

Molecular Signatures from Gene Expression Data

Ramón Díaz-Uriarte

Abstract

“Molecular signatures” or “gene-expression signatures” are used to predict patients’ characteristics using data from coexpressed genes. Signatures can enhance understanding about biological mechanisms and have diagnostic use. Nevertheless, available methods to search for signatures fail to address key requirements of signatures, especially the discovery of sets of tightly coexpressed genes. After suggesting an operational definition of signature, we develop a method that fulfills these requirements, returning sets of tightly coexpressed genes with good predictive performance. This method can also identify when the data are inconsistent with the hypothesis of a few, stable, easily interpretable sets of coexpressed genes. When applied to simulated data, this method recovers fairly well the existing signatures; nevertheless, identification of molecular signatures in some widely used real data sets is questionable under this simple model, which emphasizes the needed for further work on the operationalization of the biological model and the assessment of the stability of putative signatures. An R package that implements the procedure is available from <http://ligarto.org/rdiaz/Software/Software.html>. Supplementary information available in <http://ligarto.org/rdiaz/Papers/signatures-supl.mat.pdf>.

Index Terms

Multivariate statistics, Statistical computing, Statistical software, Biology and genetics

I. INTRODUCTION

“**M**OLECULAR SIGNATURES” or “gene-expression signatures” are a key feature in many studies that use microarray data in cancer research [1]–[5]. In p. 375 [6] refer to signatures as “(...) genes that are **coordinately expressed** in samples related by some identifiable criterion such as cell type, differentiation state, or signaling response” (emphasis is ours). Molecular signatures are often used to model patients’ clinically relevant information (e.g., prognosis, survival time, etc) as a function of the gene expression data, but instead of using individual genes as predictors, the predictors are the signature components or “metagenes”.

If we are successful searching for a signature, then we will be able to model, for instance, the probability of developing a metastasis as a function of a few signature components or metagenes where each signature component is made of genes that show strong coexpression. Thus, molecular or gene expression signatures can be important both for diagnostic purposes and for providing information about the biological mechanisms underlying certain conditions by highlighting genes that both coexpress and are related to that condition.

In spite of the widespread use of the term “molecular signature”, no explicit definition is available. Following the conventions of the literature [1]–[5], [7], and building upon the definition above [6], we will consider a signature to be composed of one or more **signature components** or **metagenes**, where each signature component is a weighted combination of one or more coexpressed genes, and such that statistical models that use signatures both have good predictive performance and are easy to interpret biologically. Interpretation is made easier because the prediction is based on signature

Manuscript received ...; revised ...

R. Díaz-Uriarte is at the Spanish Cancer Research Center (CNIO), Madrid, Spain. His email is rdiaz@ligarto.org

- 1) Genes of a signature component should show tight coexpression. We can make this more explicit by requiring that each gene of a signature component should show a strong correlation with the signature component.
- 2) For a given classification/prediction problem only a few signature components should be needed to obtain reasonable predictive performance.
- 3) Signature components could have many genes; additionally, it often seems more desirable to include a gene in a signature component even if it does not belong to that signature, than to exclude a gene that does belong to that signature.
- 4) The same genes are used for a signature component over all samples (i.e., the signature components are the same for all groups).

Fig. 1. Requirements of signatures and signature components (see text for details).

components that are weighted averages of **subsets of tightly coexpressed genes**, which can help when attempting to relate specific biological features to, for example, particular alterations on a metabolic pathway. Based upon the above references, we can try to formalize these goals by requiring that signatures and signature components satisfy the conditions shown in Figure 1.

The conditions in Figure 1 reflect a very specific biological model. Our objective is to develop a statistical method appropriate for this biological model. By using a method that tries to fulfill those conditions we can also provide evidence that, for any particular case, the underlying biological assumptions behind this attempt are inconsistent with the data or, in other words, the assumptions embodied in Figure 1 are inappropriate. As will be discussed later, our method is an attempt to map a particular biological model into a statistical method, but other statistical approaches would be more appropriate if other biological models are regarded as appropriate. As well, we recognize that the search for molecular signatures is often pursued to provide biological insights into coexpressed genes related to conditions, and thus minimization of prediction error should not be the only goal: if there are potential trade-offs between prediction error and “biologically interpretable signatures”, it is advantageous if the researcher has the option of modifying the terms of this trade-off flexibly.

A. Limitations of alternative methods

A variety of approaches have been used to identify molecular signatures. A review is provided in the supplementary material. Briefly, most methods return either a single signature component ([2], [8], [9]) which is a weighted average of a set of genes, or several signature components ([7], [10]–[13]) which are often obtained using dimension reduction techniques (e.g., principal component analysis [PCA], partial least squares [PLS], sufficient dimension reduction) either on the complete set of genes or on a preselected subset.

The most common problems of available methods are:

- Genes within signature components do not necessarily show tight coexpression: no method makes tight coexpression a requirement to be fulfilled.
- The interpretation of components is very difficult for most methods that use PCA or PLS, since all the genes to which PCA or PLS is applied have loadings on each component.
- The search for components in many PCA or clustering of genes methods is carried out without incorporating information from the dependent variable.

- 1) Find the seed gene for a signature component:
 - a) Seed gene is gene with smallest cross-validated (CV) prediction error among available genes. (The CV prediction error is obtained using as predictive model the chosen predictor [e.g., DLDA], including all previous signatures, if any).
 - b) If CV prediction error $<$ (CV prediction error of the previous signature - c_1 standard error), continue; otherwise, terminate signature finding.
- 2) If signature component = 1, eliminate all genes with (resubstitution) prediction error $>$ prediction error from always betting on the most frequent class.
- 3) Build an initial signature with all the genes j where $abs(cor(gene_j, seed.gene)) \geq r_{seed}$.
- 4) Obtain the signature component as the 1st PC of a PCA on the initial signature.
- 5) Reduce signature component:
 - a) Eliminating, one by one, from the signature the gene with the smallest absolute correlation with the seed gene, until $abs(correlation(\mathbf{x}_{pr_{i,j}}, \mathbf{pr}_i)) > r_{min}$ is met.
 - b) Eliminate, one by one, any gene for which its exclusion from the signature component leads to a CV prediction error $<$ last prediction error - c_2 s.e.(prediction error).
- 6) Exclude from further consideration all genes that belong to the signature component just built.
- 7) Return to 1. until no further components are needed.

Fig. 2. Basic steps of the signature algorithm.

- Most methods are designed for a specific type of task (e.g., classification or survival, but not both) and would be difficult to extend to other types of dependent variables.

Our objective in this paper is to propose a method that overcomes these problems. Our method returns signature components of tight coexpression (and thus, signature components that should ease interpretation) with good predictive performance. In addition, the potential trade-offs between prediction error and “biologically interpretable signatures” can be flexibly modified by the user: first, through the r_{min} parameter the user controls how tight the coexpression of each signature component should be (all the genes in a component will be required to show a correlation larger than r_{min} with the signature component); second, the user controls the penalties of adding new signature components and of eliminating genes from signature components relative to changes in prediction error rates.

II. METHODS

A. Algorithm

1) *Key elements of the proposed method:* Our objective is to directly fulfill the conditions in Figure 1. We start our search with a seed gene that will be the core of the first signature component; this first signature component is found so that genes within the component show tight coexpression and the prediction error is acceptable. We repeat this process (find seed gene for a component and then obtain the whole component) greedily, until no further components are needed. The main steps of the algorithm are shown in Figure 2. In this section we explain how the conditions in Figure 1 can be fulfilled and provide a geometrical interpretation of the algorithm.

2) *Fulfilling signature requirements:* A common and simple way of characterizing a signature component is to use linear combinations (weighted averages) of the genes that belong to that signature component ([3], [4], [7], [14]). Although we could characterize a signature component using several

different linear combinations of the genes of that component, most methods (but see [15]) characterize a signature component using only one linear combination or “metagene”. A single metagene per signature component simplifies interpretation, and is implicit in the requirement that each gene of a signature components should show a strong correlation with the signature component.

Thus, to fulfill requirement one in Figure 1, we can use Principal Component Analysis (PCA — which is closely related to Singular Value Decomposition [SVD]). PCA yields “the best” representation (or “least distorting” representation, in the least squares sense) of the original data [16]–[18] in a subspace of reduced dimensions. The first PC is the best 1-dimensional representation of the original genes of this signature component. If the genes of the signature component are tightly coexpressed, then each of these genes should show a high correlation with the signature component, as we required above (this will also mean that the percentage of variance in the original gene expression data explained by this first PC will be high). After running the procedure, each signature component will be made of tightly coexpressed genes (we require that all the genes in a component show a correlation larger than a pre-specified threshold of r_{min}).

