Survival prediction using gene expression data: a review and comparison

Wessel N. van Wieringen *

Department of Mathematics, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

David Kun

Department of Mathematics, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Regina Hampel

Institute for Medical Statistics and Epidemiology, Technical University of Munich, Ismaningerstr. 22, D-81675 Munich, Germany,
Institute for Epidemiology, GSF, Ingolstädter Landstrasse 1, D-85764 Neuherberg, Germany

Anne-Laure Boulesteix

Institute for Medical Statistics and Epidemiology, Technical University of Munich, Ismaningerstr. 22, D-81675 Munich, Germany,
Sylvia Lawry Centre for Multiple Sclerosis Research, Hohenlindenerstr. 1, D-81677 Munich, Germany

---

Abstract

Preprint submitted to Elsevier *25 May 2008*

Knowledge of the transcription of the human genome might greatly enhance our understanding of cancer. In particular, gene expression may be used to predict the survival of cancer patients. Microarray data are characterized by their high-dimensionality: the number of covariates ($p \sim 1000$) greatly exceeds the number of samples ($n \sim 100$), which is a considerable challenge in the context of survival prediction. An inventory of methods that have been used to model survival using gene expression is given. These methods are critically reviewed and compared in a qualitative way. Next, the methods are applied to three real-life data sets for a quantitative comparison. The choice of the evaluation measure of predictive performance is crucial for the selection of the best method. Depending on the evaluation measure, either the L_2-penalized Cox regression or the random forest ensemble method yields the best survival time prediction using the considered gene expression data sets. Consensus on the best evaluation measure of predictive performance is needed.

Key words: Proportional hazard model, Microarray, Brier score.

1 Introduction

Knowledge of the human genome and its expression might greatly enhance our understanding of cancer [15]. In particular, the gene expression of tissue of cancer patients may be used to predict their survival time. A microarray measures the expression of thousands of genes simultaneously (which genes are expressed and to what extent). The reader is referred to [43] for an overview of the biological and technical aspects of the microarray technology.

In survival analysis one studies survival time, which is defined as the time

* Corresponding author.

length from the beginning of observation until the death (or some other event) of the observed patient or until the end of observation. The main goal is to predict the time to event (commonly denoted as survival time even if the considered event is not death) using the expression of the genes as explanatory variables. Because the event is not observed for all observations, standard regression techniques cannot be applied: censoring has to be taken into account. The comparison presented here is limited to methods that use the proportional hazard model [18] or tree-based ensemble methods [27] to link the survival time to gene expression.

Traditionally, the Cox proportional hazard model is applied in a situation where the number of samples greatly exceeds the number of covariates ($n > p$). When predicting survival with gene expression data, one runs into the problem of the high-dimensionality of microarray data: the number of genes ($p \sim 1000$) exceeds by far the number of samples ($n \sim 100$), yielding the well-known "$p >> n$" paradigm. In addition to high-dimensionality, gene expression data are often highly correlated, which further increases the collinearity between explanatory variables.

Here we give an inventory of methods that have been used to predict survival time using gene expression. These methods are first critically reviewed and compared in a qualitative way. They are then applied to three real-life cancer data sets for a quantitative comparison. We find that the choice of the evaluation measure of predictive performance is crucial for the selection of the best method. Depending on the evaluation measure, either the L_2-penalized Cox regression or the random forest ensemble method yields the best survival time prediction with the data sets considered in this study (see results section). Consensus on the best evaluation measure of predictive performance is

needed.

Many methods have been proposed in the emerging field of high-dimensional biological data analysis. Hence, the need for insightful review articles and fair comparison studies is probably as stringent as the need for new methodological developments [8]. Boulesteix et al. [11] define a fair comparison study of statistical prediction methods as follows.

- The title includes explicitly words such as "comparison" or "evaluation", but no specific method is mentioned in the title, thus excluding articles whose main aim is to demonstrate the superiority of a particular (new) method.
- The article is written at a high statistical level. In particular, the methods are described precisely (including, e.g. the chosen variant or the choice of parameters) and adequate statistical references are provided.
- The comparison is based on at least two data sets.
- The comparison is based on at least one of the following evaluation strategies: CV, MCCV, bootstrap methods (see Section 3.2).

Even if the above rules are respected, different teams are expected to obtain different results, for instance because they do not use the same evaluation design, the same implementation or the same parameter tuning procedure. Moreover, optimistically biased results are likely to be obtained with the method(s) from the authors' expertise area [11]. For example, authors are aware of all available implementations of that method and will quite naturally choose the best one, with the best parameter settings. Similarly, an unexperienced investigator might overestimate the error rate of methods involving many tuning parameters by setting them to values that are known to the experts as sub-

optimal.

While the relative performance of class prediction methods has been investigated in several high-quality neutral studies, e.g. [19,57], there are to our knowledge very few published neutral comparison studies of survival prediction methods [12,53]. Corroboration of findings from independent studies, which is crucial in medical research [29] would strengthen the conclusions. Moreover, the present article includes several important aspects which not considered in the previous studies. We use different implementations of several other methods (SuperPC and PLS) and study additional approaches (bagging and random forest). Moreover, we consider different evaluation measures to assess the predictive power of the methods (for instance, [12] does not include the Brier score). The three comparison studies have one real-life data set in common, but we also analyze two additional benchmark data sets.

Notation and the proportional hazard model

Let $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ denote the times of observation of the n available patients and T the corresponding random variable: t_i is either the time until the death of the i-th patient or the time until the end of the observation (in which case the observation is right-censored: death of the patient did not happen before the censoring). The censoring variable δ_i indicates whether the patient died at time t_i ($\delta_i = 1$), or the observation was censored ($\delta_i = 0$). Further, D denotes the set of indices of the events (i.e. such that $\delta_i = 1$), and R_r, $r \in D$ are the sets of indices of the individuals at risk at time $t_r - 0$. Finally, let $\mathbf{X}$ be the $(n \times p)$ gene expression matrix, where X_{ij} is the expression level of gene j in sample i.

The proportional hazard model [18] models the hazard rather than the survival time. The hazard is defined as:

$$h_T(t) = \lim_{r \to 0} \frac{S_T(t) - S_T(t+r)}{r S_T(t)} = \lim_{r \to 0} \frac{P(t \leq T \leq t+r | T \geq t)}{r},$$

where $S_T(t) = \mathrm{P}(T > t) = 1 - \mathrm{P}(T \leq t)$ is the survival function describing the probability of surviving after time point t. Then, $h_T(t) \cdot \Delta t$ can be interpreted as the probability of an event occurring at the next instance, given that the patient has survived until t [33]. The survival function and the hazard are related through $S_T(t) = \exp\left(-H_T(t)\right) = \exp\left(-\int_0^t h_T(u)du\right)$, where $H_T(t)$ is the cumulative hazard. The proportional hazard model is given by $H(t, \mathbf{X}) = H_0(t) \exp\left(f(\mathbf{X}; \boldsymbol{\beta})\right)$, where $H_0(t)$ is an unspecified baseline hazard function that is assumed to be the same for all patients, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the parameter vector to be estimated and f is a function of the explanatory variables, often $f(\mathbf{X}; \boldsymbol{\beta}) = \beta_1 X_1 + \ldots + \beta_p X_p$. The parameter vector $\boldsymbol{\beta}$ and the baseline $H_0(t)$ are estimated using partial likelihood maximization and the Breslow-estimator, respectively. Given the estimated parameters and baseline hazard, the hazard and survival for a new sample with expression profile $\tilde{\mathbf{X}}$ are predicted by $\widehat{H}_0(t_i, \tilde{\mathbf{X}}) \exp(\tilde{\mathbf{X}}^T \widehat{\boldsymbol{\beta}})$ and $\exp\left(-\widehat{H}_0(t_i, \tilde{\mathbf{X}}) \exp(\tilde{\mathbf{X}}^T \widehat{\boldsymbol{\beta}})\right)$, respectively.

2 The inventory

Here we present an inventory of methods that have been proposed to predict survival using high dimensional microarray data. The articles presented in this chapter were found using the search engines `www.pubmed.gov` and `www.scholar.google.com`. References in the articles found were also checked.

