

UCSF

Recent Work

Title

Relating amino acid sequence to phenotype: Analysis of peptide binding data

Permalink

<https://escholarship.org/uc/item/4597061w>

Authors

Segal, Mark R
Cummings, Michael P
Hubbard, Alan E

Publication Date

2000-12-01

Peer reviewed

Relating Amino Acid Sequence to Phenotype: Analysis of Peptide Binding Data

Mark R. Segal

Division of Biostatistics, University of California, San Francisco, CA 94143-0560

email: mark@biostat.ucsf.edu

Michael P. Cummings

Josephine Bay Paul Center for Comparative Molecular Biology and Evolution,

Marine Biological Laboratory, Woods Hole, MA 02543-1015

email: mike@bluefish.mbl.edu

and

Alan E. Hubbard

Division of Biostatistics, University of California, Berkeley, CA 94720

email: hubbard@stat.berkeley.edu

SUMMARY. We illustrate data analytic concerns that arise in the context of relating “genotype”, as represented by amino acid sequence, to phenotypes (outcomes). The present application examines whether peptides that bind to a particular major histocompatibility complex (MHC) class I molecule have characteristic amino acid sequences. However, the concerns identified and addressed are considerably more general. It is recognized that simple rules for predicting binding based solely on preferences for specific amino acids in certain (anchor) positions of the peptide’s amino acid sequence are generally inadequate and that binding is potentially influenced by all sequence positions as well as between-position interactions. The desire to elucidate these more complex prediction rules has spawned various modeling attempts, the shortcomings of which provide motivation for the methods adopted here. Because of (i) this need to model between-position interactions, (ii) amino acids constituting a highly (20) multilevel unordered categorical covariate, and (iii) there frequently being numerous such covariates (i.e. positions) comprising the sequence standard regression/classification techniques are problematic due to the proliferation of indicator variables required for encoding the sequence position covariates and attendant interactions. These difficulties have led to analyses based on (continuous) properties (e.g. molecular weights) of the amino acids. However, there is potential information loss in such an approach if the properties used are incomplete and/or do not capture the mechanism underlying association with the phenotype. Here we demonstrate that handling unordered categorical covariates with numerous levels and accompanying interactions can be done effectively using classification trees and recently devised bump-hunting methods. We further tackle the question of whether observed associations are attributable to amino acid properties as well as addressing the assessment and implications of between-position covariation.

KEY WORDS: Bump hunting; Classification trees; Prediction rules; Unordered categorical covariates.

1 Introduction

A wide variety of biomedical problems can be viewed as attempts at relating genotype to phenotype. This is apparent from the loose definitions: *genotype* – the class to which an organism or entity belongs based on its genes; *phenotype* – the class to which an organism or entity belongs based on its physical characteristics (Lewontin, 1992). A common illustration is provided by studies seeking to relate viral or bacterial mutations (genotypes) to resistance. The ultimate specification of an organism's genotype is given by its complete DNA sequence or genome. Necessarily, we deal with partial genotypes corresponding, for example, to select genes or markers. The sequence itself can be based on either nucleotides or amino acids; here we focus on the latter. Also of interest are bio-physico-chemical properties of individual amino acids, examples of which include molecular weight, volume, hydrophobicity, and polarity. The physical characteristics denoting phenotype have the familiar typology being nominal, ordinal, or continuous variables.

So, given the objective of relating genetic level information to physical properties, why can't standard regression methodologies be employed? What is it, if anything, that distinguishes this setting? We contend that the nature of genotype data, in particular when represented by amino acid sequence, precludes use of many familiar regression techniques. It is the occurrence of several unordered categorical covariates, each having (potentially) numerous levels, that mandates alternative approaches. To make these concerns concrete we immediately turn to a simple motivating example that is the subject of our subsequent analyses. We emphasize the simplicity of this illustration – only 8 covariates (sequence positions) are featured – the difficulties cited would amplify appreciably with longer sequences which are commonplace.

Milik et al., (1998) are concerned with predicting the amino acid sequences of peptides that bind to the particular class I MHC molecule, K^b. Here, the peptides of interest are 8-mers – molecules composed of an ordered sequence of 8 amino acids – which may result from proteolysis of invading viral particles. Some of these peptides bind to class I MHC molecules which, in turn, present them on the surface of the infected cell. There these complexes are recognized by cytotoxic T lymphocytes that destroy the infected cell. Hence, MHC binding is essential for any peptide to induce an immune response and the problem of identifying peptides that bind to particular MHC molecules is of

utmost immunologic importance (Gulukota, 1998). Here the peptide's amino acid sequence constitutes genotype and its binding is the phenotype. Select data giving peptide amino acid sequence and corresponding binding to K^b as a binary (yes/no) outcome are given in Table 1. The single letter designations are the standard abbreviations for the various amino acids. The complete dataset has 310 such observations and is available from <http://newfish.mbl.edu/Lab/Resources/>. Barplots giving specific amino acid frequencies for binding and non-binding peptides are given in Figure 1. It is important to note that the data is obtained by random sampling from a large ($> 10^7$) library of synthetic peptides, so that there is no evolutionary history linking the peptides. Scott and Smith (1990) provide details about the construction of such libraries.

Studies (Rammensee et al., 1995) revealed that peptides binding class I MHC molecules typically have specific amino acids at specific positions, called *anchor* positions, in the sequence. However, the use of simple rules for predicting binding based solely on such anchor position preferences (so-called motifs) are inadequate; binding is known to be influenced by both the presence of secondary anchor positions and interactions between amino acids within the peptide. It is this search for more complex structure that spawns the association problem that we will examine further, both with regard to analyzing this particular dataset as well as discussing related statistical issues. Analogous to Milik et al., (1999) this is undertaken without using any structural information about the MHC-peptide complex; see Zhang et al., (1998) for such an approach that uses known crystal structures of class I MHC molecules to construct pocket models.

In the next section we outline shortcomings with standard regression tools applied to such data. In section 3 we introduce tree-structured and bump-hunting methods that hold promise for overcoming these difficulties. Section 4 describes two statistical concerns surrounding amino acid sequence data: sequence position covariation and the role of amino acid properties. Section 5 returns to the peptide data for detailed analysis while section 6 presents some concluding discussion.

2 Standard Method Difficulties

By “standard methods” we intend the suite of estimation, inference and diagnostic machinery subsumed by the generalized linear model (GLM) framework (McCullagh and Nelder, 1989), as well as various extensions thereof such as generalized additive models (Hastie and Tibshirani, 1990). In analyzing the peptide binding data, Milik et al., (1998) employ artificial neural networks (ANNs) using amino acid property variables. Our immediate concern is not the choice of regression method (ANNs) but rather the use of the property variables in lieu of genotype. This was done since the use of the amino acids themselves would require (approximately) 19 indicator variables for each of the 8 positions and it was contended that the resultant large numbers of covariates would (i) be unmanageable; and (ii) lead to overfitting. We now demonstrate that similar concerns – the inability to readily handle unordered categorical covariates with numerous levels – applies to standard methods.

For a continuous outcome a classical approach for dealing with unordered categorical covariates would be multi-way ANOVA. For example, if peptide binding had not been dichotomized, we could entertain an 8 way ANOVA, with dimensions corresponding to the 8 sequence positions. However, since this represents $\approx 20^8$ cells, all but very low-order models will be inestimable due to sparseness. And, as will be seen below, there is interest in at least second-order interaction terms – even this low an order proves problematic. Furthermore, many studies will feature much longer sequences. Thus, there is a clear need for model/variable selection methods, which we discuss later.

