Mining Nuggets of Activity in High Dimensional Space from High Throughput Screening Data

Yuanyuan Wang Hugh A. Chipman William J. Welch

March 7, 2002

High throughput screening (HTS) is often used in drug discovery to screen large numbers of compounds against a biological target. Statistical analysis of HTS data is aimed at uncovering the relationship between chemical structure, as quantified by various numerical descriptor variables, and biological activity. Estimation of this structure-activity relationship enables prediction of biologically active molecules in huge collections of potential drug compounds. Based on some empirical work with two data sets from the National Cancer Institute, we will compare the predictive performances of some statistical methods. We will also point out some discrepancies between the objectives in drug discovery and classical statistical measures such as deviance or misclassification error.

MSC 2000: Primary 62G08, 62H30; Secondary 62P10, 92C40
Running head: High Throughput Screening Data

1 Introduction:

In the last twenty years, more and more scientists in biology and chemistry have come to believe that the biological activity of a compound is a consequence of its chemical structure (e.g. Livingstone 1995, Section 1.2; King et al. 1992; Klopman 1984). This knowledge has already been applied to the drug discovery process successfully (e.g. Young and Hawkins 1998; Lam 2001 Chapter 3 and Chapter 4).

Biotechnology advances such as newly developed synthesis methods and better assay techniques make it possible to screen tens of thousands to hundreds of thousands of compounds at the early stages of drug discovery. In the main applications of this paper, for instance, two measures of biological activity in protecting human cells from HIV infection were assayed for two databases of about 30,000 compounds each. To find the most promising drug candidates, biochemists would like to examine as many compounds as possible. However, it is impractical to test all of the huge number of compounds potentially available. Research pharmaceutical companies now have up to two million compounds in their databases, and combinatorial chemistry (Service 1996) can potentially generate similar numbers of new compounds. There is a great need to optimize this HTS process. One important approach is to use the data from assayed compounds to relate biological activity (the response) to molecular descriptors of chemical structure (explanatory variables). Uncovering this structure-activity relationship (SAR) helps biologists and chemists make decisions on which compounds are most likely to be highly active, so that they can speed up the searching process (e.g. Jones-Hertzog et al. 2000).

Empirical modeling of the SAR has many challenges. First, although the data generated by HTS may have an enormous number of screened compounds, active compounds are often rare. Second, quantifying a compound's structure is difficult, since there is no natural way to turn the three-dimensional chemical structure into numerical descriptors. Third, it is thought that SAR data inevitably involve threshold and nonlinear effects. Fourth, large amounts of random or systematic measurement errors may be present. Fifth, the screening process itself may produce chemical databases with strong local clustering in the descriptor space. Previous screens aimed at other biological responses may have led to synthesis and hence availability of compounds in concentrated regions of high activity. These regions may or may not be relevant to the current screen.

In this paper, we compare the performances of a few classification methods on a binary-response data set and corresponding methods for continuous data. Then we investigate further the most successful approaches, namely classification and regression trees (Breiman et al. 1984) and K-nearest neighbors (KNN) (see Dasarathy, 1990 for a review). This leads to some understanding of the special features of HTS data.

The paper is structured as follow. Section 2 introduces the data sets we use for illustration. Section

3 gives more details about the modeling methods and compares their performances according to a criterion specific to the efficiency of the drug discovery process. Sections 4 and 5 investigate the most successful methods (trees and KNN) further and shows that typical goodness-of-fit criteria may be inappropriate for HTS data. Section 6 summarizes our findings and discusses future work.

2 The data and objectives

2.1 NCI data

For illustration, we will use two data sets on AIDS anti-viral screening from the National Cancer Institute (NCI) chemical data base. Both data sets include about 30,000 compounds. One of them has a categorical response measuring how a compound protects human CEM cells from HIV-1 infection; the other has a continuous response recording the concentration of the compound that gives 50% protection on infected cells. The two studies investigate different biological targets, but most compounds (over 75%) are common to the two datasets. The compound activities are in the public domain (**http://dtp.nci.nih.gov/docs/aids/aids_data.html**).

Six chemical descriptor variables were generated by GlaxoSmithKline chemists. They are continuous variables called BCUT numbers (Burden 1989, Pearlman and Smith 1998) which describe the structure of the compounds such as their surface area, bonding patterns, charges, and hydrogen bond donor and acceptor ability. Pearlman and Smith (1999) showed that it may be possible to find fairly low-dimensional (2-D or 3-D) subsets of BCUT variables such that active compounds are clustered in the relevant subspace. Thus, BCUT values are good candidates as explanatory variables for modeling the relationship between activity and structure.

2.1.1 Categorical data

For the categorical-response data, the activity measure for each compound has three levels: 0 (inactive), 1 (moderately active), and 2 (active). The data are very unbalanced: there are 215 active compounds, 393 moderately active compounds and the rest (29,204) are inactive. Figure 1 plots BCUT3 and BCUT4, two variables that we will show to be important in Section 4. Figure 1(a) displays all the active compounds. Octagons and triangles indicate active and moderately active compounds, respectively. Figure 1(b) has the same scale and shows all inactive compounds. Both distributions are very complex. The active compounds are located in regions where there are many inactive compounds. Similar patterns are seen for any pair of descriptors. Thus, no obvious distinct clusters of active regions are apparent by looking at the descriptors two at a time in this simple way. However, there are some promising signals. The density plots in Figure 2 compare the distributions of active, moderately active, and inactive compounds in BCUT space. It shows us that BCUT3 and BCUT4 seem to be related to activity, since the peaks of the density curves for the active compounds stand out.

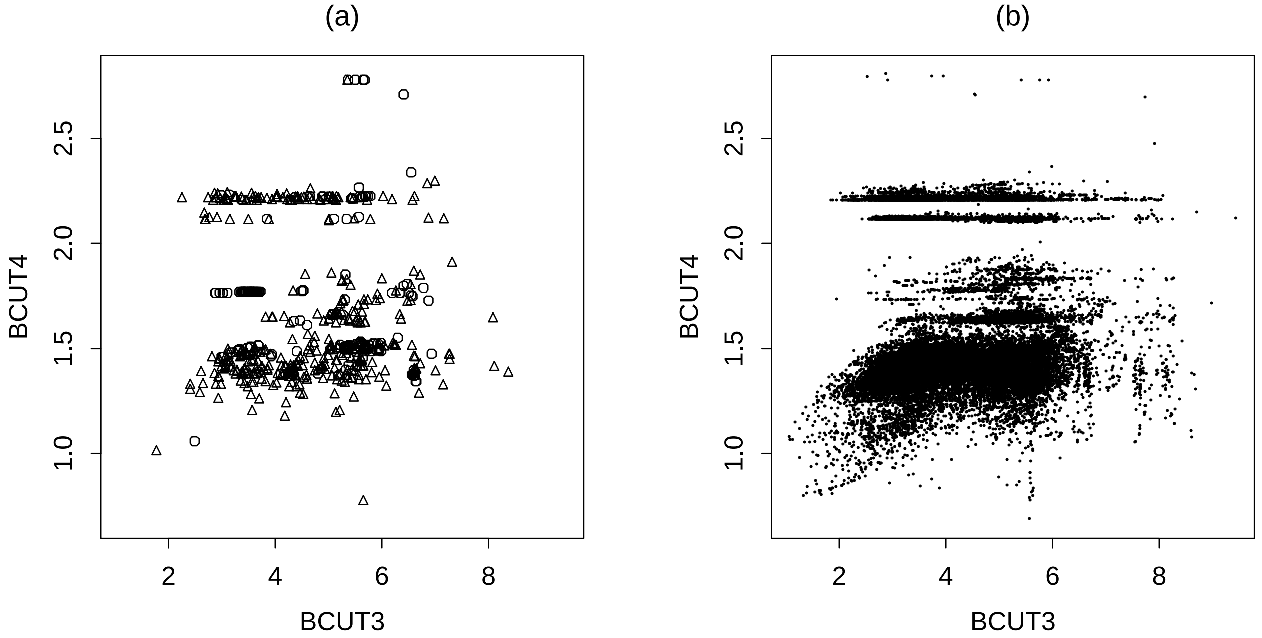


Figure 1: Plot of BCUT3 and BCUT4 values: (a) active compounds, with octagons and triangles indicating active and moderately active compounds, respectively, and (b) inactive compounds.