Nevertheless, we do not claim our search has been exhaustive. In the comparison study, we focus on methods which have had wide impact: comparing all the methods proposed in the literature in a single study with the same amount of precision would be an impossible task. The methods are described at a high level only: the reader is referred to the original article for a more detailed description. Some methods have been proposed by several authors with only minor differences. We have chosen to describe only one of them and point out the differences with the others.

Clusters as predictors

Alizadeh et al. [1] and many others thereafter first group the samples by means of (hierarchical) clustering and use this grouping as a prognostic factor in the Cox proportional hazard model. They cluster the samples by means of the gene expression profiles. The resulting clustering labeling is then considered as a summary of the predictive information contained in the expression data, and is used in survival analysis.

Formally, this is not a prediction method, but could be used as such by assigning each new sample to one of the found clusters. The estimated Kaplan-Meier curve for this cluster yields the predicted survival time of the new sample. The relation between the survival time and the original explanatory variables, however, is obscured by an intricate link between the gene expression and the cluster label. Moreover, it is unclear whether this unsupervised procedure makes efficient use of the available predictive information of the gene expression, and the choice of the number of clusters and of the clustering algorithm may be problematic. Further, it is now well-established that summarizing continuous

variables as categories (or clusters) is often a bad idea [52]. We therefore do not include this method in our comparison.

Supervised harvesting of expression trees

Hastie et al. [24] use the (summarized) expression of groups of genes as a predictor for survival. The groups are found by (hierarchical) clustering of the genes. The expression of the genes in a cluster is summarized, e.g. by taking the sample-wise average. The averaged expressions are then used to predict survival.

The method starts by clustering the p genes hierarchically, resulting in $p + p - 1 = 2p - 1$ clusters. Next, the corresponding average expression profile is calculated for every cluster giving $2p - 1$ new predictors. Using the $2p - 1$ resulting predictors, one builds a proportional hazard model also including first-order interactions [24]. The authors suggest that higher order interactions could be included in the model, but for computational reasons limit themselves to pairwise interactions. Hastie et al. [24] decide on the clusters to be used by means of a forward stepwise/backward deletion method, where the final model is chosen by K-fold cross-validation.

The supervised harvesting method has been heavily criticized. Hastie et al. [24] themselves suggest that the method might need a large number of samples to successfully discover interactions. Two other drawbacks are pointed out by Segal et al. [55]: 1) the model is very sensitive to the clustering method applied in the first step, and 2) gene harvesting may include heterogenous clusters whose average expression correlates with the outcome, despite that the individual genes in the cluster exhibits little association. We thus do not

take the supervised harvesting of expression trees along in the comparison.

Univariate gene selection

The most straightforward and intuitive approach to handle high-dimensional data consists of carrying out univariate gene selection and using the obtained (small) subset of genes as covariates in a standard Cox model. Such an approach was adopted by, e.g., Jenssen et al. [30]. We include univariate gene selection in our comparison study as a baseline reference method. In contrast to Jenssen et al. [30], we do not group the expression of each gene into three categories (low, median, high) but rather consider gene expression as a metric variable. Consequently, we order genes based on the p-value obtained using Wald's test in univariate Cox regression rather than the logrank test. Contrary to Jenssen et al [30], we select a pre-fixed number of genes ($\tilde{p} = 10$ in the present study) rather than genes whose p-values fall below a threshold. This warrants that we will have a set of genes of convenient size for any training set. Furthermore, the univariate Cox regression model is estimated based on the training data only, which is a universally recommended approach [9,57].

Univariate feature selection does not account for the correlation between genes. Consequently, many highly correlated genes may be selected. This is indeed observed. As a consequence of high correlation, many genes which are univariately significant have insignificant p-values in the multivariate proportional hazard model.

Supervised principal component analysis

Bair and colleagues [3,4] propose supervised principal component analysis (SuperPC) to predict survival using gene expression data. SuperPC is a modification of the conventional principal components analysis (PCA). It handles the problem of high-dimensionality by using not all genes, but only those with the strongest estimated correlation with the survival time for the principal component analysis. These components are then used in the proportional hazard model.

SuperPC first calculates the (absolute) Cox score statistic as a measure of the univariate association between each gene and the survival time. A threshold for the Cox scores is then chosen by cross-validation. A reduced expression matrix consisting of only those genes whose Cox score statistic exceeds the threshold is formed. Predictors for survival are constructed by means of singular value decomposition from the reduced expression matrix. The resulting principal component(s) is (are) used in the proportional hazard model to predict the survival.

The original papers on SuperPC do not suggest any method for selecting the number of principal components to be used in survival prediction. The examples in [3] and [4] use only one or two principal component(s).

The principal components are a weighted average of the original expression profiles, which can be interpreted as 'eigengenes', 'supergenes' or 'meta-genes' [2]. This interpretation is however a label without content, because it is neither linked to a biological entity nor to a theoretical construct. The interpretation of components is generally not straightforward, especially if the number of genes contributing to the component is large. Regardless of their interpretation

principal components may be excellent survival time predictors. Furthermore, as a dimension reduction method, the SuperPC approach allows simple low-dimensional visualization. The implementation of SuperPC available from the R package `superpc` is used in the remainder with one or two component(s) following [3] and [4].

Partial Cox regression

Nguyen and Rocke [44] propose to use the partial least squares (PLS) algorithm for the prediction of survival time with microarray data. Their procedure, however, does not handle the censoring aspect of the survival data properly. Extensions of the PLS algorithm that can handle censored data are suggested by Nguyen and other researchers [6,7,38,45,46]. See [10] for a review of these methods. In the present article, we focus on the method by Bastien [6,7], because its statistical properties are well-documented [7] and the algorithm does not involve any critical iterative optimization step, in contrast to [46].

In a nutshell, PLS is a supervised dimension reduction technique which can be used to relate a response variable to the explanatory variables $\mathbf{X}$, see [41] for an historical overview of the development of the different PLS variants and [10] for an overview of recent applications to genomic data. When applied to regression problems without censoring, the method constructs the new components as mutually orthogonal linear combinations of the covariates having maximal covariance with the response. PLS differs from PCA in that the constructed components have maximal covariance with the response instead of maximal variance. In contrast to PCA, PLS is thus a supervised dimension reduction

method. The loadings of the explanatory variables on the components are non-linear functions of the explanatory variables and the response. They are found by a computationally efficient iterative algorithm.

Bastien [7] modifies the standard PLS method by replacing the linear regression step by Cox regression for the derivation of the PLS coefficients. The first PLS component is a weighted sum of the centered expression values. In Bastien's method, these weights equal the regression coefficients of the univariate proportional hazard model up to a scaling constant. Next, the expression values are regressed against the formed component. The residuals are used for the construction of the next component, which is also a weighted sum with weights chosen in a similar fashion. The process is continued until K components are constructed. K could be determined by cross-validation.

PLS has been critized for having undesirable shrinkage properties [17]. See [34] for a discussion. It is however unclear whether this criticism applies to the generalized PLS method under study. As with the SuperPC method the resulting components may be hard to interpret. Nonetheless, they still may be excellent survival time predictors. Further, as a dimension reduction method, PLS approaches allows simple low-dimensional visualization. In the present paper, we focus on the method by [7] which we have re-implemented in R. The number of PLS components is fixed to one or two, like for the SuperPC method.

L_2-penalized Cox regression

Pawitan et al. [48], Hastie et al. [25] and Van Houwelingen et al. [60] all propose to use the Cox-model with a quadratic penalty (ridge regression) in

order to predict survival time based on gene expression data. However, this particular penalized approach is computationally too intensive for the $p >> n$ situation. The penalized Cox regression is thus combined with a dimension reduction technique for reducing the number of computational operations. All three aforementioned methods use cross-validation to find the optimal penalty parameter.

All methods replace the expression matrix by the $(n \times n)$ matrix of principal components. This reduces the dimension of the covariate space from p to n dimensions. It is motivated by a theorem stating that we can replace the p covariates by the n principal components, use the latter in the penalized regression problem, and obtain the solution of the original problem by retracing the singular value decomposition [25].

Van Houwelingen et al. [60] initiate their estimation algorithm by estimating the baseline hazard (using $\boldsymbol{\beta} = 0$). Next, they estimate $\boldsymbol{\beta}$ by maximizing the penalized total likelihood, whereas Pawitan et al. [48] and Hastie et al [25] maximize the partial log-likelihood. The resulting estimates of $\boldsymbol{\beta}$ are used to update the estimate of the baseline hazard. The latter two steps are alternated until convergence. The optimal value for λ is chosen by leave-one-out cross-validation using the cross-validated partial log-likelihood [62]. Pawitan et al. [48] use the adjusted profile-likelihood for cross-validation, and Hastie et al [25] propose the score statistic for cross-validation.

