Protein Classification Based on Text Document Classification Techniques

Betty Yee Man Cheng,[1] Jaime G. Carbonell,[1] and Judith Klein-Seetharaman[1,2]*
[1]*Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, Pennsylvania*
[2]*Department of Pharmacology, University of Pittsburgh School of Medicine, Biomedical Science Tower E1058, Pittsburgh, Pennsylvania*

ABSTRACT **The need for accurate, automated protein classification methods continues to increase as advances in biotechnology uncover new proteins. G-protein coupled receptors (GPCRs) are a particularly difficult superfamily of proteins to classify due to extreme diversity among its members. Previous comparisons of BLAST, k-nearest neighbor (k-NN), hidden markov model (HMM) and support vector machine (SVM) using alignment-based features have suggested that classifiers at the complexity of SVM are needed to attain high accuracy. Here, analogous to document classification, we applied Decision Tree and Naïve Bayes classifiers with chi-square feature selection on counts of *n*-grams (i.e. short peptide sequences of length *n*) to this classification task. Using the GPCR dataset and evaluation protocol from the previous study, the Naïve Bayes classifier attained an accuracy of 93.0 and 92.4% in level I and level II subfamily classification respectively, while SVM has a reported accuracy of 88.4 and 86.3%. This is a 39.7 and 44.5% reduction in residual error for level I and level II subfamily classification, respectively. The Decision Tree, while inferior to SVM, outperforms HMM in both level I and level II subfamily classification. For those GPCR families whose profiles are stored in the Protein FAMilies database of alignments and HMMs (PFAM), our method performs comparably to a search against those profiles. Finally, our method can be generalized to other protein families by applying it to the superfamily of nuclear receptors with 94.5, 97.8 and 93.6% accuracy in family, level I and level II subfamily classification respectively. Proteins 2005;58:955–970.**

Key words: Naïve Bayes; Decision Tree; chi-square; *n*-grams; feature selection

INTRODUCTION

Classification of Proteins

Advances in biotechnology have drastically increased the rate at which new proteins are being uncovered, creating a need for automated methods of protein classification. The computational methods developed to meet this demand can be divided into five categories based on sequence alignments (categories 1–3, Table I), motifs (category 4) and machine learning approaches (category 5, Table II).

The first category of methods (Table I-A) searches a database of known sequences for the one most similar to the query sequence and assigns its classification to the query sequence. The similarity search is accomplished by performing a pairwise sequence alignment between the query sequence and every sequence in the database using an amino acid similarity matrix. Smith–Waterman[1] and Needleman–Wunsch[2] are dynamic programming algorithms guaranteed to find the optimal local and global alignment respectively, but they are extremely slow and thus impossible to use in a database-wide search. A number of heuristic algorithms have been developed, of which BLAST[3] is the most prevalent.

The second category of methods (Table I-B) searches against a database of known sequences by first aligning a set of sequences from the same protein superfamily, family or subfamily and creating a consensus sequence to represent the particular group. Then, the query sequence is compared against each of the consensus sequences using a pairwise sequence alignment tool and is assigned the classification group represented by the consensus sequence with the highest similarity score. The third category of methods (Table I-C) uses profile hidden Markov model (HMM) as an alternative to consensus sequences but is otherwise identical to the second category of methods. Table III shows some profile HMM databases available on the Internet.

The fourth category of methods searches for the presence of known motifs in the query sequence. Motifs are short amino acid sequence patterns that capture the conserved regions, often a ligand-binding or protein–protein interaction site, of a protein superfamily, family or subfamily. They can be captured by either multiple sequence alignment tools or pattern detection methods.[4,5]

Alignment is a common theme among the first four categories of classification methods. Yet alignment as-

Abbreviations: *n*-gram, short peptide sequences of length *n*; GPCR, G-protein coupled receptors; SVM, support vector machine; HMM, hidden Markov model; k-NN, k-nearest neighbor

*Correspondence to: J. Klein-Seetharaman, Department of Pharmacology, Biomedical Science Tower E1058, University of Pittsburgh, Pittsburgh, PA 15261. E-mail: judithks@cs.cmu.edu

Received 25 May 2004; 17 August 2004; Accepted 28 September 2004

Published online 11 January 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.20373

TABLE I. Tools Used in Common Protein Classification Methods

A. Pair-wise Sequence Alignment Tools

Tool	Reference
BLAST	Altschul et al., 1990[3]
FASTA	Pearson, 2000[37]
ISS	Park et al., 1997[38]
Needleman-Wunsch	Needleman and Wunsch, 1970[2]
PHI-BLAST	Zhang et al., 1998[39]
PSI-BLAST	Altschul et al., 1997[40]
Smith-Waterman	Smith and Waterman, 1981[1]

B. Multiple Sequence Alignment Tools

Tool	Reference
BLOCKMAKER	Henikoff et al., 1995[41]
ClustalW	Thompson et al., 1994[42]
DIALIGN	Morgenstern et al., 1998;[43] Morgenstern, 1999[44]
MACAW	Schuler et al., 1991[45]
MULTAL	Taylor, 1988[46]
MULTALIGN	Barton and Sternberg, 1987[47]
Pileup	Wisconsin Package, v. 10.3[48]
SAGA	Notredame et al., 1996[49]
T-Coffee	Notredame et al., 2000[50]

C. Profile HMM Tools

Tool	Reference
GENEWISE	Birney et al., 2004[51]
HMMER	HMMER, 2003[52]
META-MEME	Grundy et al., 1997[54]
PFTOOLS	Bucher et al., 1996[55]
PROBE	Neuwald et al., 1997[56]
SAM	Krogh et al., 1994[57]

sumes that order is conserved between homologous segments in the protein sequence,[6] which contradicts the genetic recombination and re-shuffling that occur in evolution.[7,8] As a result, when sequence similarity is low, aligned segments are often short and occur by chance, leading to unreliable alignments when the sequences have less than 40% similarity[9] and unusable alignments below 20% similarity.[10,11] This has sparked interest in methods which do not rely solely on alignment, mainly machine learning approaches[12] (Table II) and applications of Kolmogorov complexity and Chaos Theory.[6]

There is a belief that classifiers with simple running time complexity on alignment-based features are inherently limited in performance due to unreliable alignments at low sequence identity and that complex classifiers are needed for better classification accuracy.[13,14] Thus, while classifiers at the higher end of running time complexity are being explored, classifiers at the lower end are neglected. To the best of our knowledge, the simplest classifier attempted on protein classification is the k-nearest neighbor (k-NN) classifier. Here we describe the application of two classifiers simpler than k-NN that performs comparably to HMM and support vector machine (SVM) in protein classification: Decision Trees and Naïve Bayes.

G-Protein Coupled Receptors

With the enormous amount of proteomic data now available, there are a large number of datasets that can be used in protein family classification. We have chosen the G-protein coupled receptor (GPCR) superfamily in our experiments because it is an important topic in pharmacology research and it presents one of the most challenging datasets for protein classification. GPCRs are the largest superfamily of proteins found in the body;[15] they function in mediating the responses of cells to various environmental stimuli, including hormones, neurotransmitters and odorants, to name just a few of the chemically diverse ligands to which GPCRs respond. As a result, they are the target of approximately 60% of approved drugs currently on the market.[16] Reflecting its diversity of ligands, the GPCR superfamily is also one of the most diverse protein families.[17] Sharing no overall sequence homology,[18] the only feature common to all GPCRs is their seven transmembrane α-helices separated by alternating extracellular and intracellular loops, with the amino terminus (N-terminus) on the extracellular side and the carboxyl terminus (C-terminus) on the intracellular side, as shown in Figure 1.

The GPCR protein superfamily is composed of five major families (classes A–E) and several putative and "orphan" families.[19] Each family is divided into level I subfamilies and then further into level II subfamilies based on pharmacological and sequence-identity considerations. The extreme divergence among GPCR sequences is the primary reason for the difficulty in classifying them, and this diversity has prevented further classification of a number of known GPCR sequences at the family and subfamily levels; these sequences are designated "orphan" or "putative/unclassified" GPCRs.[17] Moreover, since subfamily classifications are often defined chemically or pharmacologically rather than by sequence homology, many subfamilies share strong sequence homology with other subfamilies, making subfamily classification extremely difficult.[13]

Classification of G-Protein Coupled Receptor Sequences

A number of classification methods have been studied on the GPCR dataset. Lapinsh et al.[20] extracted physical properties of amino acids and used multivariate statistical methods, specifically principal component analysis, partial least squares, autocross-covariance transformations and z-scores, to classify GPCR proteins at the level I subfamily level. Levchenko[21] used hierarchical clustering on similarity scores computed with the SSEARCH[22] program[1] to classify GPCR sequences in the human genome belonging to the peptide level I subfamily into their level II subfamilies. Liu and Califano[23] used unsupervised, top-down clustering in conjunction with a pattern-discovery algorithm, a statistical framework for pattern analysis, and HMMs to produce a hierarchical decomposition of GPCRs down to the subfamily level.