As some of the methods we investigate are intended for a binary response, and there are relatively few

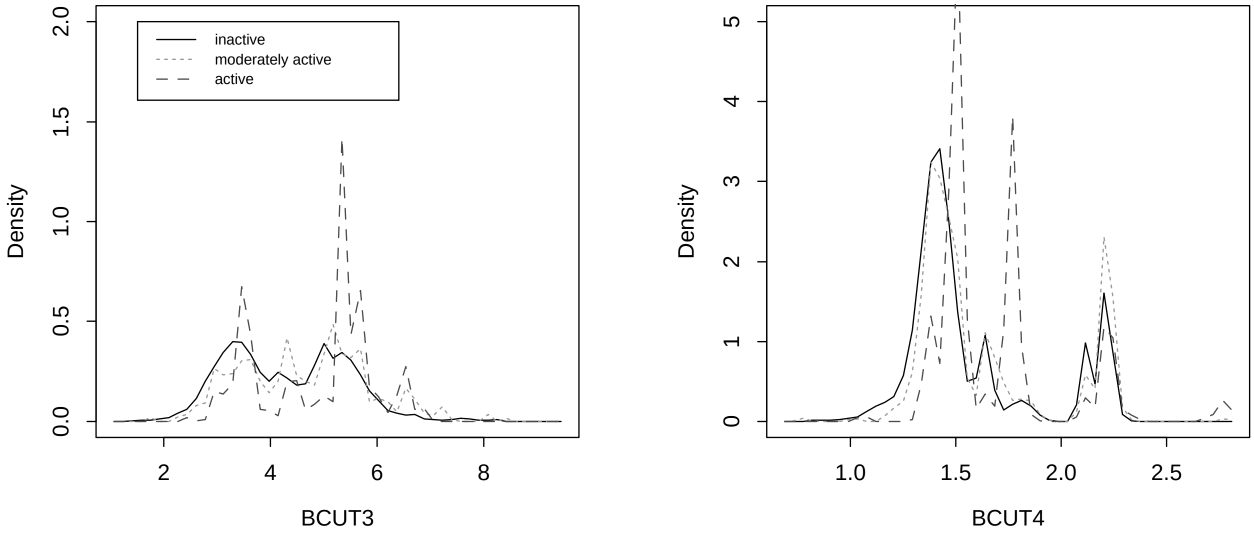


Figure 2: Density of BCUT3 and BCUT4 by activity

compounds in the two active categories, we combined the active and moderately active cases into one group. The resultant data set has 98% inactive and 2% active (608) compounds.

2.1.2 Continuous Data

There are 28596 compounds in total. The dependent variable is defined as the negative logarithm of the compound concentration that protects infected cells by 50% ($-\log(EC50)$). In this setting, larger values indicate more potent compounds. The plots in Figure 3 are grey-scale 2-d density images of the response conditional on each predictor bin. Some obvious outliers in BCUT space were not included. The density of the points is indicated by grey-level and the line is a fitted local regression curve. The flatness of the curves suggests that no single BCUT value is a good predictor.

2.2 Objectives

The object of analyzing such data is to understand the structure-activity relationship (SAR). Specifically, scientists want to use a fitted SAR from a relatively small screen (thousands to tens of thousands of compounds) to guide the selection of further compounds from a database. Recall that these

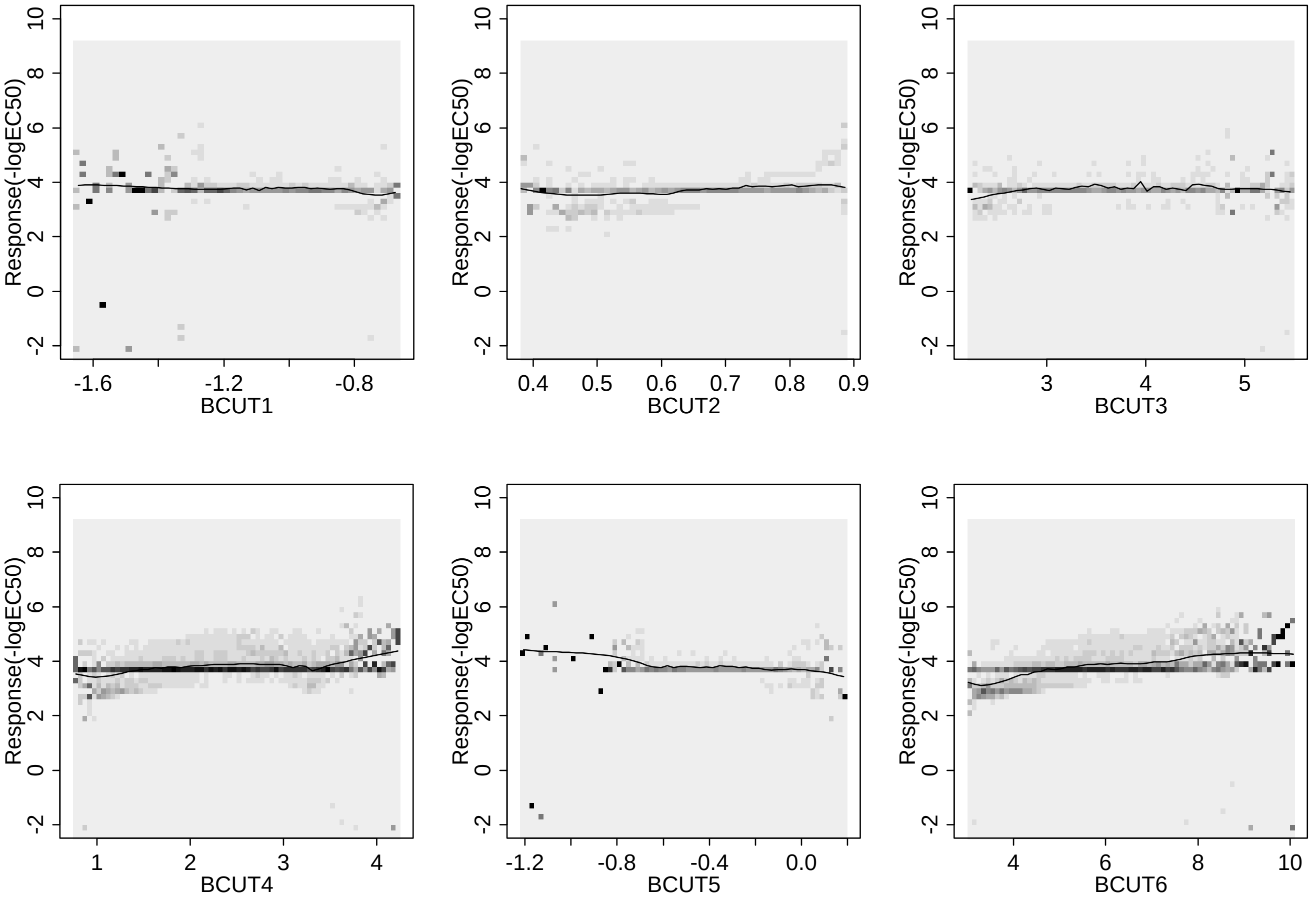


Figure 3: Image plots of the response versus predictors. The curve in each plot represents a fitted local regression.

databases are often huge, but the active compounds are rare, and screening all possible compounds is economically prohibitive. Thus, chemists and biologists want to select and assay a relatively small number of compounds. Further compounds will be chosen based on the fitted SAR. Hopefully, they will include many potent molecules, which will be passed on to the next stage of drug development.

To compare the modeling methods, we randomly divide the data into training (model building) and test (validation) sets of equal size. For categorical data, separate splits are made of the active and inactive compounds, so the training and test sets are both comprised of 304 active compounds and 14602 inactive compounds. For the continuous data, a random split is applied such that the training and test set each has 14298 observations. The training-test split simulates the situation where a limited number of assays are used for model fitting (the training data) and activity is to be predicted for the rest of the collection (test data).