Binding, as provided, is binary. So, one natural modeling framework is logistic regression. There are 20 naturally occurring amino acids. Here, the number of distinct amino acids at each of the eight positions is 18, 20, 20, 20, 20, 20, 19, and 20 respectively. Arguably, the default starting model would include each position. This entails estimating 149 coefficients corresponding to the respective indicators. Immediately we see that just assimilating the resultant output will be difficult. Simple tasks such as appraising individual position and/or amino acid importance, grouping amino acids with similar effects within position, and comparing across positions become daunting when just the output coefficients span several pages. Remember, too, that this is a simple model for a very small (8 positions) problem.

The most consequential shortcoming arises in accommodating between position interactions. For the MHC - peptide binding example, it is necessary to consider such interactions because the nature of binding and the structure of particular amino acids can effect adjacent and/or second nearest neighboring amino acids' ability to bind to MHC (Gulukota et al., 1997). This suggests that models including at least select third order interactions be entertained. Such considerations are commonplace when analyzing sequence - phenotype associations, where the situation is compounded by the need to include interactions between non-adjacent positions. This arises, for example, in determining quantitative trait loci (Fridyland and Speed, personal communication). When dealing with nucleotide or amino acid sequence data fitting difficulties ensue due to the combinatorial explosion in the number of indicators needed to encode these interactions. For a set of sequences of length k of an n level residue ($n = 4$ nucleotides or $n = 20$ amino acids) with each level represented at all positions we have (a) $\binom{k}{2}(n-1)^2$ terms for all second order interactions; (b) $(k-1)(n-1)^2$ for adjacent second order interactions; (c) $\binom{k}{3}(n-1)^3$ for all third order interactions; and (d) $(k-2)(n-1)^3$ for adjacent third order interactions. Here, for the very small ($k = 8$) peptide example further limited to adjacent second order interactions we require according to (b) above 2,527 terms (the exact number is 2,451 since some positions do not exhibit all $n = 20$ possible amino acids) and, by (a), 10,108 (exact 9,711) terms to encode all second order interactions. In either case, fitting proves prohibitive for all standard software packages due to insufficient dynamic memory (on a Sun Ultra 5 Model 360 with 128 MB RAM). This breakdown is not remedied by either employing forward stepwise selection or attempts at memory expansion. Clearly, all ($\approx 384,104$ terms) or adjacent ($\approx 41,154$ terms) third order interactions cannot be handled.

As suggested by Milik et al., (1998), it is to avoid such difficulties that properties of the amino acids are used in place of genotype. These property variables are ordered and so can be handled much more readily. Even entertaining nonlinearities and interactions, we would be unlikely to spend more than a few degrees of freedom per variable. Further, the resulting models will be more succinct and so more readily interpretable. However, there are potential losses associated with making recourse to a property variable representation. Principally, if the collection of property variables does not capture how the amino acids affect phenotype, then there is obvious and crucial information loss. That this can occur is exemplified by (i) Milik et al., (1998) opting to augment their property variable set with indicators for particular amino acids, (ii) Kidera et al., (1985) itemizing some 188 properties with the

implication that using an exhaustive list is problematic, and (iii) in light of the above arguments re the need for interaction between amino acids at neighboring positions, there would be an attendant need to consider properties of interacting amino acids which may not correspond to the usual multiplicative terms(s) of individual properties.

Indeed, it may well be that the simple amino acid information itself, as given by the linear sequence (position) representation, is deficient since it omits essential spatial information deriving from the three dimensional structure of the peptide-MHC complex; see Zhang et al., (1998). While such matters are beyond the scope of the present paper we do, however, return to contrasting property and amino acid based analyses. Any property necessarily derives from a many-to-one mapping from the amino acids. This is manifest for discrete properties such as charge, but also occurs for continuous properties. For example, in Table 1 we observe that the position 8 amino acid for the 3rd peptide (row 3) is Leucine (L) with a molecular weight of 131.17 (last column, row 3). This coincides with the molecular weight for Isoleucine (I), the amino acid at position 8 for the 4th peptide (row 4). We address questions as to whether the effect of amino acids on phenotype is via property variables in light of these relationships. But first we describe the utility of tree-structured and bump-hunting methods for handling unordered categorical covariates, with obvious applicability to genotype-phenotype analyses.

3 Handling Unordered Categorical Covariates

3.1 Tree-Structured Methods

The definitive reference describing tree-structured methods is “Classification and Regression Trees” by Breiman et al., (1984), hereafter denoted by CART. A more recent overview of numerous extensions and refinements to the basic paradigm is provided by Segal (1995). While the methodology handles many differing problem types and is supported by interactive companion software (e.g. Therneau and Atkinson, 1997), our emphasis here will be on how tree-structured techniques handle unordered categorical covariates.

CART tree construction involves four components. These are: (1) A set of binary (yes/no) questions, or *splits*, phrased in terms of the covariates that serve to partition the covariate space. A tree structure derives from splitting recursively. The subsamples created by assigning cases according to these splits are termed *nodes*; (2) A *split function* $\phi(s, t)$ that can be evaluated for any split s of any node t which is used to compare competing splits; (3) A means for determining appropriate tree size; and (4) Statistical summaries for the nodes of the tree.

Item (1) deals with handling covariates. Allowable splits are defined as follows: (a) each split depends upon the value of only a *single* covariate; (b) for ordered (continuous or categorical) covariates, x_j , only order preserving splits of the form “Is $x_j \leq c$?” for $c \in \text{domain}(x_j)$ are considered; (c) for unordered categorical covariates all possible splits into disjoint category subsets are allowed.

So, for the covariate type of interest, unordered categorical, no constraints on possible subdivisions are imposed. If such a covariate has M categories then there are $2^{M-1} - 1$ splits to examine leading to combinatorial explosion for large M . However, by generalizing a result from Fisher (1958), CART (§8.8, 9.4) establishes a theorem that reduces this to an eminently feasible $M - 1$ splits: if we rank the levels of the unordered categorical covariate by mean response value, then we only need examine splits that preserve this ranking. When dealing with the amino acid alphabet, in polymorphic (variable) settings such as the peptide binding example, we have $M = 20$ so that without recourse to Fisher’s result we would need to evaluate a prohibitive 524,287 splits per position as opposed to 19.

It is by appropriately defining the split function ϕ (Item 2) that classification is effected. Choice of suitable split functions is discussed extensively in CART (Chapter 4). Here we are interested in the two class (binder, non-binder) problem. Let g be some impurity function and define the impurity of a node t by $i(t) = \sum_{j=1}^2 g(p_{jt})$ where p_{jt} is the proportion of cases in t that belong to class j . Requirements for g are (i) $g(0) = g(1) = 0$; (ii) $g(p) = g(1 - p)$, and (iii) $g''(p) < 0$ (i.e. g is concave). Two natural candidates for g are the Gini diversity index $g(p) = p(1 - p)$, and the information index (which is equivalent to the binomial deviance) $g(p) = -p \log(p)$ which are almost equivalent (Therneau and Atkinson, 1997) as is evidenced in section 5.4. For a split s partitioning t into t_L and t_R the split function is then defined as

$$\phi(s, t) = i(t) - p_L i(t_L) - p_R i(t_R) \tag{1}$$

where p_L is the proportion of node t cases assigned to t_L and $p_R = 1 - p_L$. The best split s^* maximizes (1), giving the greatest reduction in impurity. Recursive application to the resultant daughter nodes (t_L^*, t_R^*) and so on gives rise to progressively smaller nodes of decreasing impurity or increasing homogeneity. Thus, here we will (hopefully) be creating nodes that contain predominantly binding (or non-binding) peptides – the classification objective.