As with the SuperPC method the resulting components may be hard to interpret. Nonetheless, they still may be excellent survival time predictors. Further, the components allow simple low-dimensional visualization. By retracing the singular value decomposition, one could also obtain the estimates for the indi-

vidual genes. A drawback of this approach is that all coefficients are allowed to be non-zero, thus yielding complex models. In the present study, we use an R implementation of the method of Van Houwelingen et al. [60] which was provided by Jelle Goeman.

L_1-penalized Cox regression

As opposed to Van Houwelingen et al. [60], Park and Hastie [47] (but also Gui and Li [23] and Segal [56]) use the Cox model with an L_1-penalty. The L_1-penalty has the advantage (over the L_2-penalty) of shrinking some of the coefficients to zero. Hence, it has an in-built feature selection procedure. The use of an L_1-penalized Cox model is proposed by Tibshirani (1997) [59]. His algorithm to fit the model is inefficient for the situation under study ($p >> N$, with $p \approx 10000$). Park and Hastie [47] propose a computationally efficient algorithm to fit the Cox model with an L_1-penalty. As opposed to [23] and [56], they use a penalty given as a linear combination of the L_1- and L_2-norm, which stabilizes the fitting procedure in the presence of strong correlations between covariates. It also leads to grouping of highly correlated features [64].

Park and Hastie [47] formulate the estimation equation for $\boldsymbol{\beta}$ as:

$$\widehat{\boldsymbol{\beta}}(\lambda) = \text{argmin}_{\boldsymbol{\beta}} - \ln L(\mathbf{t}, \mathbf{X}, \delta, \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1,$$

where $\lambda > 0$ is the regularization parameter and $\|.\|_1$ stands for the L_1 norm. They introduce an algorithm that determines the entire path of the coefficient estimates as λ varies, i.e. that finds $\{\widehat{\boldsymbol{\beta}}(\lambda) : 0 < \lambda < \infty\}$. The algorithm starts with calculating $\widehat{\boldsymbol{\beta}}(\infty)$, which results in setting $\boldsymbol{\beta}$ equal to zero. Then it computes a series of solution sets $\{\lambda_i, \boldsymbol{\beta}(\lambda_i)\}$, each time estimating the

coefficients with a smaller λ based on the previous estimate. Hence, with each step the penalty is lowered, encouraging more coefficients to become non-zero. The estimates of $\boldsymbol{\beta}$ and the step size of λ between iterations are determined by the previous estimates. The iteration stops if the set of non-zero coefficients is not augmented anymore. The optimal λ is chosen by cross-validation or by minimization of the Bayesian information criterion.

The in-built feature selection preserves the original interpretation of the genes. However, we observe that the added L_2-penalty does not prevent highly correlated features to be included, as claimed in [64]. The set of features may thus contain some redundancy, which does not necessarily affect the predictive power of the method. Further, the computation time of the method is among the highest of the methods in the comparison. In this study, we use the R package `glmpath`.

Tree-based ensemble methods

The considered ensemble methods offer the possibility to include hundreds or thousands of covariates. However, it is recommended to carry out preliminary variable selection instead of including all covariates in order to reduce computation time. In this article, we set the number of covariates to $\tilde{p} = 200$. In our experience larger numbers of covariates yield similar results. Variable selection is always performed using the training data set only, as commonly recommended in the context of prediction using high-dimensional data [9]. Performing variable selection using both the training and test data sets would indeed lead to an underestimation of the prediction error.

The term "bagging" is the abbreviation of **b**ootstrap **agg**regat**ing** and is first

introduced by Breiman [14]. The basic idea of bagging is to aggregate a large number of estimators in order to improve the prediction compared to one single estimator. Bagging of survival trees is investigated by Hothorn et al. [26,27].

A tree is built by successively splitting into two groups (nodes). If X_j denotes the splitting covariate, one node contains all observations with $X_j \leq c$, while the other node contains all observations with $X_j > c$. The covariate X_j and the threshold c are selected based on a splitting criterion. Nodes which are not splitted anymore (according to the so-called "stopping criterion") are denoted as leaves. For details see [13,32,54]. Conditional inference trees [28] are an important class of decision trees based on statistical tests and include survival trees as a special case. In this article, we focus on such trees which are reported to be less biased than previous approaches [28]. In survival trees, prediction is performed as follows. For each leaf, the Kaplan-Meier estimation of the survival function is carried out based on the training observations forming this leaf. A new observation is dropped down the tree and predicted by the Kaplan-Meier curve of the leaf it falls into.

Bagging of survival trees is carried out as follows. A total of B bootstrap samples of size n are drawn from the training sample. A survival tree is generated for each of the B bootstrap samples. Prediction is performed by dropping the new observation down the B trees successively. The observations from the B leaves in which the new observation falls are combined into one single (large) sample, from which the predicted Kaplan-Meier curve is estimated.

Random Forests are another ensemble method used for survival analysis by Hothorn et al. [27]. This method is based on the same principle and aggregation

scheme as bagging. However, the trees are built in a different way. Not all covariates are used to generate a tree. Instead, only a subset of randomly selected covariates are considered as candidate covariates at each splitting, while the others are ignored. The number of selected covariates is a parameter of the method. Note that random forests are equivalent to bagging if the number of selected covariates is set equal to the total number of covariates.

The non-parametric tree-based ensemble methods are not based on a particular stochastic model and can thus be applied in a wide range of situations. An inconvenience is that they involve many tuning parameters such as the number of trees, the number of candidate covariates at each split or the stopping criterion. However, our experience is that i) the results do not depend much on the parameters, ii) the default settings of the package `party` do not need to be changed. A further inconvenience of ensemble methods is the random component induced by bootstrap sampling: two forests grown using the same training data usually yield slightly different results. Variability can only be decreased at the price of computing time, by increasing the number of trees. Finally, ensemble methods share the pitfall of many machine learning approaches: in contrast to single trees, the output of ensembles is hard to interpret for non-experts.

In the current study, we use the implementations of bagging and random forests available from the `R` package `party` [28] with the default parameter values.

Other methods

To keep the scope of the study manageable, we only include a limited number of methods in the comparison. Some arbitrariness is unavoidable, but we feel that the current selection covers the spectrum of methods reasonably well.

Many other, but less widely used methods linking survival times and gene expression have been proposed [36,37,39,31,40,58,63]. The more complex problem of survival analysis based on longitudinal data is addressed by Rajicic et al. [50]. A related approach is the `globaltest` method by Goeman et al [21], which tests whether a group of genes is associated with the outcome variable, for instance, survival. This approach is not a prediction method, but, as pointed out in [60], it may be used as a preliminary step before prediction to assess whether the expression data have predictive potential.

3 Results: Comparison of the methods

Here we compare the methods described above qualitatively and quantitatively. In the qualitative comparison we use high-level characteristics to group and discriminate the methods on the basis of how they produce predictors in the "$p >> n$"-paradigm. The quantitative comparison consists of the assessment of predictive performance in real-life data sets.

3.1 Qualitative comparison

All methods handle the problem of high-dimensionality by some form of dimension reduction in order to use the expression data to predict survival. The

aim of dimension reduction is to find a set of $\widetilde{p}$ new features based on the input set of p features improving prediction accuracy or decreasing the number of features without significantly decreasing prediction accuracy of the predictor built using only the new features.

Two strategies of dimension reduction can be distinguished: feature selection and feature extraction. *Feature selection* consists of selecting the best possible subset of the input feature set. This preserves the interpretability of the original data. *Feature extraction* consists of finding a transformation to a lower dimensional space. The new features are then a linear or non-linear transformation of the original features. This may improve the prediction ability, but may not have a clear physical meaning. Note that feature selection is in fact a special case of feature extraction.

The dimension reduction strategies are further characterized by univariate versus multivariate approaches and supervised versus unsupervised approaches. *Univariate approaches* for dimension reduction consider each individual gene separately from the other, whereas *multivariate approaches* take into account the covariance or/and interactions between genes. *Supervised approaches* for dimension reduction take into account the response (survival) information of the samples within the training set. *Unsupervised approaches* make no use of survival data in the dimension reduction.

