Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Statistical Planning and Inference 130 (2005) 277–298

journal of statistical planning and inference

www.elsevier.com/locate/jspi

Vector quantization of amino acids: Analysis of the HIV V3 loop region

A.B. Olshen[a,*], P.C. Cosman[b], A.G. Rodrigo[c], P.J. Bickel[d], R.A. Olshen[e,f,g]

[a] *Department of Epidemiology and Biostatistics, Memorial Sloan-Kettering Cancer Center, 307 East 63rd St., Third Floor, New York, NY 10021-6094, USA*

[b] *Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407, USA*

[c] *School of Biological Sciences, University of Auckland, Auckland 1020, NZ, USA*

[d] *Department of Statistics, University of California at Berkeley, Berkeley, CA, 94720-3860, USA*

[e] *Department of Health Research and Policy of the Stanford University School of Medicine, Stanford, CA 94305, USA*

[f] *Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA*

[g] *Department of Statistics, Stanford University, Stanford, CA 94305, USA*

Received 4 August 2003; accepted 7 October 2003

In Honor of Herman Chernoff's 80th birthday

Available online 29 July 2004

Abstract

This paper is about techniques for clustering sequences such as nucleic or amino acids. Our application is to defining viral subtypes of HIV on the basis of similarities of V3 loop region amino acids of the envelope (*env*) gene. The techniques introduced here could apply with virtually no change to other HIV genes as well as to other problems and data not necessarily of viral origin. These algorithms as they apply to quantitative data have found much application in engineering contexts to compressing images and speech. They are called *vector quantization* and involve a mapping from a large number of possible inputs into a much smaller number of outputs. Many implementations, in particular those that go by the name generalized Lloyd or k-means, exist for choosing sets of possible outputs and mappings. With each there is an attempt to maximize similarities among inputs that map to any single output, or, alternatively, to minimize some measure of *distortion* between input and output. Here, two standard types of vector quantization are brought to bear upon the cited problem

[*] Corresponding author.
E-mail address: olshena@mskcc.org (A.B. Olshen).

0378-3758/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.jspi.2003.10.010

of clustering V3 loop amino acid sequences. Results of this clustering are compared to those of the well known UPGMA algorithms, the unweighted pair group method in which arithmetic averages are employed.

Keywords: Vector quantization; HIV; Clustering

1. Introduction

In electrical engineering, algorithms for clustering quantitative data known as *vector quantization*, or VQ, have been applied for the most part to compressing images and speech. Vectors of data that correspond to image pixels or sampled speech are clustered into groups, and a centroid is chosen to represent each group. Regardless of the particular area of application, always subsequent vectors are "quantized" into a more compact set by being mapped to one of the allowed group of representatives. Here we adapt VQ to qualitative data, with particular application being made to sequences of amino acids of the V3 loop region of the HIV *env* gene. The purpose of clustering is not to compress data, but rather to explore the covariation of amino acids at different sites. Insights gained thereby may help with understanding: the epidemiology of the disease, the interactions between the human immune system and virus that result in genetic diversity, and possibly even the relationships between genomic structure and function.

Because of its critical functionality and high variability (Wain-Hobson, 1994), the envelope gene has been studied intensively. It harbors variable and conserved regions. The variable regions include many sites of antibody or cytotoxic T-lymphocyte recognition; in addition, sequence variation in some regions correlates with phenotypic variation of the virus (Hoffman et al., 2002). In particular, the third variable region, or V3 loop, has been shown to be the principal neutralizing domain of the envelope gene in T-cell line adapted viruses. (Hwang et al., 1992); and differences in the V3 loop are the primary determinants of cell tropism and cytopathicity. Each of these properties is now understood to result from the ability of the virus to use alternate co-receptors for cell entry (Bjorndal et al., 1997). Amino acids from sequences of the V3 loop region have been examined for covariation in several different ways. Korber et al. analyzed a set of 308 sequences of the first 32 V3 loop amino acids from the 1991 AIDS database (Korber et al., 1993). Their goal was to identify pairs of sites where mutations would "with high confidence be identified as covarying." Using a statistic with an information theoretic basis, they advanced a set of seven pairs of covarying sites that seemed to merit further analysis. This work was extended in several ways in (Bickel et al., 1996), in which the 308 sequences and a set of 440 sequences from the 1993 AIDS database were examined with the statistic of (Korber et al., 1993), as well as with two other related measures of covariability. This later work also approached covariation primarily in a pair-wise fashion, but it included some results on the existence of cliques (sets of more than two sites that appear to act in concert) and on the extent to which some particular sites are critical to interactions. For example, an apparently strong pairwise interaction between sites 13(H)[1] and 25(D) in the 308 data set was found to be likely due to the two interactions between 11 and 13 and 11 and 25.

The methodology and results of this paper continue the study of variation and covariation of the V3 loop region residues, but they are not based upon pairs of sites. Our new methods attempt to cluster sequences using *all* sites. Unlike methods based on phylogenetic trees, our algorithms are concerned with maximizing similarity rather than reconstructing evolutionary relationships. While the final results may be similar, our methods do not explicitly take into account differential rates of evolutionary change (or substitution) across sites, the possibility of multiple substitutions that mask changes, etc. In designing microbial vaccines, it is common practice to use antigens derived from a single laboratory strain of the agent. If such a vaccine is to work well, it is likely that the antibodies it raises would be effective against viral variants that are structurally similar to the original strain. Classification by evolutionary relationships does not a priori guarantee that members of the same group will share a high degree of structural identity. The methods by which we cluster are chosen explicitly to maximize a particular measure of similarity; of course, other such measures could be employed. If a particular measure of similarity entails also similarity of efficacy for a vaccine, then, presumably, when a vaccine works well against one member of a group it is likely to do the same against other members of the same group. In view of these considerations, it seems better here to cluster without regard to the known phylogenetic subtypes of HIV-1 (Korber et al., 1997). Though we are unable to claim that our simple measure of similarity for sequences necessarily reflects their structural closeness, it is probably still best if our methods are used incorporating a measure of similarity but without being confounded with cladistic similarity.

Phylogenetic trees were used by Korber, Myers and others to subdivide the data sets (Myers et al., 1991; Seiller-Moiseiwitsch et al., 1994), because it was recognized that the covariation that was observed statistically could be the result not of the constraints of protein structure or functional relation driven by selection but rather of “an evolutionary heritage from distinct founder viruses.” A phylogenetic analysis based on long (883) site stretches of the *env* gene produced largely consistent trees ending in seven clades whose geographical clustering is consistent with the history of the epidemic. The same caveat of “founder viruses” applies to this study. However, the purpose of the methods described here is simply to achieve a classification based on similarity, and not to attempt to explain the “why” or “how” underlying the clusters. This goal is also seen in (Korber et al., 1994), where sequences of amino acids from the V3 loop were clustered by a phenetic principle whereby amino acid identities and similarities are evaluated without regard to evolutionary relationships. Various trends in V3 loop protein sequences were revealed by carrying out the phenetic clustering in conjunction with phylogenetic classifications of subtypes.

The set of 440 sequences from 1993 and the 308 sequences studied in (Korber et al., 1993) had 152 sequences in common (identical for the 32 residues considered). However, the two sets differ in many ways, which is not surprising. The epidemic is dynamic; and neither set can be viewed as a random sample from the populations of HIV viruses as they existed on or before 1991 and 1993, respectively. Biases from epidemiological clustering,

[1] Our numbering adds one to that of Korber et al. (1993) and Bickel et al. (1996). The letter following the number gives the consensus value at the site for their set of 308 sequences.

differential sampling of patient populations, and variability of viruses within patients are undoubtedly substantial. Thus, covariability can be an artifact of sampling. This places severe limitations on any conclusions that we or earlier investigators can draw from these data. We consider here the 440 sequence data set (Korber et al., 1993), reduced to 434 sequences by elimination of six sequences for which some values were missing. Our goal in this work is only partly to provide meaningful analyses of covariation within the V3 loop; one major goal is to introduce novel statistical techniques that may prove useful in other applications.