For future reference, in connection with appraising anchor positions in the face of between position correlation, we introduce *surrogate* and *competitor* splits. A surrogate split best reproduces the optimal split s^* but on a different covariate and has utility for handling missing observations and determining covariate importance; see CART §5.3. The first competitor split is just the split that has the second best (to s^*) reduction in impurity (1), again based on a different covariate.

3.2 Bump-Hunting Methods

By casting classification as function optimization, Friedman and Fisher (1999) (FF) develop flexible procedures that appropriately handle unordered categorical covariates and perform well in high dimensions. We give a brief outline of their development and estimation strategy before applying this methodology to the peptide binding problem. Bump-hunting resembles tree-structured methods in seeking covariate-defined subregions over which the outcome is extreme. The methods differ in that tree methods are recursive – the subregions are related via a tree structure – whereas no such constraint is imposed by bump-hunting, increasing flexibility.

Given a target function $f(\mathbf{x})$, where \mathbf{x} represents a vector of covariates, the goal of finding maxima of f can be generalized to seeking subregions of the covariate space within which the average value of f is much larger than the overall average. The search corresponds to the “hunt” and the elevated f average values the “bumps”. Let S_j be the set of all possible values for the j^{th} covariate, which may be ordered real values (continuous or discrete) or unordered categories. The entire covariate space can be then be represented by the p dimensional outer product $S = S_1 \times \dots \times S_p$. We seek a subregion R of the covariate space S , $R \subset S$, for which

$$\bar{f}_R = \text{ave}_{\mathbf{x} \in R} f(\mathbf{x}) = \int_{\mathbf{x} \in R} f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} / \int_{\mathbf{x} \in R} p(\mathbf{x}) d\mathbf{x} \gg \bar{f}$$

where \bar{f} is the overall average and $p(\mathbf{x})$ is the (unknown) joint covariate density. Designate the support of the subregion R by β_R : $\beta_R = \int_{\mathbf{x} \in R} p(\mathbf{x}) d\mathbf{x}$. There is typically a trade-off between \bar{f}_R and β_R – larger subregion averages will be associated with smaller support and vice versa.

For our two class problem define the indicator outcome $y = I\{\text{class} = \text{binder}\}$ and take $f(\mathbf{x}) = E[y|\mathbf{x}] = \Pr(y = 1|\mathbf{x})$. Thus, the regions sought correspond to those with a relatively large proportion of binders. We now define what constitutes the allowable regions and indicate how they are obtained. FF favor rules that can be readily described and interpreted even if this sacrifices power. Accordingly, they require that the solution region R be specified by simple statements involving individual covariates. These rules have the form $R = \bigcup_{k=1}^K B_k$ so that the solution region is the union of a set of simply defined subregions B_k . Let v_{jk} represent a subset of the possible values of the j^{th} covariate, x_j . Then each B_k is taken to be a “box”: $B_k = v_{1k} \times v_{2k} \times \cdots \times v_{pk}$ within the entire covariate space. Thus, each box can be described via the intersection of subsets of values of each covariate: $\mathbf{x} \in B_k = \bigcap_{j=1}^p (x_j \in v_{jk})$.

For ordered covariates the allowable subsets are contiguous intervals $v_{jk} = [a_{jk}, b_{jk}]$ so that the projection of a box onto just the continuous covariates yields a hyper-rectangle. For unordered categorical covariates, any subset of levels is allowable. Hence, we have flexibility akin to CART’s in handling unordered categorical covariates.

In order to obtain good boxes a two-phase strategy is employed. Initially, a box is produced by iteratively removing unimportant (e.g., low proportions of binders) regions of the covariate space. This stage is termed “top-down peeling”. Constraining that each successive region removed only eliminate a small ($\leq 10\%$) of the total sample mitigates against “greediness” deriving from optimizing without look ahead; see also Hastie et al., (2000). However, additional improvements are pursued by readjusting (enlarging) the boundaries of the resultant box. This constitutes the second stage, termed “bottom-up pasting”. Complete algorithmic details are deferred to FF. Our peptide binding application focuses on single box solutions, given in Tables 2 and 3, which illustrates the flavor of bump-hunt output and are further discussed in section 5.3.

4 Issues for Amino Acid Sequence Data

In this section we briefly discuss two concerns pertaining to amino acid sequence data. These are (i) assessing covariation between sequence position, and (ii) assessing the importance of properties of the amino acids. These issues, along with the techniques described in section 3, are then featured in re-analyzing the peptide binding in the following section.

4.1 Position Covariation

As collinearity is to linear models, *masking* is to trees. Loosely, masking refers to the phenomenon whereby a selected split precludes an alternative, almost-as-good split from emerging. The variable associated with this unseen split is said to be masked. Both CART and Therneau and Atkinson (1997) provide detailed information on masked variables by outputting lists of surrogate splits; see CART §5.3. We will appeal to surrogates when interpreting the results of tree-structured analysis of the peptide binding data below. But, additionally and relatedly, we investigate measures of correlation between the amino acid position variables, using methods of Bickel et al., (1996), which we briefly describe next. In general, such correlation is anticipated with sequence data due to linkage disequilibrium – the non-random association of alleles at different loci due to processes including co-ancestry, gene flow, genetic drift and selection.

Categorical covariates require customized measures of correlation, several of which have been proposed. We focus on the “P-statistic” of Bickel et al., (1996) defined and interpreted as follows. Consider a set of n sequences that are aligned and of the same length, l . For the peptide binding data we have $n = 310$ and $l = 8$. Define

$$\begin{aligned}\hat{p}_i(a) &= n^{-1}\#\{\text{sequences with amino acid } a \text{ at position } i\} \\ \hat{p}_{ij}(a,b) &= n^{-1}\#\{\text{sequences with amino acid } a \text{ at position } i, \text{ amino acid } b \text{ at position } j\} \\ M_{ij} &= \sum_{a,b} \hat{p}_{ij}(a,b) \log \left(\frac{\hat{p}_{ij}(a,b)}{\hat{p}_i(a)\hat{p}_j(b)} \right)\end{aligned}\tag{2}$$

where the (double) sum in (2) is over all amino acids a, b at positions i, j respectively. The P-statistic

is then given by

$$P_{ij} = \max_{a,b} P_{ij}(a,b)$$

where $P_{ij}(a,b)$ is the M_{ij} statistic obtained by replacing the 20 letter amino acid alphabet at positions i and j with binaries $\{a, \text{not } a\}$ and $\{b, \text{not } b\}$ respectively. As described by Bickel et al., (1996) $M_{i,j}$ is the likelihood ratio statistic for testing the hypothesis of independence of positions i and j against arbitrary covariation and, in large samples, is roughly equivalent to the usual Pearson chi-squared statistic for testing independence. $P_{i,j}(a,b)$ is the likelihood ratio test specialized to the alternative that it is the amino acid pair (a,b) that drives the dependence. $P_{i,j}$, the maximum of the $P_{i,j}(a,b)$, is intended to detect situations where only one pair of amino acids exhibits covariation but without prespecifying which pair.

Evaluation of the significance of the M or P statistics makes recourse to permutation. A large number of permuted data sets of the same structure and with the same marginal probabilities for each amino acid at each position are created by independently permuting the amino acids at each position. By computing M or P on each dataset we can obtain a permutation test significance level in the standard fashion. As per Bickel et al., (1996) we adopt this permutation approach since (i) the asymptotic chi-squared approximations are known to be poor with the sparse tables that almost necessarily arise with sequence data, and (ii) we are interested in simultaneous inference for all possible pairs of positions so the large number of permuted data sets provides protection.

