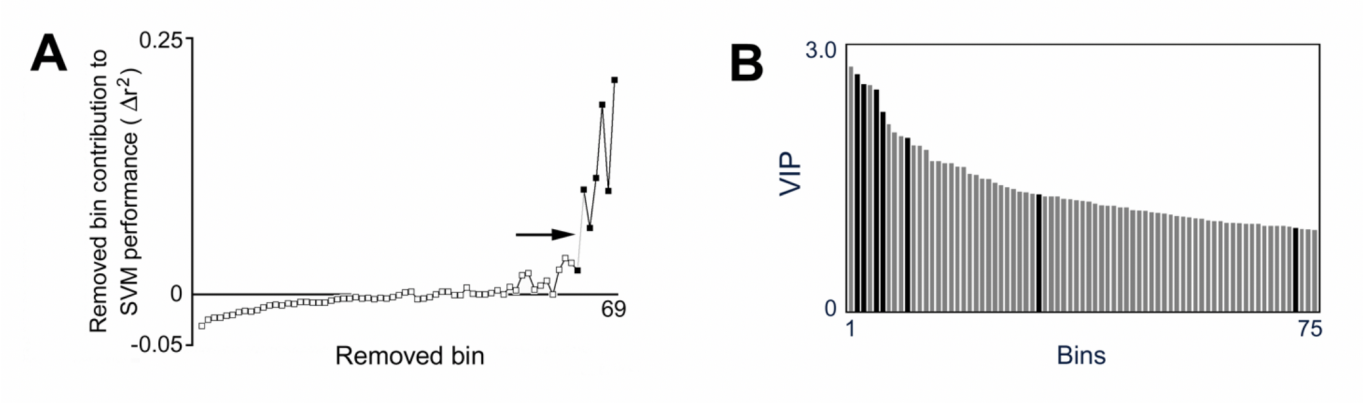

Figure 8. *Identification of score-significant bins in the "NMR" spectra of the demonstration dataset.* A: SVM-based bin-elimination procedure. The contribution of a bin to SVM predictive performance, Δr^2, is estimated by the difference in SVM predictive accuracy with vs. without an input from that bin. The contribution of each bin removed during the backward sequential elimination procedure (see the main text and Figure 7A) is plotted in the order of its removal, revealing a very slow upward creep in contributions of successively removed bins (starting from negative to eventually positive), which is characteristic of insignificant bins. The last 6 bins to be removed, however, break with this slow trend and have their own, elevated and fast rising, trend. The point of this trend transition – between the 7th and 6th remaining bins – identifies the last 6 bins as most likely to be significant. B: PLS identification of score-significant bins. The outcome of the PLS analysis is shown in the standard Umetrics format, plotting VIP (variable influence on projection) of the 75 bins most important to the developed PLS model. In both A and B the seven truly significant bins are shown in black. Note that while the SVM-based procedure correctly kept all seven significant bins until the end and identified six of them as potentially significant, the PLS-based procedure was no more successful than simply using bin correlations with the score (Figure 3).

Using such a difference plot (Figure 8A), we can select the bins that are most likely to be the significant ones. Starting from the left in Figure 8A, obviously we can discard as insignificant all the bins for which $\Delta r^2 < 0$ (i.e., their removal improved the SVM performance). Also can be clearly discarded the bins in the middle of the plot, where Δr^2 values hover around 0. That leaves us with 10 rightmost bins (defining this group by the rightmost bin with $\Delta r^2 \leq 0$; i.e., the 59th removed bin). To further narrow down the list of potentially significant bins, we can look for a point of transition in the Δr^2 curve from, on the left, a gradually rising trend (with some minor jitter) to, on the right, a distinctly different curve segment. In Figure 8A this point, marked by an arrow, happens to be between the sixth and seventh bins from the right. Thus we can define as the "potentially significant bins" the six rightmost bins.

How successful was our bin selection process? Were we able to correctly identify the truly significant bins? It turns out that we did very well. The seven truly significant bins (i.e., bins #1-7) are shown in black in the Figure 8A plot, revealing that the bin-elimination procedure correctly recognized them and removed all the other (insignificant) bins before this group. Then, with the use of the point-of-transition test, we picked all but one of the seven significant bins.

To contrast the bin-selection performance of this bin-elimination procedure with the standard PLS-based approach, Figure 8B shows the plot generated by the PLS software, in which the bins are

sorted from left to right in the descending order of their importance to the PLS model of the data (only 75 most important bins are plotted). The truly significant bins are shown in black, irrelevant bins are shown in gray, revealing that the standard PLS method was much less effective than the SVM method in identifying the significant bins. In fact, the order of the bins in Figure 8B mostly reflects their correlation with the score.