2. Methodology

In this paper, we develop tools to explore interactions among groups of sites. Rather than examining pairs or triples of sites, we take an approach in which all the sites are considered together. The algorithms are our adaptations of what in the electrical engineering literature is called *pruned tree-structured vector quantization* (PTSVQ or TSVQ) (Riskin and Gray, 1991; Gersho and Gray, 1992) and *full search vector quantization* (FVQ) (Gersho and Gray, 1992). They have been applied with considerable success to the lossy (i.e., not invertible) compression of radiographic and satellite images, as well as of digitized speech (Abut, 1990; Cosman et al., 1994, 1996). The algorithms require as their raw material Euclidean vectors, that is, sequences of known, fixed length of ordinary numbers. Yet our amino acid sequences are qualitative; there is no natural numeric ordering of amino acids. So we must translate the sequences of residues to the requisite vectors. With each site j we associate an Euclidean vector of dimension $\#A(j)$, the number of different residues seen in the data set at that site. Therefore, each sequence is associated with a point in the Euclidean space $R^{|A|}$, where

$$|A| = \sum_{j=1}^{32} \#A(j).$$

For a particular amino acid sequence of length 32, the coordinates that correspond to site j are coded to 0 or 1; 1 for the coordinate that corresponds to the residue that appears and 0 for all other coordinates associated with that site. (The order by which amino acids are assigned to coordinates is immaterial because our algorithms and inferences do not change if we permute that order in any fashion.) Fig. 1 is an example of the translation process for the first 4 residues of each of 5 sequences. In this example, site 1 (S1) occupies 3 coordinates of the binary vector. In the 5 sequences shown, only two amino acids appear (C and Y); but in the full data set, S appears as well, and so all vectors must maintain a coordinate position for this residue as well. Site 2 (S2) has 8 different residues appearing (T,I,S,M,A,V,E,L) in the full data set and thus contributes 8 components to the binary vector, site 3 (S3) has 2 (R,S) and site 4 (S4) has 3 (P,L,H). Thus for these 4 sites, the binary vector has $3+8+2+3=16$ coordinates. For the 32 sites in our data, the numbers of different residues (the values of $\#A(j)$) ranged from 2 to 12, and the binary vector had a total length of 259 coordinates.

The distance $d_j(\mathbf{x},\mathbf{y})$ between two vectors $\mathbf{x}$ and $\mathbf{y}$ at site j is simply the ordinary Euclidean distance. Thus, if $\mathbf{x}_j$ and $\mathbf{y}_j$ denote, respectively, the subvectors of length $\#A(j)$

$$\begin{array}{lcllll}
\text{CTRH}\ldots & \longrightarrow & 100 & 10000000 & 10 & 001\ldots\\
\text{CMRP}\ldots & \longrightarrow & 100 & 00010000 & 10 & 100\ldots\\
\text{YTRP}\ldots & \longrightarrow & 001 & 10000000 & 10 & 100\ldots\\
\text{C I SP}\ldots & \longrightarrow & 100 & 01000000 & 01 & 100\ldots\\
\text{CTRL}\ldots & \longrightarrow & \underbrace{100}_{\text{S1}} & \underbrace{10000000}_{\text{S2}} & \underbrace{10}_{\text{S3}} & \underbrace{010}_{\text{S4}}\ldots
\end{array}$$

Fig. 1. The translation process for the first 4 sites of 5 different sequences.

of $\mathbf{x}$ and $\mathbf{y}$ corresponding to the coordinate positions for site j, then

$$d_j(\mathbf{x},\mathbf{y}) = \{(\mathbf{x}_j - \mathbf{y}_j)^{\mathrm{t}}(\mathbf{x}_j - \mathbf{y}_j)\}^{1/2},$$

where $\mathbf{x}^{\mathrm{t}}$ denotes the transpose of a column vector $\mathbf{x}$. These differences between coordinates are all equal to zero, or else at most two of them have magnitude one. In general, we have

$$d_j(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{x} \text{ and } \mathbf{y} \text{ match at site } j,\\ \sqrt{2} & \text{if } \mathbf{x} \text{ and } \mathbf{y} \text{ differ at site } j.\end{cases}$$

Any positive definite symmetric matrix $\mathbf{B}_j$ of size $\#A(j) \times \#A(j)$ could give rise to a distance measure for the subvectors:

$$\rho_j(\mathbf{x},\mathbf{y}) = \{(\mathbf{x}_j - \mathbf{y}_j)^{\mathrm{t}}\mathbf{B}_j(\mathbf{x}_j - \mathbf{y}_j)\}^{1/2}.$$

This formulation would allow one to incorporate refinements such as declaring that at site j, amino acids R and D are farther apart than are R and K. This could be useful, for example, to reflect the fact that R and K both have positively charged side chains, whereas D has a negatively charged side chain. Suppose for the moment that there existed a site j with only 3 different residues appearing (R,K,D). A sequence that had R at that site would have a binary subvector $\mathbf{x}_j = 100$ for that site. Similarly, sequences with K or D there would have, respectively, binary subvectors $\mathbf{y}_j = 010$ and $\mathbf{z}_j = 001$. If we wished to incorporate the idea that R and D are, say, 2 times farther apart than are R and K, we could use a positive definite weighting matrix such as

$$B_j = \begin{bmatrix} 3 & 2 & -1\\ 2 & 3 & -1\\ -1 & -1 & 3 \end{bmatrix}$$

in which case the distance between R and K at the site would be

$$d_j(\mathbf{x}_j,\mathbf{y}_j) = \{(\mathbf{x}_j - \mathbf{y}_j)^{\mathrm{t}}\mathbf{B}_j(\mathbf{x}_j - \mathbf{y}_j)\}^{1/2} = \sqrt{2},$$

whereas the distance between R and D at the site would be

$$d_j(\mathbf{x}_j,\mathbf{z}_j) = \{(\mathbf{x}_j - \mathbf{z}_j)^{\mathrm{t}}\mathbf{B}_j(\mathbf{x}_j - \mathbf{z}_j)\}^{1/2} = 2\sqrt{2}.$$

In our work thus far we have used only the identity matrix, which is to say that all differences at a given site are considered equivalent. They are characterized by the directions and relative lengths of their principal axes. The actual numerical values that appear in **B** depend upon the base coordinitization of the underlying Euclidean space. In particular, that some values may be negative in particular implementations is immaterial. The overall distance $D(\mathbf{x}, \mathbf{y})$ between vectors **x** and **y** can be obtained as a weighted sum of the distances for the separate sites. Highly variable regions of eukaryotic, prokaryotic, and presumably viral, genomes are generally more susceptible to the accumulation of mutational noise that masks the evolutionary or functional relationships with other such genomes. It is plausible to suggest that a match between two sequences at a highly variable site would more likely be the result of chance than would a match at a highly conserved site. This would suggest choosing weights that vary inversely with the number of amino acids seen at the site. We propose something very close. If $p_1, \ldots, p_{\#A(j)}$ are the probabilities of the acids at site j, the probability of mismatch between two sequences at the site is

$$P(\text{mismatch}) = 1 - \sum_{k=1}^{\#A(j)} p_k^2 \leqslant 1 - \frac{1}{\#A(j)}.$$

Thus, we focused on a weighting that was inversely proportional to $1 - 1/\#A(j)$, and

$$D(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{32} \frac{d_j^2(\mathbf{x}, \mathbf{y})}{1 - 1/\#A(j)} \right)^{1/2}.$$

As it turned out, the weighting scheme was subtle enough that it had little impact on our results.