The importance of assessing position covariation in this manner is at least two-fold: (i) we can elicit relationships amongst the positions themselves that can then inform model specification – necessary in light of the software breakdowns described in section 2 when no constraints were imposed on first-order interactions, and (ii) we can gauge how much additional predictive ability will derive from considering an expanded set of positions beyond putatively established markers. We highlight this second feature in the next section with reference to the anchor positions.

4.2 Association via Property Variables

Model selection/comparison concerns arise in the context of relating genotype to phenotype in at least two obvious ways. Firstly, within a particular modeling methodology is the phenotype-genotype association via property variables. That is, does a set of amino acid derived property variables explain as much of the relationship as the amino acids themselves? Is it, for instance, the hydrophobicity of the amino acids that influences binding? Since the relationship between an amino acid position variable and a property position variable is many-to-one we can't obtain a meaningful answer to the above question by allowing the variables to compete head-to-head in conjunction with some variable selection scheme. However, we could (i) fit a model using property variables, then (ii) fit a subsequent model to residuals from (i) using amino acid variables. If this second fit revealed structure we could infer that the property variables did not fully capture association with the outcome. But, should no structure emerge (i.e. a null model result from step (ii)), then any claim that the property variables do explain association would need to be tempered by power (to detect a non-null model) considerations.

Tree-based models allow further comparison. The set of possible splits utilizing a property variable is a small subset of the set of possible splits utilizing an amino acid variable. By quantifying how "far" the optimal split based on a property variable is from the optimal split based on an amino acid variable we can, under a null assumption that all splits are equally likely, assess whether association is via the property variable. We operationalize the distance between splits on unordered categorical covariates (amino acids) and continuous covariates (properties) by how many "moves" (transpositions) from one resultant node to the complementary node of an unordered categorical covariate split are necessary to yield a split achievable on a continuous covariate. This is best illustrated by example.

Consider a 9 level unordered categorical covariate and let a derived property variable take ordered values 1 through 9. Consider, too, a split of the unordered categorical covariate that partitions the 9 levels (3,6). Using the labels corresponding to the property variable the following possibilities arise:

Illustrative Partition	Moves Required to Achieve Order	Number of Such Partitions
$\{1, 2, 3\} \{4, 5, 6, 7, 8, 9\}$	0	2
$\{1, 3, 4\} \{2, 5, 6, 7, 8, 9\}$	1	36
$\{1, 4, 5\} \{2, 3, 6, 7, 8, 9\}$	2	45
$\{4, 5, 6\} \{1, 2, 3, 7, 8, 9\}$	3	1

The sum of entries in the rightmost column is $84 = \binom{9}{3}$; the number of (3,6) partitions.

For an unordered categorical covariate with n levels the maximum number of moves required is $\lfloor (n/3) \rfloor$. The number of splits or partitions requiring 0, 1 and 2 moves to achieve order is $n - 1$, $(n^3 - 7n - 24)/6$, and $(n^5 - 5n^4 + 5n^3 - 55n^2 - 1026n + 480)/120$ respectively. We apply this result to the peptide binding data in the following section.

5 Peptide Binding Revisited

5.1 Classification Trees

Tree-structured classification, as described in section 3.1, was applied to the peptide binding data. Training and test sets having the same dimensions as those used by Milik et al., (1998) were obtained via random selection. We present results using data priors, unit misclassification costs and the information index (deviance) as split function. Results were not sensitive to the (random) selection of training and test datasets or the choice of split function. Alternative priors and/or costs were not explored since we had no basis for specifying these differently.

The initial large tree grown on the training data and using solely the 8 amino acid position variables is depicted in Figure 2. The predicted class (based on simple majority rule) at each node (ellipses for internal nodes, rectangles for terminal nodes) is given by the 0 (non-binding) or 1 (binding) indicated within each node, while the ratio below gives number misclassified / node size. Numerals above each node are used for identification purposes. Thus, the topmost node (Node 1) contains 223 cases

of which the majority ($131 = 223 - 92$) are binders, there being 92 misclassifications (non-binders). The splits are indicated on the branches of the tree. So, for example, Node 1 is partitioned on the basis of position 8 with those cases having amino acids F, I, L, M, or Y in the 8th position being assigned to the right daughter node. The large initial tree is grown so as to capture all potentially important splits. This is then collapsed back up using cost-complexity pruning, with selection from the resultant nested sequence of trees being based on either cross-validation or an independent test sample; see CART for motivation and details of this approach.

Basing tree size selection on CART's "1 SE rule" the best pruned subtree is one with 4 terminal nodes – it is the smallest tree within one standard error of the minimally attained deviance, this holding irrespective of whether test data or 5- or 10- fold cross-validation with the training sample data is used. Figure 3 presents the best 4 terminal node tree with predictions from running the test set shown. For the test set data so classified we achieve a sensitivity of 86% and a specificity of 94%. The over-optimistic values corresponding to re-using the training data for prediction purposes for this four terminal node tree are 85% and 99% respectively.

The explanation for the apparent data loss corresponding to the Node 3 split (Node 3 sample size = 50; Nodes 6 and 7 sample size = $45 = 44 + 1$) is that the splits are determined on the training data. Ambiguity in classifying test data can arise when a split does not utilize levels represented in the test data; note that only 12 of the 20 possible position 5 levels are present in Node 3 training data. This ambiguity can be resolved using surrogate splits. So doing improves sensitivity (94%) while not appreciably impacting specificity (92%).

Thus, we obtain good performance using a very simple classification scheme. This contrasts with ANN and standard method results. Further, the tree method illuminates the predictive structure of the data as is further discussed in connection with position covariation (section 5.4).

5.2 Property Variables

The predictive ability of some 12 (amino acid) property variables was also investigated. These included volume, bulkiness, flexibility, polarity, aromaticity and charge as considered by Milik et al.,

(1998), as well as additional measures of hydrophobicity and mass. Using the same training and test datasets and the same tree growing, pruning and selection strategy as above, again a tree with four terminal nodes (Figure 4) was selected.

One interesting point illustrated by comparing the property and amino acid trees concerns greedy splitting. As noted, by virtue of the many-to-one mapping of amino acids to properties and the unconstrained nature of splits on unordered categorical covariates, any property-based split can be reproduced by an amino acid split. However, the converse is not true. We might anticipate that the amino acid tree would be superior as appreciably more splits are being evaluated. Indeed, for the training data, this is the case for the first split: there are 25 ($= 17 + 8$) misclassifications for the amino acid tree and 27 ($= 17 + 10$) for the property tree. However, when we make overall comparisons between the selected, four terminal node trees there are more misclassifications with the amino acid tree ($21 = 0 + 17 + 3 + 1$) than for the property-based tree ($20 = 5 + 1 + 5 + 9$). While it is the case that the split criteria deliberately (see CART §4.2) do not minimize misclassification totals, these results highlight the fact that one-step look-ahead does not necessarily produce overall optimal results.

Node 1 is split based on polarity at position 8. This split closely approximates the Node 1 split based on amino acids which also used position 8 (Figure 3): only one test sample and two training sample cases are assigned differently and the deviances are correspondingly comparable. Given the multitude of possible amino acid based splits ($2^{19} - 1$) it may appear that this agreement suggests that the binding - genotype association is captured via a property variable, polarity, which we next explore.