Sensitivity of bin-elimination procedure to dataset size and bin significance.

The performance of the bin-elimination procedure shown in Figure 8A is very impressive, especially considering that many of the significant bins had relatively low correlation with the score and did not stand out among other, insignificant bins (see Figure 3). It should be pointed out, however, that such success is not guaranteed, when dealing with small numbers of data samples (relative to much larger numbers of bins), coupled with low correlations between significant bins and the score. To provide an example, we generated another dataset of 40 random samples, in which the score was computed from the first seven bins plus a hidden bin according to the same formula used for the first dataset. The only difference between the two datasets was in using a different Random Seed Number to start the random number generator in the Matlab program. Using our score-permutation procedure for testing whether the bins carry information about the score, we establish that SVM predictive accuracy on this dataset (r^2 = 0.67) was never approached or exceeded by 50 randomly permuted datasets. Next, we perform the bin-elimination procedure and plot its outcome in Figure 9A. Relying on the point-of-transition test, we select the three rightmost bins as the potentially significant bins. How well did we do? The truly significant bins are plotted in black, revealing that (1) two of the seven significant bins had such low correlations with the score that they were omitted from the analysis, and (2) the bin-elimination procedure was less than 100% effective in recognizing the five bins that were analyzed – among the three “potentially significant bins” only two are truly significant and the third one is False Positive. Three significant bins were misclassified as insignificant.

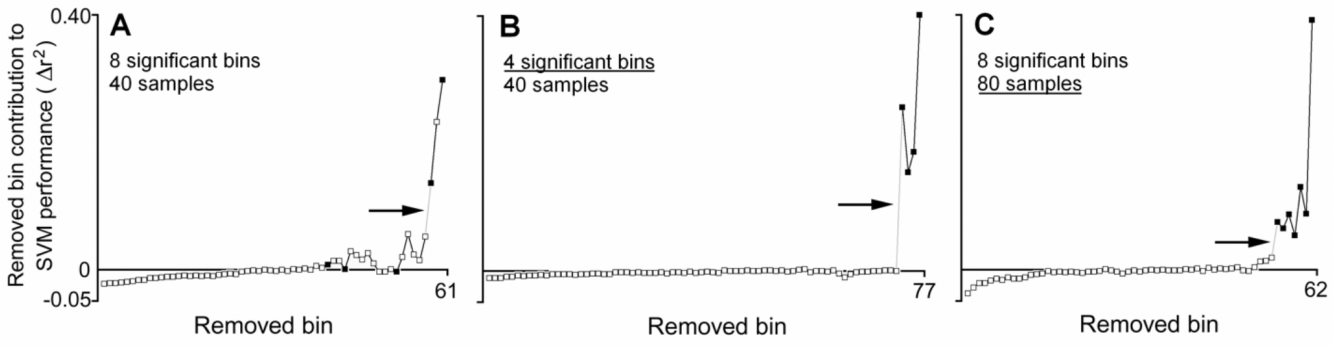

Figure 9. *Dependence of the effectiveness of the bin-elimination procedure on the dataset size and bin significance.* A: performance of the bin-elimination procedure on a dataset more “difficult” than the one in Figure 8. Note that only five significant bins (black points) happened to have enough score correlations relative to all other bins to be included in the analysis and only two of them were identified as potentially significant. B: performance of the bin-elimination procedure on the same dataset made less “difficult” by increasing the strength of association between significant bins and the score (by computing the score using the first four, rather than seven + one hidden, significant bins). C: performance of the bin-elimination procedure on the same dataset as in A, but made less “difficult” by doubling the number of data samples to 80. Note that in both B and C the relatively small “improvements” in the dataset made the bin-elimination procedure 100% successful in identifying all the significant bins.

The probability of False Positives and False Negatives is reduced (1) when significant bins happen to have stronger correlations with the score (since then they will stand out more among all other bins), or (2) when we can use more data samples in SVM training and bin selection (since this will give us more accurate estimates of true correlations of bins with the score, again making significant bins stand out more among all other bins).

To illustrate the first point, we use the same “difficult” dataset shown in Figure 9A, but compute the score using only the first four bins, rather than the first seven bin plus a hidden bin. Consequently, the now four significant bins in this case account each for 25% of the behavior of the score, rather than for only 12.5% in the original case. The outcome of the bin-elimination procedure applied to this dataset is plotted in Figure 9B, showing that now the bin-elimination procedure was able to correctly classify as “potentially significant bins” all four truly significant bins.