In tracing evolutionary relationships, one would similarly advance the argument that a match between two sequences at a highly variable site is not particularly indicative of a common ancestor remaining unchanged. As two sequences diverge, increasing numbers of changes are masked by multiple substitutions, so that it may appear that two taxa or species are closely related when in fact they are truly distantly related. In phylogenetic reconstruction, correction factors based on a Poisson model of substitution can be used to correct the apparent distance between pairs of species. Our preliminary approach to weighting is a crude attempt at accounting for the actual degrees of “similarity” represented by matches between sequences at variable or conserved sites. But the framework is general. More general weightings for combining distances from different sites, and potentially, weightings for distances between residues at any given site, can be incorporated directly into the algorithm.

Our clustering methods divide the sequences into subsets that provide an exhaustive partition of “sequence space.” The goal is to describe the probability mechanism by which the sequences might have been generated without explicit regard for evolution, and to search for a simple model. This model should approximate nature closely enough to lend insight, on the one hand, but should allow for arbitrary covariation of sites on the other. Suppose that for our distinct clusters of sequences, we could compute both a centroid for each cluster and a probability weight associated with the cluster. (The weights would be non-negative and sum to 1.) We would, thereby, have represented the joint distribution by which the sequences were generated as a “mixture” of probability distributions, each a point mass concentrated on its respective cluster centroid, with “mixing weights” the probability associated with the cluster. We do not pretend that these models are entirely accurate or descriptive; but they, like others of our statistical approaches, are intended to simplify the difficult process of summarizing the data, and to do the best we can given the sparseness of the available observations and the highly multivariate nature of any "correct" model.

The empirical distribution of our $N = 434$ vectors can be viewed as a histogram of the sequences. If we want to summarize the histogram by one vector, an obvious choice is the centroid, that is, the average. It has the well known property of minimizing over all possible choices of a single vector the average over the sample of the squared distance from it to points in the sample. This average squared distance is called *distortion* in the engineering literature. In the language of the previous paragraphs, we are approximating the probability mechanism by which the set of sequences was generated by a single point mass at the centroid. If the $|A|$-dimensional Euclidean space is partitioned into m disjoint regions which each contain some of the N sequences, one can calculate a centroid for each component of the partition. Tree-structured VQ is a hierarchical or nested series of partitions in which we attempt to minimize the average distortion between the vectors in the data set and their partition centroids. Full search VQ is an arbitrary (non-nested) partitioning of the space. Both tree-structured and full search vector quantizers can be designed by the *generalized Lloyd algorithm* (GLA), an iterative method for improving a set of clusters (Linde et al., 1980; Gersho and Gray, 1992). It begins with a set of N vectors and m cluster centroids. The cluster centroids can be obtained in a variety of ways, which will be outlined later. The GLA alternates between the two steps of assigning sequences from the sample to the nearest cluster centroid, and relocating the old centroids to become the centroids of the sequences that were most recently assigned to them.

2.1. Pruned tree-structured vector quantization

The output of these analyses are binary trees. Binary trees are familiar from other statistical scenarios and from phylogenetic analyses with PAUP and other software (Swofford and Olsen, 1990). With the latter, one tries to describe the evolutionary history of a set of sequences so that the branching that led to their observed distribution is summarized as parsimoniously as possible. Our approach does not preclude trees with topologies like those of cladistic origin, but the motivation for them and the algorithms by which they are produced are based solely on empirical distributions—in this case of amino acid sequences—and how they cluster.

The design of the tree-structured classifier begins with the global centroid, and we denote by D_{root} the value of the distortion for the entire data. Now suppose that we partition the $|A|$-dimensional Euclidean space of sequences in any fashion by a hyperplane, that is, by whether a fixed linear combination of the coordinates of each vector is positive or negative. (There is no loss in restricting ourselves to fixed *linear* combinations of coordinates. If our task is to assign points to cluster centers so as to minimize distortion, then for any assignment of centers and fixed matrices $\mathbf{B}_j$ and fixed *nonlinear* boundaries to our partition, there will always be a partition with hyperplane boundaries that has distortion at least as small as has the partition with nonlinear boundaries.) On each side of the partition we find the centroid. Each sample point is considered to belong to the centroid to which it is closer. This partition is called the *Voronoi* partition. We ascribe the empirical relative content of each component of

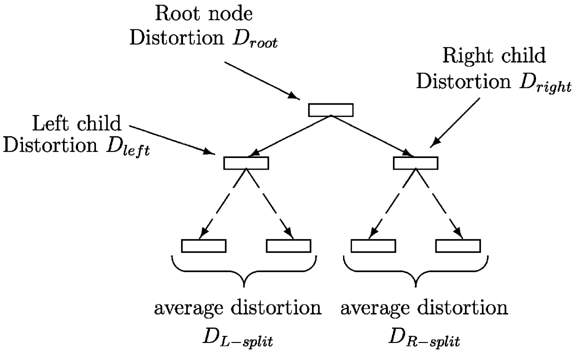

Fig. 2. Successive binary partitioning of the sequence space.

the partition to its corresponding centroid. Thereby, we will have represented the probability mechanism that generated the sample by a two-component probability mixture of the two centroids. We denote by D_{left} and D_{right} the average squared distances from respective centroid within each of the two subsets of the resulting partition. A simple argument based on convexity shows that no matter what linear combination of coordinates we have taken, the average over the sample of these new squared distances will be less than D_{root}, the average squared distance to the overall centroid. This observation leads us to attempt to find the partition (the linear combination of coordinates) for which the reduction in distortion is maximized. The linear combination given by the generalized Lloyd algorithm provides a useful and computationally feasible approximation to the best partition (and sometimes to the actual best partition). Note that while there will be a reduction in the average over the sample of the squared distance to the closest centroid, it need not be the case that there is a reduction within each of the two subsets. These algorithms were adaptations of methods used in decision tree design; they are related to the CART^{R} algorithms for classification and regression (Breiman et al., 1984).

The process of successive partitioning corresponds to the formation of a binary tree. The first step is a *two-means* clustering of the data. The *root node* of the tree can be characterized by its distortion D_{root} and by specification of the linear combination of coordinates that determine its split into daughter nodes. The process continues by taking that daughter node for which we can effect the maximal reduction in distortion by partitioning it and forming new centroids for its two subsets. See Fig. 2. The best partitioning of the left daughter node results in an average distortion $D_{\text{L-split}}$ for those vectors that map into the left daughter node, and similarly for the right daughter node. Of the two candidate splits depicted in Fig. 2 by dashed lines, we make the split that gives the larger reduction in distortion. Thus, the left daughter node is partitioned if

$$D_{\text{left}} - D_{\text{L-split}} \geqslant D_{\text{right}} - D_{\text{R-split}}$$

and otherwise the right daughter node is split.

We now have a rooted binary tree with three terminal nodes (leaves). Next, recurse the algorithm on the three current subsets by choosing the best of the candidate partitions of

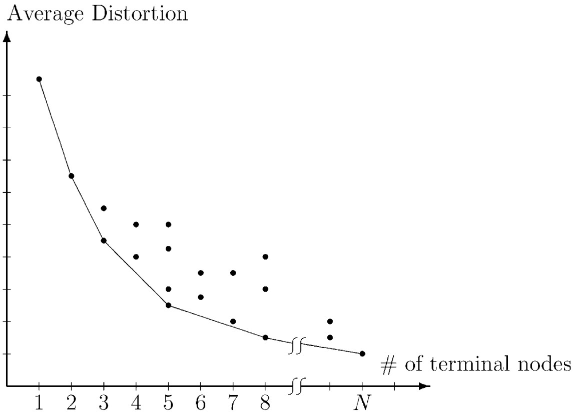

Fig. 3. Distortion vs. # terminal nodes for all possible subtrees of a large initial tree.

the current three leaves. Successive application of the algorithm results in "growing" a successively larger binary tree. Stop when further partitioning reduces the distortion by less than a preassigned small amount. Typically, the process yields a tree larger than what we want for further use. Even though the leaves of the tree might correspond to biologically relevant groups of HIV variants, and although nature's "best" clustering may be very complex, any attempt to find all such groups would result in our having little possibility of useful description. It follows that we must decide which subtrees of the large tree should be retained for further investigation and summary.