In contrast to some previous methods which use PCA or PLS over the complete set of genes, there is no need for our method to return components which are uncorrelated or orthogonal: there is no biological argument that requires that signature components be orthogonal, uncorrelated, or independent (see discussion). For ease of interpretation we will additionally require that no gene belongs to more than one signature component. (In other words, each gene in the original data matrix belongs to either one, and only one, signature component or to none.)

The second and third requirements of Figure 1 can be incorporated by adding new signature components only if they result in a relevant reduction of prediction error, and retaining genes in a signature unless they produce large increases in prediction error. In other words, we will penalize adding signature components, but will try to obtain large signature components, allowing the user to modify the relative penalties.

3) *Searching for signature components and a geometrical interpretation:* Our objective is, thus, to maximize predictive performance using signature components that satisfy that the correlation of each gene in a signature component with the signature component is larger than a given threshold. However, the discussion so far does not indicate how to find the signature components and, given the dimensionality of the problem, an exhaustive search for the optimal solution is not possible. Since we require that each component be highly correlated with the genes of that component, we can start the search with genes that have good predictive abilities on their own. Once we find an initial “**seed gene**”, we build an initial candidate signature component by including all “promising genes” (e.g., all those with a minimum correlation with the seed gene), and later reduce the signature component eliminating genes until the conditions of minimum correlation with 1st PC (all genes have a correlation with the 1st PC $> r_{min}$) and predictive performance are met. (If this elimination eliminates all genes except the seed gene, then, of course, the two requirements are met).

The method proposed here is a heuristic search that has an intuitive geometrical interpretation. We require that each component be highly correlated ($> r_{min}$) with the genes in the component, which is equivalent to saying that the vector of the component must have a similar direction as the vectors of each gene in variable space (the space where subjects are the axes). Therefore, no matter which genes belong to a signature component, the component will have a similar direction as any of its genes. Then, it seems reasonable to start the search with the direction that has the best predictive ability, the seed gene; this seed gene is the direction in space that most contributes to separation of the groups in a classification problem; analogous for regression or survival analysis. When we form the complete signature component, all other genes of the signature component have directions

that are similar to that of the seed gene. Together, all the genes of a signature component move the direction slightly, but this shift is possibly towards directions that contribute more to separation of groups (or that at least do not degrade the separation too much) and never moves us far away from the original seed gene. This process is repeated until the addition of new signature components does not achieve any relevant decrease in prediction rate, or until a maximum pre-specified number of signature components is reached. The algorithm is shown in Figure 2. Further details are given below.

4) *Choice of underlying classifier:* In this paper we will be dealing with a classification problem. Each signature component is used as a predictor variable for a classifier. Of the available classification methods, we have used DLDA (Diagonal Linear Discriminant Analysis), a version of linear discriminant analysis which assumes the same diagonal variance-covariance matrix for all the classes [19], and NN (K-Nearest Neighbor, with $k = 1$), a simple non-parametric rule that assigns a test sample to the class of the closest training sample (where closeness is measured using Euclidean distance in the space whose dimensions are the signature components). KNN and DLDA have been repeatedly shown to perform as well as, or better than, many competing methods with microarray data [19], [20]. In addition, DLDA and KNN are simple to implement and interpret. [19] used an adaptive procedure to estimate the optimal number of neighbors to use with KNN; that can be time consuming, and we have fixed $K = 1$, since this is often a successful rule [21], [22]. As discussed in the supplementary material, other classifiers can be used.

5) *Commented algorithm:*

0. **Gene filtering.** These gene filtering procedures have little effect on the outcome of the signature algorithm but can contribute to speed up the process. We eliminate all genes that have constant values across all samples. This filtering step is independent of class values, but obviously genes with constant value cannot contribute to discrimination among classes. Next, we exclude those genes with an F-ratio of between classes to within classes sums of squares smaller than 2. Elimination of genes which are extremely unlikely to contribute anything to a possible classifier is common in the literature (e.g., [19]). Our criterion is not very strict and, in our experience, excluding these genes has rarely any effect on the final signatures, but contributes to speed up the process (since step 1 is time consuming).

1) **Find the seed gene for a signature component.**

- a) The seed gene is the gene with smallest prediction error among available genes. (The prediction error is obtained using as predictive model the chosen predictor [e.g., DLDA], including all previous signatures, if any).
- b) If the prediction error $<$ (prediction error of the previous signature - c_1 standard error), continue; otherwise, terminate signature finding.

Notes:

- When we start the search for signatures, there are no previous components. Additionally, to allow the process to start, we arbitrarily set “prediction error with previous signature” to 1. Therefore, we always obtain at least one signature (if there are any genes left after the preprocessing steps).
- We use the prediction error only as a guide to select genes, not as an estimate of prediction error.
- The “prediction error” actually used is obtained as follows. First, we compute the resubstitution error rate if DLDA, or the leave-one-out error rate if NN (the resubstitution error rate of NN is always 0). Then, we select all those genes with a resubstitution error rate of 0, or the 10 genes with the smallest error rate, and compute their 10-fold CV error rate.

The reason for this two-step process is to use the more computationally intensive 10-fold CV on only genes that are likely to be the best performers, but do an initial ranking using resubstitution, which is appropriate when we are only interested in ranking genes [23].

- Varying c_1 we control how much reduction in prediction error we require before considering the addition of a new signature; $c_1 = 1$ yields the usual “1 standard error rule”, common with classification trees [21], [24], and resembles forward variable selection methods.
 - The CV prediction error of the gene with the smallest error is not a good estimate of prediction error, and is biased down [25] since the same data that are used for the selection of genes in the prefiltering step are used in this step and we are ordering genes based on CV error. The objective of this step is not to estimate prediction error, but to allow ranking of genes to select the most promising candidate.
- 2) Since the previous step provides optimistic estimates of prediction error, we can filter the data excluding from further consideration all genes that have a (CV) prediction error (minus 1 s.e.) larger than the smallest prediction error that can be achieved by chance (always betting on the most frequent class). As in the prefiltering step, this is not a very strict filter, which rarely affects final signature composition but that can speed up the process. Because the main objective is to minimize time spent in step 1, this filtering is only applied right after the search for the first signature component, when the largest number of “unimportant” genes is likely to be removed.
 - 3) We build an **initial signature component** using all the genes j where $abs(cor(gene_j, seed.gene)) \geq r_{seed}$.
 - To minimize the impact of outlying and extreme values, we require: a) that for each gene the above criterion be satisfied using both Pearson’s product-moment correlation and Spearman’s rank correlation; b) that the sign of the correlation coefficient be the same for each gene for Pearson’s and Spearman’s rank correlations.
 - The value of r_{seed} has very little effect on the procedure, whenever $r_{seed} < r_{min}$ (see details on r_{min} in 5a). In fact, it is easily seen that, if we use no restriction related to the Spearman rank correlation (i.e., we only use Pearson’s correlation as criterion), then the resulting signature component does not depend on r_{seed} whenever $r_{seed} < r_{min}$. For the range of values of r_{min} often considered, $r_{seed} = 0.4$ is small enough.
 - As the criterion for inclusion is the absolute value of the correlation, we are not restricted to finding only genes with positive correlation with the seed gene.
 - As mentioned in the discussion, the algorithm can include in a signature component genes that show no correlation within groups or genes that have very different patterns of correlation in different groups; see discussion.
 - 4) We carry out a PCA on the genes of the initial signature component. This is done using a SVD on the mean-centered data, which is analogous to doing an eigenvalue-eigenvector decomposition of the covariance (not correlation) matrix. We do not scale all genes to a common variance (equivalent to doing the PCA or eigenvalue decomposition on the correlation matrix) because we assume that all genes are in comparable units, and thus the relative differences in variances among genes are important and should be taken into account in the PCA (see discussions about prescaling data in PCA in [16, pp. 21 and ff.], [18, pp. 314 and ff.], [17, pp. 64 and ff.]); in the present case, scaling all genes to a common variance would probably result in magnifying the effects of noise.
 - 5) This initial signature component might not fulfill the conditions above (predictive performance and $abs(correlation(\mathbf{x}_{pr_{i,j}}, \mathbf{pr}_i)) \geq r_{min}$). We consider whether a gene needs to be removed from

the signature component by:

- a) Eliminating, one by one, from the signature component the gene with the smallest absolute correlation with the seed gene, until $abs(correlation(\mathbf{x}_{pr_{i,j}}, \mathbf{pr}_i)) > r_{min}$ is met for all genes in the signature component.
- b) Ensuring that the predictive accuracy of the model so far cannot be appreciably improved by removing any gene from the signature component. For each gene, i , in the current signature component, we fit the predictive model (using either DLDA or NN) that includes the previous signature components (if any) and the current component without this gene (this also involves obtaining a new principal component). Using cross-validation, we determine the predictive error of the model without each gene i ($prediction\ error_{-i}$), and compare it with the prediction error of the model without excluding any genes from this signature component. We eliminate the gene with the largest prediction error if $prediction\ error_{-i} < last\ prediction\ error - c_2\ s.e.(prediction\ error)$. We repeat the process until no further gene is eliminated, and at each iteration the last prediction error is updated to be the one achieved after eliminating a gene. (This process is somewhat analogous to backwards variable elimination methods).