In terms of amino acids, the position 8 polarity split corresponds to assigning C, F, I, L, M and W to one daughter node in contrast with the amino acid split of F, I, L, M and Y. So, in terms of moves as defined in section 4.2, two moves are required to map the unordered (amino acid) split to the ordered (property) split. Using the formulae presented there with $n = 20$, corresponding to the 20 letter amino acid alphabet, there are 19,983 splits two moves removed from an ordered split, 1,306 splits one move removed, and 19 splits zero moves (i.e. already ordered) removed. Thus, the probability due to chance of observing this degree of agreement is $(19,983 + 1,306 + 19) / (2^{19} - 1) = 0.04$. Of course, the notion of chance here presupposes that all splits are equally likely. This is not the case asymptotically

with, in null situations, extreme “end-cut” splits being favored; see CART §11.8. While tempting to conclude that the above “p-value” nonetheless suggests a possible role for amino acid polarity in peptide binding, further consideration (section 5.4) in terms of surrogate and competitor splits diminishes this possibility. We do note, however, that the importance of polarity in the context of DNA and protein evolution was demonstrated by Xia and Li (1998).

5.3 Bump-hunting

Results from using the bump-hunting approach of section 3.2 are presented in Tables 2 and 3 for amino acid and property variables respectively. The software does not allow a prescribed training set to be specified. Rather, the relative sizes of training and tests sets are provided. This explains the slight discrepancy between the proportion binding in the training set here (54.4%) and at Node 1 of the classification tree (58.7%). Like the classification tree analyses, the bump-hunting results were not sensitive to random selection of alternative training sets.

As indicated in section 3.2, we focus on a single box solution. Table 2(a) gives box summaries for both the training and test datasets, while 2(b) gives the box definition. From (a) we note that the solution box has support of 0.74 for the training data and 0.84 for the test data. In other words, 74% of the training and 84% of the test data are contained within the selected box. Further, the percentage of binders in this (large) box is 70.4% (training) and 74.7% (test), appreciably greater than the overall percentages of 54.4% and 66.3% respectively.

From Table 2(b) we see that the solution box is very simply defined involving just two positions, 8 and 5. These positions also figured prominently in the classification tree and are, in fact, two of the three anchor positions. However, the levels (i.e. specific amino acids) involved in the box definition display minimal overlap with the tree. Note that the over-bar notation represents set complement. Thus, the box is defined by

$$\mathbf{x} \in B = \begin{cases} \text{position 8} \in \{A, D, E, F, G, H, I, K, L, M, N, Q, R, T, V, W, Y\} \ \& \\ \text{position 5} \in \{A, C, D, E, F, G, H, I, K, L, M, N, Q, R, S, T, V, Y\}. \end{cases}$$

This contrasts with the definition of the classification tree’s sole terminal node that is classified as

binders:

$$\mathbf{x} \in \text{Node 7} = \begin{cases} \text{position 8} \in \{F, I, L, M, Y\} \ \& \\ \text{position 5} \in \{A, F, I, L, M, N, Y\} \end{cases}$$

for which the training and test sample percentage binders are 99% and 95.5% respectively. We discuss these differences after presenting the bump-hunting results for property variables.

The remainder of Table 2(b) is interpreted as follows: if the solution box was to be enlarged by sequentially removing the defining variables, the change in box support and percent binders would be as given. That is, if position 5 is eliminated, the box support increases to 0.86 but the percent binders decreases (very slightly) to 74.2%. Thus, position 5 does not meaningfully add to the solution. Further removal of position 8 reverts to the entire sample so we have support of 1.00 and percent binders equal to the overall test set percentage (66.3%).

An interesting contrast is obtained using property variables, the results for which are presented in Table 3. In comparison with the amino acid variable box, we obtain a smaller box (training support 0.59; test support 0.67) having appreciably higher percent binders (training set 85.0%; test set 94.3%). Given (i) the flexibility with which unordered categorical covariates are purportedly handled, and (ii) the fact that the property variables derive from the amino acid variables, that such an improved binding percentages are obtained using property variables is attributable to algorithm limitations in handling unordered categorical covariates.

Further comparing the property box with the classification tree based on properties, we again see an overlap of variables. The box is defined using volume at position 5, flexibility at position 1 and polarity at position 8. The classification tree has two terminal nodes classified as binders, the larger of which (support 0.49 (training set), 0.56 (test set)) is defined using volume at position 5 and polarity at position 8 and has binding percents of 99% (training set) and 88% (test set). Thus, there is much greater concordance between tree and bump-hunting results for property variables than there is for amino acid variables.

5.4 Anchor Positions and Position Covariation

We return to consideration of one of the motivating concerns: the elicitation of more complex rules for binding than afforded by solely using the anchor positions. Here, the anchor positions are 3, 5 and 8. The latter two figure prominently in both the classification tree and bump-hunt rules. There is a suggestion of a role for position 1 in that (i) the classification tree using amino acid variables features a position 1 split, and (ii) the property variable box obtained using bump-hunting features a position 1 property (flexibility). However, before inferring a role for this non-anchor position, it is important to appraise position covariation and, in the tree context, surrogate splits.

The results of applying the Bickel et al., (1996) permutation-based assessment of pairwise position covariation can be summarized as follows. For non-binders, there are no significantly covarying sites, in accord with the random sampling of synthetic peptides. Conversely, for binders, almost all $\binom{8}{2} = 28$ pairs of sites significantly (and comparably) covary using a *simultaneous* p-value of $p = 0.05$. Implications of this are (i) there are likely alternative split and box descriptions based on other positions that provide competitive classifications, and (ii) the ability to elicit rules based on positions beyond the anchors is diminished. These conclusions are reinforced by a consideration of surrogate and competitor splits.

The amino acid based classification tree (Figure 3) features only one split (Node 2) using a non-anchor position (position 1). But, the best surrogate and competitor splits for this node are both based on positions 5 and 3 respectively, both anchor positions. Further, (i) the competitor splits are comparable in terms of impurity improvement to the optimal (selected position 1) split, and (ii) if we adopt the Gini split criterion, instead of using the information index, position 5 is used for the optimal split. Conversely, for the two splits that use anchor positions (Node 1, position 8; Node 3, position 5) the primary competitor and surrogate splits are again based on anchors: position 5 for Node 1, and position 3 for Node 3.

We now turn to examination of surrogates and competitors for the classification tree based on property variables (Figure 4) and revisit the Node 1 split on position 8 polarity, the subject of section 5.2. The definition of surrogate and competitor splits stipulate that they be based on a different covariate

to the optimal split. There are numerous (12) property variables per position. Without exception (for each node), the top (first four) competitor and surrogate splits use the same position as the optimal split. Further, these splits are either highly competitive or strong surrogates with concordances (overlap) exceeding 96%. So, a tree with very similar performance could be obtained by splitting Node 1 on either volume or hydrophilicity at position 8. As indicated, this mitigates against claims that polarity is the important property re binding association.

Analogous diagnostics for appraising alternative variable selections / model formulations are available for the bump-hunting methodology. These are based on relative frequency distributions: ratios of the within box to overall density for each covariate are plotted (FF, section 16.2). As anticipated from the strong position covariation, these plots (not shown) demonstrate that alternative box definitions, using other positions and levels (specific amino acids) would yield similar support and percent binders.