Plot a graph in which y is distortion and x is the number of terminal nodes of the tree. The large tree corresponds to a particular point in the graph, in which y is small and x is big. (If all the sequences were identical, the root node alone would be enough; and it would correspond to the point (1,0) on the graph.) Consider all subtrees of the large tree than could be formed by "pruning off" branches at any internal node of that tree. The "subtrees" obtained correspond to various points in the graph, one per subtree. See Fig. 3, in which the large tree has N leaves and the tree that consists only of the root node has largest distortion. The lower convex hull of the set of points is given in Fig. 3 by connected straight line segments. A subtree will be of special interest if it lies on this lower convex hull since there will be no other subtree that has both smaller distortion and not more terminal nodes. These "admissible" trees that comprise the convex hull are nested (Gersho and Gray, 1992, Section 17.4; Breiman et al., 1984, Section 10.2). Obviously, there may be many such admissible trees.

The two references cited make clear that one can give an algebraic interpretation to the previous discussion and Fig. 3. Thus, for some positive number λ and candidate tree $\mathscr{T}$, one can form the criterion

$$D_\lambda = D(\mathscr{T}) + \lambda|\tilde{\mathscr{T}}|,$$

where $D(\mathscr{T})$ is the distortion of the tree $\mathscr{T}$ and $|\tilde{\mathscr{T}}|$ is its number of terminal nodes. For any fixed λ we can find the subtree of the large tree for which D_λ is maximized. For a graphical interpretation see Fig. 17.6 and its surrounding prose in the book by Gersho and Gray (1992, p. 649). The original large tree will be “best” for small but positive values of λ, and it will remain the best tree as λ increases. But, as λ increases, so, too, does D_λ; and at a certain value specific to the data at hand, the large tree will no longer minimize D_λ; instead, an optimally pruned, admissible subtree will. Ultimately, further increase of λ will entail another optimally pruned subtree of the previous subtree, and so on.

In an engineering context, the single tree of most interest is often chosen by its *bit rate*, that is, by its *depth* averaged over the empirical distribution of the data to which it applies. By *depth* is meant the number of steps from root node to terminal node. For (typically lossy, that is, not invertible) data compression, each input vector is represented by the centroid of the leaf in which it lies, and only the binary path map through the tree need be retained in the compressed format. In our case of clustered amino acids, the choice among admissible subtrees, that is, the choice of best λ, is difficult. We want a tree that leads to a simple partition of feature space, one for which some biological meaning can be given to the terminal nodes. Parsimony methods for inferring phylogenies select trees that minimize the total length of the tree, that is, the number of evolutionary steps required to explain a given set of data. In our context, choosing a real minimum for tree size would leave us only with the one cluster at the root node.

One approach to finding a “best” tree is based on finding the break-points of a plot of distortion versus number of terminal nodes, although by itself this approach may be naive (Sugar and James, 2003). There can be large holes in the plot in the sense that trees of many sizes may not be admissible. We chose instead to attempt to use the *gap statistic* (Tibshirani et al., 2001). The technique was developed in a k-means context and chooses the number of clusters for a set of data to maximize the difference (“the gap”) between the observed distortion and the expected distortion. Specifically, if there are k clusters denoted by $C_1, C_2, \ldots, C_k$ of respective sizes $n_1, n_2, \ldots, n_k$, then the measure of distortion used in the gap statistic is

$$W_k = \sum_{i=1}^{k} \frac{1}{2n_i} D_i,$$

where D_i is the sum of pairwise distances for all points in the ith cluster. The statistic W_k is computed for a range of ks. Then the *gap statistic* is defined as

$$\mathrm{Gap}_n(k) = E_n^*(\log(W_k)) - \log(W_k),$$

where E_n^* denotes the expectation based on a sample of size n from an appropriate reference distribution. For us, k is the number of terminal nodes, and the reference distribution is the permutation distribution of independent permutations of amino acids within sites. The uniform distribution over the range of the data, widely used and chosen for certain minimax reasons, is patently inappropriate for our problem. The cited difficulty that owes to there not being admissible trees for many values of k leads to some difficulty in estimating $E_n^*(\log(W_k))$. Our current and not altogether satisfactory solution to this problem is to impute the missing values of the log distortion statistic in every permutation via linear interpolation. Another approach could be to base inference on the “penalty” term λ. There are difficulties, too, with this approach. They are not reported here.

2.2. Full search vector quantization

The tree-structured quantizer is constructed by a constrained search process. Therefore, the subset reached by a test vector as it traverses the tree may not be the best, that is to say closest, one possible for any fixed number of subsets (leaves). To compute the distortion between the test sequence and each one of the leaf centroids would be a *full search* method. Any given set of centroid locations could be improved by running the generalized Lloyd algorithm on the entire data set simultaneously. That is, the final result of the tree-structured clustering with m terminal nodes can itself be treated as the initial starting point for running the generalized Lloyd algorithm, which can only improve the average distortion, but eliminates the simple tree structured relationship among the nodes. A variety of other techniques can also be used for generating an initial set of m clusters (Gersho and Gray, 1992). As is discussed in the next section, there were strong similarities found between the clusters of the tree-structured and full search methods.

Given an initial tree with too many terminal nodes, there exist other ways for obtaining candidate smaller trees, besides looking at the pruned subtrees. We note that one could agglomerate nodes from disparate branches of the initial large tree; this ignores the hierarchical aspect of our TSVQ approach, but much might be gained in parsimony and biological plausibility. Such combining would amount to forming what are called *trellis codes* by engineers and *reticulated classifications* by systematic biologists. This approach might be a post-processor, not only to the tree-structured vector quantizers, but also to the full search quantizers whose descriptions follow. It may be that some systematic biologists would prefer hierarchical to reticulated classifications.

3. Results

3.1. Tree-structured vector quantization

We first grew the tree as large as possible, with the only restriction being that overall distortion must be reduced by at least 1% to continue splitting and that splits that would result in nodes with fewer than 10 sequences could not be made because such small subgroups would be difficult to interpret. These small subgroups did not appear until late in the splitting process. The fully grown tree consisted of 35 nodes, 18 of which were terminal nodes. The leaves ranged in size from 13 to 51. After pruning, the admissible trees had 1, 4, or 10 through 17 terminal nodes. The lack of admissible trees of between 5 and 9 nodes made difficult the task of finding a "best" tree. The plot of distortion vs. number of terminal nodes was not informative for this reason.

The gap statistic, discussed in Methods, was used to help determine the optimal tree. One thousand times, the data were permuted within each site, and the same growing and pruning process utilized on the original data was repeated. For the trees based on permutation, the smallest trees that contained more than one terminal node always contained 18, 19, or 20 terminal nodes; the smallest permutation trees were larger than the largest original trees. To estimate the first term in the gap statistic, we linearly interpolated between the root node tree and the second smallest tree. The gap statistic increased 40% between trees with 4 and

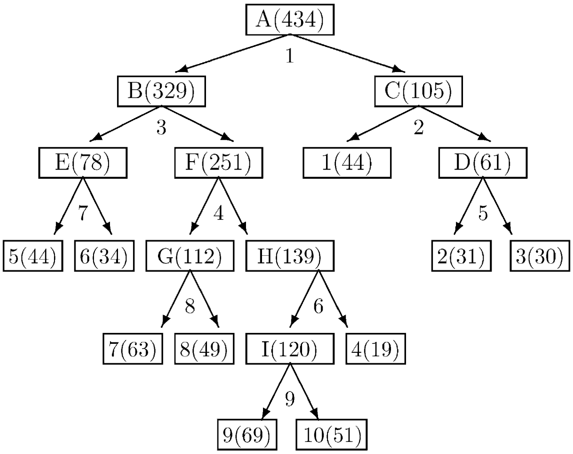

Fig. 4. The 10-leaf tree for the 434 sequences: internal nodes are labeled A–I, and terminal nodes are labeled 1–10. Numbers in parentheses indicate number of sequences belonging to a node. The number associated with a split of the tree indicates the order in which that split was made.