Notes:

- As in 1, the CV prediction error is too optimistic, and we only use it as a way of excluding genes that seriously degrade predictive performance (not to assess prediction error of the method).
 - Varying c_2 we can control how much we penalize each gene for their effects on decreasing prediction error. We often use $c_2 = 1$.
 - The genes that are eliminated in 5a need not be the ones with smallest correlation with the 1st PC. The method used prevents that a large set of genes weakly correlated with the seed gene, but strongly correlated among themselves, might drive the 1st PC away from the direction of the seed gene.
 - The condition $abs(correlation(\mathbf{x}_{pr_{i,j}}, \mathbf{pr}_i)) > r_{min}$ ensures that the % of variance accounted for by the 1st PC is, at least, $\geq r_{min}^2$ and often much larger. (This can be shown easily using the standard facts about PCA, e.g., [18, p. 315 to 317], that the sum of the eigenvalues from the PCA = trace of variance-covariance matrix, that $\mathbf{a}_1' \mathbf{a}_1 = 1$, [where \mathbf{a}_1 is the vector of coefficients of the first PC, or the first eigenvector from the PCA], and that the correlation of any variable with the 1st PC is $a_{i1} \sqrt{l_1} / s_i$ [where l_1 is the first eigenvalue and s_i is the standard deviation of the variable i]).
- 6) Exclude from further consideration all the genes that are part of the signature component just built.
 - 7) Return to 1 until no further components are needed.

B. Simulation study of recovery of existing signatures

1) *Parameters of the simulations:* Experimental data are unsuitable to evaluate if the proposed procedure can recover signatures in the data, since we do not know whether or which signatures are present in the data. Thus, we have used simulations to evaluate if we can recover signatures when they are present in the data. We have simulated data under different numbers of “true” signature components (1, 2, 3), different numbers of distinct classes of patients (2, 3, 4) and different number of genes that belong to each signature component (5, 20, 100). In all cases, the number of subjects per class has been set to 25 (a number which is similar to, or smaller than, that of many microarray

studies). The data have been simulated from a multivariate normal distribution. All “genes” have a variance of 1, and the correlation between genes within a signature component is 0.9, whereas the correlation between genes among signature components is 0. In other words, the variance-covariance matrix is a block-diagonal matrix as:

$$\Sigma = \begin{bmatrix} \mathbf{a} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{a} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{a} \end{bmatrix},$$

where

$$\mathbf{a} = \begin{bmatrix} 1 & 0.9 & \dots & 0.9 \\ 0.9 & 1 & \dots & 0.9 \\ \vdots & \vdots & \vdots & \vdots \\ 0.9 & 0.9 & \dots & 1 \end{bmatrix}.$$

The class means have been set so that the unconditional prediction error rate (see [26]) of a DLDA with a gene from each signature component is approximately 5%; and each signature component has the same relevance in separation. Specifically, the class means used are:

- One signature component:
 - Two classes: $\mu_1 = -1.65, \mu_2 = 1.65$.
 - Three classes: $\mu_1 = -3.58, \mu_2 = 0, \mu_3 = 3.58$.
 - Four classes: $\mu_1 = -3.7, \mu_2 = 0, \mu_3 = 3.7, \mu_4 = 7.4$.
- Two signature components:
 - Two classes: $\mu_1 = [-1.18, -1.18], \mu_2 = [1.18, 1.18]$.
 - Three classes: $\mu_1 = [0, 0], \mu_2 = [3.88 \cos(15), 3.88 \sin(15)],$
 $\mu_3 = [3.88 \cos(75), 3.88 \sin(75)]$.
 - Four classes: $\mu_1 = [1, 1], \mu_2 = [4.95, 1], \mu_3 = [1, 4.95], \mu_4 = [4.95, 4.95]$.
- Three signature components:
 - Two classes: $\mu_1 = [-0.98, -0.98, -0.98], \mu_2 = [0.98, 0.98, 0.98]$.
 - Three classes: $\mu_1 = [2.76, 0, 0], \mu_2 = [0, 2.76, 0], \mu_3 = [0, 0, 2.76]$.
 - Four classes: $\mu_1 = [2.96, 0, 0], \mu_2 = [0, 2.96, 0],$
 $\mu_3 = [0, 0, 2.96], \mu_4 = [2.96, 2.96, 2.96]$

After the genes that belong to the signature components are generated, we add another 4000 $\mathcal{N}(0, 1)$ variables to the matrix of “genes”.

For each combination of number of signature components * number of classes * number of genes per signature component we generate 40 data sets and run the the DLDA-based procedure on 20 of the samples and the NN-based procedure on the other 20.

2) *Statistics to evaluate performance in simulation runs:* The summary statistics used to evaluate performance are:

- **Gene overlap: overall** The overall gene overlap is defined as

$$|recovered.genes \cap true.genes| / \sqrt{|recovered.genes| |true.genes|},$$

where *recovered.genes* are the genes in the signature components, and *true.genes* are the genes that are true members of the signature components. The overall gene overlap thus refers to the overlap over all genes, regardless of signature component.

- **Gene overlap: mean of signature components** Denominated “Gene overlap: comp. mean” in the figures. We calculate gene overlap for each estimated signature component and return the mean over all the signature components. (Each estimated signature is associated with the true signature with which the overlap is largest.) The gene overlap for each signature component, therefore, measures if the genes are assigned to the correct component.
- **Number of components** The number of signature components returned by the algorithm.

C. Comparing predictive performance with established methods

Here we compare the predictive performance of our method with that of three well established methods, support vector machines, KNN, and DLDA, using several “real data” sets.

Predictive performance is evaluated using 10-fold cross-validation (i.e., the complete algorithm shown in Figure 2 is applied to each of the 10 “training sets”). This 10-fold cross-validation was repeated 20 times under each condition. The error rates shown are not the CV error rates obtained in steps 1 and 5 of the signature algorithm (see Figure 2), since those are biased down; the error rates shown are the error rates obtained from cross-validating the complete procedure.

Because an important parameter of our method might be r_{min} , the minimal absolute correlation between each gene in a signature component and the signature component, we have evaluated the performance of the signature method using a set of values of r_{min} that covers a “biologically interesting” range: $\{.60, .65, .70, .75, .80, .85, .90, .95\}$. In addition, we have also examined the differences between using $c_1 = c_2 = 1$ compared to $c_1 = c_2 = 0$; the first corresponds to the usual “1 se rule” and should lead to more interpretable results (c_1 and c_2 are related to how much we penalize adding a new signature component and how much we penalize eliminating genes from signature components).

1) The data sets:

- **Leukemia dataset** From [2]. The original data, from an Affymetrix chip, comprises 6817 genes, but after filtering as done by the authors we are left with 3051 genes. Filtering and preprocessing is described in the original paper and in [19]. We used the training data set of 38 cases (27 ALL and 11 AML) in the original paper (the observations in the “test set” are from a different lab and were collected at different times). This data set is available from [<http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi>] and also from the Bioconductor package “multtest” ([<http://www.bioconductor.org>]).
- **Adenocarcinoma dataset** From [14]. We used the data from the 12 metastatic tumors and 64 primary tumors. The original data set included 16063 genes from Affymetrix chips. The data (DatasetA_Tum_vsMet.res), downloaded from [<http://www-genome.wi.mit.edu/cgi-bin/cancer/>], had already been rescaled by the authors. We took the subset of 9376 genes according to the UniGene mapping, thresholded the data, and filtered by variation as explained by the authors. The final data set contains 9868 clones (several genes were represented by more than one clone); of these, 196 had constant values over all individuals.
- **NCI 60 dataset** From [27]. The data, from cDNA arrays, can be obtained from [<http://genome-www.stanford.edu/sutech/download/nci60/index.html>]. The raw data we used, which is the same as the data used in [19], [28], is the one in the file “figure3.cdt”. As in [19], [28] we filtered out genes with more than two missing observations and we also eliminated, because of small sample size, the two prostate cell line observations and the unknown observation. After filtering, we were left with a 61 x 5244 matrix, corresponding to eight different tumor types (note that, as done by previous authors, we did not average the two observations with triplicate hybridizations). As in [19] we used 5-nearest neighbor imputation of missing data using

the program GEPAS [29] (<http://gepas.bioinfo.cnio.es/cgi-bin/preprocess>); unlike [19], however, we measured gene similarity using Euclidean distance from the genes with complete data, instead of correlation: [30] found Euclidean distance to be an appropriate metric. Finally, as in [19, p. 82] gene expression data were standardized so that arrays had mean 0 and variance 1 across variables (genes).