A systematic comparison of the performance of different classifiers ranging in complexity has been carried out recently by Karchin et al.[13] for GPCR classification at the superfamily level (i.e., whether or not a given protein is a

TABLE II. Some Machine-Learning Approaches to Protein Classification

Classifier	Features	Reference
Bayesian inference using Gibbs sampling	Number of conserved columns, size and number of classes and motifs in them	Qu et al., 1998[69]
Bayesian neural networks	Bigram counts, presence and significance of motifs found using an automated tool Sdiscover	Wang et al., 2000[70]
Clustering	Digraph representation of the sequence space where the weight of each edge between two sequences is the similarity score of the sequences from Smith–Waterman, BLAST and FASTA	Yona et al., 1999[71]
	Sequence and topological similarity	Mitsuke et al., 2002[72]
Discriminant function analysis (non-parametric, linear)	Frequency of each amino acid, average periodicity of GES hydropathy scale and polarity scale, variance of first derivative of polarity scale	Kim et al., 2000[73]
Neural networks	n-gram counts with SVD	Wu et al., 1995[74]
	Matrix patterns derived from bigrams	Ferran & Ferrara, 1992[75]
Sparse Markov transducers	All subsequences of the protein inside a sliding window	Eskin et al., 2000[76] & 2003[77]
Support vector machines	Fisher scores with Fisher kernel	Jaakkola et al., 1999[78] & 2000;[79] & Karchin et al., 2002[13]
	Set of all possible k-grams (fixed k) with spectrum kernel and mismatch kernels	Leslie et al., 2002[80] & 2004[81]
	String subsequence kernel	Vanschoenwinkel et al., 2002[82]

TABLE III. Databases Providing Protein Family Classification Information

Motifs / Profiles Databases	
Database	Reference
BLOCKS+	Henikoff et al., 1999;[58] Henikoff et al., 2000[59]
eMOTIF	Huang and Brutlag, 2001[60]
PFAM	Bateman et al., 2004[34]
PRINTS	Attwood et al., 2002[61]
PRODOM	Servant et al., 2002;[36] Corpet et al., 2000[35]
PROSITE	Falquet et al., 2002;[62] Sigrist et al., 2002[63]
SMART	Ponting et al., 1999;[64] Letunic et al., 2002[65]
Superfamily	Gough et al., 2001[66]
SWISSPROT	Apweiler et al., 1997;[67] Boeckmann et al., 2003[68]

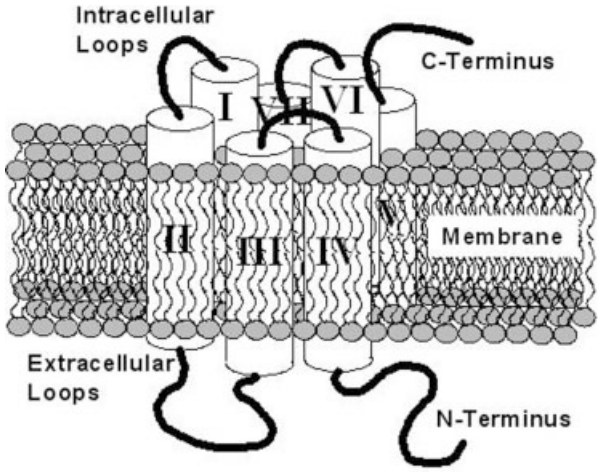

Fig. 1. Schematic of a GPCR. The seven cylinders represent the transmembrane α-helices. The membrane lipid environment is indicated in gray.

GPCR) and level I and II subfamily levels. Note that family-level classification was not examined by this study. The methods tested include a simple nearest neighbor approach (BLAST), a method based on multiple sequence alignment generated by a statistical profile HMM, a nearest-neighbor approach with protein sequences encoded into Fisher Score Vector space (kernNN) and SVM.

In the HMM method, a model is built for each class in the classification, and a query sequence is assigned to the class whose model has the highest probability of generating the sequence. Karchin et al. investigated two implementations of support vector machines, SVM and SVMtree, where the latter is a faster approximation to a multi-class SVM. Fisher Score Vectors were also used with SVM and SVMtree. To derive the vectors, Karchin et al. built a profile HMM model for a group of proteins and then computed the gradient of the log likelihood that the query sequence was generated by the model. A feature reduction technique based on a set of pre-calculated amino acid distributions was used to reduce the number of features from 20 components per matching state in the HMM to nine components per matching state. Both SVM and the kernNN method made use of radial basis kernel functions. The results from this study are reproduced in Table IV.

Karchin and coworkers' study concluded that while simpler classifiers (specifically HMM) perform better at the superfamily level, the computational complexity of SVM is needed to attain "annotation-quality classification" at the subfamily levels. However, the simplest classifiers, such as Decision Trees and Naïve Bayes, have not been applied in this context. In this study, we investigated in further detail the performance of simple classifiers in the task of GPCR classification at the family and subfamily levels. We first optimized these simple classifiers using feature selection and then compared our results to those reported by Karchin et al.[13] To our surprise, we found that using only a simple classifier on counts of n-grams in conjunction with a straightforward feature-selection algo-

TABLE IV. Classification Results Reported in Previous Study on Complexity Needed for GPCR Classification

Superfamily Classification	
Method	Accuracy at MEP (%)
SAM-T99 HMM	99.96
SVM	99.78
FPS BLAST	93.18
Level I Subfamily Classification	
Method	Accuracy at MEP (%)
SVM	88.4
BLAST	83.3
SAM-T2K HMM	69.9
kernNN	64.0
Level II Subfamily Classification	
Method	Accuracy at MEP (%)
SVM	86.3
SVMtree	82.9
BLAST	74.5
SAM-T2K HMM	70.0
kernNN	51.0

Karchin et al. reported their results in terms of "average errors per sequence" at the minimum error point (MEP). Through e-mail correspondence with the first author, we verified that "average errors per sequence" is equivalent to the error rate. Thus, the accuracy results shown above are converted from those in their paper by the formula "1 - average errors per sequence."

rithm, specifically chi-square, is sufficient to outperform all of the classifiers in the previous study.[13]

MATERIALS AND METHODS

In this study, we applied the Decision Tree and the Naïve Bayes classifier to the protein classification task. Analogous to document classification in language technologies, each protein sequence is represented as a vector of n-gram counts where n-grams are extracted from the sequence at each of the n possible reading frames. For instance, the sequence "ACWQRACW" has two counts each of bigrams AC and CW, and one count each of bigrams WQ, QR and RA. Instead of using only n-grams of a single fixed length n, we used n-grams of lengths 1,2…n.

Decision Tree

Decision Tree is one of the simplest classifiers in machine learning. One of its advantages lies in its ease of interpretation as to which are the most distinguishing features in a classification problem. It has been used previously with biological sequence data in classifying gene sequences.[24] We used the C4.5 implementation of Decision Tree by J. R. Quinlan.[25]

Given a dataset of training instances, each represented by a vector of feature values, the Decision Tree algorithm grows the tree downwards from the root utilizing the information gain maximization criterion:

$$IG(C,f) = H(C) - p(f)H(C|f) - p(\bar{f})H(C|\bar{f}) \qquad (1)$$

where C is the class label variable, $H(C|f)$ is the entropy of C given having the feature f, $H(C|\bar{f})$ is the entropy of C given not having the feature f, and $p(f)$ and $p(\bar{f})$ are the probability of having and not having the feature f respectively. A feature is selected for each node in the tree where the most discriminative features, as determined by the above criterion, are located close to the root of the tree. For example, the trigram DRY is one of the most discriminative features for Class A GPCR and would therefore be located closer to the root of the tree than the unigram D, which is a very common feature among all GPCRs.

When deciding on the label of a sequence in the test set, the Decision Tree algorithm proceeds down the tree from the root, taking the feature at each node in turn and examining its value in the test sequence's feature vector. The value of the feature determines which branch of the tree the algorithm takes at each step. The process continues until the algorithm reaches a leaf node in the Decision Tree and assigns the test sequence the class label of the leaf node.

To build the Decision Tree from the training set, the algorithm computes the information gain of each feature (i.e. n-gram count) with respect to the class label (i.e. either the protein family or subfamily) using the above formula. The most discriminating feature x, defined as the one with the highest information gain, is taken to be the root node of the Decision Tree, and the dataset is split into subsets based on its value. For each subset, the information gain of each of the remaining features is computed using only the subset of data, and the feature x' with the highest information gain is taken as the root node of the subtree represented by the subset of data (and thus a child node of x). The subset is then further divided into smaller subsets based on the value of x', and the process repeats until each training instance in the subset has the same class label or each training instance is identical in all its feature values.

Since our features, n-gram counts, lie on a continuous range, information gain is also needed to determine the most discriminative threshold for each feature. Using the threshold, we can divide our dataset into two subsets based on feature x by considering whether the value of x is greater or less than that of y. Thus, we do not need to divide the dataset into as many subsets as there are different values of x in the data, thereby avoiding overfitting our classifier to the training data. More detailed information can be found in machine learning textbooks such as refs. 26,27.

To further reduce overfitting of the Decision Tree, the algorithm uses a confidence interval to prune away some bottom portions of the tree once it has been fully built. The confidence interval is used to give a pessimistic estimate of the true error rate at each tree node from the training data. If the estimated error rate at a node is lower than the combined estimated error rate of its child nodes, then the node is made into a leaf node by pruning its child nodes away. We examined the effect of different confidence intervals by repeating our cross-validation experiments varying only the confidence interval. Since the difference

in accuracy was not significant (1.2% at maximum) we used the default 75% confidence level in the remainder of our experiments.

Naïve Bayes

Naïve Bayes is the other simple machine learning classifier used in this study. Its naïve assumption that all of its features are independent clearly did not hold when we allowed overlaps in extracting n-grams of length greater than 1 from a sequence. Nonetheless, the classifier worked remarkably well in our task, as described below.