The hit rate (proportion of active compounds or "hits" amongst those selected) is a popular measure (e.g., Tatsuoka, Gu, Sacks, Young, 1998) to evaluate predictive performance for classification models. Furthermore, the hit rate need only be high for a few hundred compounds ranked highest by any method and hence chosen. For instance, a 50% hit rate for 200 compounds selected will generate 100 active compounds. These compounds would usually be examined for their chemical structures. It is desirable to have several "leads" from different chemical classes for further optimization of activity, toxicity analysis, etc. The hit curve of the highest ranked compounds selected by a model depicts the number of active compounds, or hits, versus the number of compounds selected. We will then visually compare the hit curves of the various methods.

Similarly, for continuous data, we use a statistical method to predict activity and select the compounds with the highest predicted activities. We then compare the distributions of measured activity for the compounds selected by the methods.

3 Comparison of the methods

After collapsing the categorical data set to two inactive/active categories, we have binary-response data. The $-log(EC50)$ response is continuous. Most of the methods we listed below have versions for both types of response data.

Regression models For the binary data, we consider a Logistic Regression Model (LRM) on the six BCUT numbers. LRM is a special case of generalized linear models and it is a popular model for binary response data. It assumes

$$\log\left(\frac{p}{1-p}\right) = b_0 + b_1 BCUT1 + b_2 BCUT2 + \cdots + b_6 BCUT6, \tag{1}$$

where p indicates the probability that the observed case is active.

For $-\log(EC50)$, a Linear Regression on the six BCUT numbers is used. It assumes

$$E(-\log(EC50)) = b_0 + b_1 BCUT1 + b_2 BCUT2 + \cdots + b_6 BCUT6. \tag{2}$$

GAM In either logistic regression or linear regression, each BCUT predictor can be replaced by a smooth function of the BCUT, yielding a generalized additive model (Hastie and Tibshirani 1990). These smooth functions have 4 degrees of freedom by default in the S implementation and are estimated from the training data.

NN Neural networks (see Ripley 1993 and 1996 for general references) are very flexible and can be useful when we pursue good prediction instead of interpretation. Specifically, a feed-forward neural network (Venables and Ripley 1999, p296-302) with one hidden layer (9 hidden units) is implemented here. It is the simplest but most common form of neural network. For the classification problem, the response is the estimated probability of being active.

MARS (Multivariate Adaptive Regression Splines) A multiple regression function is constructed from linear splines and their tensor products (Friedman 1991). Generalized cross validation (GCV), a kind of residual sum of squares of the model penalized by the model size, is used to select the best candidate model. The default POLYMARS implementation in R (Kooperberg 1997) is used.

Tree The S-Plus implementation (Clark and Pregibon 1991), which is similar to CART (Breiman et. al. 1984), is used. Using binary recursive partition, a tree successively splits the data along coordinate axes of the predictors such that at each division, the resulting two subsets of data are as homogeneous as possible with respect to the response of interest. At each step in the construction algorithm, an optimal split is identified. This local optimality does not guarantee that the optimal tree will be found.

For the categorical data we use classification trees (see Section 4 for details); regression trees are built for the continuous data. The deviance is used as a splitting criterion. Initially, we work with the tree generated by the S-Plus defaults: a node must have at least 10 observations and its deviance at least 1% of the root node deviance to be split. Pruning the default tree is considered in Section 4.

C4.5: C4.5 (Quinian 1993) is a popular classification method. It also uses recursive partitioning to generate a classification tree from a set of data. A collection of rules is automatically constructed from the tree, removing insignificant conditions. Rules are also simplified by deletion of rules that do not contribute to overall accuracy. Finally, the sets of rules are ordered to minimize false positive errors and a default class is chosen. Therefore, the misclassification rates are taken as the scores to prioritize the derived rules.

KNN: K-nearest neighbors is a very simple but powerful method. The algorithm is as follows: for each observation in the test set, the k nearest points in the training set based on some proximity measure are found and the prediction is made by either majority vote among the selected closest points or the mean of response of the selected closest points. Euclidean distance is the usual proximity measure. The parameter k stands for the number of neighbors we want to examine and can be determined by leave one out cross-validation on the training data. For the classification problem, the votes for the winning class relative to k can be regarded as the probability that the test case is in the predicted class. For the regression problem, the predicted response for a test compound is the average response of its k neighbors.

After dividing the data, we constructed each model based on the training set and evaluated it on the test set. The estimated probability of activity (categorical data) or the estimated response (continuous data) is used as a score to select compounds from the test set. Note that for some methods such as trees, C4.5, and KNN, a large number of ties may be present when ranking the compounds in the test set.

3.1 Comparison of methods for the categorical data

Figure 4 displays the hit curves for one random training-test data split. The horizontal axis represents the number of compounds selected; the vertical axis represents the number of actives actually obtained.

For example, to select 100 compounds using one of the classifiers, every compound in the test data is scored and the 100 highest ranked compounds are chosen. The tree, KNN and C4.5 methods all give about 50 actives out of the 100 selected. Since the rankings provided by these three methods have many ties, their hit curves consist of discrete points connected by straight lines. The points indicate the actual hits we can obtain when groups of tied compounds are simultaneously selected, and the lines provide the expected number of hits in between. Figure 4 shows that the tree model and KNN are the most successful techniques, dominating all the other methods. These methods are more local, flexible and they make minimal assumptions about the underlying relationship. They are able to focus on very

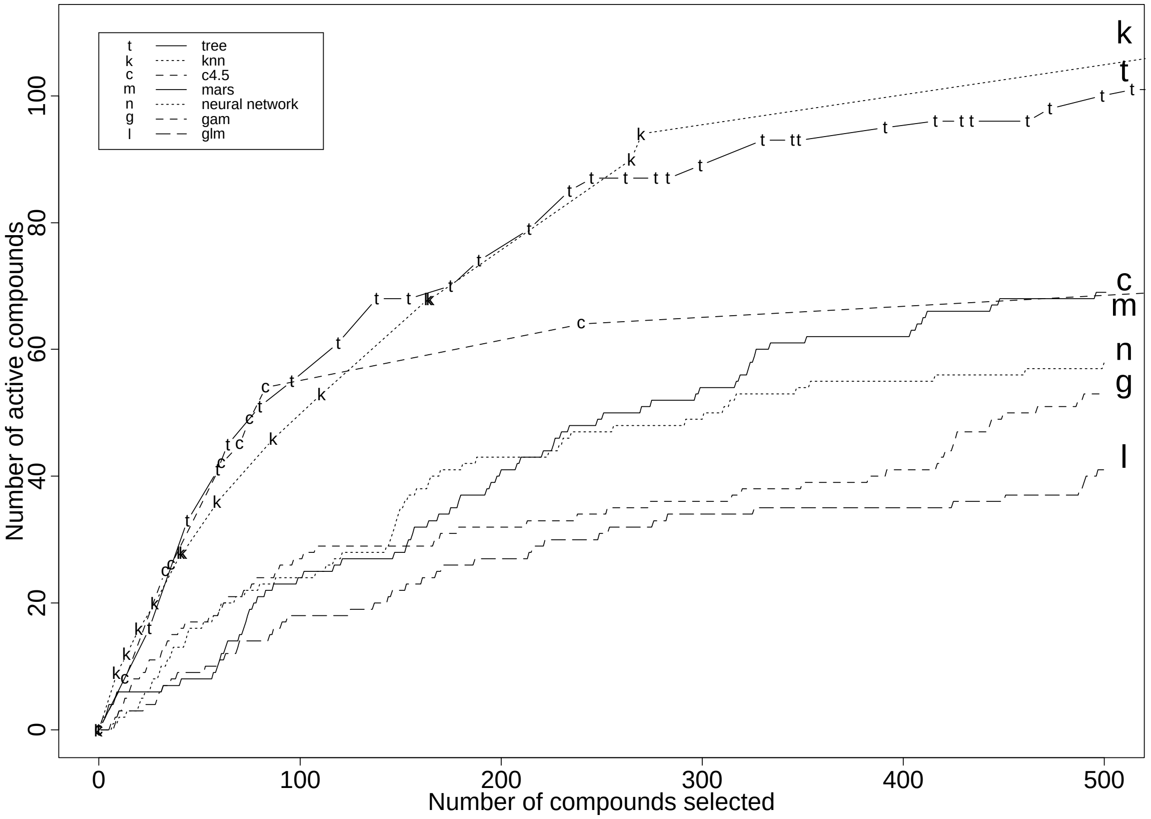


Figure 4: Number of active compounds versus number of compounds selected.

local regions containing concentrations of active compounds (e.g. Hawkins et al. 1997) and capture interactions, and thresholds which often exist in HTS Data (e.g. Young and Hawkins 1998; Pearlman and Smith 1999). The performance of a technique as simple as KNN is impressive. Unlike GLM, GAM, MARS and NN, the tree model and KNN do not assume that the relationship between the probability of biological activity (the response) and the measurable features of the chemical structure (BCUT's) can be approximated by a continuous function.