To illustrate the positive effect of the size of the dataset, we again use the same “difficult” dataset shown in Figure 9A, but generate 40 additional new data samples. The score in this case is still computed according to the original formula as a weighted sum of values of the first seven bins plus a hidden bin. Thus, in this case each significant bin accounts for only 12.5% of the behavior of the score, but there are twice as many data samples available for the analysis. The outcome of the bin-elimination procedure applied to this dataset is plotted in Figure 9C, showing again that the bin-elimination procedure was able to correctly recognize as “potentially significant bins” all seven truly significant bins.

Examples of performance of bin-elimination procedure.

Additional examples of the performance of the bin-elimination procedure are shown in Figure 10. Eleven new datasets were generated of the same degree of difficulty as the first two datasets (shown in Figures 8 and 9A). That is, these datasets also consist of 40 randomly generated 200-bin samples and for each sample a score is computed from the first seven bins and one hidden bin according to the same formula used before. Each new dataset was evaluated using our permutation test for the presence of score-related information in the spectra. Only six of the datasets passed this test. Thus, the overall probability of correctly detecting that such spectra contain score-predictive information is 8 out of 13, or ~60%, indicating that such datasets (having only 40 samples of spectra in which only 7 out of 200 bins carry some – and not much, only 12.5% each – information about the score) are very challenging and are close to the limit of analyzable spectra. In other words, such spectra cannot be made much more difficult (e.g., by reducing the number of samples or reducing information content of significant bins) before it will become completely impossible to detect score-related information in them.

The six datasets that passed the test were subjected to the bin-elimination procedure, the outcomes of which are shown in Figure 10A. The plots show a full range of outcomes, from correctly finding all seven significant bins to identifying – incorrectly – only one bin as significant. Overall, across all eight datasets that passed the first test, 35 bins were identified as potentially significant (out of 8x7=56 maximally possible, or again ~60%). Most importantly, only 5 out of 35 identified bins were in fact not significant, indicating the False Positive rate of only 14%. In contrast, for example, if we selected significant bins based on the statistical significance of their correlation with the score, we would have a much higher False Positive rate of 44%. Thus, even on the least analyzable datasets, the bin-elimination procedure still performs quite reliably.