- **Breast cancer dataset** From [9]. The data, from Affymetrix arrays, were downloaded from [<http://www.rii.com/publications/2002/vantveer.htm>] (we used the files `ArrayData_less_than_5yr`, `ArrayData_greater_than_5yr.zip`, `ArrayData_BRCA1.zip`, corresponding to 34 patients that developed distant metastases within 5 years, 44 that remained disease-free for over 5 years, and 18 with BRCA1 germline mutations and 2 with BRCA2 mutations). As did by the authors, we selected only the genes that were “significantly regulated” (see their definition in the paper and supplementary material), which resulted in a total of 4869 clones. Because of the small sample size, we excluded the 2 patients with the BRCA2 mutation. We used 5-nearest neighbor imputation for the missing data, as for the NCI 60 data set. Finally, we excluded from the analyses the 10th subject from the set that developed metastases in less than 5 years (sample 54, IRI000045837, in the original data files), because it had 10896 missing values out of the original 24481 clones, and was an outstanding outlying point both before and after imputation. The breast cancer dataset was used both for two class comparison (those that developed metastases within 5 years vs. those that remain metastases free after 5 years) and for three group comparisons.

Therefore, we use three datasets in which the problem is classification into two classes (leukemia, adenocarcinoma, breast cancer), one dataset with a three class problem (breast cancer) and one dataset with an eight class problem.

2) *The competing methods:* We have used three methods that have shown good performance in reviews of classification methods with microarray data [19], [20].

- **Diagonal Linear Discriminant Analysis (DLDA)** DLDA is the maximum likelihood discriminant rule, for multivariate normal class densities, when the class densities have the same diagonal variance-covariance matrix (i.e., variables are uncorrelated, and for each variable, its variance is the same in all classes). This yields a simple linear rule, where a sample is assigned to the class k which minimizes $\sum_{j=1}^p (x_j - \bar{x}_{kj})^2 / \hat{\sigma}_j^2$, where p is the number of variables, x_j is the value on variable (gene) j of the test sample, \bar{x}_{kj} is the sample mean of class k and variable (gene) j , and $\hat{\sigma}_j^2$ is the (pooled) estimate of the variance of gene j [19]. In spite of its simplicity and its somewhat unrealistic assumptions (independent multivariate normal class densities), this method has been found to work very well.
- **K nearest neighbor (KNN)** KNN is a non-parametric classification method that predicts the sample of a test case as the majority vote among the k nearest neighbors of the test case [21], [22]. To decide on “nearest” here we use, as in [19], the Euclidean distance. The number of neighbors used (k) is chosen by cross-validation as in [19]: for a given training set, the performance of the KNN for values of k in $\{1, 3, 5, \dots, 21\}$ is determined by cross-validation, and the k that produces the smallest error is used.
- **Support Vector Machines (SVM)** SVM are becoming increasingly popular classifiers in many areas, including microarrays [10], [31], [32]. SVM (with linear kernel, as used here) try to find an optimal separating hyperplane between the classes. When the classes are linearly separable, the hyperplane is located so that it has maximal margin (i.e., so that there is maximal distance between the hyperplane and the nearest point of any of the classes) which should lead to better performance on data not yet seen by the SVM. When the data are not separable, there is no separating hyperplane; in this case, we still try to maximize the margin but allow some

classification errors subject to the constraint that the total error (distance from the hyperplane in the “wrong side”) is less than a constant. For problems involving more than two classes there are several possible approaches; the one used here is the “one-against-one” approach, as implemented in “libsvm” [33]. Reviews and introductions to SVM can be found in [22], [34].

For each of these three methods we need to decide which of the genes will be used to build the predictor. Based on the results of [19] we have used the 200 genes with the largest F -ratio of between to within groups sums of squares. [19] found that, for the methods they considered, 200 genes as predictors tended to perform as well as, or better than, smaller numbers (30, 40, 50 depending on data set).

We evaluated predictive performance using 10-fold cross-validation; the results shown are from 20 replications of the 10-fold cv process. In all cases, cross-validation includes gene selection [25], [35]; in other words, for the three competing methods and the signature algorithm the selection of genes is carried out within each of the 10 “training sets” of the cross-validation. Thus, we insure that the subjects for which prediction is performed have not been used for the gene selection process.

III. RESULTS

A. Can we recover signatures when they are present?

We have used simulations to evaluate if we can recover signatures when they are present in the data, where the simulated data include both a “signal”, and noise random variables not related to the outcome. For the signal part, data have been simulated using different numbers of true signature components (1 to 3), classes of patients (2 to 4), and number of genes per signature component (5, 20, 100). In addition to this “signal”, we have added 4000 normal random variates not related to the dependent variable. The parameters used for the signature algorithm were $r_{min} = 0.8$ (the minimal absolute correlation between each gene in a signature component and the signature component) and $c_1 = c_2 = 1$ (c_1 and c_2 are related to how much we penalize adding a new signature component and how much we penalize eliminating genes from signature components; $c_1 = 1$ and $c_2 = 1$ correspond to the usual “1 standard error” rule). We run the simulations 20 times under each condition. To represent the results of the simulations, we show three summary statistics, **Gene overlap: overall**, **Gene overlap: mean of signature components**, and **Number of components**—see Methods for complete definition of each statistics—, averaged over the 20 replicate data simulations.

The results using Nearest Neighbor (NN) as the underlying classifier are shown in Figure 3; the results using the Diagonal Linear Discriminant Analysis (DLDA) classifier are shown in the supplementary material and are similar, but show better performance (the data are simulated under a model that is closer to the one assumed by DLDA). The **overall gene overlap** is often very close to 1, indicating that most of the “true genes” have been recovered. The **mean overlap** per signature is lower, because there are cases of over- or under-estimation of the **number of signatures** which preclude overlaps close to 1. In many cases such as, for example, I.5, the first signature component captures exactly all the true genes, but occasionally there are additional components estimated, which results in overlap of mean signature components of 0.5 ($= (1 + 0)/2$) or 0.33 ($= (1 + 0 + 0)/3$). This explains that, in many cases, the mean overlap over components falls in a few discrete classes. A similar phenomenon explains results in III.100, with three classes, where the number of components is underestimated; for the components found, the overlap per component is 1, or very close to 1, but since a component is missing, the overall overlap is 0.8 ($= 200/\sqrt{200 * 300}$).

With three and four classes, and with both two and three signature components, the procedure never returns only one signature component because it would be impossible, with this simulated