The characteristics of the methods considered in this article are summarized in Table 1. This is done to facilitate a comparison of the methods on a conceptual level, but also to interpret performance differences on the real-life data later. Supervised principal component analysis [3] is an element of every group, be-

Method	Feature selection/extraction	Uni/multi-variate approach	Supervised approach
Univariate gene selection	sel	uni	y
Supervised PCA	ext/sel	uni/multi	y/n
Partial Cox regression	ext	multi	y
L_1-penalized Cox regression	sel	multi	y
L_2-penalized Cox regression	ext	multi	y
Bagging	ext	uni/multi	y
Random forests	ext/sel	uni/multi	y

Table 1
Dimension reduction strategies of the compared methods.

cause it contains two dimension reduction steps. First a univariate supervised feature selection is done, followed by a feature extraction, when the principal components are constructed multivariately and unsupervisedly. Bagging and random forests are used in combination with preliminary univariate gene selection, but tree construction is essentially multivariate, since it takes interactions into account. Hence, these methods can be seen as a combination of feature selection and extraction.

¿From Table 1 it is obvious that all methods are (at least partially) supervised approaches, and perform dimension reduction mostly multivariately. Within those characteristics it is the feature selection/extraction that sets the methods apart.

3.2 Predictor evaluation for the quantitative comparison

The true evaluation of a predictor's performance is to be done on independent data. In the absence of independent data (the situation considered here) the predictive accuracy can be estimated as follows [20]. The samples are splitted

into mutually exclusive training and test sets. The gene expression and survival data of the samples in the training set are used to build the predictor. No data from the test set are used in the predictor construction (including variable selection) by any of the methods compared. This predictor is considered to be representative of the predictor built on all samples (of which the training set is a subset). The test set is used to evaluate the performance of the predictor built from the training set: for each sample in the test set, survival is predicted from gene expression data. The predicted survival is then compared to the observed survival and summarized into an evaluation measure (discussed below). To avoid dependency on the choice of training and test set, this procedure is repeated for multiple splits. The average of the evaluation measures resulting from each split is our estimate of the performance of the predictor built using the data from all samples.

We now discuss evaluation measures of predictive performance. It is not straightforward to evaluate and compare prediction methods in the presence of censoring. The standard mean-squared-error or misclassification rate criteria used in regression or classification cannot be applied to censored survival times. In this section we describe the three measures (p-value, R^2 and Brier score) used in the present comparison study to evaluate the prediction of the methods compared. The first two measures are based on the Cox model, while the third measure (Brier score) uses the predicted survival curves, which can also be derived via other approaches such as tree-based procedures. For applying the two Cox-based measures to tree-based prediction methods (bagging and random forests), we simply extract the predicted median survival time from the predicted survival curves and use it as a predictor in a univariate Cox model. This approach, though possibly suboptimal, allows to compare all the

prediction methods with all evaluation measures.

Here we elaborate on the three evaluation measures, as each is a different operationalization of predictive performance.

- *p-value (likelihood ratio test):* To assess whether the built predictor has significant predictive power, we use the likelihood ratio test [35]. This is a well-known statistical test used to make a decision between two models, (here) a null model having no predictive power and the built predictor. More formally, the test evaluates the null hypothesis $H_0 : \boldsymbol{\beta} = 0$, i.e. the gene expression predictor has no effect on survival. The null hypothesis is evaluated using the likelihood ratio test statistic $LLR(\widehat{\boldsymbol{\beta}}) = -2\,(l(0) - l(\widehat{\boldsymbol{\beta}}))$, with $l(.)$ denoting the value of the log-likelihood function. Under the null hypothesis this test statistic has a χ^2 distribution, which is used to calculate the p-value. The p-value summarizes the evidence against H_0: the lower the p-value the more probable that H_0 is not true. In other words, the lower the p-value, the more evidence that the built predictor is a good predictor of survival. Note that this p-value is derived from the test data set: the data that were used to construct the predictor are not used for its evaluation. The p-value of the likelihood ratio test has been used as an evaluation measure for predictive performance of gene expression based predictors of survival by many others [3,12,46,56].
- R^2 *criterion:* To quantify the proportion of variability in survival data of the test set that can be explained by the predictor, we use the coefficient of determination (henceforth called R^2). A predictor with good predictive performance explains a high proportion of variability in the survival data of the test set, and vice versa a poor predictor explains little variability in the test set. In a traditional regression setting the R^2 statistic is one mi-

nus the ratio of the residual sum of squares and the total sum of squares. Consequently, it ranges from 0 (no explained variation), to 1 (all variation explained). This definition can however not be used in the context of censored data. Nagelkerke [42] gives a general definition of the R^2 statistic that can be used for Cox proportional hazard models:

$$R^2 = 1 - \exp\left(-\frac{2}{n}(l(\widehat{\boldsymbol{\beta}}) - l(0))\right), \tag{1}$$

where $l(.)$ denotes the log-likelihood function. Others have also used the R^2 statistic to assess predictive performance of gene expression based predictors of survival [3,56].

- *Brier score:* The goodness of a predicted survival function can also be assessed based on the integrated Brier-Score introduced by Graf et al [22]. The Brier-Score $BS(t)$ [26,49] is defined as a function of time $t > 0$ by

$$BS(t) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\widehat{S}(t \mid \mathbf{X}_i)^2 \mathrm{I}(t_i \leq t \wedge \delta_i = 1)}{\widehat{G}(t_i)} + \frac{(1 - \widehat{S}(t \mid \mathbf{X}_i))^2 \mathrm{I}(t_i > t)}{\widehat{G}(t)}\right], \tag{2}$$

where $\widehat{G}(\cdot)$ denotes the Kaplan-Meier estimate of the censoring distribution which is based on the observations $(t_i, 1 - \delta_i)$ and I stands for the indicator function. The values of the Brier-Score range between 0 and 1. Good predictions at time t result in small Brier-Scores. The numerator of the first summand is the squared predicted probability that individual i survives until time t if he actually died (uncensored) before t, or zero otherwise. The better the survival function is estimated, the smaller is this probability. Analogously, the numerator of the second summand is the squared probability that individual i dies before time t if he was observed at least until t, or zero otherwise. Censored observations with survival times smaller than t are weighted with 0. The Brier-score as defined in Eq. (2) depends on t. It

makes sense to use the integrated Brier-Score (IBS) given by

$$IBS = [\max(t_i)]^{-1} \int_0^{\max(t_i)} BS(t)\mathrm{dt}\,. \tag{3}$$

as a score to assess the goodness of the predicted survival functions of all observations at every time t between 0 and $\max(t_i)$, $i = 1, \ldots, N$. Note that the IBS is also appropriate for prediction methods that do not involve Cox regression models: it is more general than the R^2 and the p-value criteria and has thus become a standard evaluation measure for survival prediction methods [27,53]. The Brier score is implemented in the function `sbrier` from the package `R` package `ipred`, which we use in this article.

As an alternative measure of predictive performance we also considered the variance of the martingale residuals in the Cox model. However, we found that this measure is not able to discriminate well between between good and poor predictors in the considered setting (data not shown). It is therefore omitted here.

As a preliminary step to our real data study, we have simulated two data sets, one mimicking a situation with no predictive information for survival contained in the gene expression and the other mimicking a situation with predictive information for survival in the gene expression. Our goal was to assess whether the above measures are indeed able to distinguish predictors with poor predictive performance and from predictors with good predictive performance. Comparing the average of the performance measures over the simulated data sets, we observed, as expected, a dramatic decrease of the p-value and Brier score from simulated data set 1 to simulated data set 2, and similarly a dramatic increase of the R^2 statistic (data not shown). This indicates that the metrics are indicative of predictive performance.

The integrated Brier score may be the metric of choice as it is based on predicted survival curves, which are output by most survival prediction methods. Hence, it can be used even when the Cox model does not hold. There is however a practical argument in favor of the p-value and R^2 as measures of predictive performance, which lies in the fact that they are well understood by life scientists. Life scientists have initiated the development of gene expression predictors of clinical outcome and they will also be the end users. It is therefore important that they understand the evaluation of a predictor's performance. This may be achieved by explaining, e.g. the Brier score in detail. Our experience is that they often find this too abstract a notion to understand. Although they are willing to accept our findings based on, e.g. the Brier score, they highly appreciate the communication of results in terms of a metric for which they have developed intuition, like the p-value and R^2.