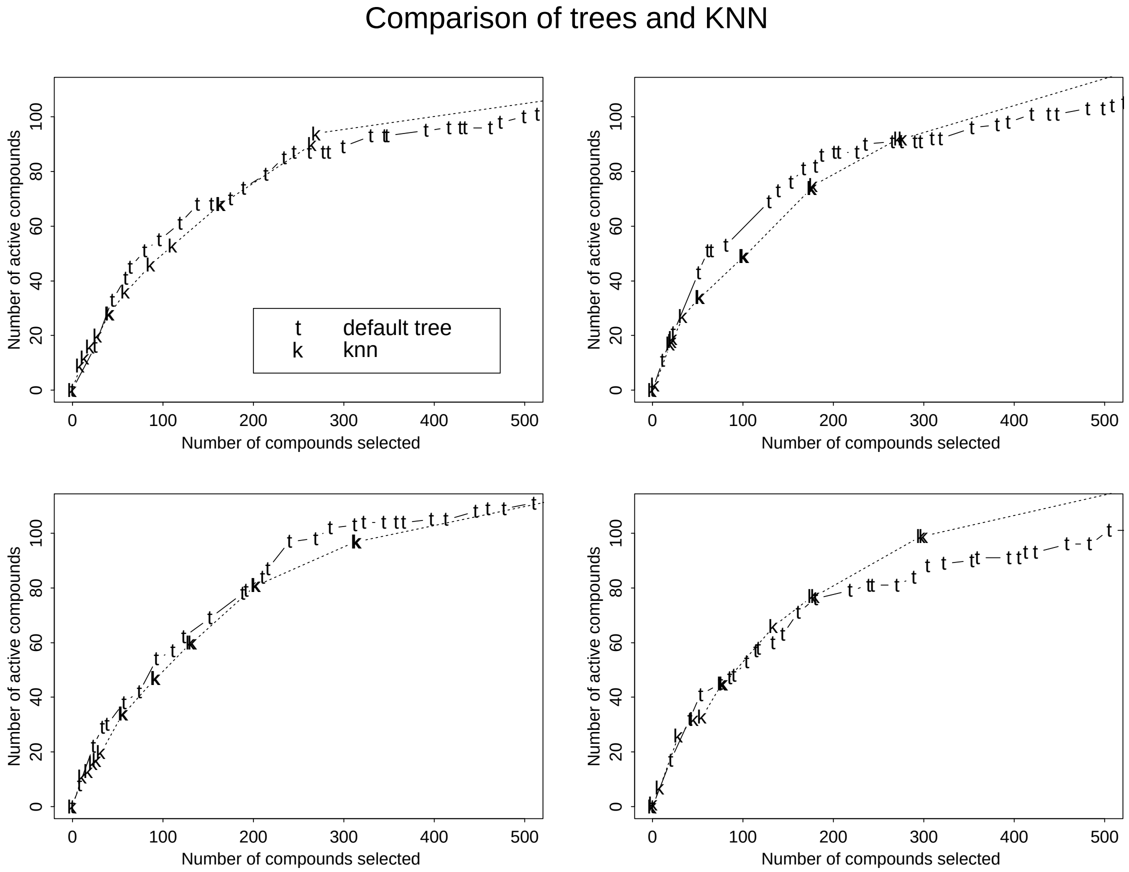


Figure 5: Number of active compounds found by the default tree and KNN, respectively, in the test data versus number of compounds selected based on four random splits.

C4.5 is a method similar to classification trees. It grows the decision tree to produce rules and then simplifies them. Figure 4 shows it is among the best models at the very beginning (selecting less than 100 compounds) but finds few further actives after that. This may be due to the way that C4.5 deletes rules, leaving large number of compounds grouped together with tied scores. This is evident in Figure 4: from 100 to 500 compounds, only two large groups are selected. In Section 4, we show why larger trees with more nodes (rules) perform better when assessed as in Figure 4.

To check whether these results depend on the training/test data split, we randomly split our data four times. Figure 5 gives hit curves for KNN (with the symbols "k") and the default tree model (with the symbols "t") for each split. In terms of the hit curves, trees and KNN are competitive with each other. One deficiency in KNN is that it gives very little usable information regarding how each predictor (the BCUT's here) relates to activity. Ideally, a good classifier not only provides accurate prediction, but also provides some insight into the important chemical features. In this respect, classification trees are good candidates: the important BCUT numbers are used to generate splits during the tree-growing procedure and the most promising terminal nodes of the final tree correspond to small regions of high activity in the BCUT metric space (e.g. Rusinko et al. 1999).

3.2 Comparison of methods for the continuous data

The five quantile plots in Figure 6 summarize the activity distributions of the compounds in the test set that are ranked highest by the various methods. The number of compounds selected is 100, 200, 300, 400, and 500 respectively. As a baseline, a quantile plot of the whole test set is drawn at the left of each plot. The symbols $\times$, $+$, $\triangle$ and $\circ$ indicate the 90%, 75%, 50% and 25% quantile respectively. The points above the quantile curves are the upper 10% of measured activities for the selected compounds. The 90% quantile and the plotted points are two important criteria to evaluate the model performance. The higher the quantile and the more large values, the better the performance. Figure 6 shows the distributions of $-\log(EC50)$ values of selected compounds vary considerably by model type, and the ranking of models by performance is almost the same as for the categorical data set. The ranking from LM, GAM, MARS and NN to Tree and KNN also mirror the complexity ranking of the assumptions that each model demands. As in the classification case, the local methods (trees and KNN) dominate.

Note that only these two methods identified the very best compounds in the test set. Strong local behavior in the continuous data is also indicated by the fact that the optimal k equals 7 in KNN and the optimal tree size is over 150 (Section 4).