We used the Rainbow implementation of the Naïve Bayes classifier by Andrew K. McCallum. (The Rainbow implementation is part of Bow,[28] a toolkit for statistical language modeling, text retrieval, classification and clustering.) Given a test instance d_i, the Naïve Bayes algorithm predicts the class $\hat{c}$ such that

$$\hat{c} = \underset{j}{\operatorname{argmax}} \mathrm{P}(c_j|d_i) = \underset{j}{\operatorname{argmax}} \frac{\mathrm{P}(d_i|c_j)P(c_j)}{P(d_i)} \qquad (2)$$

where

$$P(d_i|c_j) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|c_j)^{N_{it}}}{N_{it}!} \qquad (3)$$

$$|d_i| = \sum_{t=1}^{|V|} N_{it} \qquad (4)$$

c_j is the j^{th} protein family, d_i is the i^{th} sequence, w_t is the t^{th} n-gram in the set of all n-grams V and N_{it} is the count of w_t in d_i. Thus, $P(d_i)$ is the prior probability that the sequence is d_i and $P(c_j)$ is the prior probability that the sequence belongs to protein family c_j. Likewise, $P(c_j|d_i)$ represents the probability that the sequence belongs to protein family c_j given the sequence d_i, and similarly, $P(d_i|c_j)$ is the probability that the sequence is d_i given that it belongs to protein family c_j. $P(d_i|c_j)$ is estimated using a multinomial distribution of n-grams trained specifically for the protein family c_j where $P(w_t|c_j)$, the probability of the n-gram w_t occurring in the sequence given that the sequence belongs to family c_j, is computed from the training data as the count of w_t in all sequences belonging to c_j divided by the total number of n-grams in all sequences belonging to c_j. More details about the multinomial distribution model used for the n-gram features can be found in ref. 29. To prevent zero probabilities for $P(w_t|c_j)$ from insufficient data, LaPlace smoothing is used.

Since single amino acids (unigrams) are too short to be motifs and occur too frequently to be good indicators of motifs, we have excluded all unigrams as features in our experiments with the Naïve Bayes classifier. Thus, we used only n-grams of lengths 2,3…n.

Chi-Square

Most machine-learning algorithms do not scale well to high-dimensional feature spaces,[30] and the Decision Tree and Naïve Bayes classifiers are no exceptions. Thus, it is desirable to reduce the dimension of the feature space by removing non-informative or redundant features that would hide the informative ones or otherwise confuse the classifier.[31] A large number of feature selection methods have been developed for this task, including document frequency, information gain, mutual information, chi-square and term strength. These methods were used as a pre-processing step on our data prior to applying the classifiers. We chose to use chi-square in our study because it is one of the most effective feature selection methods in document classification,[31] slightly surpassing information gain (whose formula is given in the Decision Tree section).

The chi-square statistic measures the discriminating power of a binary feature x in classifying a data point into one of the possible classes. It quantifies the lack of independence between a given binary feature x and a classification category c by computing the difference between the "expected" number of objects in c with that feature and the observed number of objects in c actually having that feature. By "expected," we mean the number of instances of c we would find with feature x if the feature were not dependent on the category and had a uniform distribution over all the categories instead. Thus, the formula for the chi-square statistic for each feature x is as follows:

$$\chi^2(x) = \sum_{c \in C} \frac{[e(c,x) - o(c,x)]^2}{e(c,x)} \qquad (5)$$

where C is the set of all categories in our classification task, and $e(c,x)$ and $o(c,x)$ are the "expected" and observed number of instances in category c with feature x, respectively.

The "expected" number $e(c,x)$ is computed as

$$e(c, x) = n_c \frac{t_x}{N} \qquad (6)$$

where n_c is the number of objects in category c, N is the total number of objects and t_x is the number of objects with feature x.

To obtain binary features from counts of n-grams, we divided each n-gram feature into 20 "derived" features with binary values by considering whether the n-gram occurs at least i times in the sequence, where i represents the first 20 multiples of 5 (i.e., 5,10,15…100) for unigrams and the first 20 multiples of 1 (i.e., 1,2,3…20) for all other n-grams. Then we computed the chi-square statistic for each of these binary features.

For instance, for the trigram DRY, we computed the chi-square statistic for the 20 binary features of whether DRY occurs at least 1,2,3…20 times. The expected number of protein sequences in class c having at least i occurrences of the n-gram DRY is computed as

$$\#\text{ of sequences in class } c \times \frac{\#\text{ of sequences with at least } i \text{ occurences of DRY}}{\text{total \# of sequences}}$$

The chi-square statistic for DRY occurring at least i times is the square of the difference between the expected and

TABLE V. Organization of Chi-Square Values for 20 Binary Features Associated with Each *n*-Gram

Unigram	*i*	5	10	15	···	100
A						
C						
···						
***n*-Gram**	***i***	**1**	**2**	**3**	**···**	**20**
AA						
AC						
···						
AAA						
AAC						
···						

the observed number of sequences in each class having at least i occurrences of DRY, normalized by the expected number and summed over all classes.

Next, for each n-gram j, we found the value $i_{\max}$ such that the binary feature of having at least $i_{\max}$ occurrences of n-gram j has the highest chi-square statistic out of the 20 binary features associated with n-gram j. This is equivalent to finding the column $i_{\max}$ where the maximum value in each row occurs as shown in Table V. Note that the selected column for each n-gram could be different. The n-grams were then sorted in decreasing order according to the maximum chi-square value in their row, that is, the chi-square value in their $i_{\max}$ column. The top K n-grams were selected as input to our classifiers, where K is a parameter that was tuned to achieve maximum accuracy. Each protein sequence was represented as a vector of length K where the elements in the vector were the counts of these top K n-gram features. We also investigated the effect on accuracy of having each vector component be the binary feature of having at least $i_{\max}$ occurrences of the selected n-gram j versus being the count of n-gram j. The vectors with chi-square selected features (n-gram counts or binary values) were used in the classification procedures for Naïve Bayes and Decision Trees in an identical fashion to that described above for the entire set of n-grams without chi-square selection.

Dataset Used for GPCR Family-Level Classification

In family-level classification, we gathered our own dataset by taking all GPCR sequences and bacteriorhodopsin sequences with SWISS–PROT entries found in the September 15, 2002 release of GPCRDB.[19] GPCRDB is an information system specifically for GPCRs, containing all known GPCR sequences, classification information, mutation data, snake-plots, links to various tools for GPCRs and other GPCR-related information. It contains both sequences with SWISS–PROT entries and those with TREMBL entries. These entries contain important information such as the protein's classification, function and domain structure. SWISS–PROT entries are computer-generated annotations that have been reviewed by a human, while TREMBL entries have not yet been reviewed. For this reason, we have chosen to use only those sequences with SWISS–PROT entries in our evaluation.

TABLE VI. Distribution of Classes in Dataset Used in GPCR Family Classification

Family	# of Proteins	% of Dataset
Class A	1081	79.72%
Class B	83	6.12%
Class C	28	2.06%
Class D	11	0.81%
Class E	4	0.29%
Class F - frizzled / smoothened family	45	3.32%
Drosophila odorant receptors	31	2.29%
Nematode chemoreceptors	1	0.07%
Ocular albinism proteins	2	0.15%
Plant Mlo receptors	10	0.74%
Orphan A	35	2.58%
Orphan B	2	0.15%
Bacteriorhodopsins	23	1.70%
Total	1356	100.00%

Notice the majority class, Class A, comprises almost 80% of the dataset.

According to GPCRDB, the GPCR superfamily is divided into 12 major and putative families. Bacteriorhodopsin is a non-GPCR family of proteins that is often used as a structural template for modeling three-dimensional structures of GPCRs.[32] Thus, we have decided to include them in our dataset as a control. Hence, there were 13 classes in our family classification dataset, 12 GPCR families and one non-GPCR family, as shown in Table VI. A ten-fold cross-validation was used as our evaluation protocol.

Datasets Used for GPCR Level I Subfamily Classification

Since we used the results of the various classifiers studied by Karchin et al.[13] as the baseline for our subfamily classifications, we used the same datasets and evaluation protocol in our experiments at the level I and II subfamily classification. This is independent of the dataset and the process used to create the dataset in family-level classification. In level I subfamily classification, shown in Table VII, there were 1269 sequences from subfamilies within Classes A and C, as well as 149 non-GPCR sequences from archaea rhodopsins and G-alpha proteins. Note that subfamilies not in Class A and C are not included. The non-GPCR sequences were grouped together as a single class of negative examples for our classifier evaluation. We performed a two-fold cross-validation using the same training–testing data split as in the study by Karchin et al.[13] The dataset and training-testing data split are available at www.soe.ucsc.edu/research/compbio/gpcr/subfamily_seqs.