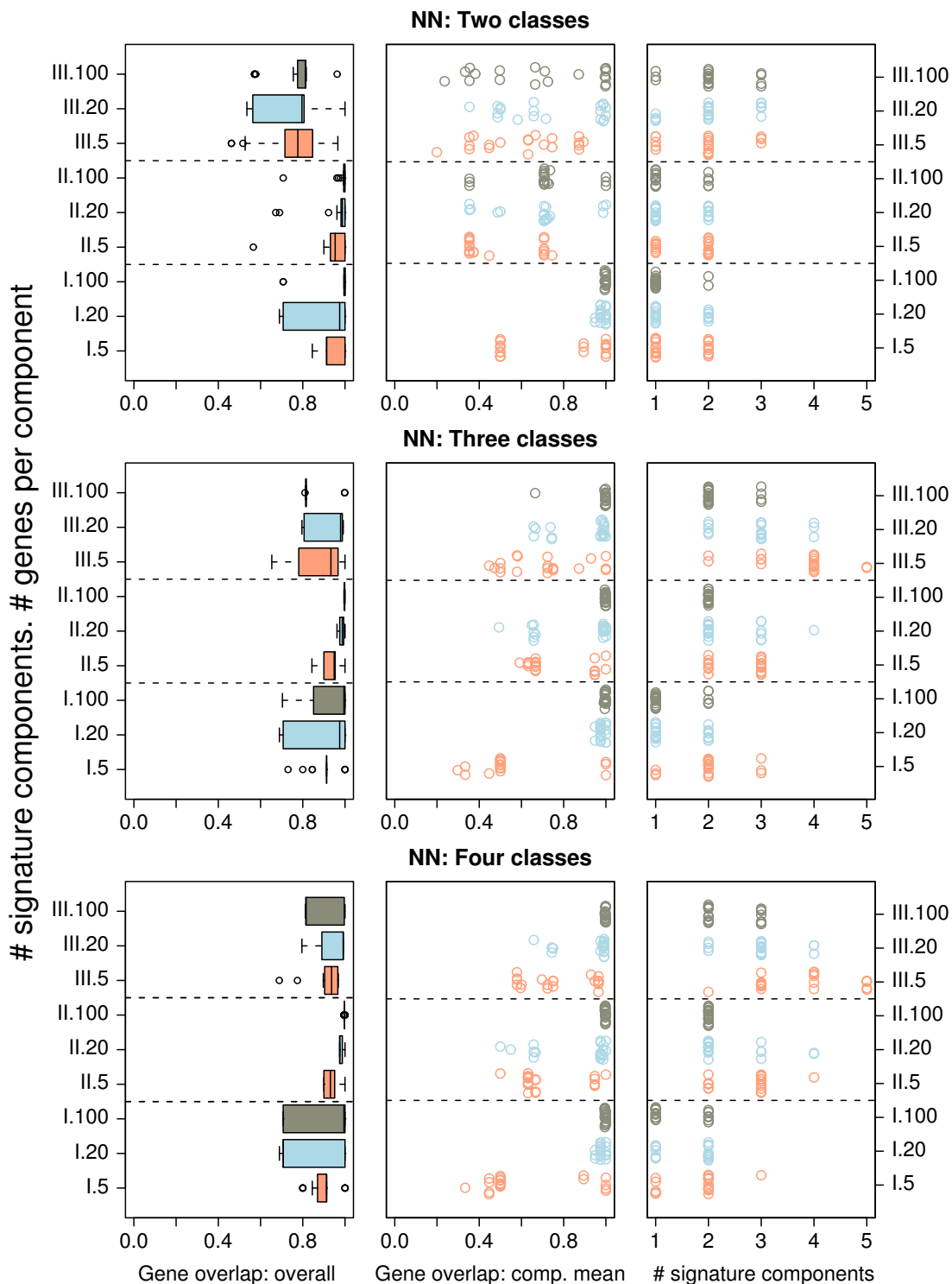


Fig. 3. Signature recovery when using the signature method with the NN predictor. Each line represents a combination of number of signatures and number of genes per signature. For instance, II.5 denotes II signature components with 5 genes per signature component. Based on 20 replicate simulations. To facilitate distinguishing data, points have been jittered vertically in the center and right panels. See text for explanation of variables.

data, to achieve reasonable separation of the classes with only one component. When there is one true signature component, the modal value of the estimated number of components tends to be one. When there are two true signature components, the modal value is generally two with three or four classes. But when there are two classes the modal value is one. When we examine the latter cases, we see that most of the genes that belong to the true signature are recovered (note that the overall gene overlap is close to 1) but they are all recovered in a single signature component. The reason is that, even if the genes of different signature components are uncorrelated within groups, when the correlation is computed over the whole sample there is a correlation between all the genes as the two groups are pulled apart in 2D space. Finally, the performance of the DLDA-based method (see Figure in supplementary material) is often better than that of the NN-based one, which is to be expected since the simulated data were generated under a model that meets the assumptions of DLDA. In summary, then, the proposed procedure is able to recover existing signature components.

B. Predictive performance: comparison with other classification methods on real microarray data sets

Predictive performance, and comparison with three well established classification methods, using several real microarray data sets, is shown for one set of parameters in Figure 4; figures for other combinations of classifier and values of c_1, c_2 (parameters that control the penalties of adding new signature components and eliminating genes from signature components) are shown in the supple-

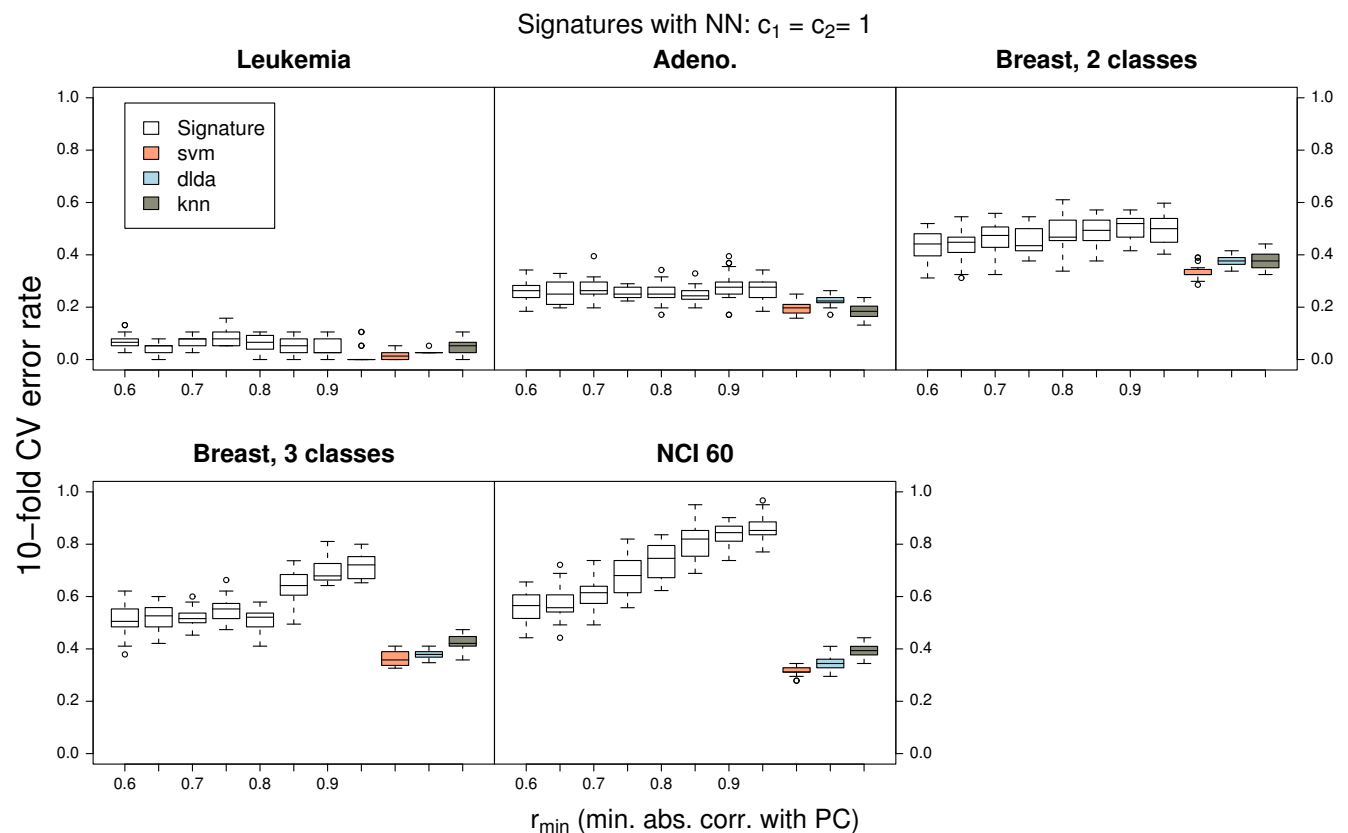


Fig. 4. Predictive performance, as a function of r_{\min} , of the signature method using NN as classifier and comparison with SVM, KNN, and DLDA. Figures based on 20 replicates of the 10-fold-CV procedure. Results for $c_1 = c_2 = 1$.

mentary material and show the same patterns. Predictive performance changes very little from using discriminant analysis vs. nearest neighbor (DLDA vs. NN) as the underlying classifier. Comparing $c_1 = c_2 = 1$ with $c_1 = c_2 = 0$ (see Figures in supplementary material) does not show any relevant differences in predictive performance; of course, there are differences in the outcome because, not surprisingly, using $c_1 = c_2 = 0$ tends to result in more signature components of smaller numbers of genes per component, and higher correlations between components.

Changes in r_{min} , the parameter that sets the minimum correlation required between a gene in a signature component and that signature component, have little effects on predictive performance, except for the NCI data set, and slightly for the Breast cancer with 3 classes data set. In Table I we show the median number of components, median total number of genes in a signature, and median average number of genes per component obtained in 200 bootstrap runs using $c_1 = c_2 = 1$ with $r_{min} = 0.85$ and $r_{min} = 0.6$, and using NN as the classifier (see also section III-C). It can be seen that changes in r_{min} do affect the outcome in terms of number of genes per component. These results would indicate that choice of r_{min} can probably be guided more by interpretability concerns (whether we want larger signature components of looser coexpression or smaller signature components of tight coexpression) than by concerns over predictive ability, therefore providing the user with added flexibility.