6 Discussion

The thrusts of this paper have been to (i) demonstrate that specialized techniques are needed to handle multi-level unordered categorical covariates, as constituted by amino acid sequence data, and that standard methods are deficient for this purpose, and (ii) illustrate this and other analysis issues in the context of peptide binding. Despite the focus on peptide binding, we believe that the tree and bump-hunt methods featured here have great generality in terms of analyzing phenotype-genotype association. For example, we have used these approaches in determining which combinations of point mutations in the tuberculosis *rpoB* gene are associated with resistance (quantified by minimum inhibitory concentration) to the anti-tuberculin rifampin. Another setting where tree methods, albeit extended via bagging (Breiman, 1996) are useful, and standard approaches are inadequate, is in detection of quantitative trait loci. Using both simulation and real world examples, Fridyland and Speed (personal communication) present successful tree bagging applications, in contrast to failures of standard regression techniques. An important note here is that the genotype information, while still unordered categories, is not highly multi-level. Rather, there are only a few alleles at each

locus. What makes such problems challenging for standard regression/classification methods is the combinatorial explosion deriving from the need to identify interactions (between loci) of order ≥ 3 with ≥ 20 (frequently hundreds) of loci, little if any prior knowledge re important loci and weak main effects.

Arguably (Ripley, 1996; Hastie, 1997) artificial neural networks (ANNs) are overused in some domains. We believe this to be the case for many peptide binding applications where the combination of relatively small sample sizes, 20 level unordered categorical covariates, and the desirability of interpretable illumination of predictive structure makes ANNs seemingly inappropriate. For instance, it is difficult to assess the respective contributions of anchor and non-anchor positions. Unlike Milik et al's (1998) resorting to a property representation, citing overfitting and management concerns when using ANNs with amino acids themselves, others (Gulukota et al., 1997; Honeyman et al., 1998) have used ANNs with amino acid inputs. While direct comparisons are clearly necessary, the fact that the latter authors obtain sensitivities and specificities $\leq 80\%$ on an independent test sample as well as not gaining any insight into position importance and/or interactions, supports use of tree or bump-hunt approaches.

The salient feature of tree methods re unordered categorical covariates is the flexible, indeed exhaustive, and automated handling of *groups* of levels. This is appealing in that it bypasses the need for computing, examining and grouping individual regression coefficients corresponding to the myriad indicators needed. Further, variable integrity is preserved, interactions accommodated, and easy interpretation/prediction facilitated via the associated tree scheme.

An often noted deficiency of tree-structured methods is that, by virtue of fitting piecewise constant response surfaces, they perform poorly with respect to prediction when faced with smooth response surfaces. This, in part, motivated Friedman's (1991) multivariate adaptive regression spline (MARS) extension of regression trees. However, here such concerns are moot. The very notion of a smooth response surface presupposes the existence of *ordered* covariates – otherwise there is nothing to be smooth with respect to. So, when dealing solely with genotype information represented by unordered categorical covariates the above criticism does not apply.

Finally, both more experience with, and software refinement of, the recently devised and highly

promising Friedman and Fisher (1999) bump-hunt methodology is indicated. For example, the performance differences exhibited when using amino acid versus property variable sets (Tables 2 and 3) reflect implementation limitations. The many-to-one mapping from amino acids to their properties means that we ought not do worse using amino acids as is seemingly the case.

Acknowledgements

This work was supported by NIH grants AI40906 and AI39932. The authors thank Mariusz Milik for providing data and Phil Spector for providing software. Professors Bickel, Friedman, Hastie and Tibshirani and two anonymous referees provided very helpful comments.

References

1. Bickel PJ, Cosman PC, Olshen RA, Spector PC, Rodrigo AG, Mullins JI. (1996) Covariability of V3 loop amino acids. *AIDS Research and Human Retroviruses*, **12**:1401-11.
2. Breiman L, Friedman JH, Olshen RA, Stone CJ. (1984). *Classification and Regression Trees*. Belmont: Wadsworth.
3. Breiman L. (1996). Bagging predictors. *Machine Learning*, **24**:123-40.
4. Fisher WD. (1958). On grouping for maximum heterogeneity. *Journal of the American Statistical Association*, **53**:789-98.
5. Friedman JH. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, **19**:1-67.
6. Friedman JH, Fisher NI. (1999). Bump hunting in high-dimensional data. *Statistics and Computation*, In Press.
7. Gulukota K, Sidney J, Sette A, DeLisi C. (1997). Two complementary methods for predicting peptides binding major histocompatibility complex molecules. *Journal of Molecular Biology*, **267**:1258-67.
8. Hastie TJ, Tibshirani RJ. (1990). *Generalized Additive Models*. London: Chapman and Hall.

9. Hastie TJ. (1998). Neural networks. In *Encyclopedia of Biostatistics*, Armitage P, Colton T. (eds.). New York: Wiley.
10. Hastie TJ, Tibshirani RJ, Eisen M et al., (2000). Gene shaving: A new class of clustering methods for expression arrays. *Genome Biology*, **1**:research0003.1-0003.21.
11. Honeyman MC, Brusica V, Stone NL, Harrison LC. (1998). Neural network-based prediction of candidate T-cell epitopes. *Nature Biotechnology*, **16**:966-69.
12. Kidera A, Konishi Y, Oka M, Ooi T, Scheraga HA. (1985). Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *Journal of Protein Chemistry*, **4**:23-55.
13. Lewontin RC. (1992). Genotype and phenotype. In *Keywords in Evolutionary Biology*, (Keller EF, Lloyd EA, eds.). Cambridge MA: Harvard University Press.
14. McCullagh P, Nelder J. (1989). *Generalized Linear Models*, New York: Chapman and Hall.
15. Milik M, Sauer D, Brunmark AP, Yuan L, Vitiello A, Jackson R, Peterson PA, Skolnick J, Glass CA. (1998). Application of an artificial neural network to predict specific class I MHC binding peptide sequences. *Nature Biotechnology*, **16**:753-6.
16. Rammensee H-G, Friede T, Stefanovic S. (1995). MHC ligands and peptide motifs: first listing. *Immunogenetics*, **41**:178-228.
17. Ripley BD. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.
18. Scott JK, Smith GP. (1990). Searching for peptide ligands with an epitope library. *Science*, **249**:386-390.
19. Segal MR. (1995). Extending the elements of tree-structured regression. *Statistical Methods in Medical Research* **4**:219–236.
20. Therneau TM, Atkinson EJ. (1997). An introduction to recursive partitioning using the RPART routines. *technical report: Mayo Foundation*
21. Xia X, Li W-H. (1998). What amino acid properties affect protein evolution? *Journal of Molecular Evolution*, **47**: 557-64.

22. Zhang C, Anderson A, DeLisi C. (1998). Structural principles that govern the peptide-binding motifs of class I MHC molecules. *Journal of Molecular Biology*, **281**: 929-47.