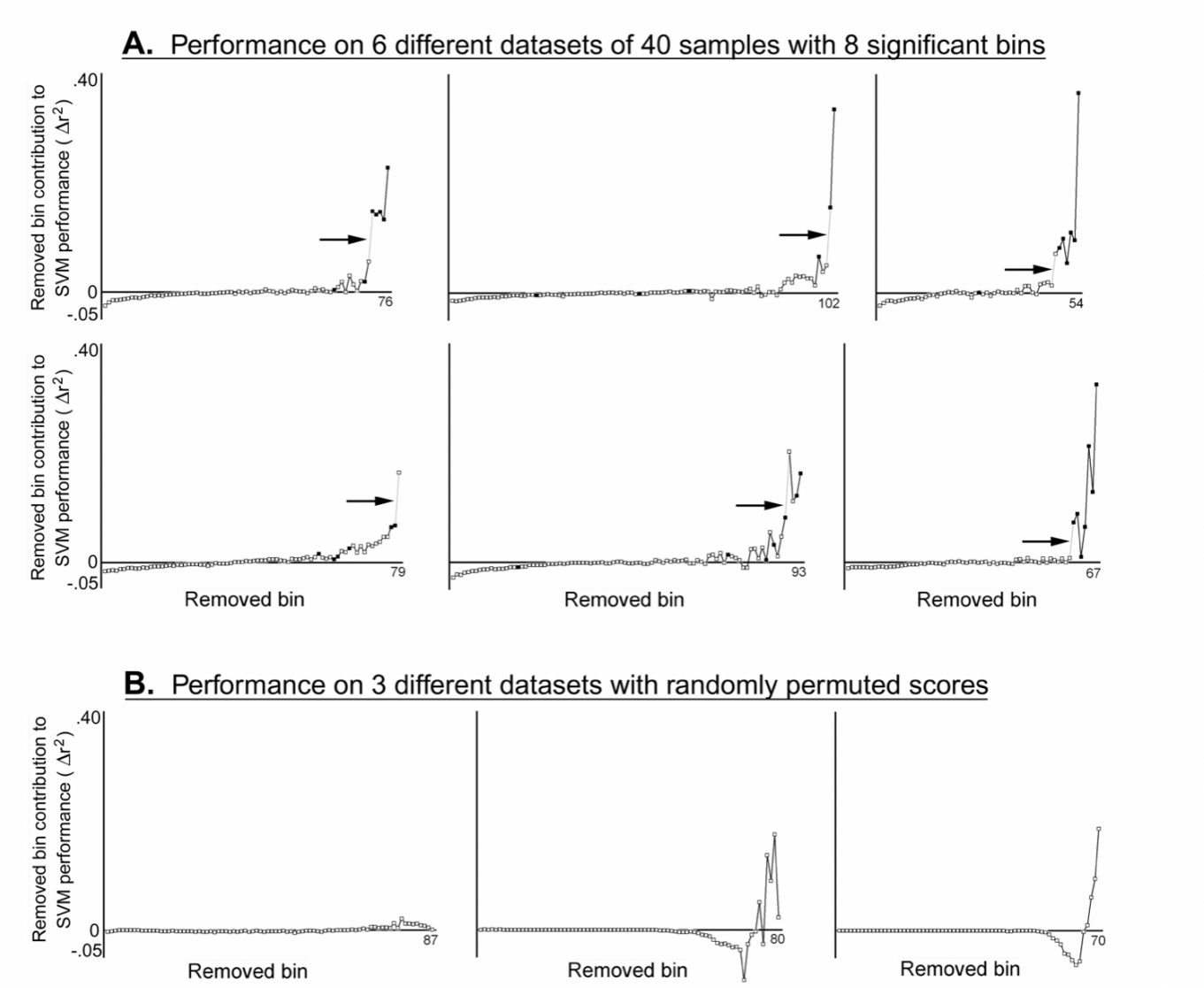

Figure 10. *Illustrative examples of the performance of the bin-elimination procedure on additional datasets generated according to the same rules as the demonstration dataset (see Methods).* **The plot format is the same as in Figure 8A. A: in each dataset shown, the spectra carry score-related information in seven significant bins. B: in each dataset shown, the scores are randomly permuted among samples, making them information-free with regard to the score.**

How different would the plots generated by the bin-elimination procedure be on datasets that carry no information about the score? Three examples are shown in Figure 10B. These examples span the range of what we have observed among 20 datasets in which scores were randomly permuted among samples, thus totally destroying any relationship between a spectrum and the score attached to it. While many plots are flat (similar to one in the left panel), others do exhibit a fast rise at the end of the procedure, when just a few bins remain in the set that is used as input to an SVM. Such plots look quite similar to the ones generated on the score information-containing datasets (see Figure 10A), but they have a distinctive feature that clearly sets them apart. To see this feature, compare plots in Figure 10A with those in Figure 10B. The first among the removed bins in information-free datasets (Figure 10B) make no contribution to the SVM performance, whereas in information-containing datasets (Figure 10A) they make negative contribution to SVM performance. Next, moving to the right in each plot before the curve begins to rise abruptly: in Figure 10B the curve dips below zero and then recovers, whereas in Figure 10A the curve gradually rises above zero. While this feature can be used as a warning flag, indicating that the last removed

bins are not significant, a more general rule should be that the bin-elimination procedure should not be used on datasets that did not pass the first, score-permutation test of their score information content.

Use of a “Validation” Dataset

STEP 4: Testing the statistical significance of the selected bin set.

Bins selected as potentially significant by the bin-elimination procedure can be further validated if a new set of spectra becomes available. To illustrate our approach, we created an “original” dataset of 40 randomly generated 200-bin spectrum samples and then generated an additional, “validation,” set of 20 samples. The “original” dataset passed the first, score-permutation test for score-related content and the bin-elimination procedure (plotted in Figure 11A) selected bins #1, #77, #5 and #7 as potentially significant. Thus, the “original” dataset yielded four potentially significant bins, one of which happened to be wrong.

Before we use the “validation” dataset, we should develop our “best” SVM model for score prediction by training the final SVM on the “original” dataset, using for its inputs only the four bins chosen as potentially significant by the bin-elimination procedure. To generate the “best” SVM model, we have to find SVM parameters C and g that are optimal specifically for these four input bins. Thus, as before, we use the “leave-8-out” SVM training/testing protocol and try 30 values of g-parameter (in steps of $g_{i+1} = 0.75g_i$ starting from $g_1 = 1.5$), followed by 15 values of C-parameter (in steps of $C_{i+1} = 0.5C_i$ starting from $C_1 = 128$). We find that at $C = 2$ and $g = 0.015$, the SVM achieves its best test accuracy ($r^2 = 0.799$). We then generate the “best” SVM model by training an SVM with $C = 2$ and $g = 0.015$ using all 40 samples of the “original” dataset.

We next turn to the “validation” dataset and obtain the responses of the “best” SVM model (developed on the “original” dataset) to the 20 new validation samples. The SVM performance on these previously unused samples drops significantly to $r^2 = 0.235$. This drop is expected; it indicates that in order to maximize SVM performance on the “original” dataset, we somewhat over-fitted the SVM parameters.