10 terminal nodes. For the trees of 10–17 terminal nodes, the gap fluctuated up and down. The largest gap was actually for the 17 terminal node tree, but we have chosen to report the 10 terminal node tree. The gap is less than 10% smaller than that of the 17 node tree, and the gap statistic is known, in some contexts, to overestimate the optimal cluster size (Dudoit and Fridlyand, 2001).

Fig. 4 contains a diagram of the 10 terminal node tree. Labeling is A–I for internal nodes and 1–10 for terminal nodes. For each node, the number in parentheses indicates the number of sequences that belong to that node. Each split of the tree is illustrated by a pair of diverging arrows and has a number associated with it that indicates the order in which that split was made. It can be seen that the tree is not completely balanced, with as few as two splits or as many as four splits leading to a terminal node. The sizes of the terminal nodes range from as few as 19 in cluster 4 to as many as 69 in cluster 9. Fig. 5 shows the raw data organized by site and the sequences randomly ordered, while Fig. 6 shows the same data with the sequences sorted into clusters. Areas with vertical black lines are homogeneous. Note the difference in homogeneity within sites 10, 22, 25 after clustering. Nevertheless, with this number of terminal nodes, the clusters are not close to being homogeneous within every site. One interesting example is site 6. For this site, 396 of the 434 residues are N. For all but cluster 3, the site is almost homogeneous with just the N residue. For that cluster, the site is very heterogeneous.

It is a difficult task to summarize the results of this clustering process. One way to begin is to examine the consensus sequences for the nodes of the tree. The consensus sequence for a node has as its jth residue that with the largest value among the corresponding $\#A(j)$ coordinates of the centroid for the node. These consensus sequences are presented in Fig. 7. As in Fig. 4, the columns corresponding to internal nodes are labeled A–I, and those for terminal nodes are labeled 1–10. Nodes are called “siblings” if they share the same parent

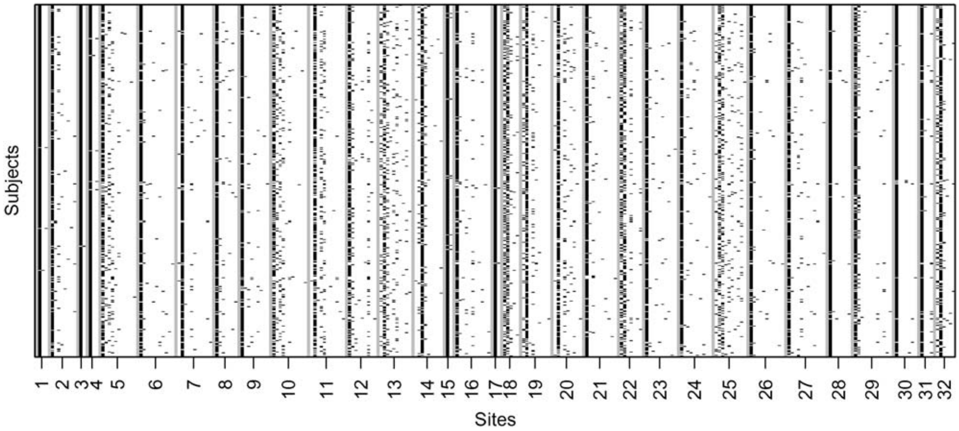

Fig. 5. Heat map of unordered data. Rows are sequences and columns are residues within sites. Vertical gray lines separate sites. Black corresponds to a residue being present and white to absent. Black vertical lines imply homogeneity.

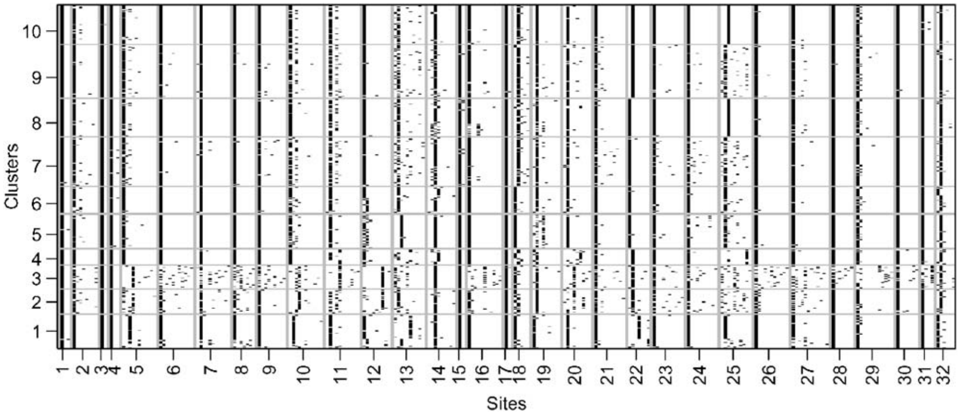

Fig. 6. Heat map of TSVQ ordered data. Details are the same as in Fig. 5, except that sequences are ordered by cluster and that horizontal gray lines separate clusters. Note the increased homogeneity at many sites.

node. Boxed entries indicate cases where the consensus amino acid at a site differs between a pair of siblings. The first column shows the overall (root node) consensus sequence. The next two columns contain data for internal nodes B and C, the two children of the root node. As shown by the boxed entries, their consensus sequences differ in 5 places: sites 5,10,12,18,22. On the simplest level, this suggests that these sites are of particular importance in distinguishing possible biologically relevant groups of HIV variants, and it also raises the possibility that these 5 sites, or some subset of them, vary together in functional ways. Examining differences in other pairs of siblings reveals that sites 12 and 25 are also frequently involved in distinguishing clusters.

	A	B	C	1	D	2	3	E	F	5	6	G	H	7	8	I	4	9	10
1	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
2	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
3	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
4	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
5	N	N	Y	S	Y	Y	Y	N	N	N	N	N	N	N	N	N	N	N	N
6	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
7	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
8	T	T	T	T	T	T	I	T	T	T	T	T	T	T	T	T	T	T	T
9	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
10	K	K	Q	T	Q	Q	Q	K	K	K	K	K	K	K	K	K	K	K	K
11	S	S	S	S	R	S	R	S	S	S	S	S	S	S	S	S	t	S	S
12	I	I	T	I	T	T	T	I	I	I	I	I	I	I	I	I	I	I	I
13	H	H	H	T	H	H	P	R	H	R	H	H	H	H	H	H	H	H	H
14	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	L	I	I
15	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
16	P	P	P	P	P	P	L	P	P	P	P	P	P	P	P	P	P	P	P
17	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
18	R	R	Q	Q	Q	Q	Q	Q	R	Q	Q	R	R	R	Q	R	R	R	R
19	A	A	A	V	A	A	A	A	A	T	A	A	A	A	A	A	A	A	A
20	F	F	F	F	L	L	L	F	F	F	F	F	F	F	F	F	W	F	F
21	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
22	T	A	T	R	T	T	T	A	T	A	A	A	T	A	A	T	T	T	T
23	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
24	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
25	D	D	D	D	R	K	R	D	E	D	D	E	E	D	E	E	Q	D	E
26	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
27	I	I	I	I	I	I	T	I	I	I	I	I	I	I	I	I	I	I	I
28	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
29	D	D	D	D	D	D	Y	D	D	D	D	D	D	D	D	D	D	D	D
30	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
31	R	R	R	R	R	R	G	R	R	R	R	R	R	R	R	R	R	R	R
32	Q	Q	Q	K	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q

Fig. 7. Consensus sequences for all nodes of the 10-leaf tree. The columns corresponding to internal nodes are labeled A–I. The columns corresponding to terminal nodes are labeled 1–10. Boxed entries indicate cases where the consensus amino acid at a site differs for two nodes that have the same parent.