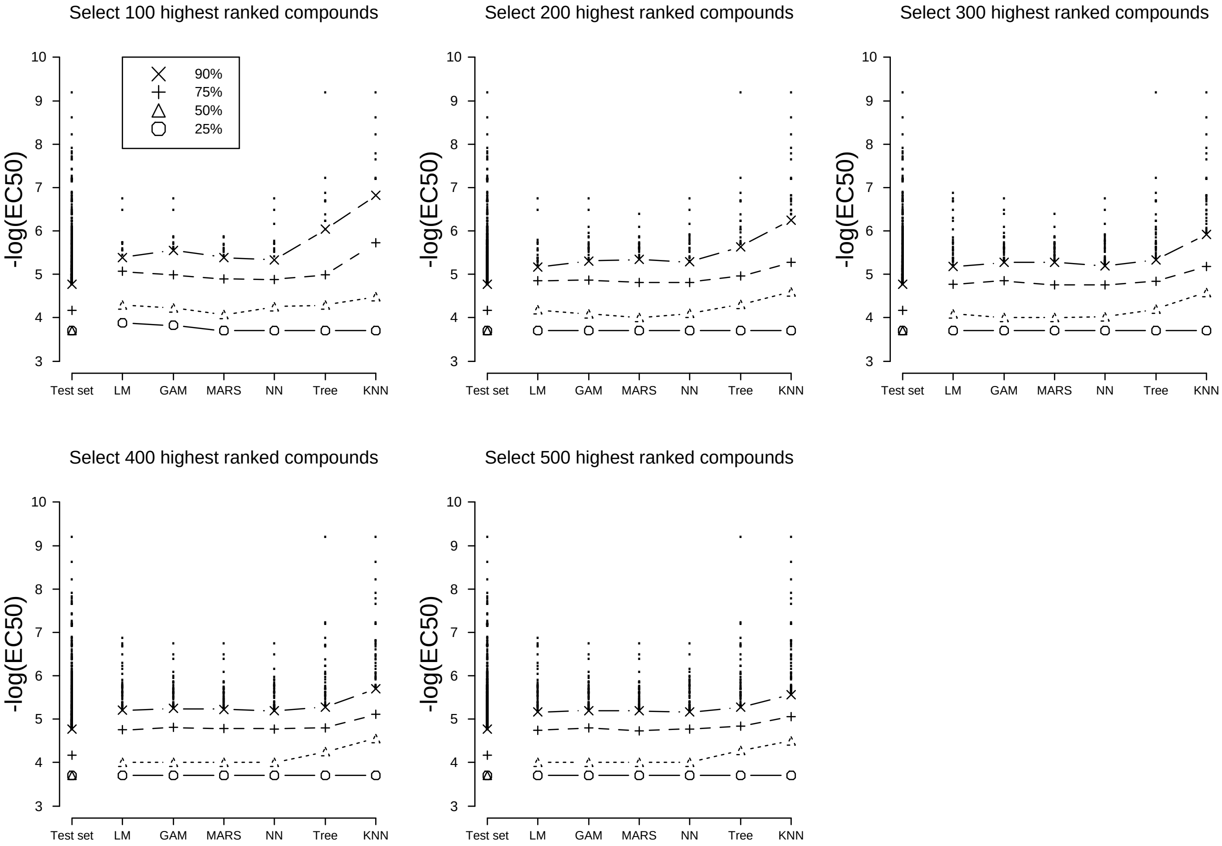


Figure 6: Quantile plots for the selected compounds (the symbols ×, +, △ and ∘ indicate the 90%, 75%, 50% and 25% quantiles respectively). The points above the quantile curves display the upper 10% of measured activities.

Even the best models are selecting some compounds which are very inactive here. One possible reason why the classification rankings are more striking is that there we are treating the bottom 98% of the data as equal, whereas in the regression setting there is a substantial difference between a relatively low score of 2 and 5. The regression models for continuous data may be modeling these unimportant distinctions. We think weighting the more active compounds more heavily is one possible way to get around this. Preliminary unpublished work by Lam shows that weighting gives some improvement, but unresolved issues remain, such as how to choose the proper weights and whether the weights can also apply to other methods (say NN, MARS). Therefore this will be investigated as future research. In the next section, we explore classification trees in detail and understand why tree pruning is ineffective in these contexts.

4 Classification and regression trees

Regression and classification trees are quite similar in concept. Both of them recursively make binary splits of the data along coordinate axes of the predictors such that at each division, the resulting two subsets of data are as homogeneous as possible with respect to the response of interest. The default trees giving the hit curves displayed in Figures 4 and 5 and the regression tree displayed in Figure 6 are constructed using two constraints to stop further splitting:

- there must be at least 10 observations in a node; and
- the node deviance must be at least 1% of the root node deviance.

These constraints are the default options of tree models defined in S-Plus (Clark and Pregibon 1991). To illustrate, Figure 7 depicts the first few nodes of the default tree built on the binary-response training set that leads to the hit curve in Figure 4. The whole training set (14906 compounds with 304 active) in the top node (root) is divided into two subsets using BCUT4. The number inside each node is the predicted class. The fraction below each node indicates the number of cases misclassified relative to the number of cases falling in the node, leading to the estimated hit rate of each node. For instance, 132 compounds in the training set reside in the right-bottom node of this tree. Of them, 25 are active. The estimated hit rate is $\hat{p}$= 25/132 =0.20, and because this is less than 0.5 the compounds in this node will be predicted inactive.

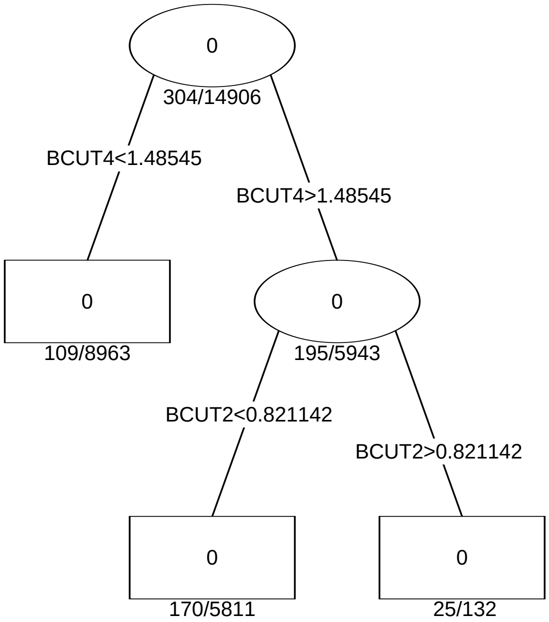


Figure 7: Part of the classification tree fitted to the NCI AIDS anti-viral binary data. The edges connecting the nodes are labeled by the left and right splitting rules. Interior nodes are denoted by ellipses and terminal nodes by rectangles, with the predicted targets centered in the node. The fraction under each node is the number of compounds misclassified relative to the number of compounds in the node.

4.1 Classification trees

4.1.1 Examining the ranking criteria

It is common to take $\hat{p}$ as a score to rank the terminal nodes and prioritize compounds in the test set. However, these scores do not always provide a reliable ranking. For example, suppose we have two terminal nodes, one having 100 compounds with 99 active ($\hat{p} = 0.99$), the other having only one compound which is active ($\hat{p} = 1$). The score for the first node is a much more reliable estimate of the true activity rate, though according to $\hat{p}$ the second node should be ranked first. To account for uncertainty in $\hat{p}$, we assume a Binomial model for responses in each terminal node and calculate a 95% one-sided confidence interval or lower bound, $\hat{p}_{lb}$, for the true hit rate p. $\hat{p}_{lb}$ is based on an exact calculation, rather than a normal approximation. The $\hat{p}_{lb}$ score will be large if $\hat{p}$ is large and there are many compounds in a node (Lam 2001, Chapter 4). The hit curves for the default trees in Figure 4 and Figure 5 results from such scores. They improve the trees' performances considerably. For example, for the data split in Figure 4, at 100 compounds selected, the hit increases from 52.0 to 56.0 expected active compounds.

A ”pure” tree can be constructed by removing the constraints on node size and deviance for the default tree, so that all the terminal nodes have compounds of one class. In many applications, this will lead to over-fitting. To our surprise, in plots analogous to Figure 4 the pure tree performs as well as the default tree for this data set. We believe that large trees perform well because they are able to identify highly localized regions of high activity (nuggets).

To illustrate, we select the best three terminal nodes respectively from the default tree and the pure tree from the training/test split in Figure 4. Table 1 shows the top-ranked nodes (training data) from the two trees. Because the nodes selected from the pure tree are just subsets of those selected from the default tree, we use A, B, C to represent those nodes. In the default tree, for example, 25 cases reside in node C with 22 hits. These 22 hits are all the cases in node C of the pure tree. For the pure tree, the $\hat{p}_{lb}$ criterion reduces to choosing the largest pure active node first.

	Default tree					Pure tree	
Node	$\frac{\#activecomp.}{\#total}$	$\hat{p}$	rank	$\hat{p}_{lb}$	rank	$\frac{\#activecomp.}{\#comp.}$	rank
A	$\frac{13}{14}$	0.93	1	0.70	3	$\frac{13}{13}$	3
B	$\frac{20}{22}$	0.91	2	0.74	1	$\frac{16}{16}$	2
C	$\frac{22}{25}$	0.88	3	0.71	2	$\frac{22}{22}$	1

Table 1: Comparison of the first 3 terminal nodes for two ranking methods, using training set frequencies.