Table 1. Selected Data

obs	bind	pos1	pos2	pos3	pos4	pos5	pos6	pos7	pos8	molwt1	molwt2	molwt8
1	1	S	S	P	S	H	P	G	M	105.09	105.09	149.21
2	1	S	M	I	T	F	T	P	L	105.09	149.21	131.17
3	1	S	M	V	A	P	P	H	L	105.09	149.21	131.17
4	1	Y	S	P	P	Y	S	S	I	181.19	105.09	131.17
307	0	S	P	S	N	P	S	V	F	105.09	115.13	165.19
308	0	T	P	Y	S	R	P	P	T	119.02	115.13	119.02
309	0	P	Y	S	R	P	P	T	P	115.13	181.19	115.13
310	0	Y	S	R	P	P	T	P	R	181.19	105.09	175.20

Table 2. Bump-Hunting: Amino Acids

Dataset	Overall Binding	Box Binding	Box Support
Training	54.4%	70.4%	0.74
Test	66.3%	74.7%	0.84

(a) Box Summaries

Box Definition	Remove Variable	
	Binding	Support
pos8 $\overline{\{C, P, S\}}$	66.3%	1.00
pos5 $\overline{\{P, W\}}$	74.2%	0.86

(b) Defining Variable

Table 3. Bump-Hunting: Property Variables

Dataset	Overall Binding	Box Binding	Box Support
Training	54.4%	85.9%	0.59
Test	66.3%	94.3%	0.67

(a) Box Summaries

Box Definition	Remove Variable	
	Binding	Support
vol5 > 0.527	66.3%	1.00
flex1 < 0.855	83.7%	0.77
pol8 < 0.685	89.3%	0.72

(b) Defining Variables

Figure Captions

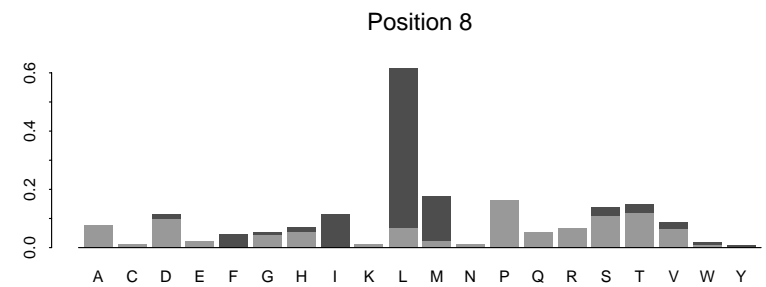
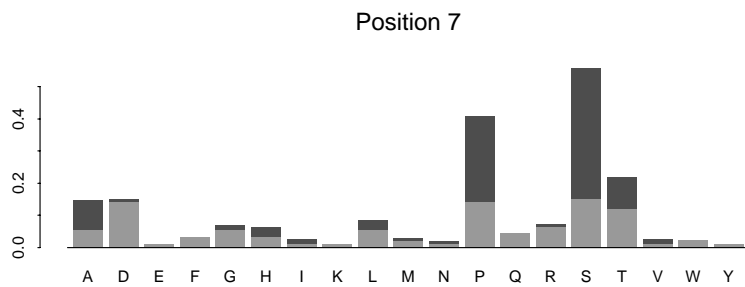
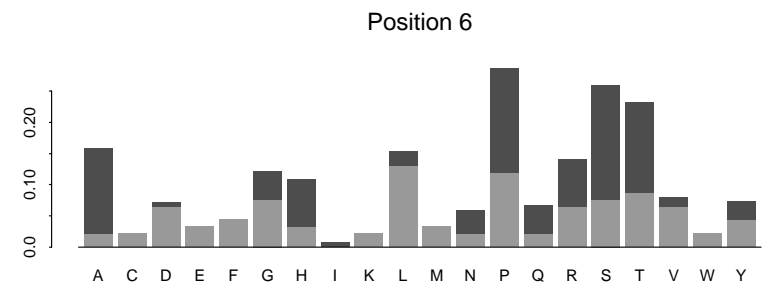
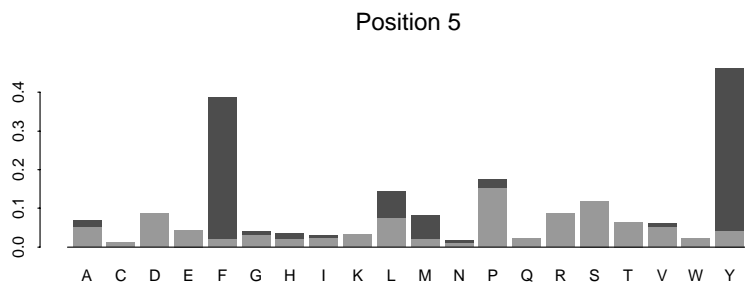
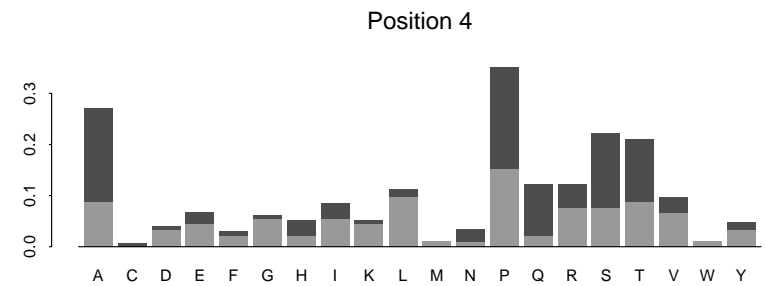
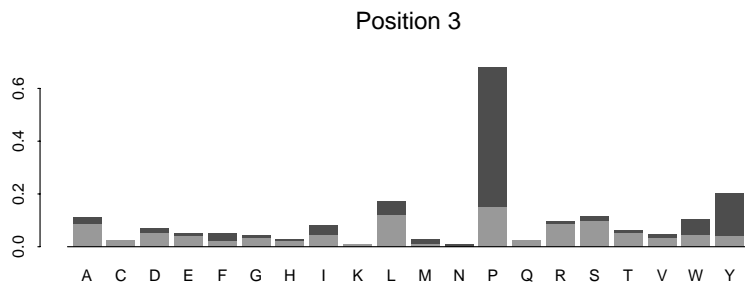
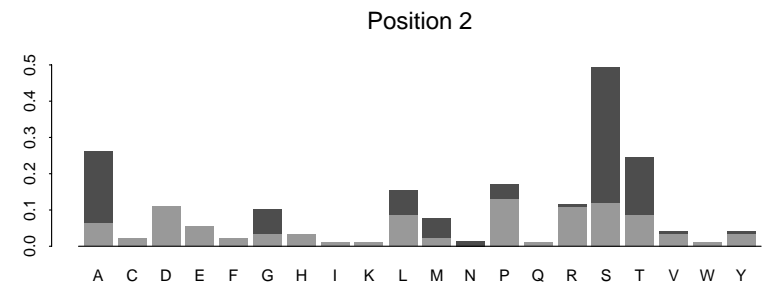
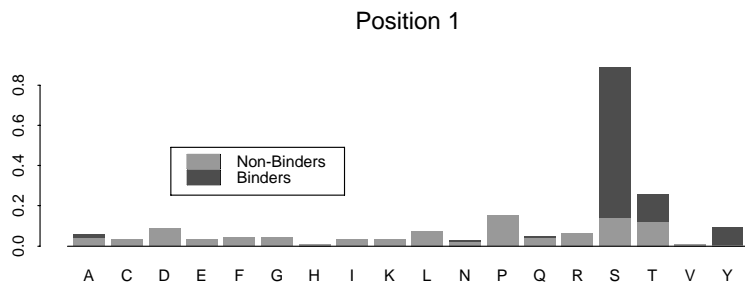
Figure 1: Frequencies for amino acids at each of the positions stratified by binding status.

Figure 2: Initial large classification tree for predicting peptide binding grown using the training data.