TABLE I
MEDIAN VALUES FROM 200 BOOTSTRAP RUNS FOR TOTAL NUMBER OF GENES IN SIGNATURES, NUMBER OF SIGNATURE COMPONENTS AND AVERAGE NUMBER OF GENES PER COMPONENT.

| Data set | $r_{min} = 0.85$ | | | $r_{min} = 0.6$ | | |
|---------------------------|------------------|-------------|------------------|-----------------|-------------|------------------|
| | Total genes | #Components | Mean Genes/Comp. | Total genes | #Components | Mean Genes/Comp. |
| Leukemia | 6 | 1 | 5 | 52.5 | 1 | 50 |
| Breast cancer (2 classes) | 2 | 2 | 1 | 10 | 2 | 3.875 |
| Breast cancer (3 classes) | 6 | 2 | 2.5 | 63.5 | 2 | 33.25 |
| Adenocarcinoma | 5 | 1 | 3 | 45.5 | 1 | 31.5 |
| NCI 60 | 4 | 2 | 1.67 | 16 | 2 | 7.1 |

Finally, the performance of the signature method is only slightly worse than that of the three competing classifiers, except for the NCI data set. As seen in Table I, most of the signatures, specially with $r_{min} = 0.85$, used very few components of very few genes each (compared to the use of 200 genes for the competing classifiers); thus, the predictive performance is achieved using a very small number of genes and thus potentially facilitating a simple biological interpretation. In the case of the NCI data set, there are eight classes with only 61 samples, and in most cases the signature component only returned between 1 and 3 components. Probably the forward sequential addition of components in the signature method has affected negatively the predictive capabilities, because any single addition was most likely incapable of resulting in a large enough decrease of prediction error to justify further addition of components (in contrast to using, directly, 200 genes as predictors).

C. Stability of results

To evaluate the stability of results, we rerun the complete procedure on all data sets using the bootstrap [36], [37] with 200 bootstrap samples, similar to what [38] do to evaluate a complex fitting procedure. We run the procedure for settings of $c_1 = c_2 = 1$ with $r_{min} = 0.85$ and $r_{min} = 0.6$, and using NN as the classifier. The results are shown in Table II. The bootstrap results indicate that when using a high value of r_{min} we rarely obtain similar solutions repeatedly; some data sets, however, seem

to yield more stable solutions (e.g., Leukemia data sets) and, not surprisingly, if the r_{min} criterion is set to less stringent values, results tend to be more repeatable.

TABLE II

STABILITY OF RESULTS USING THE BOOTSTRAP, WITH 200 BOOTSTRAP ITERATIONS. VALUES SHOWN ARE THE NUMBER OF GENES THAT ARE RETURNED, AS MEMBERS OF A SIGNATURE COMPONENT, IN AT LEAST THOSE MANY BOOTSTRAP RUNS.

| Data set | $r_{min} = 0.85$ | | | $r_{min} = 0.6$ | | |
|---------------------------|-------------------------------|----|----|-------------------------------|----|-----|
| | Genes present at least % runs | | | Genes present at least % runs | | |
| | 50 | 20 | 10 | 50 | 20 | 10 |
| Leukemia | 0 | 7 | 15 | 7 | 80 | 179 |
| Breast cancer (2 classes) | 0 | 0 | 9 | 0 | 0 | 43 |
| Breast cancer (3 classes) | 0 | 0 | 3 | 0 | 51 | 246 |
| Adenocarcinoma | 0 | 0 | 3 | 0 | 45 | 270 |
| NCI 60 | 0 | 0 | 0 | 0 | 0 | 6 |

IV. DISCUSSION

A. Similarities and differences with other methods

Our method is unique because it simultaneously searches for sets of genes that are tightly coexpressed and lead to good predictive performance. The search for the sets of genes is carried out using the information from the dependent variable (at the first stage —when selecting the seed gene—, and at the pruning stage of reducing the signature component —when genes that lead to decreased predictive performance are eliminated from the signature component —; see Figure 2).

One important difference between our proposed method and most previous approaches that use PCA is that, by performing PCA only on subsets of genes, our method returns signature components where genes show tight coexpression. Because returned components are not orthogonal, and simple components are an explicit goal, our approach is actually closer to some ideas implemented in SAS’s PROC VARCLUS, which is similar to factor analysis with oblique rotation and can be used to obtain clusters of variables, of relatively simple interpretation, to be further used in model building [39]–[41].

Using PCA on subsets of genes, instead of the complete set of genes, is crucial because it makes interpretation easier and allows for subsets of tight coexpression. Simple PCA and related methods [16], [42], [43] as well as SAS PROC VARCLUS also try to achieve components of tight coexpression, but many of these approaches cannot be applied with $p \gg n$, and all of them carry out the PCA without using the information from the dependent variable. The later is a fundamental requirement in our case since the sets of genes with tightest coexpression would be irrelevant for our purposes if they are not related to the dependent variable we are trying to model. This difference in objectives is also evident because our aim is not to explain the most variance in the genes (as in most simple PCA approaches) nor maximize variable explained across all clusters (such as in PROC VARCLUS). Overall summarization of information is not important for our problem, because we are interested in prediction, and we often have the suspicion that most of the genes in the array are not related to the outcome variable. Finally, our approach can result in signature components which are correlated (sometimes strongly), but this is not inherently a problem because there is no biological reason to suggest that the underlying biological causes or factors ought to be independent or uncorrelated. Moreover, if the true underlying causes are not orthogonal, using a method such as PCA can lead to interpretational and conceptual difficulties because each biological cause will be spread over several

orthogonal components [44]: non-orthogonal biological causes are inconsistent with procedures such as PCA and PLS.

Bayesian classification trees using the 1st PC from gene clusters [45] and block PCA [15] also used PCA on subsets of genes, instead of the complete set of genes. In both cases their subsets of genes were obtained using criteria that did not make any use of the information from the dependent variable. In addition, in [45] the metagenes are not necessarily of subsets of tightly coexpressed genes. In [15] there is an explicit criterion of % variance accounted for, but often the number of components used to summarize a subset of genes is too large to allow for easy interpretation (11 to 16 principal components per signature component).

Supervised harvesting of expression genes [11] also works with clusters of subsets of genes, which are then used in a predictive model. As before, however, the clustering is carried out without using information from the dependent variable; even if the selection of which subsets or clusters to use in the model uses the information from the dependent variable, the very first step of clustering genes does not, and can therefore be unable to recover sets of tightly coexpressed genes that are good predictors. Finally, the “Wilma” and “Pelora” methods [46], [47] do use the information from the dependent variable in the formation of clusters of genes; nevertheless, there is no explicit objective of achieving tight gene coexpression within clusters and thus how tightly coexpressed genes are in each metagene cannot be specified in advance; in addition, Wilma weights each gene equally within a cluster (only possible weights are +1 and -1) whereas we use PCA on unscaled genes, thus allowing genes to play a different role in the specification of the direction of the signature component (genes with larger among-subject variance play a more important role in determining direction).

The rest of the alternative methods differ strongly from our proposed method, either because they do not return subsets of genes but components with loadings from all genes (e.g., PLS based methods), or return subsets of genes where there is not requirement of tight coexpression [2, e.g., weighted gene voting].

After gene selection and dimension reduction (i.e., the use of only the 1st PC, that collapses all the information from the genes of a signature component onto one dimension), the predictive model of our choice is fitted. In this sense, the method as presented here is “just” a DLDA or NN that uses signature components instead of genes as the predictors. The choice of DLDA and NN was made based on published results that showed their excellent performance with microarray data. In particular, [19] showed that other forms of discriminant analysis tended to perform much worse because of the small ratio sample size/parameters needed to estimate covariances and different variances per group; nevertheless, since in most cases our method returns just a few signature components, other types of discriminant analysis that use the information from the covariance of the predictors (e.g., linear discriminant analysis and quadratic discriminant analysis) might prove useful.

B. Coexpression: across-group and within-group

The algorithm can include in a signature component genes that show no correlation within groups but that show correlation among groups because they are far apart in the multidimensional space, and the correlation coefficient is computed across the whole, pooled, sample. The algorithm might even include in the same signature component genes that have very different patterns of correlation in different groups, if they still show sufficiently strong correlation over the pooled sample.

To our knowledge, this issue has not been explicitly addressed in any other approach to the signature problem (see reviews in supplementary material). However, probably the most biologically relevant components are those where there is strong correlation within groups, because this is a more reliable indicator of coordinated expression.

A possible solution is, for example, to only accept results for a signature component if a principal component analysis over the pooled sample after centering the data with respect to the group means yields a relevant first eigenvalue; for added robustness, we might want to use the trimmed mean. This approach, however, does not directly address if there are different multivariate orientations in different groups of subjects, and how these orientations within-group relate to the across-group orientation. In particular, the case where several groups not only have the same first principal component, but are lined up along a common axis, known as “allometric extension” [48], [49], might constitute the most natural type of signature component.

Krzanowski [16], [17, see reviews and summary in] has proposed a method to directly compare the subspaces defined by the principal components of each of the groups. In our case, as we only use the first principal component of a set of genes to define a signature component, for each signature component we can compare the first principal component of each group. An example using the NCI 60 data set is shown in the supplementary material. This method only compares the orientation of the principal components (the eigenvectors) but does not compare the location of the multivariate means. The EDDA [50] and common principal components [51] approaches provide frameworks to examine differences among covariance matrices of particular interest in the context of discrimination among groups. Nevertheless, it must be remembered that biological interpretation of results is not always straightforward, specially when the underlying biological factors are not orthogonal [44], and that we need to assess the need for robust methods [52] and power issues related to the small sample sizes common in microarray data. We are currently investigating some of these issues; nevertheless, in the presence of the difficulties associated with interpretation, low statistical power, and possible lack of robustness to outlying observations, right now the best recommendation could be to conduct an examination of patterns of within vs. among coexpression of genes for the returned signature components.