3.3 Analysis of real-life data sets

In this section, three publicly available data sets are analyzed: the breast cancer data set by Van 't Veer et al. [61], the AML data set by Bullinger et al. [16] and the DLBCL data set by Rosenwald et al [51].

Design of real-life data sets

The breast cancer data set by Van 't Veer et al. [61] is available at `http://www.rii.com/publications/2002/vantveer.html`, and consists of 295 samples measured on cDNA arrays. Each gene expression profile consists of 24885 genes. There are no missing values in the data set. Following Van Houwelingen et al [60], we reduce the number of genes on the basis of the p-values from

the Rosetta error model. Genes with a p-value less than 0.01 in 45 of the 295 samples are removed, leaving 5057 genes in the data set. The overall survival (death due to any cause) is taken as the endpoint.

The AML data set of Bullinger et al. [16] can be downloaded from the GEO data base with accession number GSE425. It comprises 119 gene expression profiles, each made up of 6283 genes. We remove samples with more than 1800 missing values, 13 in total. We also remove genes who had over 20% missings. This leaves us with a data set of 103 expression profiles containing 4673 genes. Remaining missing values are imputed using the R function `impute.knn` from the `impute` package. The overall survival data is used as the endpoint in the analysis.

The DLBCL data set by Rosenwald et al. [51] is available at `http://llmpp.nih.gov/DLBCL/`. It consists of 240 samples with expression profiles of 7399 genes. Missing values are imputed using the R-package `knn.impute`. Again overall survival is used for analysis.

Real-life data sets results

Before applying the survival prediction methods to each data set, we apply the `globaltest` procedure by Goeman et al. [21] to each real-life data set. The test reveals that in all data sets there is a significant association between survival times and gene expression, which should be picked up by each method.

The real-life data sets are randomly split into training and test sets with a $2:1$ ratio. To ensure that the evaluation measures do not depend on the particular split into training and test sets, we generate 50 random training/test splits for each data set. The survival prediction methods are applied to the training

sets, and the test sets are used for the calculation of the evaluation measures.

Figures 1, 2, and 3 present box plots of the results for each evaluation measure. The box plots are grouped by method, with three box plots (corresponding to the three real-life data sets) per method. The coding of the methods underneath the box plots is explained in Tables 2, 3, and 4 (given in the Appendix). These tables also contain the three evaluation criteria for each considered method and each data set. The median and IQR are given to match the characteristic features of the box plots.

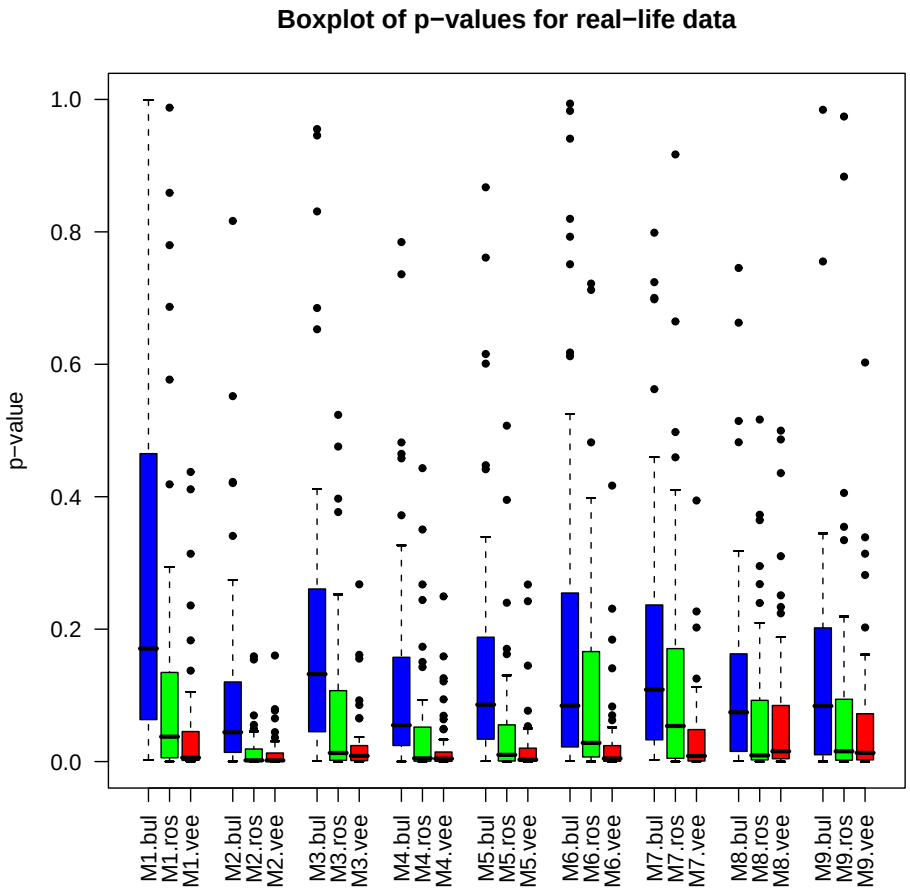

Fig. 1. Box plot of results for the real-life data sets: p-value.

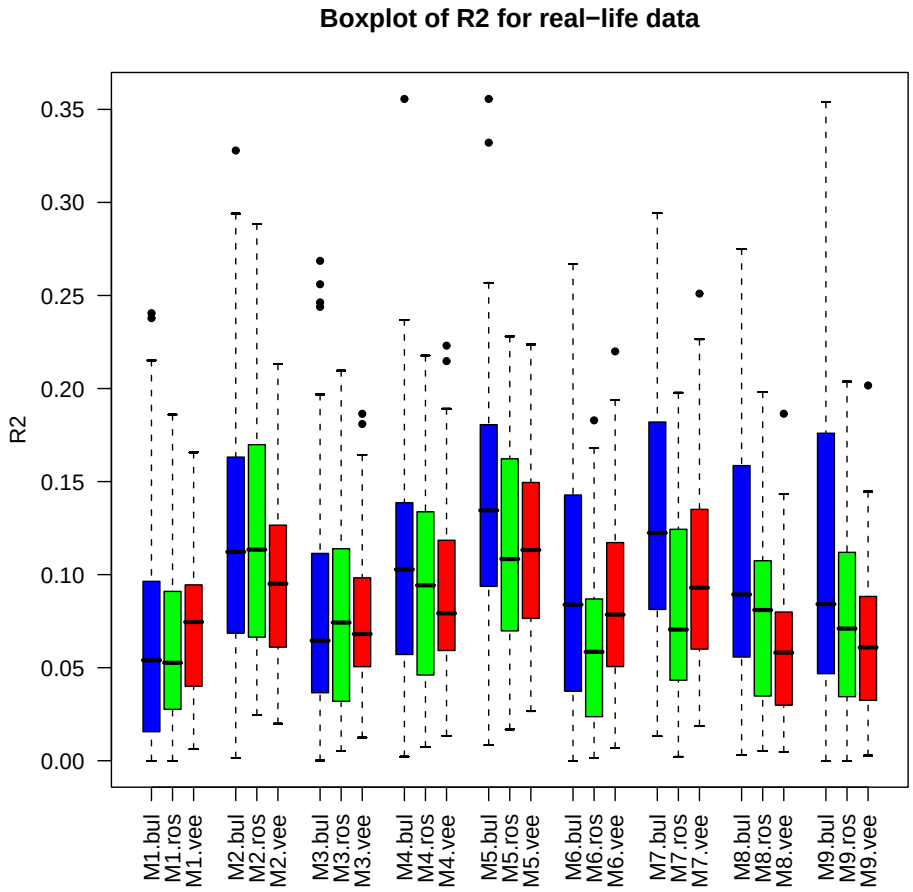

Fig. 2. Box plot of results for the real-life data sets: R^2.