Is this test performance of the SVM model statistically significant? To answer this question, we again use the score permutation strategy (randomly shuffling the scores among the 20 validation samples) and compute r^2 between these permuted scores and the outputs of the SVM model. Such permutations are repeated 100 times, yielding 100 values of r^2, which are plotted in Figure 11B in relationship to r^2 obtained on true (not permuted) scores. As Figure 11B shows, out of 100 attempts the value of r^2 on permuted scores exceeded only once the r^2 value on the true scores. Thus we can conclude with 99% confidence that the predictive performance of the SVM model is not random and the four potentially significant bins, as a group, do carry significant information about the score.

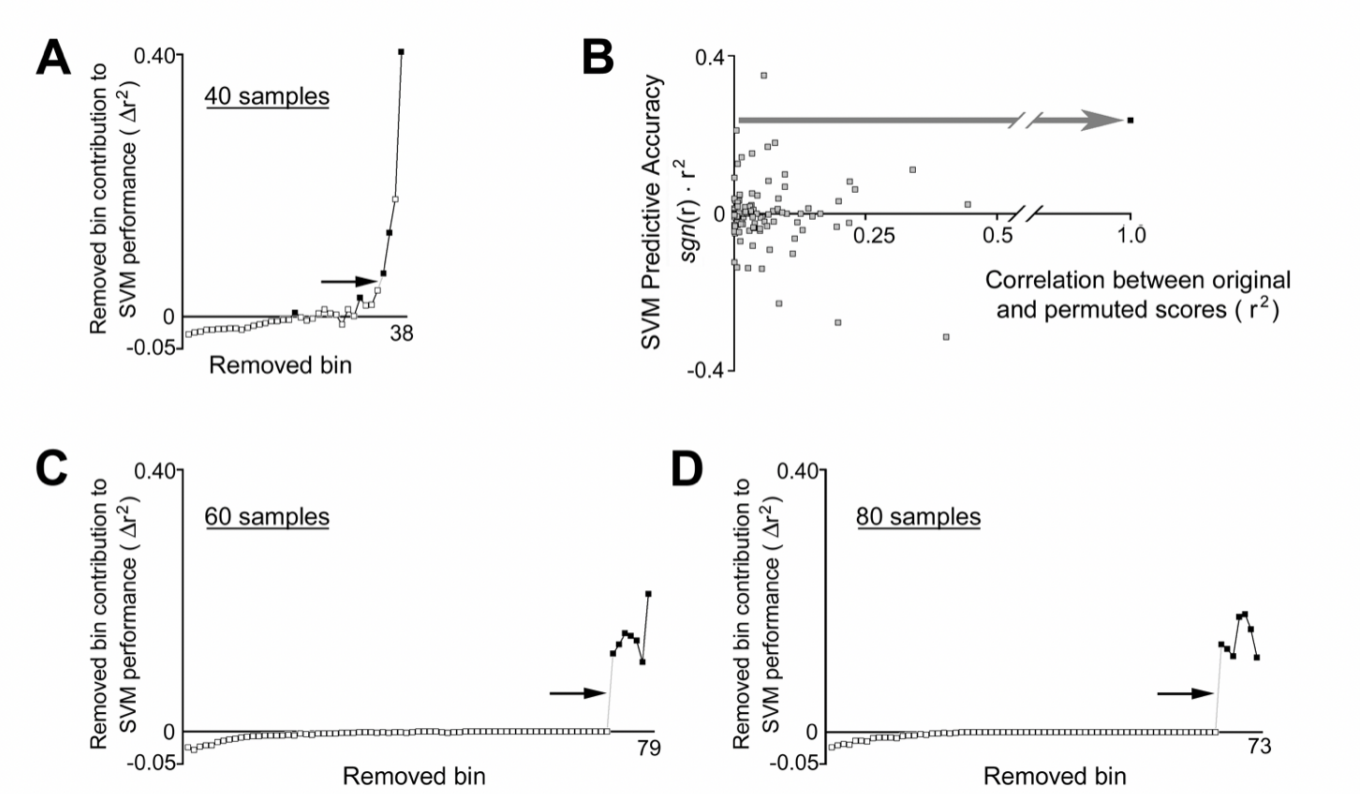

Figure 11. *Validation of the bins selected as potentially significant on a 40-sample dataset using additional data samples.* A: performance of the bin-elimination procedure on the first 40 samples. Note that only four bins are identified as potentially significant, one of them incorrectly. Thus, this "difficult" dataset presents a good challenge to validation procedures. B: score-permutation test of the statistical significance of the selected set of four potentially significant bins. The plot shows the predictive accuracy of the "best" SVM model (an SVM trained on the 40 first data samples using just these four bins) on 20 new, "validation," data samples. Black point, with an arrow pointing at it, shows the SVM accuracy on true scores, gray points show SVM accuracy on 100 different random permutations of the scores among the 20 validation samples. Note that SVM accuracy on the true scores is clearly superior to accuracy on permuted scores, thus establishing that the four chosen bins do carry information about the scores. C: test of validity of individual selected bins. Shown is the performance of the bin-elimination procedure on the expanded, 60-sample dataset, which combines 40 original and 20 validation data samples. Note that all seven truly significant bins are now identified by the procedure as potentially significant. D: final validation test of individual selected bins. Shown is performance of the bin-elimination procedure on an even larger, 80-sample dataset, which combines 40 original, 20 first validation, and 20 new – second validation – data samples. Note that the same seven bins as in C are again selected by the procedure, indication that they comprise a stable – final – selection.

STEP 5: Re-evaluating the list of selected bins.

One limitation of this verification test is that it does not assign statistical significance to individual bins: although, based on the test, we are now confident that the four selected bins together carry significant score information, we cannot conclude that each of the four bins is significant. And, in fact, bin #77 was selected mistakenly. How can we test the validity of individual selected bins? Unfortunately, a direct test of the validity of individual bins will require a much larger number of data samples than we used so far (on the order of 100s). An indirect indication of the significance of the bins can, however, be obtained by combining 20 samples of the validation dataset with 40 samples of the original dataset and performing the bin-elimination procedure on this larger set of 60 samples (see Figure 11C). The bin-elimination procedure is very efficient at utilizing additional data samples to produce more secure bin selections. As comparison of plots in Figures 11A and 11C shows, 20 additional samples were already sufficient to allow the bin-elimination procedure to correctly select all seven significant bins. Of course, normally we would not know that the newly selected bins are all correct, but we can draw confidence from the fact that the originally selected bins #1, #5, and #7 were selected again using the larger dataset, while bin #77 was dropped. Thus we can interpret the combined results of the two bin-elimination procedures as: (1) strongly