The three nodes shown in Table 1 have very good hit frequencies. Both default and pure trees define almost the same region of the BCUT space. In the test set, for instance, there are 19 compounds in node C for both the default and pure trees. Of them, 17 are active, a hit rate of 89%. Examining the binary splits defining these nodes reveals that there are many splits on BCUT3 and BCUT4. For example, 10 splits lead to node C in the default tree. Of them, 7 split on BCUT3 or BCUT4. The

BCUT metric space corresponding to these nodes is highly localized. The range of BCUT4 in node C, for instance, covers only about 0.3% of the whole range. Therefore, we believe BCUT3 and BCUT4 are very important and activity is highly localized in this subspace. We next explore in more detail the relationship between size of a classification tree (and hence localization) and performance when selecting compounds.

4.1.2 Examining the size of the classification tree

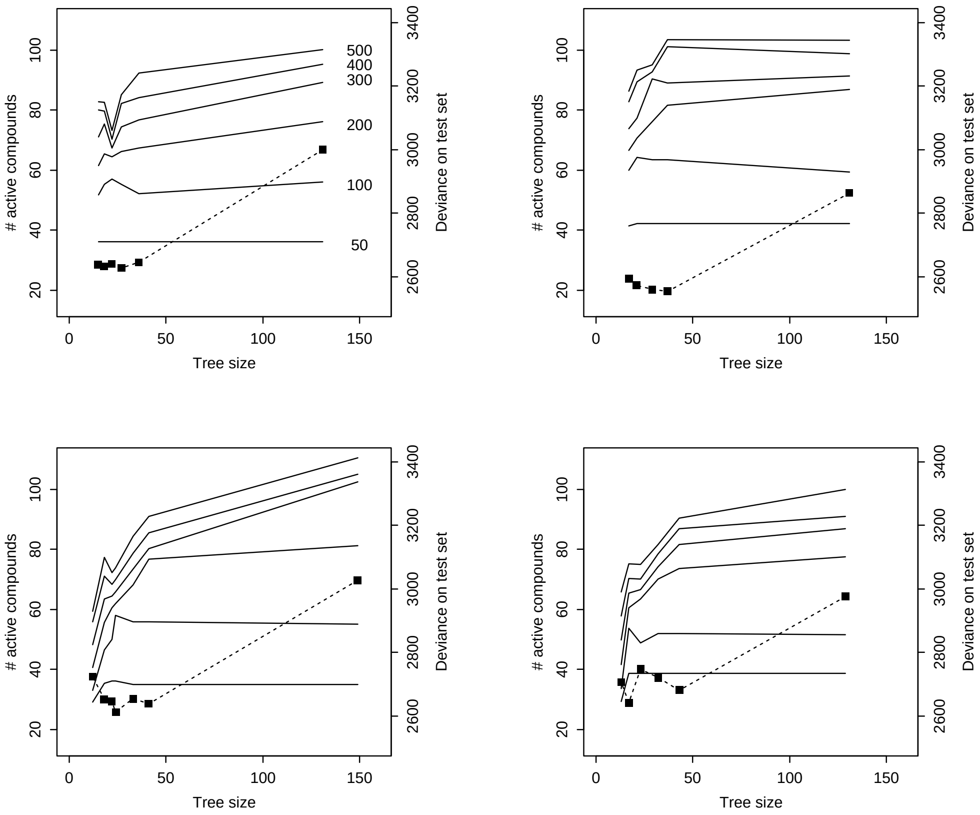


Figure 8: Each plot represents one training/test data split. It shows the number of hits versus the size of the tree for selecting 50, 100, 200, 300, 400, 500 compounds respectively (six straight lines) and the deviance versus tree size (dotted line).

The tree models leading to Figure 4 and Figure 5 are all very large. The default tree in Figure 4, for instance, has 131 terminal nodes. The corresponding pure tree has 393 terminal nodes.

Breiman et al. (1984) suggest a two-step procedure for choosing the size of a tree:

1. Pruning. A common approach is to grow the largest possible tree and then prune it back.

2. Searching amongst the many possible pruned trees using cross-validation (CV) to determine the best tree size. The most popular criteria are misclassification rate or deviance.

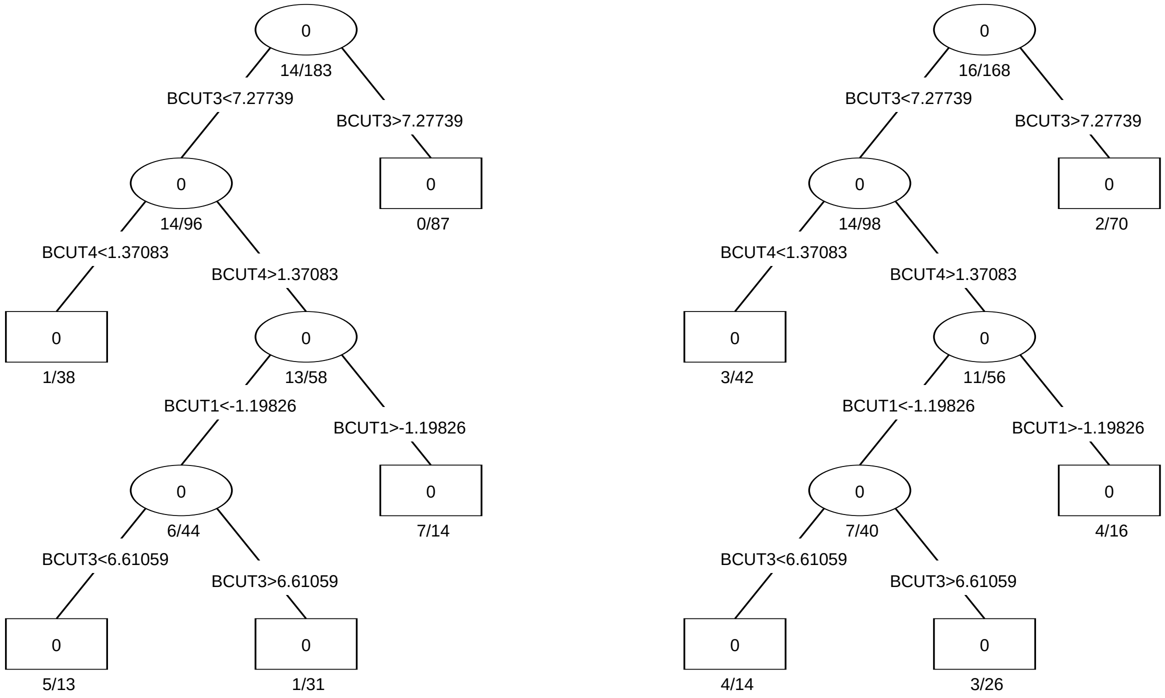


Figure 9: Part of a classification tree fitted to the NCI AIDS anti-viral data. The left one is a subset of the tree built on the training set and the right one is the same tree but evaluated on the test set.

We explore the impact of pruning by computing the hit rate for a fixed number of compounds selected as trees are pruned back. In Figure 8, the default tree is pruned. The four pictures illustrate the results for four random splits (those in Figure 5). For each picture, the six solid lines show how the number of hits in the test set varies with the tree size when selecting 50, 100, 200, 300, 400 or 500 compounds respectively, and the dotted line displays how the deviance evaluated on the test set relates to the tree size. Most of the solid lines are monotone increasing, which indicates that larger trees seem to perform as well as or better than smaller trees. This phenomenon is more obvious when more compounds are selected (say 300 instead of 50). However, the deviance (a smaller-the-better criterion), shown by the dotted lines, suggests small trees (say 20-40 terminal nodes). Similar analysis of the misclassification rate also suggests small trees are optimal. Note that here the deviance and misclassification rate are based on the test set, which would not be possible in practice. Thus, even when these classical criteria for choosing a tree size are evaluated on the compounds we are predicting, they lead to trees that identify fewer active compounds among a focussed selection.