A small p-value indicates a good performing method. It can be seen from Figure 1 and Table 2 that the L_2-penalized Cox regression has the lowest p-values for all three data sets. Moreover, its p-values also have the lowest spread of all methods. The L_2-penalized Cox regression is closely followed by partial Cox regression with one component. Most methods outperform the simple Cox regression with univariate gene selection in all three data sets. Some exceptions are the bagging and random forest methods which performs

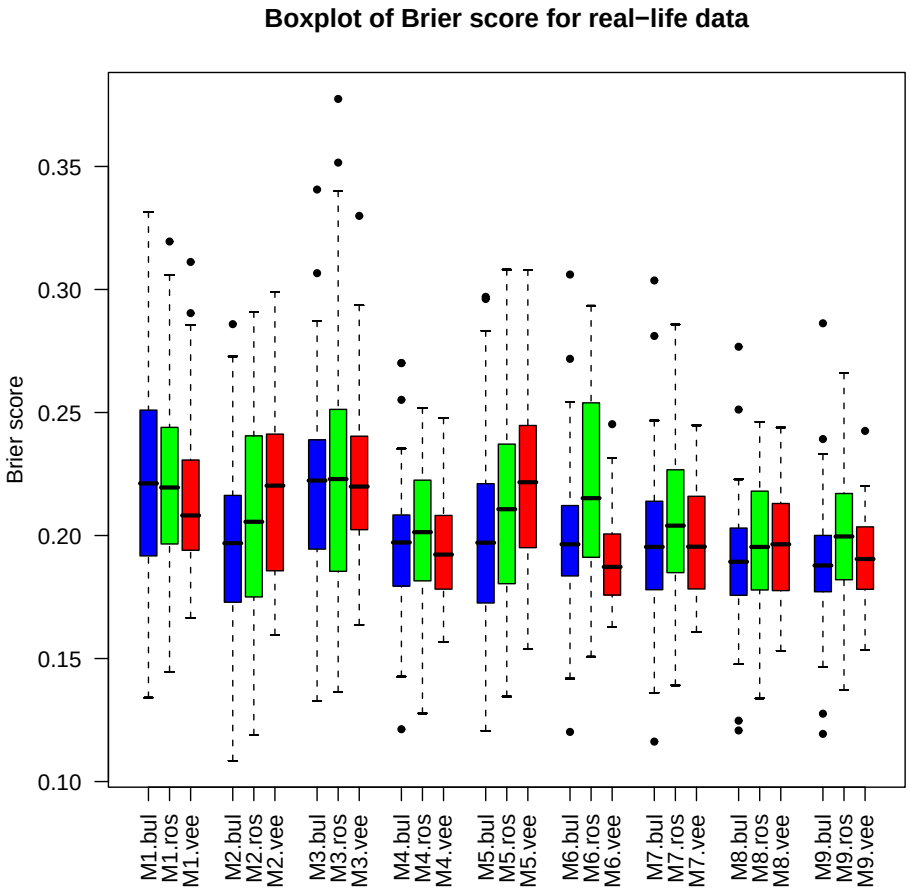

Fig. 3. Box plot of results for the real-life data sets: Brier score.

worse in the Van 't Veer breast cancer data set, and the supervised principal component method in Rosenwald's DLBCL data set. Both the partial Cox regression and the supervised principal component method perform mostly better with one component than with two.

Good survival prediction methods explain much variation, i.e. have a high coefficient of determination R^2. It can be seen that, depending on the data set, the L_2-penalized Cox regression and the partial Cox regression have the highest R^2 values. Most methods explain more variation than the simple Cox regression with univariate gene selection. Some exceptions are the L_1-penalized Cox regression and (again) the bagging and random forest methods which yield

a lower R^2 in the Van 't Veer breast cancer data set.

The better the method's survival prediction, the smaller its Brier score. Both the box plots in Figure 2 and Table 4 indicate that the tree-based ensemble methods bagging and random forest have the smallest Brier score, with random forest also having the smallest IQR. Surprising is the performance of the L_1 penalized Cox regression. It performs worse than the Cox regression with univariate gene selection for all data sets. The L_2 penalized Cox regression and two component partial Cox regression are also outperformed by the simple Cox regression in the Van 't Veer breast cancer data set.

4 Discussion and conclusion

We have given an inventory of methods that have been proposed to predict survival time using gene expression. We have reviewed them critically, and compared them in a qualitative manner. Their performance was assessed using real-life data sets, enabling a quantitative comparison of the methods.

The conclusion from the quantitative comparison are not clear cut. The best method varies depending on the data set and on the considered evaluation measure. Based on the real-life data sets results with the p-value and the R^2 coefficient we conclude that the L_2 penalized Cox regression performs best. This is in line with the findings presented in [12]. However, the results with the Brier score indicate that the ensemble methods, bagging and in particular random forest, are best. It is not surprising that random forests and bagging are comparatively better with the Brier score than with other criteria. For the other criteria, we had to use the median of the predicted Kaplan-Meier curve

as predictor in a Cox model, which is somewhat disputable and overlooks a part of the information yielded by the prediction method.

Both the simple Cox regression with univariate gene selection and L_1 penalized Cox regression select features from the gene expression data (as opposed to feature extraction). This gives their features the benefit of a clear interpretation, which is missing in the other methods. However, all evaluation measures indicate that the former perform worse than methods with feature extraction. This leads us to believe that survival prediction methods benefit from aggregation of gene expressions. The extracted features may get a clearer interpretation if the aggregation is steered by biological principles. This may even improve their predictive power.

Our comparison used statistical arguments to assess the best method for survival prediction. Practical arguments may also prevail, in particular if there is only a marginal difference between the methods' performance. For instance, if the ultimate goal is to design a diagnostic chip, a cost argument will favor the simplest model as it leads to the smallest number of genes on the chip. Hence, methods like LASSO that have an in-built feature selection may be favored over methods like Ridge Regression.

The presented comparison also leads to recommendations for future comparisons. The best survival prediction method varies with the data set and the evaluation measure. Therefore, future comparisons should include multiple data sets, with varying characteristics such as tissue type, microarray platform, sample size, et cetera. Survival prediction methods should perform well over all conditions. Also, several evaluation measures should be used until a consensus is reached on the choice of the criterion for assessing the predictive

performance.

Acknowledgements

This work was partly supported by the Center for Medical Systems Biology (CMSB) established by the Netherlands Genomics Initiative / Netherlands Organization for Scientific Research (NGI/NWO) and by the Porticus Foundation in the context of the International School for Technical Medicine and Clinical Bioinformatics. The authors thank Jelle Goeman for kindly providing the R code of the L_2-penalized Cox regression, and Martin Daumer for carefully reading a preliminary version of the manuscript.

References

[1] Alizadeh, A.A., Eisen, M.B., Davis, R.,E., Ma, C. Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson, J., Lu, L., Lewis, D.B., TibshiraniR., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, L.M., 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511.

[2] Alter, O., Brown, P.O., Botstein, D., 2000. Singular value decomposition for genome-wide expression data processing and modeling. PNAS 97, 10101–10106.

[3] Bair, E., Tibshirani, R., 2004. Semi-Supervised Methods to Predict Patient Survival from Gene Expression Data. PLoS Biology 2, 511–522.

[4] Bair, E., Hastie, T., Paul, D., Tibshirani, R., 2006. Prediction by supervised principal components. Journal of the American Statistical Association 101, 119–137.

[5] Barlow, W.E., Prentice, R.L., 1988. Residuals for relative risk regression. Biometrika 75, 65–74.

[6] Bastien, P., 2004. PLS-Cox model: Application to gene expression. In: COMPSTAT 2004, Section: Partial Least Squares.

[7] Bastien, P., Vinzi, E., Tenenhaus, M., 2005. PLS generalised linear regression. Computational Statistics and Data Analysis 48, 17–46.

[8] Boulesteix, A.-L, 2006. Reader's reaction to "Dimension reduction for classification with microarray gene expression data" by Dai et al (2006). Statistical Applications in Genetics and Molecular Biology 5, 16.

[9] Boulesteix, A.L., 2007. WilcoxCV: An efficient R package for variable selection in cross-validation. Bioinformatics 23, 1702-1704.

[10] Boulesteix, A.L., Strimmer, K., 2007. Partial Least Squares: A versatile tool for the analysis of high-dimensional genomic data. Briefings in Bioinformatics 8, 24–32.

[11] Boulesteix, A.L., Strobl, C., Augustin, T., Daumer, M., 2008. Evaluating microarray-based classifiers: an overview. Cancer Informatics 3, 77-97.