suggesting that bins #1, #5, and #7 – but not bin #77 – are significant, and (2) identifying with less confidence bins #2, #3, #4, and #6 as potentially significant.

To obtain greater confidence in these identifications, yet another set of data samples should be collected and the bin-elimination procedure should be performed on all the available data. Figure 11D plots the outcome of the bin-elimination procedure on 60+20 new data samples. As the plot shows, the bin-elimination procedure again unambiguously selected bins #1, #2, #3, #4, #5, #6, and #7 as potentially significant. Such a re-selection of the same set of bins indicates that already with 60 samples the bin-elimination procedure had reached a stable outcome, converging on bins #1-7 as the final choice of significant bins.

Discussion

SVMs have already been used successfully by Masoum et al. (2007) to analyze ^{1}H NMR spectra derived from fish oil. The oils were extracted from the salmon caught at a variety of geographic locations and were used to determine whether a given specimen was wild or farm-raised and what country did it come from. This study acquired a relatively large set of spectral samples, which allowed the authors to divide the dataset into three non-overlapping subsets: 74 samples were used exclusively to train SVMs, 45 other samples were used exclusively to optimize SVM parameters, and yet 22 other samples were used exclusively for final validation of SVM classification performance. A highly successful classification performance of SVMs made it unnecessary to devise any special procedures for more economical use of the data samples.

SVMs have been used more extensively in gene expression studies, which employ the DNA microarray technology (e.g., Mukherjee et al., 1998; Furey et al., 2000; Guyon et al., 2002; Rakomamonjy, 2003; Niijima and Kuhara, 2006). Microarray and NMR studies have similar tasks: learning to recognize in their measurements the signs of biological conditions of experimental or clinical importance. Both types of studies are also similarly constrained in how many biosamples they can afford to obtain. However, a typical microarray generates orders of magnitude more measured variables (gene expressions) than does an NMR spectrum. Consequently, microarray studies face even more severe analytical challenges than do NMR studies.

The published microarray SVM studies (e.g., Mukherjee et al., 1998; Furey et al., 2000; Guyon et al., 2002; Rakomamonjy, 2003; Niijima and Kuhara, 2006) have been quite successful in achieving their goals of diagnosing various cancers based on gene expression patterns and identifying genes that can be used as biomarkers of those cancers. However, it is difficult to evaluate the full diagnostic and biomarker-identification powers and limitations of the SVM-based analytical procedures employed in those papers, since the particular problems to which those procedures were applied were not very challenging, and also since we lack an objective knowledge of which genes should have been identified as biomarkers in those studies.