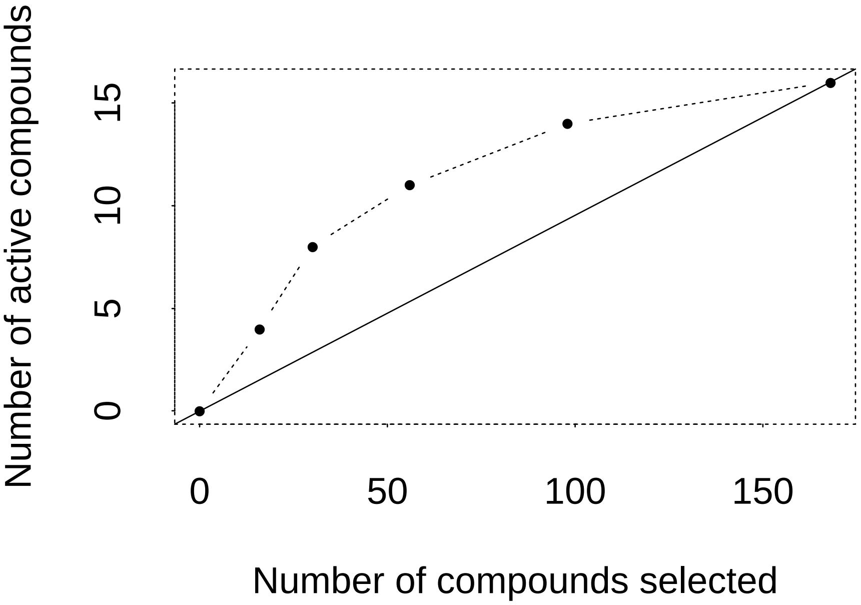


Figure 10: Number of active compounds versus number of compounds selected in the test set for the subtree in Figure 9 (dotted line) and after pruning back to the top node (solid line).

Examining the tree-pruning carefully, we found that some really good terminal nodes may be lost when we prune back a large tree. We now look at a particular node to see why pruning may lead to lower hit rates when selecting compounds. Figure 9 shows a subset of a default tree fitted to the training set (left tree) and evaluated on the test set (right tree). For example, in the training set there are 183 compounds residing in the top node with 14 actives; in the test set there are 168 compounds with 16 actives. During pruning of the default tree, this subtree is reduced to the top node, giving a test hit rate of 16/168 = 0.095, as indicated by the straight line in the hit curve of Figure 10. When the top node is not pruned, as in Figure 9, we can identify some sub-regions with higher hits. For example, the child node having 13 compounds with 4 actives in the training set has 14 compounds with 4 actives in the test set. Its hit rate (4/14 = 0.29) is much higher than the top node (0.095). Therefore, the contribution to hit curve from this subtree indicated by the dotted line in Figure 10 is above the straight line.

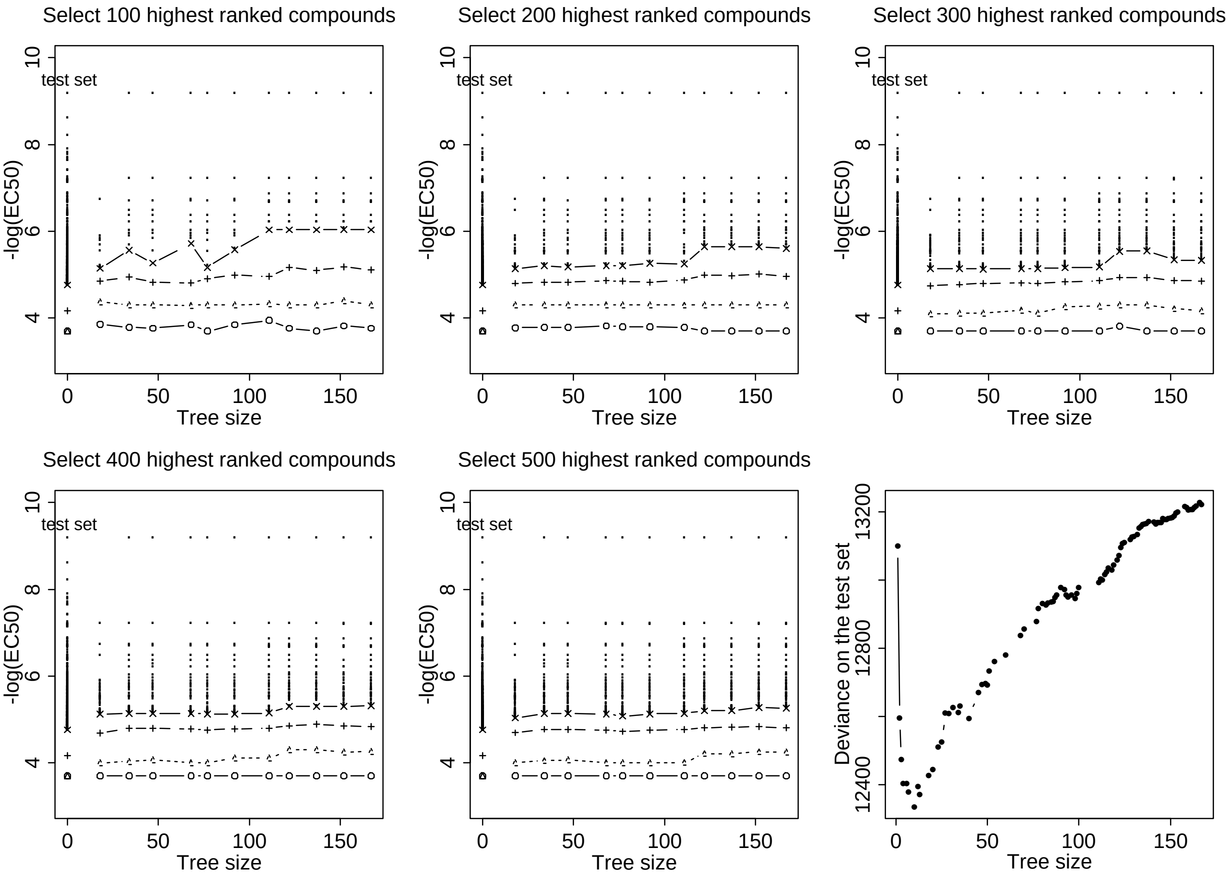
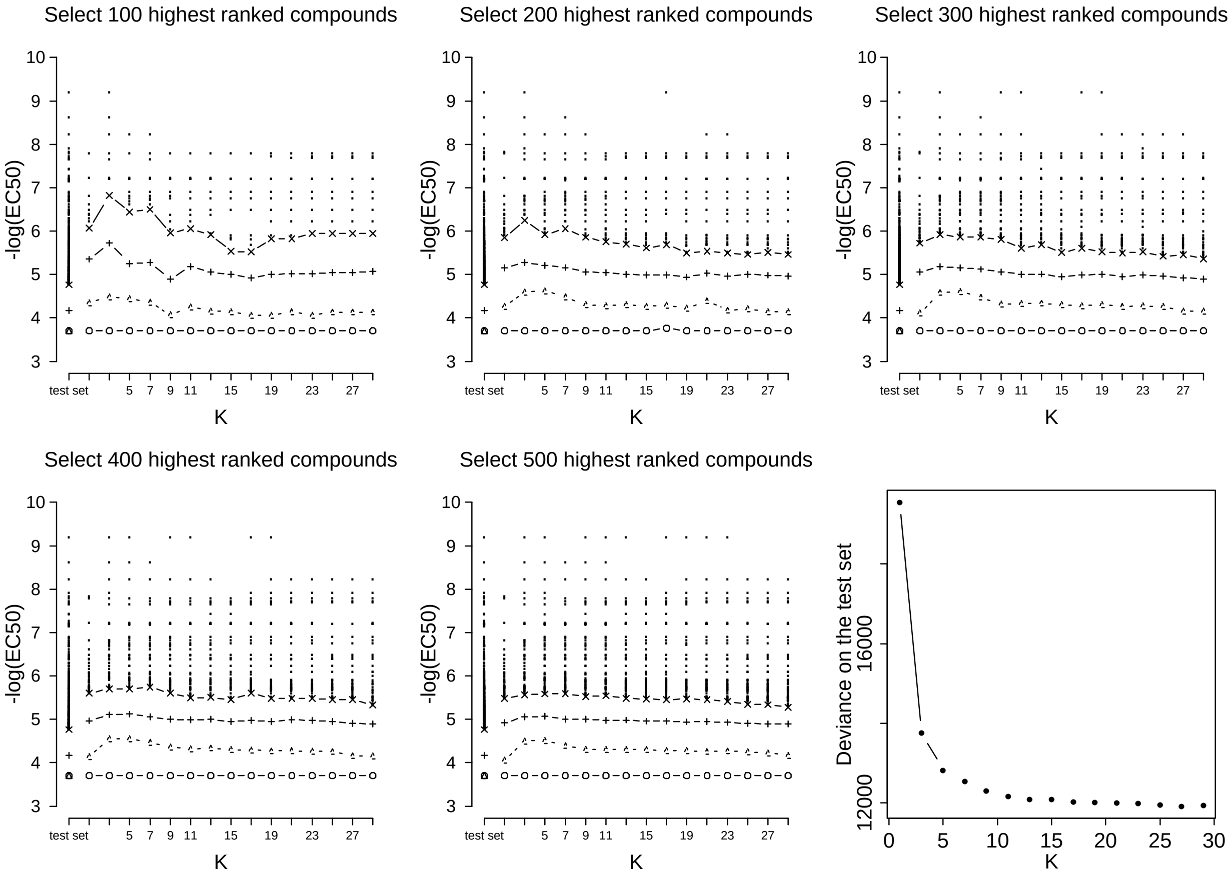