[12] Bovelstad, H.M., Nygard, S., Storvold, H.L., Aldrin, M., Borgan, O., Frigessi, A., Lingjaerde, O.C., 2007. Predicting survival from microarray data – a comparative study. Bioinformatics, doi:10.1093/bioinformatics/btm305.

[13] Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and regression trees. Wadworth, San Diego.

[14] Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

[15] Brown, P.O., Botstein. D., 1999. Exploring the new world of the genome with DNA microarrays. Nature Genetics 21, 33–37.

[16] Bullinger, L., Döhner, K., Bair, E., Fröhling, S., Schlenk, R.F., Tibshirani, R., Döhner, H., Pollack, J.R., 2004. Use of Gene-Expression Profiling to Identify Prognostic Subclasses in Adult Acute Myeloid Leukemia. New England Journal of Medicine 350, 1605–1616.

[17] Butler, N.A., Denham, M.C., 2000. The peculiar shrinkage properties of partial least squares regression. Journal of the Royal Statistical Society B 62, 585–593.

[18] Cox, D., 1972. Regression models and life tables (with discussion). Journal of the Royal Statistical Society, Series B, 74, 187–220.

[19] Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97, 77–87.

[20] Dupuy, A., Simon, R.M., 2007. Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. Journal of the National Cancer Institute 99, 147–157.

[21] Goeman, J.J., Oosting, J., Cleton-Jansen, A.M., Anninga, J.K., Van Houwelingen, H.C., 2005. Testing association of a pathway with survival using gene expression data. Bioinformatics 21, 1950–1957.

[22] Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M., 1999. Assessment and comparison of prognostic classification schemes for survival data. Statistics in Medicine 18, 2529–2545.

[23] Gui, J., Li, H., 2005. Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. Bioinformatics 21, 3001–3008.

[24] Hastie, T., Tibshirani, R., Botstein, D., Brown, P., 2001. Supervised harvesting of expression trees. Genome Biology 2, 1–12.

[25] Hastie, T., Tibshirani, R., 2004. Efficient Quadratic Regularization for Expression Arrays. Biostatistics 5, 329–340.

[26] Hothorn, T., Benner, A., Lausen, B., Radespiel-Tröger, M., 2004. Bagging survival trees. Statistics in Medicine 23, 77–91.

[27] Hothorn, T., Bühlmann, P., Dudoit, S., Molinaro, A., Van der Laan, M., 2006. Survival ensembles. Biostatistics 7, 355–373.

[28] Hothorn, T., Hornik, K., Zeileis, A., 2006. Unbiased recursive partitioning: a conditional inference framework. Journal of Computational and Graphical Statistics 15, 651–674.

[29] Ioannidis, J.P., 2005. Microarrays and molecular research: noise discovery. The Lancet 365, 488–492.

[30] Jenssen, T.K., Kuo, W.P., Stokke, T., Hovig, E., 2002. Associations between gene expressions in beast cancer and patient survival. Human Genetics 111, 411–420.

[31] Kaderali, L., Zander, T., Faigle, U., Wolf, J., Schultze, J.L., Schrade R., 2006. CASPAR: a hierarchical bayesian approach to predict survival times in cancer from gene expression data. Bioinformatics 22, 1495–1502.

[32] Keles, S., Segal, M.R., 2002. Residual-based tree-structured survival analysis. Statistics in Medicine, 21, 313–326.

[33] Klein, J.P., Moeschberger, M.L., (2003). *Survival analysis: techniques for censored and truncated data*. New York, NY: Springer.

[34] Krämer, N., 2007. An overview on the shrinkage properties of partial least squares regression. Computational Statistics 22, 249–273.

[35] Lehmann, E.L., 1986. Statistical Hypothesis Testing. 2nd Edition. New York: Springer.

[36] Li, L., Li, H., 2004. Dimension reduction methods for microarrays with application to censored survival data. Bioinformatics 20, 3406–3412.

[37] Li, H., Luan, Y., 2004. Kernel Cox model for relating gene expression profiles to censored survival data. Pacific Symposium on Biocomputing 8, 65–76.

[38] Li, H., Gui, J., 2004. Partial Cox regression for high-dimensional microarray gene expression data. Bioinformatics 20, i208–i215.

[39] Liu, H., Li, J., Wong, L., 2004. Use of extreme patient samples for outcome from gene expression data. Bioinformatics 21, 3377–3384.

[40] Ma, S., 2007. Principal component analysis in linear regression survival model with microarray data. Journal of Data Science 5, 183–198.

[41] Martens, H., 2001. Reliable and relevant modelling of real world data: a personal account of the development of PLS regression. Chemometrics and Intelligent Laboratory Systems 58, 85–95.

[42] Nagelkerke, N.J.S., 1991. A note on a general definition of the coefficient of determination. Biometrika 78, 691–692.

[43] Nguyen, D.V., Arpat, A.B., Wang, N., Carroll, R.J., 2002. DNA Microarray Experiments: Biological and Technological Aspects. Biometrics 58, 701–717.

[44] Nguyen, D.V., Rocke, D.M., 2002. Partial least squares proportional hazard regression for application to DNA microarray survival data. Bioinformatics 18, 1625–1632.

[45] Nguyen, D.V., 2005. Partial least squares dimension reduction for microarray gene expression data with a censored response. Mathematical Biosciences 193, 119–137.

[46] Park, P.J., Tian, L., Kohane, I.S., 2002. Linking Expression Data with Patient Survival Times Using Partial Least Squares. Bioinformatics 18, S120–S127.

[47] Park, M.Y., Hastie, T., 2006. L_1 Regularization Path Algorithm for Generalized Linear Models. Technical Report, Stanford University.

[48] Pawitan, Y., Bjohle, J., Wedren, S., Humphreys, K., Skoog, L., Huang, F., Amler, L., Shaw, P., Hall, P., Bergh, J., 2004. Gene expression profiling for prognosis using Cox regression. Statistics in Medicine 23, 1767–1780.

[49] Radespiel-Tröger, M., Rabenstein, T., Schneider, H.T., Lausen, B., 2003. Comparison of tree-based methods for prognostic stratification of survival data. Artifical Intelligence in Medicine 28, 323–341.

[50] Rajicic, N., Finkelstein, D.M., Schoenfeld, D.A.., 2006. Survival analysis of longitudinal microarrays. Bioinformatics 22, 2643–2649.

[51] Rosenwald, A., Wright, G., Chan, W.C., Connors, J.M., Campo, E., Fisher, R.I., Gascoyne, R.D., Muller-Hermelink, H.K., Smeland, E.B., Giltnane, J.M., Hurt, E.M., Zhao, H., Averett, L., Yang, L., Wilson, W.H., Jaffe, E.S., Simon, R., Klausner, R.D., Powell, J., Duffey, P.L., Longo, D.L., Greiner, T.C., Weisenburger, D.D., Sanger, W.G., Dave, B.J., Lynch, J.C., Vose, J., Armitage, J.O., Montserrat, E., Lopez-Guillermo, A., Grogan, T.M., Miller, T.P., LeBlanc, M., Ott, G., Kvaloy, S., Delabie, J., Holte, H., Krajci, P., Stokke, T., Staudt, L.M., 2002. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. New England Journal of Medicine 346, 1937–1947.

[52] Royston, P., Altman, D.G., Sauerbrei, W., 2006. Dichotomizing continuous predictors in multiple regression: a bad idea. Statistics in Medicine 25, 127–141.

[53] Schumacher, M., Binder, H., Gerds, T., 2007. Assessment of survival prediction models based on microarray data. Bioinformatics, Advance Access, doi:10.1093/bioinformatics/btm232.

[54] Segal, M.R., 1998. Regression trees for censored data. Biometrics 48, 35–47.

[55] Segal, M.R., Dahlquist, K.D., Conklin, B.R., 2003. Regression approaches for microarray data analysis. Journal of Computational Biology 10, 961–980.

[56] Segal, M., 2006. Microarray gene expression data with linked survival phenotypes: diffuse large B-cell lymphoma revisited. Biostatistics 7, 268–285.

[57] Statnikov, A., Aliferis, C.F., Tsamardinos, I., Hardin, D., Levy, S., 2005. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 21, 631–643.