In fact, having siblings that differ in their consensus sequences is not always of paramount importance, since the consensus is sometimes determined by only a very small difference between the most frequent and the next most frequent residues. Instead we can look at which sites, as the result of a split in the tree, experience the greatest difference in their proportions within the resulting two children nodes. These results are presented in Table 1. The nine tree splits are numbered as in Fig. 4. For each of the nine splits, Table 1 shows

Table 1
The 6 sites with the greatest absolute difference in proportions between left and right child as a result of a given split for the 434 sequences

	Sites with greatest "purification"					
Split	1	2	3	4	5	6
1	10K	5N	22A	12I	18Q	12T
2	19A	19V	10T	20F	22T	32K
3	18Q	18R	13R	22A	22T	25D
4	22T	22A	25Q	20F	16W	13H
5	16P	29D	13H	31R	11R	31G
6	20F	25Q	14I	20W	14L	11R
7	13R	13H	19A	14I	19T	32K
8	25E	25D	13H	10K	24G	10R
9	25E	25D	11S	25Q	2I	25K

the six coordinates of the binary vector that result, in order of decreasing magnitude, in the greatest difference in proportions between the children. Thus for the split of the root node, the residue K appearing at site 10 experienced the greatest "purification" as a result of the split of the root node. In actual numbers, 62.0% of the root's sequences had A at site 22. The left and right children of the root node had A appearing in, respectively, 81.1% and 1.9% of their sequences. In this sense we can say that the K residue has been somewhat "purified" by this first split. Site 12 crops up twice for the root node split, since at site 18 both the residues I and T (corresponding to distinct coordinates in the binary vector) become considerably more pure as a result of the split. The root node has 74.0% of I's at site 12, and 10.8% of T's. For the left child of the root node, those numbers become 86.3% and 0%, whereas for the right child, they are 35.2% and 44.7%. Several sites, notably 13, 18, 22, and 25 figure prominently in the table. There are numerous pairs of sites that appear to vary together in that they both register sharp increases in purity for more than one split. For example, (22,10), (22,18), (22,20), (25,23), (25,20), (25,22) and (32,19) are examples where pairs of sites figure in more than one split. Table 1 identifies one triple of sites (25,22,13) that appeared twice.

3.2. Full search vector quantization

Table 2 presents the consensus sequences for the 10 clusters of the FVQ. Here there is no hierarchical relationship between different clusters, and the clusters could be indexed from 1 to 10 in an arbitrary manner. These 10 clusters were compared against those from the tree-structured VQ, and so we chose to index the FVQ clusters to match the most similar TSVQ clusters. For each of the TSVQ clusters ($A_i, i = 1, 2, \ldots, 10$), and for each of the FVQ clusters ($B_j, j = 1, 2, \ldots, 10$) we counted the number of sequences belonging to cluster A_i which also appeared in cluster B_j, and we denote this quantity $\#(A_i \Delta B_j)$. The total number of sequences appearing in either A_i or B_j is denoted $\#(A_i \cup B_j)$. Then a

Table 2
Consensus sequences for the 10 partitions of the full search vector quantizer

	1	2	3	4	5	6	7	8	9	10
#	32	29	29	26	75	10	75	48	19	91
1	C	C	C	C	C	C	C	C	C	C
2	T	T	T	T	T	T	T	T	T	T
3	R	R	R	R	R	R	R	R	R	R
4	P	P	P	P	P	P	P	P	P	P
5	S	Y	Y	N	N	F	N	N	N	N
6	N	N	N	N	N	K	N	N	N	N
7	N	N	N	N	N	K	N	N	N	N
8	T	T	I	T	T	T	T	T	T	T
9	R	R	R	R	R	R	R	R	R	R
10	T	Q	Q	K	K	T	K	K	R	K
11	S	S	R	S	S	S	S	G	R	S
12	I	T	T	I	I	V	I	I	I	I
13	T	H	P	H	R	R	H	H	S	H
14	I	I	I	L	I	I	I	I	I	I
15	G	G	G	G	G	G	G	G	G	G
16	P	P	L	P	P	P	P	P	P	P
17	G	G	G	G	G	G	G	G	G	G
18	Q	Q	Q	R	Q	R	R	R	R	R
19	V	A	A	A	A	V	A	A	A	A
20	F	L	L	W	F	F	F	F	F	F
21	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
22	R	T	T	T	A	K	A	A	A	T
23	T	T	T	T	T	T	T	T	T	T
24	G	G	G	G	G	G	G	G	G	G
25	D	K	R	Q	D	t_1	D	E	t_2	E
26	I	I	I	I	I	T	I	I	I	I
27	I	I	t_3	I	I	I	I	I	I	I
28	G	G	G	G	G	G	G	G	G	G
29	D	D	D	D	D	D	D	D	D	D
30	I	I	I	I	I	I	I	I	I	I
31	R	R	t_4	R	R	R	R	R	R	R
32	K	Q	Q	Q	Q	K	Q	Q	Q	Q

The symbols t_1, t_2, t_3, t_4 denote cases where the consensus amino acid at the site is a tie between E/S/A, G/Q, I/K/T, and R/G, respectively. The row labeled # indicates the number of sequences that belong to the given partition.

measure of closeness between two clusters is given by

$$\rho(A_i, B_j) = \frac{\#(A_i \Delta B_j)}{\#(A_i \cup B_j)}.$$

If two clusters A and B are identical, then $\rho(A, B)=1$. If they are disjoint, then $\rho(A, B)=0$. The values for the various TSVQ and FVQ clusters are given in Table 3. The table also shows the number of sequences in each TSVQ and FVQ cluster. For example, FVQ cluster 1 has 32 sequences, and so the (1,1) entry of $\frac{32}{44}$ indicates that FVQ cluster 1 is completely contained in TSVQ cluster 1. Table entry (3,3) is $\frac{27}{32}$, indicating that TSVQ cluster 3 and FVQ cluster 3 are the two clusters that are closest together.

3.3. Agglomerative cluster analysis

Tree-structured clustering is familiar in a taxonomic context. We wished to compare the tree topology produced by TSVQ against that produced by the UPGMA algorithm (unweighted pair group method using arithmetic averages), which is also known as the average linkage method. We used the implementation provided in the *hclust* function, which is part of the **mva** library in the statistical language R. The UPGMA method operates on raw data provided as a table of distances between all pairs of sequences. We calculated the distances in exactly the same way as for the vector quantization routines. After initializing the process by considering each individual sequence to be a cluster, the UPGMA tree is constructed by successively linking the two least distant clusters. When two clusters are linked, they lose their individual identities and are subsequently referred to as a single cluster. At each stage in the process, two clusters i and j are merged into one, and the total number of clusters declines by one. The distance from the new merged cluster to another cluster k ($k \neq i, j$) is now the average of the distances from i to k and from j to k, where the distances are weighted by the number of sequences contained in i and j. The process is complete when the last two clusters are merged into a single cluster that contains all sequences. Thus the TSVQ and UPGMA clustering methods have very different philosophies. Tree-structured VQ is a divisive or “top-down” clustering method which begins by viewing the entire data set as one cluster, and it successively partitions the input space into smaller pieces. An agglomerative or “bottom-up” clustering method such as UPGMA attempts to form clusters by successively merging together the least distant of the clusters found up to that point.

A UPGMA tree was constructed for the complete data set of 434 sequences. The result is shown in Fig. 8.

4. Discussion

The three methods (TSVQ, FVQ, and UPGMA) all provide a partitioning of the sequence space. One would like to explore the degree of overlap between the sets of clusters. The TSVQ and UPGMA methods both provide a hierarchical summary of sequence similarity, whereas the FVQ does not. In principle, one would like to compare the tree topologies produced by TSVQ and UPGMA as well.