C. Stability and biological relevance of signatures

Our method follows from an operationalization of signatures and signature components (Figure 1). If the ideas embodied in Figure 1 have empirical support, then it might be possible to build good predictive models using just a few, very specific biological features that can be related to alterations on particular pathway.

On the other hand, our method can indicate that the data are inconsistent with the ideas behind Figure 1: if the signature components show high instability, this is evidence that very different models can be obtained from the data. Widely different models from a reasonably large data set cast doubt on the idea that a few, easily interpretable, signature components strongly correlated with the expression of a few key genes are associated to the clinical outcome of interest. In the context of building predictors from gene expression data, [53] [54] have emphasized that this non-uniqueness leads to interpretational difficulties and should make researchers skeptical about the biological relevance of any set of predictors; moreover, they explain how this non-uniqueness can arise from dataset sparsity. Of course, both of these issues are relevant to the present proposed methodology. None of the data sets examined in this paper yield stable signatures when we use stringent criteria of gene coexpression (see III-C and Table II with $r_{min} = 0.85$), although some of the data sets are somewhat stable from run to run when the r_{min} is set to small values. These results add to the above references in the sense that biological interpretation should be carried out very cautiously, and emphasize the difference between attempting to build good predictors and attempting interpretation [55]. More relevant to the current work, the present results indicate that simple models of molecular signatures warrant

further critical scrutiny, and that it might be extremely hard to identify molecular signatures from such sparse data sets (see [56] for a meta-analysis attempt to identify stable “signatures”).

We must recognize two limitations to these conclusions: a) establishing that two or more signature components are different probably requires additional information besides the identities of the genes; for instance, information from Gene Ontology, or known participation on certain regulatory networks; b) we have used two different r_{min} thresholds, but it is unclear what constitute “biologically reasonable” patterns of covariation between genes that are to belong to the same signature component. Nevertheless, these results emphasize the need for further work in the operationalization and explicit definition of what we mean by molecular signatures, careful consideration of the stability of results, and critical assessment of the sample sizes need to reliably identify molecular signatures.

D. Alternative statistical methods for alternative biological models

As mentioned in the introduction, we started by trying to clarify, conceptually, what is often understood by molecular signature (see Figure 1), and then devised a statistical method to fulfill those requirements. The biological model underlying the suggested method is one where most of the genes are not relevant for prediction, relevant genes are involved in one and only one signature component (i.e., non-overlapping signature components), and the signature components are common, and have similar covariance matrices, in different groups.

However, other biological models are plausible, and for those biological models other statistical methods would be more appropriate. The simultaneous clustering and classification approach in [57] could be extended by placing restrictions on the covariance matrix (i.e., require a minimum correlation between genes) but possibly allowing for different covariance matrices among groups; thus, we could address directly issues of different across vs. within-group correlations (see section IV-B), within a formal inferential framework. Biological models where signature components are not common or do not behave similarly in different groups could be investigated using modifications of the Plaid model of [58] (see also [59]). In addition, genes with the highest correlation need not be the best candidates for being in the same biological pathway; activity in a pathway might just require that precursor genes get activated, but once a threshold is reached, it might not be very important by how much the threshold is exceeded. This type of behavior could preclude strong correlation between genes that belong to the same pathway. This can be modelled building upon the latent class methods of Parmigiani and colleagues [60]–[62], where signature components are based on under-, over- or baseline expression (instead of expression levels). Work along these lines is currently in progress in our group.

E. “Just” dimensional reduction?

Even if the current method fails to identify stable features that can be associated to molecular signatures, it can be a useful dimension reduction tool. Difficulties associated with a simple mapping of the returned “signature components” to pathways, and problems derived from instability of the found components also affect any of the existing alternative methods. Thus, in the presence of instability of results, we can regard this method as a dimension reduction tool that could lead to simple biological interpretation. The simple biological interpretation could be helped not only because of the coexpression of the genes that make a signature component, but also because the dimension reduction performed is quite remarkable compared to other methods (the number of signature components and genes returned is very small for the five data sets examined; see Results) with, at most, only a slight decrease in predictive performance. Moreover, the user can control the relative trade-offs between

predictive performance and potential interpretability of results (e.g., coexpression of sets of genes) by changing the r_{min} parameter (note that if $r_{min} = 1$ the method becomes essentially either DLDA or NN with forward addition of genes to the model). This flexible modification of the trade-offs between prediction error and interpretability is of great importance in methods that are largely exploratory and oriented towards providing “biologically interpretable” output; in other words, methods for which minimization of prediction error should not be the only goal.

V. CONCLUSIONS

We have developed a method that follows directly from what are often considered as the biologically relevant signature characteristics. The method developed returns signature components of tightly coexpressed genes and thus can facilitate biological interpretation. In this paper we have applied the method to classification problems, but this approach in fact sets up a framework that allows us to find signatures regardless of the type of dependent variable. Our method not only could facilitate mapping pathological alterations to a few, tightly coexpressed sets of genes, but can also provide evidence that the underlying biological assumptions behind this attempt are inconsistent with the data. When applied to simulated data, the method can recover the existing signatures; nevertheless, in the five real data sets analyzed, our results suggest that identification of molecular signatures is questionable under this simple model. These results emphasize the needed for further work on the operationalization of the biological model and the necessity of critical assessment of the stability of putative signatures.

ACKNOWLEDGEMENT

S. Ramaswamy for answering questions about the data and processing steps for the adenocarcinoma dataset, H. Dai for explanations about the breast cancer dataset, and S. Dudoit for answering some questions about the NCI data set. A. Pérez and M. A. Piris for introducing me to, and emphasizing the importance of, molecular signatures. The Bioinformatics Unit at CNIO, C. Lázaro-Perea, Y. Benjamini and F. Falciani for discussion. D. Casado for help with the code. C. Lázaro-Perea and D. Casado for detailed and thorough comments on the ms. The author was partially supported by the Ramón y Cajal program of the Spanish MCyT (Ministry of Science and Technology); funding partially provided by project TIC2003-09331-C02-02 of the Spanish MCyT.

REFERENCES

- [1] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, and L. M. Staudt, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, pp. 503–511, 2000.
- [2] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.
- [3] A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, J. M. Giltneane, E. M. Hurt, H. Zhao, L. Averett, L. Yang, W. H. Wilson, E. S. Jaffe, R. Simon, R. D. Klausner, J. Powell, P. L. Duffey, D. L. Longo, T. C. Greiner, D. D. Weisenburger, W. G. Sanger, B. J. Dave, J. C. Lynch, J. Vose, J. O. Armitage, E. Montserrat, A. Lopez-Guillermo, T. M. Grogan, T. P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, L. M. Staudt, and the Lymphoma/Leukemia Molecular Profiling Project, "The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma," *N Engl J Med*, vol. 346, no. 25, pp. 1937–1947, 2002.
- [4] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub, "Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, no. 1, pp. 68–74, 2002.
- [5] S. Pomeroy, P. Tamayo, M. Gaasenbeek, L. Sturla, M. Angelo, M. McLaughlin, J. Kim, L. Goumnerova, P. Black, C. Lau, J. Allen, D. Zagzag, J. Olson, T. Curran, C. Wetmore, J. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. Louis, J. Mesirov, E. Lander, and T. Golub, "Prediction of central nervous system embryonal tumour outcome based on gene expression," *Nature*, vol. 415, pp. 436–442, 2002.
- [6] A. Shaffer, A. Rosenwald, E. Hurt, J. Giltneane, L. Lam, O. Pickeral, and L. Staudt, "Signatures of the immune response," *Immunity*, vol. 15, pp. 375–385, 2001.
- [7] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. J. Olson, J. R. Marks, and J. R. Nevins, "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proc Natl Acad Sci USA*, vol. 98, no. 20, pp. 11 462–11 467, 2001.
- [8] I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, O. Kallioniemi, B. Wilfond, A. Borg, and J. Trent, "Gene-expression profiles in hereditary breast cancer," *N Engl J Med*, vol. 344, no. 8, pp. 539–548, 2001.
- [9] G. van Belle, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, pp. 530–536, 2002.
- [10] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. Mesirov, T. Poggio, W. Gerald, M. Loda, E. Lander, and T. Golub, "Multiclass cancer diagnosis using tumor gene expression signatures," *Proc Natl Acad Sci USA*, vol. 98, no. 26, pp. 15 149–15 154, 2001.
- [11] T. Hastie, R. Tibshirani, D. Botstein, and P. Brown, "Supervised harvesting of expression trees," *Genome Biology*, vol. 2, pp. 0003.1–0003.12, 2001.
- [12] E. Huang, S. Ishida, J. Pittman, H. Dressman, A. Bild, M. Kloos, M. D'Amico, R. G. Pestell, M. West, and J. R. Nevins, "Gene expression phenotypic models that predict the activity of oncogenic pathways," *Nature Genetics*, vol. 34, no. 2, pp. 226–230, 2003.
- [13] A. Antoniadis, S. Lambert-Lacroix, and F. Leblanc, "Effective dimension reduction methods for tumor classification using gene expression data," *Bioinformatics*, vol. 19, no. 5, pp. 563–570, 2003.
- [14] S. Ramaswamy, K. N. Ross, E. S. Lander, and T. R. Golub, "A molecular signature of metastasis in primary solid tumors," *Nature Genetics*, vol. 33, pp. 49–54, 2003.
- [15] A. Liu, Y. Zhang, E. Gehan, and R. Clarke, "Block principal component analysis with application to gene microarray data classification," *Statist Med*, vol. 21, pp. 3465–3474, 2002.
- [16] I. T. Jolliffe, *Principal component analysis, 2nd ed.* New York: Springer, 2002.
- [17] W. J. Krzanowski, *Principles of multivariate analysis.* Oxford: Oxford University Press, 1998.
- [18] D. F. Morrison, *Multivariate statistical methods.* New York: McGraw-Hill, 1990.
- [19] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," *J Am Stat Assoc*, vol. 97, no. 457, pp. 77–87, 2002.
- [20] C. Romualdi, S. Campanaro, D. Campagna, B. Celegato, N. Cannata, S. Toppo, G. Valle, and G. Lanfranchi, "Pattern recognition in gene expression profiling using dna array: a comparative study of different statistical methods applied to cancer classification," *Hum. Mol. Genet.*, vol. 12, no. 8, pp. 823–836, 2003.
- [21] B. D. Ripley, *Pattern recognition and neural networks.* Cambridge: Cambridge University Press, 1996.
- [22] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning.* New York: Springer, 2001.
- [23] U. Braga-Neto, R. Hashimoto, E. R. Dougherty, D. V. Nguyen, and R. J. Carroll, "Is cross-validation better than resubstitution for ranking genes?" *Bioinformatics*, vol. 20, pp. 253–258, 2004.
- [24] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine Learning*, vol. 40, pp. 203–228, 2000.