Datasets Used for GPCR Level II Subfamily Classification

In level II subfamily classification (Table VIII), we used 1170 sequences from Classes A and C and 248 sequences from archaea rhodopsins, G-alpha proteins and GPCRs with no level II subfamily classification or those in a level II subfamily containing only one protein. As before, the 248 sequences were grouped together as a single class of

TABLE VII. Distribution of Level 1 GPCR Subfamilies in Classes A and C Used in Level 1 GPCR Subfamily Classification Dataset

Level I Subfamily	# of Proteins	% of Dataset
Class A	**1207**	**85.12%**
Amine	**221**	**15.59%**
Cannabis	11	0.78%
Gonadotropin releasing hormone	10	0.71%
Hormone protein	25	1.76%
Lysosphingolipid and LPA_EDG	17	1.20%
Melatonin	13	0.92%
Nucleotide-like	48	3.39%
Olfactory	87	6.14%
Peptide	381	26.87%
Platelet activating factor	4	0.28%
Prostanoid	38	2.68%
Rhodopsin	183	12.91%
Thyrotropin releasing hormone and secretagogue	13	0.92%
Viral	17	1.20%
Class C	**62**	**4.37%**
Extracellular calcium sensing	5	0.35%
GABA_B	16	1.13%
Metabotropic glutamate	21	1.48%
Putative pheromone	20	1.41%
Other Sequences	149	10.51%
Total	1418	100.00%

The total number of sequences in Class A and C (bold rows) are included here to show the relationships among the level I subfamilies. Notice the majority class, peptide subfamily in Class A, comprises 27% of the dataset.

negative examples, and a two-fold cross-validation was performed using the same training–testing data split as in the study by Karchin et al.[13]

Datasets Used for Nuclear Receptor Family, Level I Subfamily and Level II Subfamily Classification

The superfamily of nuclear receptors is divided into families, then into level I subfamilies, and finally into level II subfamilies. Tables IX, X and XI show the distribution of the datasets used in our classification at these three levels. The datasets consist of all the full sequences with SWISS–PROT entries from all the families and subfamilies in the March 2004 release (4.0) of the NucleaRDB database,[33] excluding those families and subfamilies with two or fewer members.

RESULTS

We examined classification of GPCRs at the family level and at the level I and II subfamily levels. GPCR family-level classification was used to develop our classification protocol, while the GPCR subfamily-level classifications were used to compare the performance of our protocol to that of other classifiers, particularly SVM, studied by Karchin et al.[13] Finally, a separate protein family dataset was investigated, that of nuclear receptors, to demonstrate the general nature of our findings for classification of other protein families.

GPCR Family-Level Classification without Feature Selection

The Decision Tree and the Naïve Bayes classifiers were used in a ten-fold cross-validation experiment using as features all the amino acid n-grams of a specified size, 1,2…n for Decision Tree and 2,3…n for Naïve Bayes. The dataset was split into 10 equal subsets, referred to as folds. The classifiers were tested on each fold independently while training on the remaining nine folds each time. The maximum n value for the Decision Tree was 3 ("trigrams"), since the addition of larger n-grams as features had little effect on its accuracy. In contrast, the Naïve Bayes classifier performed significantly better with bigrams and trigrams together than with bigrams alone. Thus we tested different values of n up to 5 for the Naïve Bayes classifier. However, the addition of n-grams of length greater than 3 decreased the accuracy. The results are shown in Table XII.

GPCR Family-Level Classification with Feature Selection

Next we investigated the effect of reducing the number of features on classification accuracy. Based on the results obtained without feature selection, we applied the chi-square feature selection algorithm to the set of all unigrams, bigrams and trigrams in the case of the Decision Tree classifier and to the set of all bigrams and trigrams in the case of the Naïve Bayes classifier.

To determine the optimal number of features, K, the chi-square algorithm needs to select for each classifier. We measured the accuracy of the classifier on the validation set as a function of K. Specifically, we divided our dataset into 10 folds, and in each experiment we reserved one fold as the test set and one fold as the validation set to tune the parameter K while training on the remaining eight folds. This ensured that any improvements observed in the results were not due to overfitting the classifier to the dataset. Note that this requires each family in the dataset to have at least three members so that the training set is guaranteed to have at least one member of that family. Thus, the three families with fewer than three members each (Nematode Chemoreceptors, Ocular Albinism Proteins, Orphan B) were removed from the dataset for our family-level classification experiments with chi-square feature selection.

For each classifier in each experiment, we set K to represent the number of features at which the validation set accuracy is maximized and reported the test set accuracy at that point. (Thus the value of K differs from fold to fold.) In addition, we also investigated whether there is a difference between using the binary features of having at least $i_{\max}$ occurrences of each selected n-gram j and using the counts of the selected n-grams as continuous or multinomial attributes in the Decision Tree and the Naïve Bayes classifier, respectively. The results for one of the folds are plotted in Figure 2. The accuracy of the classifier increases with K until a maximum accuracy is reached, after which the accuracy drops as K continues to increase. Using the counts of the selected n-grams instead

TABLE VIII. Distribution of Level II GPCR Subfamilies in Classes A and C Used in Level II GPCR Subfamily Classification Dataset

Level II Subfamily	# of Proteins	% of Dataset
Class A	**1133**	**79.90%**
Amine	**221**	**15.59%**
Adrenergic	66	4.65%
Dopamine	43	3.03%
Histamine	10	0.71%
Muscarinic	23	1.62%
Octopamine	12	0.85%
Serotonin	67	4.72%
Class A Orphan	**133**	**9.38%**
Bonzo	4	0.28%
Chemokine receptor like 2	3	0.21%
G10D	3	0.21%
GP40 like	4	0.28%
GPR	35	2.47%
Mas proto-oncogene	3	0.21%
Other	77	5.43%
RDC 1	4	0.28%
Hormone Protein	**25**	**1.76%**
Follicle stimulating hormone	10	0.71%
Lutropin choriogonado-tropic hormone	8	0.56%
Thyrotropin	7	0.49%
Nucleotide Like	**48**	**3.39%**
Adenosine	23	1.62%
Purinoceptors	25	1.76%
Olfactory	**87**	**6.14%**
Gustatotory odorant	2	0.14%
Olfactory type 1	9	0.63%
Olfactory type 2	6	0.42%
Olfactory type 3	6	0.42%
Olfactory type 4	9	0.63%
Olfactory type 5	18	1.27%
Olfactory type 6	10	0.71%
Olfactory type 7	3	0.21%
Olfactory type 8	2	0.14%
Olfactory type 9	4	0.28%
Olfactory type 10	9	0.63%
Olfactory type 11	9	0.63%
Peptide	**385**	**27.15%**
Angiotensin	21	1.48%
APJ like	5	0.35%
Bombesin	11	0.78%
Bradykinin	10	0.71%
C5a anaphyla-toxin	9	0.63%
CCK	14	0.99%
Chemokine chemotactic factors-like	7	0.49%
Chemokine	82	5.78%
Endothelin	14	0.99%
Fmet-Leu-Phe	10	0.71%
Galanin	9	0.63%
Gpr37-like peptide receptor	4	0.28%
Interleukin-8	13	0.92%
Melanocortin	34	2.40%
Neuropeptide Y	31	2.19%
Neurotensin	6	0.42%
Opioid	19	1.34%
Orexin	5	0.35%
Proteinase activated	7	0.49%
Somatostatin	17	1.20%
Tachykinin	21	1.48%
Thrombin	6	0.42%
Urotensin II	2	0.14%
Vasopressin	28	1.97%
Prostanoid	**38**	**2.68%**
Prostacyclin	4	0.28%
Prostaglandin	28	1.97%
Thromboxane	6	0.42%
Rhodopsin	**183**	**12.91%**
Rhodopsin arthropod	33	2.33%
Rhodopsin mollusk	6	0.42%
Rhodopsin other	12	0.85%
Rhodopsin vertebrate	132	9.31%
Thyrotropin Releasing Hormone and Secretagogue	**13**	**0.92%**
Growth hormone secretagogue	4	0.28%
Growth hormone secretagogue-like	2	0.14%
Thyrotropin releasing hormone	7	0.49%
Class C	**37**	**2.61%**
Metabotropic Glutamate	**21**	**1.48%**
Metabotropic glutamate I	4	0.28%
Metabotropic glutamate II	5	0.35%
Metabotropic glutamate III	10	0.71%
Metabotropic Glutamate Other	2	0.14%
GABA_B	**16**	**1.13%**
GABA_B1	10	0.71%
GABA_B2	6	0.42%
Other Sequences	248	17.49%
Total	1418	100.00%

The number of sequences in Classes A and C and each of their level I subfamily (bold rows) have been included here to show the relationships among the level II subfamilies. Note that the majority class, Other Sequences, comprises 17.5% of the dataset.

TABLE IX. Distribution of Nuclear Receptor Family Level Classification Dataset

Family	# of Proteins	% of Dataset
0A Knirps-like	7	2.13%
0B DAX-like	6	1.82%
1 Thyroid hormone like	117	35.56%
2 HNF4-like	54	16.41%
3 Estrogen-like	72	21.88%
4 Nerve growth factor IB-like	13	3.95%
5 Fushi tarazu-F1 like	13	3.95%
Unclassified	47	14.29%
Total	329	100.00%

The majority class, I thyroid hormone like, comprises 35.6% of the dataset.

of their binary features resulted in a higher accuracy using the Decision Tree, while no significant difference was observed for the Naïve Bayes classifier.