Table 3
Overlap between the FVQ and TSVQ clusters

			TSVQ clusters									
			1	2	3	4	5	6	7	8	9	10
		#	44	31	30	19	44	34	63	49	69	51
	1	32	$\frac{32}{44}$	0	0	0	0	0	0	0	0	0
	2	29	0	$\frac{26}{34}$	0	0	0	0	0	0	$\frac{3}{95}$	0
F	3	29	0	$\frac{2}{58}$	$\frac{27}{32}$	0	0	0	0	0	0	0
V	4	26	$\frac{1}{69}$	0	0	$\frac{18}{27}$	0	$\frac{2}{58}$	$\frac{1}{88}$	$\frac{3}{72}$	0	$\frac{1}{76}$
Q	5	75	$\frac{2}{117}$	0	0	0	$\frac{44}{75}$	$\frac{29}{80}$	0	0	0	0
	6	10	$\frac{9}{45}$	0	$\frac{1}{31}$	0	0	0	0	0	0	0
	7	75	0	0	0	0	0	$\frac{2}{107}$	$\frac{40}{98}$	$\frac{33}{91}$	0	0
	8	48	0	0	0	0	0	$\frac{1}{81}$	$\frac{13}{98}$	$\frac{13}{84}$	$\frac{14}{103}$	$\frac{7}{92}$
	9	19	0	$\frac{3}{47}$	$\frac{2}{47}$	$\frac{1}{37}$	0	0	$\frac{9}{73}$	0	$\frac{4}{84}$	0
	10	91	0	0	0	0	0	0	0	0	$\frac{48}{112}$	$\frac{43}{99}$

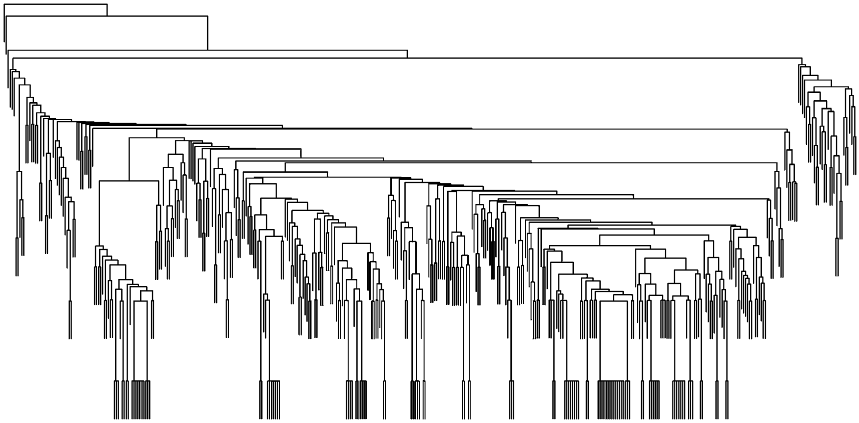

Fig. 8. The UPGMA tree.

Both the TSVQ and FVQ methods produced 10 clusters. The former was chosen, in part, by the gap statistic, while the latter was chosen to be the same size as the former. At one extreme, the UPGMA method can be considered to produce $N = 434$ clusters, where each cluster consists of a single sequence. It is not meaningful to compare the degree of overlap between this clustering and that provided by the VQ methods. In the process of agglomerating, however, the UPGMA method progressively groups those N clusters together into fewer and larger clusters. For purposes of comparison, these clusters could be used as the UPGMA output. This would correspond to a type of pruning on the UPGMA tree. This type of reduction did not prove feasible in practice, given the highly unbalanced nature of the UPGMA tree topology. For example, a reduction to two clusters would have one cluster containing a single sequence, and the other containing 433 sequences. Similarly, a reduction to three clusters would have two containing a single sequence each, and a third containing 432 sequences. This type of tree shape is sometimes referred to as "splintering" because, with a clustering method that uses splitting, it corresponds to repeatedly shaving off small splinters of data from the main group. The TSVQ tree "splinters" much less; other than the first split into 105 and 329 sequences and the third into 78 and 251 sequences, the tree maintains somewhat balanced numbers on the two sides of all splits. When the UPGMA output is pruned back from 434 to 5 clusters, a splinter of 30 sequences is found that has an overlap of 26 (86.7%) with TSVQ cluster 3 (which is also of size 30). However, it proved infeasible to prune the UPGMA down to a reasonable number of clusters without having the majority of them representing tiny splinters (individual sequences, or groups of two). When reduced to 10 clusters, the UPGMA tree had memberships of 399, 27, and 8 singletons. We note that in the analysis of (Korber et al., 1994, involving maximum-linkage clustering on the 15 amino acids spanning the crown of the V3 loop), the same phenomenon of splintering is also apparent in the resulting tree.

The splintering, of course, makes it difficult to compare the UPGMA clusters against the others in any meaningful way. When reduced to 50 clusters, the UPGMA tree had one cluster with 351 sequences (306 of which overlapped with node B of Fig. 4, the TSVQ tree), one with 11 sequences, and 48 with fewer than 10 sequences, 37 of which were singletons. With 100 clusters, there were only four with more than 10 sequences. One had 71 sequences, 69 of which were in Node E of the TSVQ tree. Another had 177 sequences, all of which were in node B. Another had 13 sequences, 10 of which overlapped with node 4. Another had 31 sequences, all of which were in node 1. Interestingly, they were the same 31 sequences that were in a daughter node of 1, which was pruned from the 10 terminal node tree. Clearly, there was some similarity between the UPGMA and TSVQ trees.

It is, of course, not strictly necessary to have a hierarchical summary of sequence similarity. The type of clustering provided by the full search vector quantizer may be all that is required. Table 3 showing the symmetric differences between the TSVQ and FVQ clusters indicates that there is, in any case, a strong similarity between the hierarchical and non-hierarchical VQ summaries of sequence similarity. The first FVQ cluster with 32 sequences is fully contained in a TSVQ cluster 44 sequences. The second and third TSVQ and FVQ clusters are very similar, with overlaps of $\frac{26}{34}$ and $\frac{27}{32}$. The third, as mentioned earlier, overlaps strongly with the second biggest cluster of the 10 cluster UPGMA. TSVQ cluster 4, has 19 sequences, 18 of which are contained in FVQ cluster 4, which is of size 26. TSVQ cluster 5 has 44 sequences, all of them contained in FVQ cluster 5, which is of size 75. As we go deeper into the TSVQ tree, the differences between the two are more substantial. Nevertheless, many of the conclusions that could be drawn from the TSVQ tree clustering could also be drawn from the FVQ clustering.

There should be some reconciliation between the paper by (Bickel et al., 1996) and this one, as it involved many of the same co-authors. That paper even refers to covariation of V3 loop amino acid sites in its title. In the cited paper several different statistics were used to quantify covariation by pairs of sites. At one point we thought that lower within cluster attained significance levels for such statistics would indicate “good” clustering. However, to look for covariation between sites is to ignore invariant sites. One intuitive notion regarding “good” clusters would be those with all invariant sites, that is to say, those that necessarily *lack* significant covariation. While it remains a challenge to combine the two appealing but rather opposing goals, some inferences can be drawn at this point. We compare Fig. 5 and Table 1 here with the previous Table 3 (p. 1407), noting that our numbering scheme is not the same (24 now was 23 in Bickel). Here, sites 3, 4, 6, 9, 15, 17, 21, 23, 24, 26, 28, and 30 have the same consensus amino acid at every node, internal or otherwise. Of these, only site 6 (there 5) has any significant variation in the earlier list of pairs, and (like most other pairs) it does not covary significantly with any other site by what is termed the P-statistic for the set of data analyzed here. Note that here we can consider two amino acids to covary if they figure in Table 1 together at the same split or along any path from root to a terminal node of our tree. Necessarily there are 10 paths. Some, like splits 1–3–7, are quite different from, say, splits 1–2, while others are similar in pairs. Thus, 1–2–5 leads to terminal nodes 2 and 3.

Substitutions at what here are termed sites 11 and 13 have been implicated as determinants of cell tropism, and this was one of the most important connections found before. See Chesboro et al. (1992), Fouchier et al. (1992), Westervelt et al. (1992) and Milich et al. (1993). In Table 1, sites 11 and 13 are listed at 8 of the 54 among those with “greatest purification”.