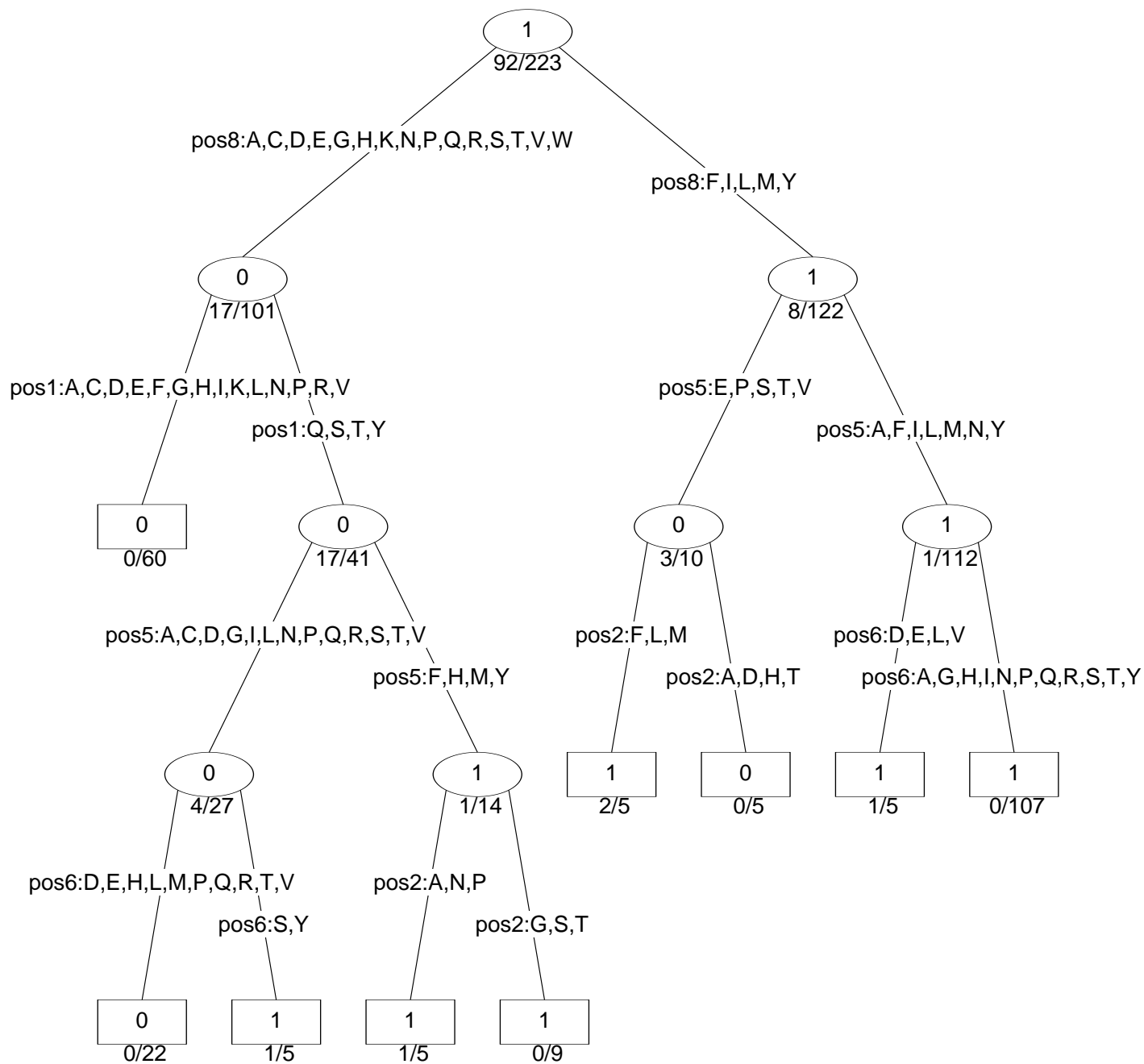
The predicted class (based on simple majority rule) at each node (ellipses – internal; rectangles – terminal) is given by the 0 (non-binding) or 1 (binding) given within each node, while the ratio below gives number misclassified / node size. Numerals above each node are used for identification purposes.

Figure 3: Results from classifying the independent test data using the selected tree.

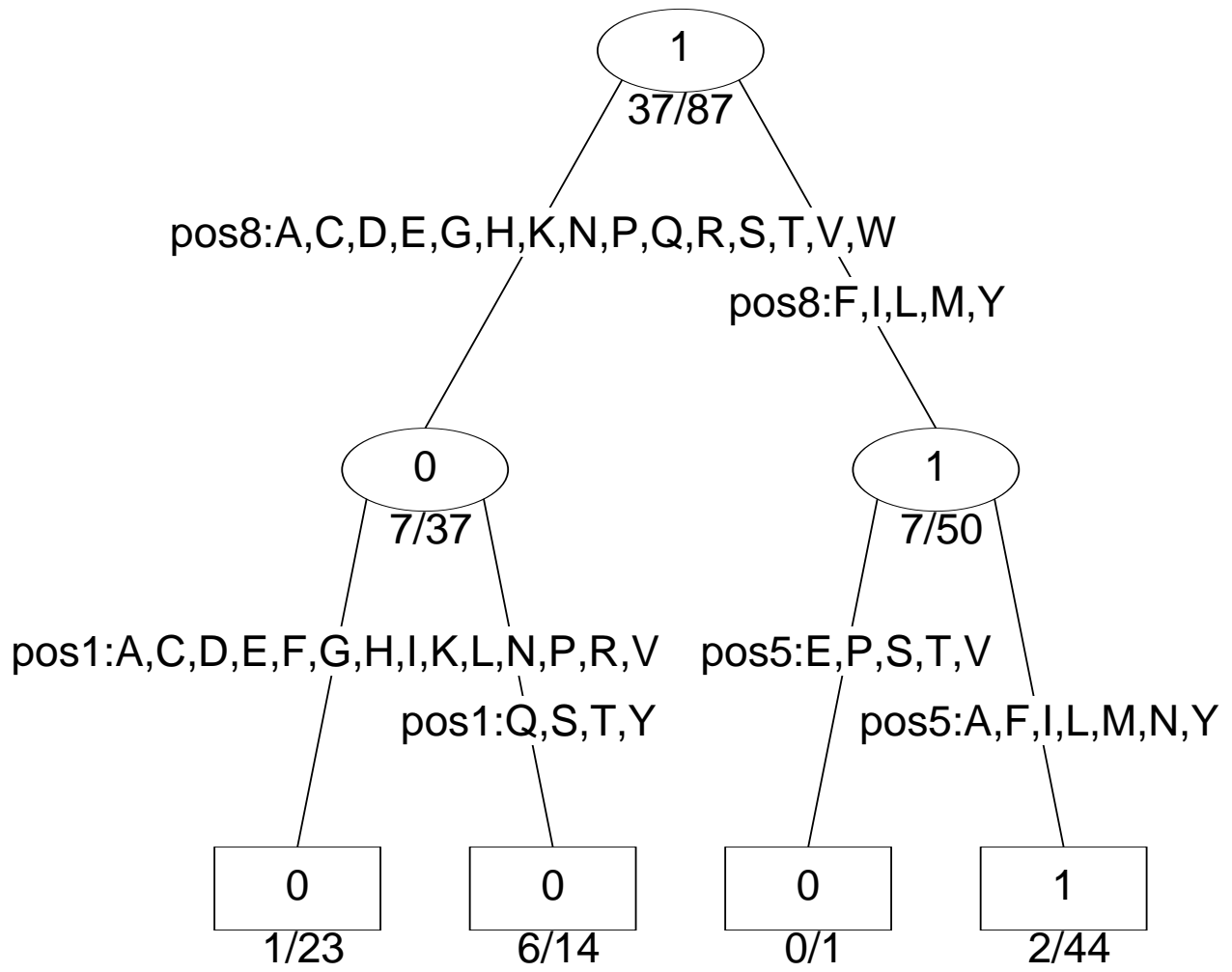
Figure 4: Results from classifying the independent test data using the tree based on property variables.



Full Tree // Training data



Predictions: test data



Predictions: test data