The validation set and test set accuracy with feature selection of the top K features averaged across the 10 folds at their respective optimal value K, determined from the validation set accuracy curve, is compared against the validation set and test set accuracy without feature selection in Table XIII. While chi-square feature selection improved the accuracy of the Decision Tree with unigrams,

TABLE X. Distribution of Nuclear Receptor Level I Subfamily Classification Dataset

Level I Subfamily	# of Proteins	% of Dataset
0B1 DAX	4	1.27%
1A Thyroid hormone	26	8.28%
1B Retinoic acid	17	5.41%
1C Peroxisome proliferator activated	19	6.05%
1D REV-ERB	12	3.82%
1F RAR-related orphan receptor	10	3.18%
1H Ecdysone-like	16	5.10%
1I Vitamin D3-like	13	4.14%
2A Hepatocyte nuclear factor 4	10	3.18%
2B Retinoic acid X	15	4.78%
2C Orphan nuclear receptors TR2, TR4	4	1.27%
2E Tailless-like	9	2.87%
2F COUP-TF-like	15	4.78%
3A Estrogen	36	11.46%
3B Estrogen-related	5	1.59%
3C Glucocorticoid-like	31	9.87%
4A NGFI-B-like	13	4.14%
5A Fushi tarazu-F1 like	12	3.82%
Unclassified nematode	47	14.97%
Total	314	100.00%

The majority class, unclassified nematode, comprises 15% of the dataset.

bigrams and trigrams, it had little effect on the Naïve Bayes classifier with bigrams and trigrams. Despite the improvement in accuracy of the Decision Tree, its accuracy is still lower than the accuracy of the Naïve Bayes classifier. With both classifiers, chi-square feature selection reduces the number of features needed to achieve their respective optimal accuracy.

Protein Length as a Feature for Classification

GPCR sequences vary significantly in length. For example, the rhodopsin sequence in Class A is one of the shortest GPCR sequences, with 348 amino acids, while the longest GPCR sequences, having several thousand amino acids, belong to Class B. Thus there may be a correlation between sequence length and GPCR family. We therefore investigated whether the protein sequence length is a useful feature in GPCR classification. Figure 3 shows a logarithmic plot of the counts of a given protein sequence length versus the sequence length, for each GPCR family and the control group bacteriorhodopsins. While this plot confirms that there is significant variation in sequence length providing some information for family classification, it also shows that the range of sequence lengths within each GPCR family overlaps significantly, potentially confusing the classifier. We confirmed this hypothesis by training a Decision Tree using the sequence length as a feature in addition to the counts of all unigrams, bigrams and trigrams. The resulting tree gave an insignificant improvement of 0.1% in test set accuracy over using the Decision Tree with only n-grams. Moreover, the sequence length appeared as a node of the Decision Tree in only one of the 10 trials in a ten-fold cross validation, and the node was at the 11^{th} level. Both histogram and experimental results therefore suggest that sequence length does not provide any information on the GPCR family classification task that is not already encoded in the n-gram counts.

TABLE XI. Distribution of Level II Nuclear Receptor Subfamily Classification Dataset

Level II Subfamily	# of Proteins	% of Dataset
1A1 Thyroid hormone alpha	14	5.96%
1A2 Thyroid hormone beta	12	5.11%
1B1 Retinoic acid alpha	6	2.55%
1B2 Retinoic acid beta	4	1.70%
1B3 Retinoic acid gamma	7	2.98%
1C1 PPAR alpha	7	2.98%
1C2 PPAR beta	3	1.28%
1C3 PPAR gamma	9	3.83%
1D2 REV-ERB beta	3	1.28%
1D3 E75	7	2.98%
1F4 HR3, CHR3	4	1.70%
1H1 Ecdysone	7	2.98%
1H2 Oxysterol LXR beta	3	1.28%
1H3 Oxysterol LXR alpha	3	1.28%
1H4 Farnesoid FXR	3	1.28%
1I1 Vitamin D3	7	2.98%
1I2 Pregnane X	3	1.28%
1I3 Constitutive androstane alpha	3	1.28%
2A1 HNF4 alpha	4	1.70%
2B1 RXR alpha	5	2.13%
2B3 RXR gamma	4	1.70%
2B4 USP	4	1.70%
2C2 TR4	3	1.28%
2E1 Tailless homolog	5	2.13%
2F1 COUP-TFI	4	1.70%
2F2 COUP-TFII	5	2.13%
2F6 V-erbA related protein	3	1.28%
3A1 ER alpha	17	7.23%
3A2 ER beta	19	8.09%
3B2 ERR beta	3	1.28%
3C1 Glucocorticoid-like	13	5.53%
3C2 Mineralocorticoid	5	2.13%
3C3 Progesterone	5	2.13%
3C4 Androgen	8	3.40%
4A1 NGFI-B	4	1.70%
4A2 NURR1	4	1.70%
4A3 NOR1	3	1.28%
5A1 Steroidogenic factor-1	5	2.13%
5A2 Fetoprotein TF	3	1.28%
5A3 Fushi tarazu-F1	4	1.70%
Total	235	100.00%

The majority class, 3A2 ER beta, comprises 8.09% of the dataset.

TABLE XII. Result of Ten-Fold Cross-Validation on GPCR Classification at Family Level Using Decision Tree and Naïve Bayes Classifier on All n-Grams of Specified Sizes

Decision Tree		Naïve Bayes	
N-grams Used	Accuracy	N-grams Used	Accuracy
1-gram	89.4%	2-grams	80.7%
1,2-grams	89.5%	2,3-grams	96.3%
1,2,3-grams	89.3%	2,3,4-grams	95.6%
		2,3,4,5-grams	94.8%

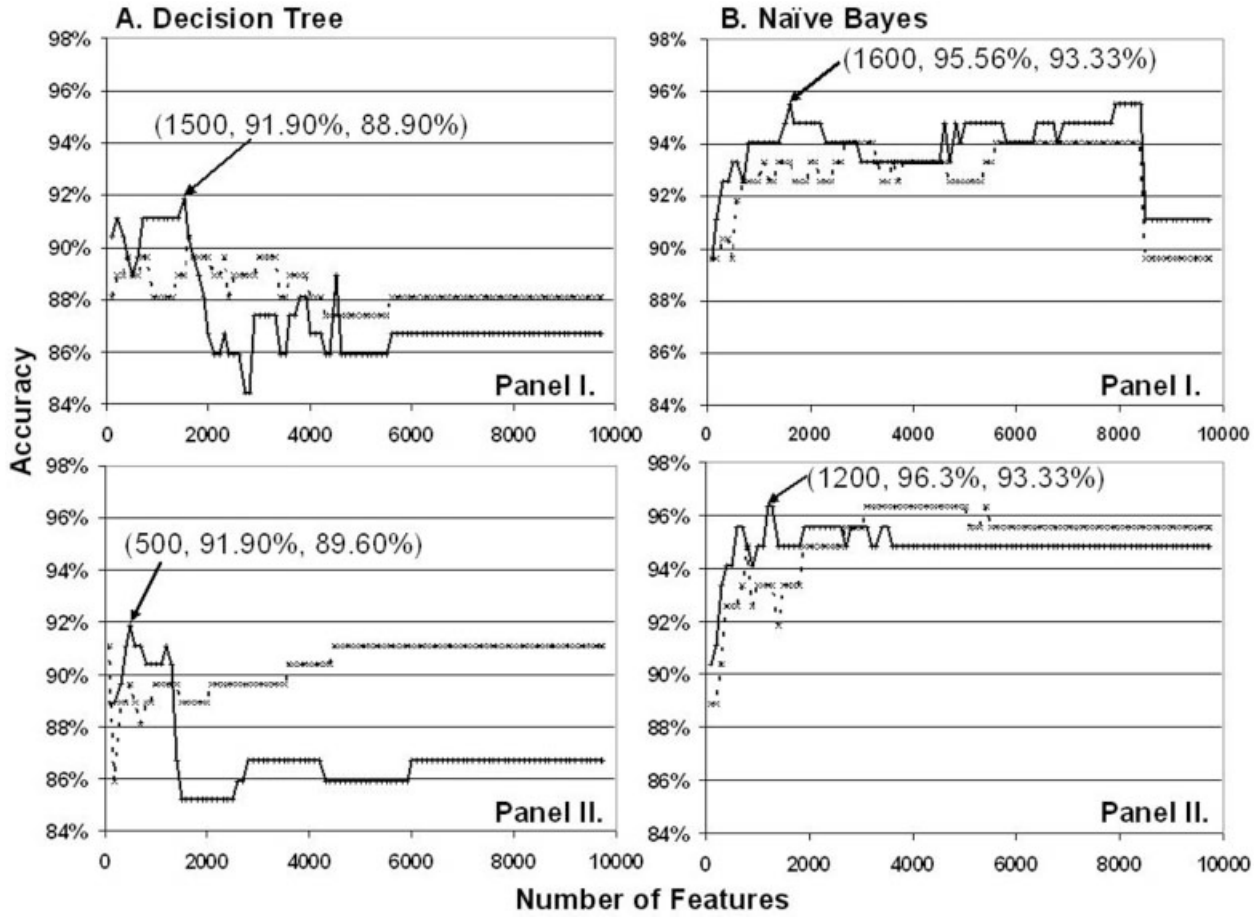

Fig. 2. Validation set (solid line) and test set (dotted line) accuracy of (a) the Decision Tree and (b) the Naïve Bayes classifier in GPCR family-level classification plotted as a function of the number of binary features (Panels I) or *n*-gram counts (Panel II), *K,* given to the classifier. The point at which the validation set accuracy reaches the maximum in each plot has been labeled with (number of features, validation set accuracy %, test set accuracy %).

TABLE XIII. Comparison of Performance of Classifiers with and without Feature Selection in GPCR Family Classification

			Accuracy	
Classifier	# of Features	Type of Features	Validation	Testing
Decision Tree	All (9723)	N-gram counts		89.00%
	100–3300	Binary	90.75%	88.68%
	100–900	N-gram counts	92.15%	90.61%
Naïve Bayes	All (9702)	N-gram counts		94.89%
	600–7100	Binary	95.41%	93.85%
	600–3300	N-gram counts	95.93%	94.30%

Unigrams, bigrams and trigrams were used with the Decision Tree, while bigrams and trigrams were used with the Naïve Bayes classifier. Note that the accuracy without feature selection here is different from that in Table XII because it came from training on eight folds as opposed to nine, to remain consistent with the other results in the table. The range of optimal values for *K* in the ten folds is shown in the "# of Features" column.