Chesboro et al. (1992) showed that a single change at site 13 from S to H created a non-infectious virus. Note that S is not the consensus amino acid for any of our nodes, while H is for nearly all. Chesebro et al. also say that altering site 13 in conjunction with sites 21–30 causes a phenotypic switch from T tropic to macrophage tropic. But now study both Fig. 4 and Table 1. Notice that site 13 covaries substantially with sites 21–30 at both splits 3 and 8, and also that split 3 precedes split 8 in a path from the root node to terminal nodes 7 and 8, the second and fourth largest of our terminal nodes. Split 3 is in the path from root node to terminal nodes 9 and 10, respectively the largest and third largest of our terminal nodes.

5. Conclusions

There are many limitations to the analyses presented here. These include the problems associated with our data sets: haphazard rather than representative data ("founder virus effects," unknown epidemiological clustering, unrepresentive sampling from different groups of individuals in terms of geography, disease status, treatment with antivirals, etc., see (Korber et al., 1993)), and the sparseness of the data (because many sites in the V3 loop are not so variable, and the amino acids other than the consensus may appear rarely). It is difficult in any case to summarize the results of the clustering methods in a simple and meaningful way, and it would be difficult to establish significance levels for the various assertions one can make. We also have little theoretical justification for our choice of the 10 terminal node tree. Nevertheless, we believe that the framework of vector quantization is very general, and it is likely to prove a useful tool in the search for simple summaries of genetic variability and covariability. There is flexibility in designing the measure of distortion, in terms of weights at one site (that is, distortion as it applies to two amino acids at a single site) and in terms of combining distortions from different sites. Because the algorithm for design explicitly tries to minimize the distortion at each step, it will tend to maximize the degree of similarity among the sequences in a cluster. If our data are typical, then the algorithm also tends to produce fewer "splinters" than does UPGMA.

Acknowledgements

Research by P.C. Cosman was supported in part by grants from the National Science Foundation; research by A.G. Rodrigo was supported in part by grants from the U.S. Public Health Service; research by P.J. Bickel was supported in part by Grants NSF DMS95-04955 and NSA MDA-904-94-H-2020 to the University of California, Berkeley; research by R.A. Olshen was supported in part by NIBIB/NIH Grant 5ROl EB002784-28 and NSF Grant CCR-073050 to Stanford University. We are grateful to Dr. Bette Korber and Dr. Jim Mullins for helpful discussions.

References

Abut, H., (Ed.), 1990. Vector Quantization. IEEE Reprint Collection. IEEE Press, Piscataway, NJ, pp. 207–331.

Bickel, P., Cosman, P., Olshen, R., Spector, P., Rodrigo, A., Mullins, J., 1996. Covariability of V3 loop amino acids. AIDS Res. Human Retroviruses 12, 1401–1411.

Bjorndal, A., Deng, H., Jansson, M., Flore, J.R., Colognesi, C., Karlsson, A., Albert, J., Scarlatti, G., Littman, D.R., Fenyo, E.M., 1997. Coreceptor usage of primary human immunodeficiency virus type 1 isolates varies according to biological phenotype. J. Virol. 71, 7478–7487.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth, Belmont, CA, pp. 1–173, 216–317. (Since 1993 this book has been published by Chapman & Hall, New York).

Chesboro, B., Wehrly, K., Nashio, J., Perryman, S., 1992. Macrophage-tropic human immunodeficiency virus isolates from different patients exhibit unusual V3 envelope sequence homogeneity in comparison with T-cell-tropic isolates: definition of critical amino acids involved in cell tropism. J. Virol. 66, 6547–6554.

Cosman, P.C., Gray, R.M., Olshen, R.A., 1994. Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. Proc. IEEE 82 (6), 919–932.

Cosman, P.C., Gray, R.M., Vetterli, M., 1996. Vector quantization of image subbands: a survey. IEEE Trans. Image Process. 5, 202–225.

Dudoit, S., Fridlyand, J., 2001. Application of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method. Technical Report 600, Department of Statistics, UC Berkeley.

Fouchier, R.A.M., Groenik, M., Koostra, N.A., Temsette, M., Husiman, H.G., Miedema, F., Schuitemaker, H., 1992. Phenotype-associated sequence variation I and third variable domain of the human immunodeficiency virus type 1 gp120 molecule. J. Virol. 66, 3183–3187.

Gersho, A., Gray, R.M., 1992. Vector quantization and signal compression. Kluwer Academic Publishers, Boston, pp. 184–194, 341–369.

Hoffman, N.G., Seillier-Moiseiwitsch, F., Ahn, J., Walker, J.M., Swanstrom, R., 2002. Variability in the human immunodeficiency virus type 1 gpl20 Env portein linked to phenotype-associated changes in the V3 loop. J. Virol. 76, 3852–3864.

Hwang, S.S., Boyle, T.J., Lyerly, H.K., Cullen, B.R., 1992. Identification of the envelope V3 loop as the primary determinant of cell tropism in HIV-1. Science 257, 535–537.

Korber, B.T.M., Farber, R.M., Wolpert, D.H., Lapedes, A.S., 1993. Covariation of mutations in the V3 loop of human immunodeficiency virus type 1 envelope protein: an information theoretic analysis. Proc. Natl. Acad. Sci. 90, 7176–7180.

Korber, B.T., MacInnes, K., Smith, R.F., Myers, G., 1994. Mutational trends in V3 loop protein sequences observed in different genetic lineages of human immunodeficiency virus type 1. J. Virol. 68, 6730–6734.

Korber, B., Hoelscher, M., McCutchan, F., Williamson, C., von Sonnenburg, F., Mullins, J.I., Pletschette, M., Weber, J., van der Groen, G., Osmanov, S., 1997. HIV-1 subtypes: implications for epidemiology, pathogenicity, vaccines and diagnostics. AIDS 11, UNAIDS17–UNAIDS36.

Linde, Y., Buzo, A., Gray, R.M., 1980. An algorithm for vector quantizer design. IEEE Trans. Comm. COM-28, 84–95.

Milich, L., Margolin, B., Swanson, R., 1993. V3 loop of the human immunodeficiency virus type 1 Env protein: interpreting sequence variability. J. Virol. 67, 5623–5634.

Myers, G., Korber, B.T.M., Berzofsky, J.A., Smith, R.F., Pavlakis, G.F., 1991. Human retroviruses and AIDS. Theoretical Biology and Biophysics Group. Los Alamos Matl. Labs, Los Alamos, NM.

Riskin, E.A., Gray, R.M., 1991. A greedy tree growing algorithms for the design of variable rate vector quantizers. IEEE Trans. Signal Process. 39, 2500–2507.

Seiller-Moiseiwitsch, F., Margolin, B.H., Swanstrom, R., 1994. Genetic variatibility of the human immunodeficiency virus. Ann. Rev. Genet. 28, 559–596.

Sugar, C.A., James, G., 2003. Finding the number of clusters in a data set: an information-theoretic approach. J. Amer. Stastist. Assoc. 98, 750–763.

Swofford, D.L., Olsen, G.J., 1990. Phylogeny reconstruction. in: Hillis, D.M., Moritz, C. (Eds.), Molecular Systematics. Sinauer Associates, Sunderland, MA, pp. 411–501.

Tibshirani, R., Walther, G., Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. J. Roy. Statist. Soc. Ser. B 32, 411–423.

Wain-Hobson, S., 1994. Is antigenic variation of HIV important for AIDS and what might be expected in the future?. in: Morse, S. (Ed.), The Evolutionary Biology of Viruses. Raven Press, New York, pp. 185–209.

Westervelt, P., Trowbridge, D.B., Epstein, L.G., Blumberg, B.M., Li, Y., Hahn, B.H., Shaw, G.M., Price, R.W., Ratner, L., 1992. Macrophage tropism determinants of human immunodeficiency virus type 1 in vivo. J. Virol. 66, 2577–2582.