- [25] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proc Natl Acad Sci USA*, vol. 99, no. 10, pp. 6562–6566, 2002.
- [26] G. J. McLachlan, *Discriminant analysis and statistical pattern recognition*. New York: Wiley, 1992.
- [27] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. V. de Rijn, M. Waltham, A. Pergamenschikov, J. C. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown, "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genetics*, vol. 24, no. 3, pp. 227–235, 2000.
- [28] M. Dettling and P. Bühlmann, "Boosting for tumor classification with gene expression data," *Bioinformatics*, vol. 19, no. 9, pp. 1061–1069, 2003.
- [29] J. Herrero, R. Díaz-Uriarte, and J. Dopazo, "Gene expression data preprocessing," *Bioinformatics*, vol. 19, no. 5, pp. 655–656, 2003.
- [30] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, "Missing value estimation methods for dna microarrays," *Bioinformatics*, vol. 17, pp. 520–525, 2001.
- [31] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.
- [32] Y. Lee and C.-K. Lee, "Classification of multiple cancer types by multicategory support vector machines using gene expression data," *Bioinformatics*, vol. 19, no. 9, pp. 1132–1139, 2003.
- [33] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," Department of Computer Science, National Taiwan University, URL: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, Tech. Rep., 2003.
- [34] C. J. C. Burgues, "A tutorial on support vector machines for pattern recognition," *Knowledge Discovery and Data Mining*, vol. 2, pp. 121–167, 1998.
- [35] R. Simon, M. D. Radmacher, K. Dobbin, and L. M. McShane, "Pitfalls in the use of dna microarray data for diagnostic and prognostic classification," *Journal of the National Cancer Institute*, vol. 95, no. 1, pp. 14–18, 2003.
- [36] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. London: Chapman and Hall, 1993.
- [37] A. C. Davison and D. V. Hinkley, *Bootstrap methods and their application*. Cambridge: Cambridge University Press, 1997.
- [38] B. Efron and G. Gong, "A leisurely look at the bootstrap, the jackknife, and cross-validation," *Am Stat*, vol. 37, no. 1, pp. 36–48, 1983.
- [39] I. SAS Institute, *SAS/STAT User's guide*. Cary, NC: SAS Institute Inc, 1999.
- [40] B. D. Nelson, "Variable reduction for modeling using proc varclus," in *Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference*, S. Institute, Ed. Cary, NC: SAS Institute, 2001.
- [41] J. F. E. Harrell, *Regression modeling strategies*. New York: Springer, 2001.
- [42] V. Rousson and T. Gasser, "Simple component analysis," Department of Biostatistics, University of Zürich, Switzerland, Tech. Rep., 2003.
- [43] S. K. Vines, "Simple principal components," *Applied Statistics*, vol. 49, no. 4, pp. 441–451, 2000.
- [44] D. Houle, J. Mezey, and P. Galpern, "Interpretation of the results of common principal components analyses," *Evolution*, vol. 56, pp. 433–440, 2002.
- [45] E. Huang, S. H. Cheng, H. Dressman, J. Pittman, M.-H. Tsou, C.-F. Horng, A. Bild, E. S. Iversen, M. Liao, C.-M. Chen, M. West, J. R. Nevins, and A. T. Huang, "Gene expression predictors of breast cancer outcomes," *Lancet*, vol. 361, pp. 1590–1596, 2003.
- [46] M. Dettling and P. Bühlmann, "Supervised clustering of genes," *Genome Biology*, vol. 3, no. 12, pp. 0069.1–0069.15, 2002.
- [47] —, "Finding predictive gene groups from microarray data," *J. Multivariate Anal.*, vol. 90, pp. 106–131, 2004.
- [48] M. Hills, *Encyclopedia of statistical sciences, Volume I*. New York: Wiley, 1982, ch. Allometry, pp. 48–54.
- [49] S. Bartoletti, B. D. Flury, and D. G. Nel, "Allometric extension," *Biometrics*, vol. 55, pp. 1210–1214, 1999.
- [50] H. Bensmail and G. Celeux, "Regularized gaussian discriminant analysis through eigenvalue decomposition," *Journal American Statistical Association*, vol. 91, pp. 1743–1748, 1996.
- [51] B. Flury, *Common principal components and related techniques*. New York: John Wiley & Sons, 1988.
- [52] G. Boente, A. M. Pires, and I. Rodrigues, "Influence functions and outlier detection under the common principal components model: a robust approach," *Biometrika*, vol. 89, pp. 861–875, 2002.
- [53] R. L. Somorjai, B. Dolenko, and R. Baumgartner, "Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions," *Bioinformatics*, vol. 19, pp. 1484–1491, 2003.
- [54] H. Zhang, C.-Y. Yu, and B. Singer, "Cell and tumor classification using gene expression data: construction of forests," *Proc Natl Acad Sci USA*, vol. 100, no. 7, pp. 4168–4172, 2003.
- [55] L. Breiman, "Statistical modeling: the two cultures (with discussion)," *Statistical Science*, vol. 16, pp. 199–231, 2001.
- [56] D. Rhodes, J. Yu, K. Shanker, N. Deshpande, R. Varambally, D. Ghosh, T. Barrette, A. Pandey, and A. Chinnaiyan, "Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression." *PNAS*, vol. 101, pp. 9309–9314, 2004.
- [57] R. Jörnsten and B. Yu, "Simultaneous gene clustering and subset selection for sample classification via MDL," *Bioinformatics*, vol. 19, pp. 1100–1109, 2003.
- [58] L. Lazzeroni and A. Owen, "Plaid models for gene expression data," *Statistica Sinica*, vol. 12, pp. 61–86, 2002.

- [59] H. Turner, T. Bailey, and W. Krzanowski, "Improved biclustering of microarray data demonstrated through systematic performance tests." *Comput. Statist. Data Anal.*, p. In press, 2004.
- [60] G. Parmigiani, E. Garrett, R. Anbazhagan, and E. Gabrielson, "A statistical framework for expression-based molecular classification in cancer," *J. Royal Statistical Society, Series B*, vol. 64, pp. 717–736, 2002.
- [61] E. Garrett and G. Parmigiani, *The analysis of gene expression data: methods and software*. New York: Springer, 2003, ch. POE: Statistical methods for qualitative analysis of gene expression, pp. 362–387.
- [62] R. Scharpf, E. S. Garrett, J. Hu, and G. Parmigiani, "Statistical modeling and visualization of molecular profiles in cancer," *Biotechniques*, vol. Suppl, pp. 22–29, 2003.