Comparison of GPCR Family Classification Against PFAM Baseline

To evaluate the performance of our classifier against that of the current state-of-the-art models, we compared our classifier against the profile HMMs in the PFAM database.[34] The PFAM database is divided into two parts, PFAM-A and PFAM-B. PFAM-A contains the curated families in PFAM, while PFAM-B contains the families automatically generated from the PRODOM[35,36] database to give PFAM a more comprehensive coverage. Thus, PFAM-A is of a higher quality than PFAM-B and is the focus of this evaluation. Since our ten-fold cross validation results (Table XIII) show the Naïve Bayes classifier to be better than the Decision Tree at the family-level classifica-

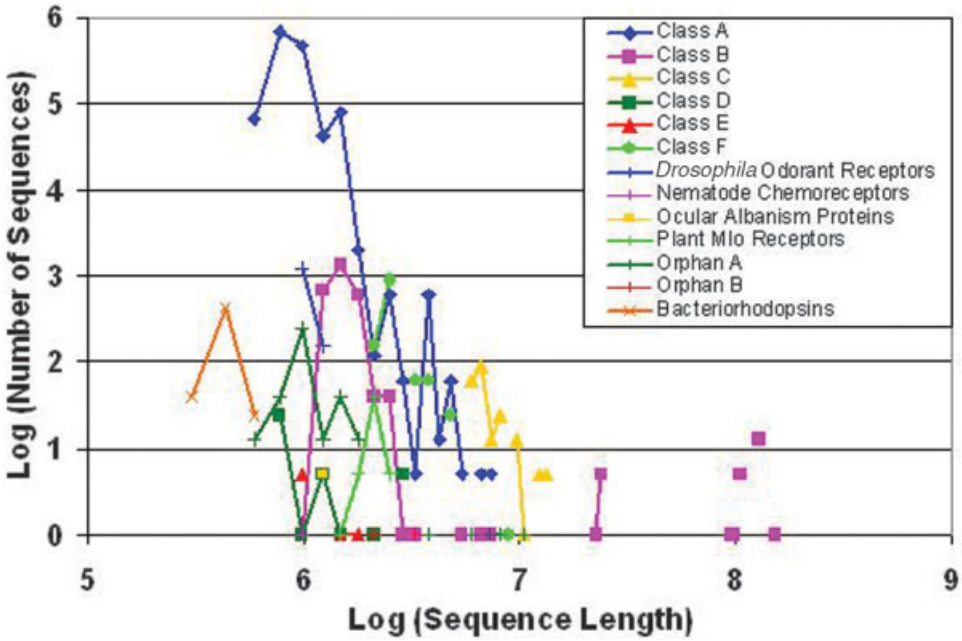

Fig. 3. Histogram of sequence length in GPCR family-level classification dataset, by class or GPCR family, on a log–log plot. Notice that, with the exception of bacteriorhodopsins, all of the curves overlap in the range between 5.75 and 7.2, suggesting that the sequence length is not a very useful indicator of the proteins membership in a particular GPCR family or a subset of GPCR families.

tion task and indicate little difference in performance from chi-square feature selection, this evaluation against PFAM focuses on the Naïve Bayes classifier without chi-square.

Each of the curated families in PFAM-A has a profile HMM created from a human-edited multiple sequence alignment, called the seed alignment, of representative sequences from that family. We take PFAM's family prediction for a given sequence as the family whose HMM model returns the highest probability for the sequence. Because of the huge effort required in building these HMMs, only four of the GPCR families are contained within PFAM-A, Classes A, B, C and *Drosophila* Odorant Receptors. Thus, only these four families are considered in the evaluation.

The training set consisted of the sequences in the seed alignment, while the test set consisted of the SWISSPROT sequences in these four families as classified by GPCRDB (September 2002 release)[19] that are not part of the seed alignment. Note that TREMBL sequences were not included in the training set for the Naïve Bayes classifier but were used in constructing the seed alignments of the profile HMMs. The test sets for the Naïve Bayes classifier and PFAM were identical and did not contain any TREMBL sequences. The distribution of the dataset in this evaluation is shown in Table XIV.

Out of 1100 test sequences, PFAM classified three sequences into non-GPCR families and was unable to classify six sequences into any PFAM-A families. This corresponds to an accuracy of 94.91%. If a constraint is placed upon PFAM to classify into a GPCR family, the accuracy increases to 99.18%. The Naïve Bayes classifier (which is constrained to classify into a GPCR family by default) has an accuracy of 97.64%. Thus, our classifier's performance is comparable to PFAM but does not require the human intervention involved in creating the PFAM database.

TABLE XIV. Distribution of GPCR Family Classification Dataset in Evaluation Against PFAM Baseline

Class	PFAM-A Training Seq.	Naïve Bayes Training Seq.	Testing Seq.
Class A	64	62	1019
Class B	36	26	57
Class C	30	12	16
Drosophila odorant receptors	40	23	8

GPCR Level I Subfamily Classification

The ten-fold cross-validation experiments on family-level classification described above demonstrate that chi-square feature selection is beneficial, not only in reducing the number of features needed but also in improving the classification accuracy. We therefore tested whether a similar improvement could be obtained at the subfamily level. As before, we measured the classification accuracy as a function of the number of features, K, using unigrams, bigrams and trigrams with the Decision Tree and bigrams and trigrams with the Naïve Bayes classifier. The accuracy was computed from a two-fold cross-validation using the same dataset and training–testing data split as in the study by Karchin et al.[13] for ease of comparison to the classifiers presented in their study.

To prevent our results from being an effect of overfitting the classifier to the dataset, we divided the testing fold in half, using one half as a validation set to tune K and the

TABLE XV. Comparison of Accuracy of Various Classifiers at GPCR Level I Subfamily Classification

Classifier	# of Features	Type of Features	Accuracy	
			Validation	Testing
Decision Tree	All (9723)	*n*-Gram counts		77.2%
	900–2800	Binary	79.9%	77.3%
	700–5600	*n*-Gram counts	80.2%	77.3%
Naïve Bayes	All (9702)	*n*-Gram counts		90.0%
	5500–7700	Binary	93.5%	93.0%
	3300–6900	*n*-Gram counts	91.3%	90.6%
SVM	Nine per match state in the HMM	Gradient of the log-likelihood that the sequence is generated by the given HMM model		88.4%
BLAST	Local sequence alignment			83.3%
SAM-T2K HMM	A HMM model built for each protein subfamily			69.9%
kernNN	Nine per match state in the HMM	Gradient of the log-likelihood that the sequence is generated by the given HMM model		64.0%

Unigrams, bigrams and trigrams are used with the Decision Tree, while bigrams and trigrams are used with the Naïve Bayes classifier. Results of SVM, BLAST, HMM and kernNN from the study by Karchin et al.[13] are reproduced above for ease of comparison.

other half as a test set. Specifically, we ran the following four experiments. First, we trained the classifier on one of the two folds and divided the other fold into halves A and B. We used A to tune the parameter K and B to test the classifier. In the second experiment, we reversed the roles of A and B while keeping the same training set. The third and fourth experiments were the same as the first and second but with the roles of the two folds reversed.

Similar to the family-level classification, a graph of the accuracy plotted against K (data not shown) showed that accuracy increases as K increases until a maximum is reached, after which the accuracy decreases. Therefore, an improvement can be obtained by using only a subset of the features selected by chi-square. The accuracy of each classifier is shown in Table XV, along with a reproduction of the results reported by Karchin et al.[13] on the same dataset and using the same evaluation procedure.

Table XV shows that chi-square feature selection can improve the accuracy of the Naïve Bayes classifier while not harming the performance of the Decision Tree in level I subfamily classification. In either case, the optimal number of features selected by chi-square is much lower than the full set of all n-grams. Using the binary features as opposed to the n-gram counts seemed to be more beneficial to the Naïve Bayes classifier. The Naïve Bayes classifier outperforms all other classifiers in level I subfamily classification, achieving an accuracy of 93.0%. This is a 39.7% reduction in residual error from the reported 88.4% accuracy of SVM, a much more complicated classifier whose computational complexity was previously believed to be needed to achieve "annotation-quality" accuracy in GPCR subfamily classification.[13]

GPCR Level II Subfamily Classification

Next, using the Decision Tree and the Naïve Bayes classifier, we repeated the two-fold cross-validation evaluation with the same training–testing data split on the level II subfamily classification used by Karchin et al.[13] Ideally, the study would use independent training, validation and testing sets, as in level I subfamily classification. Unfortunately, this was not possible because some level II subfamilies have only two sequences. However, since we were using the averaged accuracy from a cross-validation to tune K, the effect from overfitting our classifiers to the dataset should be minimal.

Plotting the accuracy of the Decision Tree and the Naïve Bayes classifier as a function of the number of features K produced graphs similar to those in family-level classification (data not shown). The accuracy of our classifiers with and without chi-square feature selection is shown in Table XVI, along with a reproduction of the results reported by Karchin et al.