[58] Tadesse, M.G., Ibrahim, J.G., Gentleman, R., Chiaretti, S., Ritz, J., Foa, R., 2005. Bayesian error-in-variable survival model for the analysis of GeneChip arrays. Biometrics 61, 488–497.

[59] Tibshirani, R., 1997. The LASSO method for variable selection in the Cox model. Statistics in Medicine 16, 385–395.

[60] Van Houwelingen, H.C., Bruinsma, T., Hart, A.A.M., Van 't Veer, L.J., Wessels, L.F.A., 2006. Cross-validated Cox regression on microarray gene expression data. Statistics in Medicine 25, 3201–3216.

[61] Van 't Veer, L.J., Dai, H., Van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., Van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., Friend, S.H., 2002. Gene expression profiling predicts clinical outcome of breast cancer. Nature 415, 530–536.

[62] Verweij, P.J.M., Van Houwelingen, H.C., 1993. Cross-validation in survival analysis. Statistics in Medicine 12, 2305–2314.

[63] Xu, J., Yang, Y., Ott, J. 2005. Survival analysis of microarray expression data by transformation models. Computational Biology and Chemistry 29, 91–94.

[64] Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B, 67, 301–320.

Appendix: tables of results

Method	Method coding	Data set name	Data set coding	Median	IQR
Cox regression with 10 best genes	M1	Bullinger	bul	0.171	0.387
Cox regression with 10 best genes	M1	Rosenwald	ros	0.038	0.124
Cox regression with 10 best genes	M1	Van 't Veer	vee	0.006	0.041
L_2-penalized Cox regression	M2	Bullinger	bul	0.044	0.103
L_2-penalized Cox regression	M2	Rosenwald	ros	0.002	0.018
L_2-penalized Cox regression	M2	Van 't Veer	vee	0.002	0.012
L_1-penalized Cox regression	M3	Bullinger	bul	0.132	0.206
L_1-penalized Cox regression	M3	Rosenwald	ros	0.013	0.092
L_1-penalized Cox regression	M3	Van 't Veer	vee	0.009	0.022
Partial Cox regression (1 comp)	M4	Bullinger	bul	0.055	0.132
Partial Cox regression (1 comp)	M4	Rosenwald	ros	0.005	0.050
Partial Cox regression (1 comp)	M4	Van 't Veer	vee	0.004	0.013
Partial Cox regression (2 comp)	M5	Bullinger	bul	0.086	0.145
Partial Cox regression (2 comp)	M5	Rosenwald	ros	0.010	0.053
Partial Cox regression (2 comp)	M5	Van 't Veer	vee	0.003	0.019
Super PCA (1 comp)	M6	Bullinger	bul	0.084	0.220
Super PCA (1 comp)	M6	Rosenwald	ros	0.028	0.156
Super PCA (1 comp)	M6	Van 't Veer	vee	0.005	0.022
Super PCA (2 comp)	M7	Bullinger	bul	0.109	0.196
Super PCA (2 comp)	M7	Rosenwald	ros	0.054	0.155
Super PCA (2 comp)	M7	Van 't Veer	vee	0.008	0.044
Bagging (100 trees)	M8	Bullinger	bul	0.074	0.146
Bagging (100 trees)	M8	Rosenwald	ros	0.009	0.088
Bagging (100 trees)	M8	Van 't Veer	vee	0.016	0.077
Random forest (100 trees)	M9	Bullinger	bul	0.084	0.190
Random forest (100 trees)	M9	Rosenwald	ros	0.016	0.089
Random forest (100 trees)	M9	Van 't Veer	vee	0.013	0.068

Table 2
Results for the real-life data sets: p-value.

Method	Method coding	Data set name	Data set coding	Median	IQR
Cox regression with 10 best genes	M1	Bullinger	bul	0.054	0.079
Cox regression with 10 best genes	M1	Rosenwald	ros	0.053	0.062
Cox regression with 10 best genes	M1	Van 't Veer	vee	0.075	0.052
L_2-penalized Cox regression	M2	Bullinger	bul	0.112	0.091
L_2-penalized Cox regression	M2	Rosenwald	ros	0.113	0.103
L_2-penalized Cox regression	M2	Van 't Veer	vee	0.095	0.064
L_1-penalized Cox regression	M3	Bullinger	bul	0.065	0.073
L_1-penalized Cox regression	M3	Rosenwald	ros	0.074	0.077
L_1-penalized Cox regression	M3	Van 't Veer	vee	0.068	0.047
Partial Cox regression (1 comp)	M4	Bullinger	bul	0.103	0.081
Partial Cox regression (1 comp)	M4	Rosenwald	ros	0.094	0.086
Partial Cox regression (1 comp)	M4	Van 't Veer	vee	0.079	0.058
Partial Cox regression (2 comp)	M5	Bullinger	bul	0.135	0.083
Partial Cox regression (2 comp)	M5	Rosenwald	ros	0.108	0.092
Partial Cox regression (2 comp)	M5	Van 't Veer	vee	0.113	0.072
Super PCA (1 comp)	M6	Bullinger	bul	0.084	0.100
Super PCA (1 comp)	M6	Rosenwald	ros	0.059	0.062
Super PCA (1 comp)	M6	Van 't Veer	vee	0.078	0.064
Super PCA (2 comp)	M7	Bullinger	bul	0.122	0.097
Super PCA (2 comp)	M7	Rosenwald	ros	0.070	0.079
Super PCA (2 comp)	M7	Van 't Veer	vee	0.093	0.072
Bagging (100 trees)	M8	Bullinger	bul	0.089	0.101
Bagging (100 trees)	M8	Rosenwald	ros	0.081	0.072
Bagging (100 trees)	M8	Van 't Veer	vee	0.058	0.049
Random forest (100 trees)	M9	Bullinger	bul	0.084	0.126
Random forest (100 trees)	M9	Rosenwald	ros	0.071	0.077
Random forest (100 trees)	M9	Van 't Veer	vee	0.061	0.055

Table 3
Results for the real-life data sets: R^2-value.

Method	Method coding	Data set name	Data set coding	Median	IQR
Cox regression with 10 best genes	M1	Bullinger	bul	0.221	0.058
Cox regression with 10 best genes	M1	Rosenwald	ros	0.220	0.045
Cox regression with 10 best genes	M1	Van 't Veer	vee	0.208	0.036
L_2-penalized Cox regression	M2	Bullinger	bul	0.197	0.042
L_2-penalized Cox regression	M2	Rosenwald	ros	0.206	0.063
L_2-penalized Cox regression	M2	Van 't Veer	vee	0.220	0.054
L_1-penalized Cox regression	M3	Bullinger	bul	0.222	0.044
L_1-penalized Cox regression	M3	Rosenwald	ros	0.223	0.064
L_1-penalized Cox regression	M3	Van 't Veer	vee	0.220	0.037
Partial Cox regression (1 comp)	M4	Bullinger	bul	0.197	0.029
Partial Cox regression (1 comp)	M4	Rosenwald	ros	0.201	0.040
Partial Cox regression (1 comp)	M4	Van 't Veer	vee	0.192	0.029
Partial Cox regression (2 comp)	M5	Bullinger	bul	0.197	0.047
Partial Cox regression (2 comp)	M5	Rosenwald	ros	0.211	0.055
Partial Cox regression (2 comp)	M5	Van 't Veer	vee	0.222	0.049
Super PCA (1 comp)	M6	Bullinger	bul	0.196	0.028
Super PCA (1 comp)	M6	Rosenwald	ros	0.215	0.061
Super PCA (1 comp)	M6	Van 't Veer	vee	0.187	0.024
Super PCA (2 comp)	M7	Bullinger	bul	0.195	0.035
Super PCA (2 comp)	M7	Rosenwald	ros	0.204	0.041
Super PCA (2 comp)	M7	Van 't Veer	vee	0.195	0.037
Bagging (100 trees)	M8	Bullinger	bul	0.189	0.027
Bagging (100 trees)	M8	Rosenwald	ros	0.195	0.038
Bagging (100 trees)	M8	Van 't Veer	vee	0.196	0.035
Random forest (100 trees)	M9	Bullinger	bul	0.188	0.022
Random forest (100 trees)	M9	Rosenwald	ros	0.200	0.034
Random forest (100 trees)	M9	Van 't Veer	vee	0.190	0.025

Table 4
Results for the real-life data sets: Brier score.