In comparison with published SVM-based procedures for microarray data analysis, the package of SVM-based procedures we recommend in this paper for metabolomics data analysis is, we believe, more thorough in extracting significant information from mega-variable, but sample-limited experimental data. A major advantage it has over microarray methods is that it has to deal only with a small fraction of variables faced by microarray studies. Other important design features of our analytical package include the following:

1. Given only a limited number of data samples, an SVM should not be trained indiscriminately on all the available measured variables – all the bins in the NMR spectrum – because supplying an SVM with many irrelevant inputs will greatly reduce its performance (see Figure 2). The smaller the fraction of bins relevant to a given discrimination (or regression) task, the more acute will be this problem. To reduce the number of SVM-studied bins, we chose to use Pearson’s correlation coefficient, but other measures of association between

individual bins and the target score (e.g., *t*-statistic, □' of Signal Detection Theory, Fisher score, ROC-derived indices, etc.) are likely to be comparably effective (e.g., Cho, 2002; Pepe et al., 2003; Kim et al., 2006). We select the optimal number of SVM-studied bins not by a rigidly preset criterion, but by SVM performance on progressively more restricted subsets of bins. The optimal subset is then validated with the score-permutation test.

2. The bin-elimination procedure also should not be used indiscriminately on all the spectral bins. If SVMs trained during the bin-elimination procedure become confused by the presence of many irrelevant bins, their bin-selection performance will be meaningless. That is why we restrict application of the bin-elimination procedure to only a reduced – but "enriched" – subset of bins chosen and validated by the preceding procedure (STEPS 1-2). Another related restriction on the use of the bin-elimination procedure is that even on the optimal subset of bins the procedure should not be used if that subset failed the score-permutation test, else the procedure will generate spurious results.
3. Instead of our bin-elimination procedure, a number of papers developed – in the context of microarray studies – other procedures for selecting significant bins (e.g., Weston et al., 2000; Guyon et al., 2002; Guyon and Elisseeff, 2003; Rakomamonjy, 2003; Zhang et al., 2006; Niijima and Kuhara, 2006). Computational speed is the main advantage of those procedures, and it is an important issue for microarray applications, since they have to search through much larger numbers of variables, on which the bin-elimination procedure would be unacceptably slow. However, such speed optimizations are based on various theoretical assumptions or approximations, which might impair the effectiveness of those procedures on more difficult problems, such as those described in this paper. To avoid such potential failures, and since computational time required by the bin-elimination procedure on metabolomics problems (on the order of hours) is not too long, we prefer to use the bin-elimination procedure.
4. Another commonly used timesaving device is to remove more than one variable at each step of a variable-elimination procedure (e.g., Guyon et al., 2002; Weston et al., 2003; Ding and Wilkins, 2006). Again, while it greatly speeds up the selection process, such a shortcut reduces the sensitivity of the procedure and should be avoided when investigating small datasets with only weak correlations between bins and the target score.
5. Our default SVM kernel is RBF. RBF-based SVMs are more versatile than linear SVMs, making accessible to analysis metabolomics problems characterized by nonlinear relations between metabolites and the conditions of interest. It might be tempting to use linear SVMs, if there are reasons to expect that in a given problem the spectrum-score relations are linear, but even in such cases we find RBF-based SVMs to do objectively better in finding significant bins during the bin-elimination procedure than linear SVMs.

Our set of analytical procedures has certain limitations, which should be taken into consideration when interpreting the results of any given study. In particular, since in the first round of bin selection (STEP 1), bins are chosen individually based on their linear correlation with the score, any bins that might have significant, but nonlinear or combinatorial relations with the score will be removed from the further analysis. This effectively narrows the range of discoverable spectrum-score relations to those having a prominent linear component. To expand the range of discoverable relations, different approaches to initial bin selection should be investigated in future.

Another limitation of our set of analytical procedures concerns the bin selection performance of the bin-elimination procedure on excessively limited numbers of data samples. Applied under such conditions, the bin-elimination procedure is likely to miss some significant bins, and it is also likely to erroneously select some irrelevant bins. On different small subsets of data samples, the bin-elimination procedure is likely to select somewhat different subsets of significant bins. However, with more samples, the bin-elimination procedure becomes more secure in its bin selection. The basic strategy, then, is to continue obtaining more data samples and performing the bin-elimination procedure on their progressively growing numbers until the selected bins stop changing. Of course, the ultimate judgment of the true significance of the selected bins will come from other, independent lines of biological inquiry (Ioannidis, 2005).