Here, using binary features selected by chi-square with the Naïve Bayes classifier was more effective than using the counts of the corresponding n-grams, giving an improvement of 10.5% in accuracy. Comparison to the previously studied classifiers shows that the Naïve Bayes classifier with its accuracy rate of 92.4% gave a 44.5% reduction in residual error compared to the SVM, whose reported 86.3% accuracy made it the best of the previously studied classifiers. Although the Decision Tree did not perform as well as SVM, it still outperformed HMM and kernNN with the aid of chi-square feature selection. This result shows that the computational complexity of SVM, which has been considered to be necessary for high accuracy in GPCR level II subfamily classification,[13] can be avoided by using the simple feature selection algorithm, chi-square, on a different feature set, the n-grams.

Generalizing to a Different Dataset: Classification of the Nuclear Receptor Superfamily

To show that our method can be applied to other protein classification problems, we applied the Naïve Bayes classifier with chi-square feature selection to the superfamily of

TABLE XVI. Comparison of Accuracy of Various Classifiers at GPCR Level II Subfamily Classification

Classifier	# of Features	Type of Features	Accuracy
Decision Tree	All (9723)	*n*-Gram counts	66.0%
	2300	Binary	70.2%
	1200	*n*-Gram counts	70.8%
Naïve Bayes	All (9702)	*n*-Gram counts	81.9%
	8100	Binary	92.4%
	5600	*n*-Gram counts	84.2%
SVM	9 per match state in the HMM	Gradient of the log-likelihood that the sequence is generated by the given HMM model	86.3%
SVMtree	9 per match state in the HMM	Gradient of the log-likelihood that the sequence is generated by the given HMM model	82.9%
BLAST		Local sequence alignment	74.5%
SAM-T2K HMM		HMM model built for each protein subfamily	70.0%
kernNN	9 per match state in the HMM	Gradient of the log-likelihood that the sequence is generated by the given HMM model	51.0%

Unigrams, bigrams and trigrams were used with the Decision Tree, while bigrams and trigrams were used with the Naïve Bayes classifier. Results of SVM, BLAST, HMM and kernNN from the study by Karchin et al.[13] are reproduced above for ease of comparison.

TABLE XVII. Results of Naïve Bayes Classifier with Chi-Square Feature Selection Applied to Nuclear Receptor Classification

			Accuracy	
Dataset	Feature Type	# of Features	Validation	Testing
Family	Binary	1500–4200	96.96%	94.53%
	n-Gram counts	400–4900	95.75%	91.79%
Level I subfamily	Binary	1500–3100	98.09%	97.77%
	n-Gram counts	500–1100	93.95%	91.40%
Level II subfamily	Binary	1500–2100	95.32%	93.62%
	n-Gram counts	3100–5600	86.39%	85.54%

nuclear receptors to family-level as well as level I and level II subfamily classification. Nuclear receptors were chosen because of their immense influence on the metabolic pathways of diseases such as diabetes, heart diseases and cancer.

In level I and level II subfamily classification, we performed a three-fold cross-validation in which one fold was used for training, another for tuning K and the third for testing. Table XVII shows the results of the experiments.

Table XVII clearly illustrates that using the binary features as opposed to their associated *n*-gram counts is more effective in boosting the accuracy of the classifier on all three levels of classification. The testing set accuracy for all three datasets falls in the mid-90% range, supporting our hypothesis that a simple classifier like the Naïve Bayes can give "annotation-quality" classification for protein sequences in general.

DISCUSSION

In this study, we evaluated the performance of simple classifiers in conjunction with feature selection against more complex classifiers in terms of running time complexity on the task of protein sequence classification. We chose to use the superfamily of GPCRs as our dataset because of its biological importance, particularly in pharmacology, and the known difficulty it presents in the classification task due to the extreme diversity among its members. Our method, analogous to document classification in the human language technologies domain, used the Decision Tree and Naïve Bayes classifiers on *n*-gram counts.

We optimized our classification procedure with feature selection using classification at the family level. In document classification, chi-square feature selection has proven to be highly successful,[31] not only in reducing the number of features necessary for accurate classification, but also in increasing classification accuracy via the elimination of "noisy features." We applied chi-square feature selection to the GPCR family classification task and found it to be successful in this task as well. Specifically, using chi-square feature selection, the accuracy increased with the number of features until a maximum accuracy was reached, after which the accuracy dropped. Thus, an improvement in accuracy can be attained by using chi-square to reduce the dimensionality of the feature space to the point at which maximum accuracy occurs.

We then applied our method to the GPCR level I and level II subfamily classification tasks studied previously by Karchin et al.[13] in a systematic comparison of classifiers of varying complexity. For comparability, we used the same dataset and evaluation procedure as published in the previous study. First, we noted that subfamily classifications are much more difficult to predict than family level classifications, as shown by the decrease in accuracy of both the Decision Tree and the Naïve Bayes classifier. This observation is consistent with the fact that subfamilies are

defined to a greater extent than families are by chemical and pharmacological criteria as opposed to sequence homology.

Because of these difficulties, the previous study concluded that at the subfamily levels, more complex classifiers are needed to maintain high classification accuracy. In particular, the accuracies of BLAST, k-nearest neighbors in conjunction with Fisher Score Vector space, profile HMM and SVM in level I and level II subfamily classification were studied with alignment-based features, and SVM was found to perform best. Using SVM, accuracy values of 88.4% and 86.3% were achieved in level I and level II subfamily classification[13] (see Tables XV and XVI, respectively).

In level I subfamily classification, we found that the Naïve Bayes classifier using the counts of all bigrams and trigrams can reduce the residual error from SVM by 13.8%. Moreover, a greater reduction of 39.7% can be achieved if chi-square feature extraction is used in conjunction with the Naïve Bayes classifier, leading to a final accuracy of 93.0%. In level II subfamily classification, the Naïve Bayes classifier with the aid of chi-square feature selection reduced the residual error from SVM by 44.5% and achieved an accuracy of 92.4%. Thus, contrary to the conclusion of the previous study,[13] our study shows that classifiers of the complexity of SVM are not needed to attain "annotation-quality" accuracy.

The comparison of our results to those of Karchin et al.[13] also shows that the Decision Tree cannot match the performance of the Naïve Bayes classifier and SVM in either level I or II subfamily classification. However, chi-square improves the accuracy of the Decision Tree to the extent that it outperforms HMM in both of these tasks.

One interesting observation in our level I subfamily classification results (Table XV) is that, while the Naïve Bayes classifier performed better with the help of chi-square feature selection, it also outperformed all other classifiers even on its own using counts of all bigrams and trigrams. This suggests that the difference in performance between the Naïve Bayes classifier and SVM may be due to the different features used. It is known that alignment-based methods have limitations[6] because of their assumption that contiguity is conserved between homologous segments which may not be true in genetic recombination or horizontal transfer.[7,8] As a result, alignments become ambiguous when sequence similarity drops below 40%[9] and unusable below 20%.[10,11] A number of approaches to alignment-free sequence comparisons have been explored (see Introduction and ref. 6). The high accuracy we achieved in protein classification using *n*-gram features suggests that for protein classification, *n*-grams may be a better set of features than alignment-based features. In contrast to the requirement by sequence alignment that ordering of homologous segments be conserved, the use of *n*-gram counts can capture the presence of small conserved peptide fragments without posing any requirements on their sequential arrangement.

Although sequence alignment has dominated the field for many years because of its intuitive nature in understanding the evolutionary origin of protein families and subfamilies, relaxing the requirement for consecutive homologous segments is more in tune with the hallmark of protein structures. Protein structures are functional because of their arrangement in three-dimensional space, bringing about important contacts between amino acids that may be far apart in the linear amino acid sequence. These amino acids form structural motifs that are conserved but rearranged in their order through evolution. *n*-Gram counts may be able to capture the presence of these motifs when alignments cannot, as in cases in which sequence similarity is too low. Thus, an important future goal is to discriminate high classification accuracy due to suitable classifiers from that due to informative features. From our current experiments, we cannot determine whether the type of features, the feature selection process or the different classifiers used has caused the significant improvement of our simple Naïve Bayes classifier over the SVM classifier. To address this question, future work should combine strong classifiers such as SVM and boosting with well-selected *n*-gram vocabularies to see if further predictive accuracy can be attained.

CONCLUSIONS

From the study presented here, we conclude that complicated classifiers with the running time complexity of SVM are not necessary to attain high accuracy in protein classification, even for the particularly challenging GPCR subfamily classification task. A simple classifier, the Naïve Bayes classifier, in conjunction with chi-square feature selection, applied to *n*-gram counts performs soundly better than a computationally complex and generally better classifier (SVM) on alignment-based features without feature selection in GPCR family, level I subfamily and level II subfamily classification. Another simple classifier, Decision Tree with chi-square feature selection, while not as powerful as either Naïve Bayes or SVM, can still outperform profile HMM. These classifiers perform comparably to profile HMMs created from human-curated alignments. We also show that the strong classification results can be extended to other protein families by their successful application to the superfamily of nuclear receptors with accuracies in the mid-90% range. Furthermore, we show that the accuracies achieved with our automated classifiers perform comparably to the current state-of-art hand-edited protein family HMM's stored in the PFAM database. Thus, given the right features, complicated classifiers with the running time complexity of SVM are not necessary to attain high accuracy in protein classification. Automatically formulating the right vocabulary via *n*-grams and chi-square feature selection is more important than the choice of classifier in achieving high accuracy. Using simple machine learning classifiers and feature selection techniques, we have created a reliable and automatic tool for general protein family classification.