Figure 11: Quantiles of measured activity and deviance versus tree size (test set). In the first five pictures, the symbols ×, +, △ and ○ indicate 90%, 75%, 50% and 25% quantiles of measured activity, and the points above the quantile curves display the upper 10% lead compounds.

4.2 Regression trees

In Figure 11, the default regression tree is pruned. The first five pictures show how the quantiles of the measured activities vary with tree size when selecting 100, 200, 300, 400 and 500 highest ranked compounds in the test set. Again, it seems that large trees are at least as good as small trees. In contrast, plotting the deviance on the test set against tree size in the sixth plot shows a minimum at around 10 terminal nodes. As with classification trees, re-ranking terminal nodes to take account of node size might improve the performance of the regression tree. However, this is still under study. Preliminary experiments show that very large trees (over 300 terminal nodes) are also competitive.



Select 100 highest ranked compounds

Select 200 highest ranked compounds

Select 300 highest ranked compounds

Select 400 highest ranked compounds

Select 500 highest ranked compounds

Figure 12: Quantiles of measured activity and deviance versus K (test set). In the first five pictures, the symbols ×, +, △ and ○ indicate 90%, 75%, 50% and 25% quantiles of measured activity, and the points above the quantile curves display the upper 10% lead compounds.

5 KNN

When using KNN on the continuous response data, the dependence of performance on model complexity is similar to the results for trees. Figure 12 displays the quantiles and deviance versus k. It is apparent that the quantile plots pick very small number of k (3-7) as optimal. However the deviance (the smaller, the better) suggests quite large k (k=27 in this case).

We considered a similar analysis for KNN for the categorical data. However, there is no apparent discrepancy between the value of k chosen by the misclassification rate and by the hit rate. Various random splits also suggest that the optimal k is quite small (about 7-15) although a very small k (2-5) performs poorly. Using a very large k does not identify as many actives, though the difference versus small k is minor.

6 Conclusion

These data sets illustrate that HTS data are very complex, calling for specific methods to model the structure-activity relationship. Local methods such as trees and KNN are competitive in these examples. In fact, they are good candidates for SAR modeling, where localization is often apparent.

Appropriate ranking of terminal nodes is very important to improve a tree's performance. For instance, if we have two pure-active nodes, then the lower-bound criterion will choose the node with most compounds.

It is surprising that large trees outperform small trees in terms of number of hits and larger quantiles since conventional criteria (deviance or misclassification rate) suggest much smaller tree sizes. For binary data, it is probably because conventional accuracy measurements often assume that the target classes have a balanced distribution. For imbalanced HTS data, where the aim is to predict the rare active compounds, the standard criteria are dominated by the majority of the inactive compounds. For continuous data, we are more interested in the few most potent compounds. However deviance treats all the compounds equally. Therefore the optimal tree size and the value of k for KNN suggested by the hit curves and quantile plots are far from those picked by deviance.

Furthermore, if the deviance or the misclassification rate fails to determine goodness of fit of the tree, we doubt whether they are generating optimal splits. There is a need to look for alternative tree growing strategies.

References

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J., (1984). *Classification and Regression Trees.* Wadsworth International Group Belmont CA.

Burden, F.R. (1989). "Molecular Identification Number For Substructure Searches", *Journal of Chemical Information and Computer Sciences*,29, 225-227.

Clark, L.A., Pregibon, D. (1991) "Tree-Based Models" in *Statistical Models in S*, edited by J. M. Chambers and T. J. Hastie. Wadsworth & Brooks/cole Advanced Books & Software Pacific Grove, California.

Dasarathy, B.V. (1990).*Nearest neighbor (NN) Norms: NN pattern classification techniques.* Los Alamitos, Calif. : IEEE Computer Society Press, 1990.

Friedman, J.H. (1991). Multivariate adaptive regression splines (with discussion). *The Annals of Statistics*, 19, 1-141.

Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models.* Chapman and Hall, London.

Hawkins, D.M. and Kass, G.V. (1982). "Automatic Interaction Detection" in *Topics in Applied Multivariate Analysis*, ed. Hawkins, D. M., Cambridge: Cambridge University Press.

Hawkin, D.M., Young, S.S., Rusinko, A. (1997). Analysis of a Large Structure-Activity Data Set Using Recursive Partitioning *Quant. Struct.-Act. Relat.* 16, 296-302.

Jones-Herzog, D.K., Mukhopadyay, P., Keefer, C.E., Young, S.S. (2000). Use of recursive partitioning in the sequencial screening of G-protein-coupled receptors. *Journal of Pharmacological and Toxicological Methods* 42 (1999) 207-215.

King, R. D., Muggleton, S., Lewis, R. A., Sternberg, M.J.E. (1992). Drug Design by maching learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. Natl. Acad. Sci.* USA Vol. 89. pp. 11322-11326.

Klopman, G. (1984). Artificial Intelligence Approach to Structure-Activity Studies. Computer Automated Structure Evaluation of Biological Activity of Organic Molecules. *American Chemical Society* Vol. 106, No. 24.

Kooperberg, C., Bose, S. and Stone, C. J. (1997). Polychotomous Regression. *Journal of the American Statistical Association*, 92, 117-127.

Lam, R. (2001), Ph.D thesis presented to Department of Statistics and Actuarial Science, University of Waterloo, Canada.

Livingstone, D. (1995), *Data Analysis for Chemists*, Oxford University Press.

Pearlman, R. S., and Smith K. M. (1998) Noval Software Tools for Chemical Diversity, *Perspectives in Drug Discovery and Design* 1998, 9/19/11: 339-353.

Pearlman, R. S., and Smith K. M. (1999) Metric Validation and the Receptor-Relevant Subspace Concept, *Journal of Chemical Information and computer Sciences* 1999, 39, 28-35.

Quinlan, J.R. (1993). *C4.5: programs for machine learning* Morgan Kaufmann Publishers, San Mateo, California.

Ripley, B.D. (1996). *Pattern Recognition and Neural Networks* Cambridge University Press.

Rusinko A., Farmen M.W., Lambert, C.G., Brown, P.L., Young S.S. (1999). Analysis of a Large Structure/Biological Activity Data Set Using Recursive Partitioning. *Journal of Chemical Information and Computer Sciences*,39, 1017-1026.

Service, R.F. (1996). Combinatorial Chemistry Hits the Drug Market *Science* Vol. 272.

Tatsuoka, K., Gu, C., Sacks, J., Young, S. S. (1998). Predicting Extreme Values in Large Datasets, *Journal of Computational and Graphical Statistics.*

Venables, W.N., and Ripley B.D. (1999).*Modern Applied Statistics with S-plus*, 3rd edition. New York : Springer.

Young S. S., and Hawkins D. M. (1998). Using Recursive Partitioninh To Analyze A Large SAR Data Set. *SAR and QSAR in environmental Research* 1998, Vol 8, pp. 183-193.