Acknowldegments

We thank Jason Winnike, who run Umetrics PLS analyses. Support for this research was provided by NIH grant R21 GM075941-01.

References

1. Masoum A, Malabat C, Jalali-Heravi M, Guillou C, Rezzi S, Rutledge DN (2007) Application of support vector machines to ^{1}H NMR data of fish oils: methodology for confirmation of wild and farmed salmon and their origins. *Anal Bioanal Chem* 387: 1499-1510.

2. Lindon JC, Holmes E, Nicholson JK (2001) Pattern recognition methods and applications in biomedical magnetic resonance. *Prog Nucl Magnetic Res* 39: 1-40.

3. Eriksson L, Johansson E, Kettaneh-Wold N, Wold S (1999) *Introduction to Multi- and Megavariate Data Analysis Using Projection Methods (PCA $ PLS)*. Umea, Sweden: Umetrics.

4. Griffin JL (2003) Metabonomics: NMR spectroscopy and pattern recognition analysis of body fluids and tissues for characterization of xenobiotic toxicity and disease diagnosis. *Curr Opin Chem Biol* 7: 648-654.

5. Holmes E, Antii H (2002) Chemometric contributions to the evolution of metabonomics: mathematical solutions to characterizing and interpreting complex biological NMR spectra. *Analyst* 127: 1549-1557.

6. Nicholson JK, Connelly J, Lindon JC, Holmes E (2002) Metabonomics: a platform for studying drug toxicity and gene function. *Nature Rev Drug Discov* 1: 153-161.

7. Reo NV (2002) NMR-based metabolomics. *Drug and Chemical Toxicology* 25: 375-382.

8. Bennett KP, Campbell C (2000) Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations* 2: 1-13.

9. Thissen U, Pepers M, Ustun B, Melssen WJ, Buydens LMC (2004) Comparing support vector machines to PLS for spectral regression applications. *Chemom. Intell. Lab. Syst.* 73: 169-179.

10. Belousov AI, Verzakov SA, von Frese J (2002) A flexible classification approach with optimal generalization performance: support vector machines. *Chemom. Intell. Lab. Syst.* 64: 15-25.

11. Joachims T (1999) Making large-scale SVM learning practical. In: *Advances in Kernel Methods - Support Vector Learning* , Schölkopf B and Burges CJC and Smola AJ (eds), MIT Press, Cambridge, MA, pp. 169-184.

12. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Discov*. 2: 121-167.

13. Vapnik V (1998) *Statistical Learning Theory*. Wiley, New York.

14. Vapnik V (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

15. Zhang X, Lu X, Shi Q, Xu X, Leung HE, Harris LN, Iglehart JD, Miron A, Liu JS, Wong WH (2006) Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data. BMC Bioinformatics 7:197.

16. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial Intelligence 97: 273-324.

17. Cho SB (2002) Exploring features and classifiers to classify gene expression profiles of acute leukemia. International Journal of Pattern Recognition and Artificial Intelligence 16: 831-844.

18. Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. Machine Learning 46: 389-422.

19. Rakotomamonjy A (2003) Variable selection using SVM-based criteria. Journal of Machine Learning Research 3: 1357-1370.

20. Weston J, Elisseeff A, Scholkopf B, Tipping M (2003) Use of the zero-norm with linear models and kernel methods. Journal of Machine Learning Research 3: 1439-1461.

21. Niijima S, Kuhara S (2006) Recursive gene selection based on maximum margin criterion: a comparison with SVM-RFE. BMC Bioinformatics 7: 543.

22. Ding Y, Wilkins D (2006) Improving the performance of SVM-RFE to select genes in microarray data. BMC Bioinformatics 7: S12.

23. Mukherjee S, Tamayo P, Slonim D, Verri A, Golub T, Mesirov JP, Poggio T (1998) Support vector machine classification of microarray data. *Technical Report CBCL Paper 182/AI Memo 1676 MIT*.

24. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16: 906-914.

25. Kim SY, Lee JW, Sohn IS (2006) Comparison of various statistical methods for identifying differential gene expression in replicated microarray data. Statistical Methods in Medical Research 15: 3-20.

26. Ioannidis JPA (2005) Why most published research findings are false? PloS Med 2: e124.

27. Pepe MS, Longton G, Anderson GL, Schummer M (2003) Selecting differentially expressed genes from microarray experiments. Biometrics 59: 133-142.

28. Bishop CM (1995) *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford.

29. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. Journal of Machine Learning Research 3: 1157-1182.

30. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2000) Feature selection for SVMs. In *NIPS* 13.