All of the methods presented here were originally applied to the text document classification task in the human language technologies domain. The successful application of document classification techniques to the protein classi-

fication task, together with the conclusion that simple classifiers can outperform complicated classifiers in this task as a result, have important implications. There are many problems in the biology domain that can be formulated as a classification task. Many of these are considered to be more challenging by biologists than the protein classification task. This includes predicting folding, tertiary structure and functional properties of proteins, such as protein–protein interactions. Thus, these important classification tasks are potential areas for applications of human language technologies in modern proteomics.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support by National Science Foundation Large Information Technology Research grants NSF 0225636 and NSF 0225656, NIH grant NLM108730, the Sofya Kovalesvkaya Award from the Humboldt Foundation and the Zukunftsinvestitionsprogramm der Bundesregierung Deutschland.

REFERENCES

1. Smith TF, Waterman MS. Identification of common molecular subsequences. J Mol Biol 1981;147(1):195–197.
2. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 1970;48(3):443–453.
3. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215(3):403–410.
4. Smith HO, Annau TM, Chandrasegaran S. Finding sequence motifs in groups of functionally related proteins. Proc Natl Acad Sci USA 1990;87(2):826–830.
5. Neuwald AF, Green P. Detecting patterns in protein sequences. J Mol Biol 1994;239(5):698–712.
6. Vinga S, Almeida J. Alignment-free sequence comparison-a review. Bioinformatics 2003;19(4):513–523.
7. Lynch M. Intron evolution as a population-genetic process. Proc Natl Acad Sci USA 2002;99(9):6118–6123.
8. Zhang YX, Perry K, Vinci VA, Powell K, Stemmer WP, del Cardayre SB. Genome shuffling leads to rapid phenotypic improvement in bacteria. Nature 2002;415(6872):644–646.
9. Wu CH, Huang H, Yeh LS, Barker WC. Protein family classification and functional annotation. Comput Biol Chem 2003;27(1):37–47.
10. Pearson WR. Effective protein sequence comparison. Meth Enzymol 1996;266:227–258.
11. Pearson WR. Empirical statistical estimates for sequence similarity searches. J Mol Biol 1998;276(1):71–84.
12. Baldi P, Brunak S. Bioinformatics: the machine learning approach. Cambridge: MIT Press; 2001.
13. Karchin R, Karplus K, Haussler D. Classifying G-protein coupled receptors with support vector machines. Bioinformatics 2002;18(1):147–159.
14. Deshpande M, Karypis G. Evaluation of techniques for classifying biological sequences. In 6th Pacific-Asia Conference on Knowledge Discover (PAKDD). 2002. p 417–431.
15. Gether U. Uncovering molecular mechanisms involved in activation of G protein-coupled receptors. Endocr Rev 2000;21(1):90–113.
16. Muller G. Towards 3D structures of G protein-coupled receptors: a multidisciplinary approach. Curr Med Chem 2000;7(9):861–888.
17. Moriyama EN, Kim J. Protein family classification with discriminant function analysis. In: Gustafson, J.P., editor. Data mining the genomes: 23rd Stadler Genetics Symposium. New York: Kluwer Academic/Plenum. 2003.
18. Kolakowski LF, Jr. GCRDb: a G-protein-coupled receptor database. Receptors Channels 1994;2(1):1–7.
19. Horn F, Weare J, Beukers MW, Horsch S, Bairoch A, Chen W, Edvardsen O, Campagne F, Vriend G. GPCRDB: an information system for G protein-coupled receptors. Nucl Acid Res 1998;26(1):275–279.
20. Lapinsh M, Gutcaits A, Prusis P, Post C, Lundstedt T, Wikberg JE. Classification of G-protein coupled receptors by alignment-independent extraction of principal chemical properties of primary amino acid sequences. Prot Sci 2002;11(4):795–805.
21. Levchenko ME, Katayama T, Kanehisa M. Discovery and classification of peptide family G-protein coupled receptors in the human genome sequence. Genome Informatics 2001:352-353.
22. SSearch. 10.3: Accelrys Inc.; http://www.biology.wustl.edu/gcg/ssearch.html.
23. Liu A, Califano A. Functional classification of proteins by pattern discovery and top-down clustering of primary sequences. IBM Systems Journal: Deep Computing for the Life Sciences 2001;40(2):379–393.
24. Yuan X, Yuan X, Buckles BP, Zhang J. A comparison study of decision tree and SVM to classify gene sequence. 2003.
25. Quinlan JR. C4.5. Release 8; http://www.rulequest.com/Personal/c4.5r8.tar.gz.
26. Quinlan JR. C4.5: Programs for machine learning: Morgan Kaufmann; 1993.
27. Mitchell T. Machine learning, chapter 3. McGraw Hill; 1997.
28. McCallum A. Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering. February 13, 2002; http://www-2.cs.cmu.edu/~mccallum/bow/.
29. McCallum A, Nigam K. A comparison of event models for Naïve Bayes text classification. In AIII-98 Workshop on "Learning for Text Categorization." 1998.
30. Sebastiani F. A tutorial on automated text categorization. In Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence. 1999. p 7–35.
31. Yang Y, Pedersen J. A comparative study on feature selection in text categorization. In Proceedings of ICML-97, 14th International Conference on Machine Learning. 1997. p 412–420.
32. Pardo L, Ballesteros JA, Osman R, Weinstein H. On the use of the transmembrane domain of bacteriorhodopsin as a template for modeling the three-dimensional structure of guanine nucleotide-binding regulatory protein-coupled receptors. Proc Natl Acad Sci USA 1992;89(9):4009–4012.
33. Horn F, Vriend G, Cohen FE. Collecting and harvesting biological data: the GPCRDB and NucleaRDB information systems. Nucleic Acids Res 2001;29(1):346–349.
34. Bateman A, Coin L, Durbin R, Finn RD, Hollich V, Griffiths-Jones S, Khanna A, Marshall M, Moxon S, Sonnhammer EL, Studholme DJ, Yeats C, Eddy SR. The Pfam protein families database. Nucleic Acids Res 2004;32 Database issue:D138–141.
35. Corpet F, Servant F, Gouzy J, Kahn D. ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. Nucleic Acids Res 2000;28(1):267–269.
36. Servant F, Bru C, Carrere S, Courcelle E, Gouzy J, Peyruc D, Kahn D. ProDom: automated clustering of homologous domains. Brief Bioinform 2002;3(3):246–251.
37. Pearson WR. Flexible sequence similarity searching with the FASTA3 program package. Methods Mol Biol 2000;132:185–219.
38. Park J, Teichmann SA, Hubbard T, Chothia C. Intermediate sequences increase the detection of homology between sequences. J Mol Biol 1997;273(1):349–354.
39. Zhang Z, Schaffer AA, Miller W, Madden TL, Lipman DJ, Koonin EV, Altschul SF. Protein sequence similarity searches using patterns as seeds. Nucleic Acids Res 1998;26(17):3986–3990.
40. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997;25(17):3389–3402.
41. Henikoff S, Henikoff JG, Alford WJ, Pietrokovski S. Automated construction and graphical presentation of protein blocks from unaligned sequences. Gene 1995;163(2):GC17–26.
42. Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res 1994;22(22):4673–4680.
43. Morgenstern B, Frech K, Dress A, Werner T. DIALIGN: finding local similarities by multiple sequence alignment. Bioinformatics 1998;14(3):290–294.
44. Morgenstern B. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics 1999;15(3):211–218.
45. Schuler GD, Altschul SF, Lipman DJ. A workbench for multiple alignment construction and analysis. Proteins 1991;9(3):180–190.

46. Taylor WR. A flexible method to align large numbers of biological sequences. J Mol Evol 1988;28(1-2):161–169.
47. Barton GJ, Sternberg MJ. A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons. J Mol Biol 1987;198(2):327–337.
48. Wisconsin Package. 10.3: Accelrys Inc.; http://www.accelrys.com/products/gcg_wisconsin_package/.
49. Notredame C, Higgins DG. SAGA: sequence alignment by genetic algorithm. Nucleic Acids Res 1996;24(8):1515–1524.
50. Notredame C, Higgins DG, Heringa J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol 2000;302(1):205–217.
51. Birney E, Clamp M, Durbin R. GeneWise and Genomewise. Genome Res 2004;14(5):988–995.
52. Eddy S. HMMER. 2.3.2; http://hmmer.wustl.edu/.
54. Grundy WN, Bailey TL, Elkan CP, Baker ME. Meta-MEME: motif-based hidden Markov models of protein families. Comput Appl Biosci 1997;13(4):397–406.
55. Bucher P, Karplus K, Moeri N, Hofmann K. A flexible motif search technique based on generalized profiles. Comput Chem 1996;20(1):3–23.
56. Neuwald AF, Liu JS, Lipman DJ, Lawrence CE. Extracting protein alignment models from the sequence database. Nucleic Acids Res 1997;25(9):1665–1677.
57. Krogh A, Brown M, Mian IS, Sjolander K, Haussler D. Hidden Markov models in computational biology. Applications to protein modeling. J Mol Biol 1994;235(5):1501–1531.
58. Henikoff S, Henikoff JG, Pietrokovski S. Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. Bioinformatics 1999;15(6):471–479.
59. Henikoff JG, Greene EA, Pietrokovski S, Henikoff S. Increased coverage of protein families with the blocks database servers. Nucleic Acids Res 2000;28(1):228–230.
60. Huang JY, Brutlag DL. The EMOTIF database. Nucleic Acids Res 2001;29(1):202–204.
61. Attwood TK, Blythe MJ, Flower DR, Gaulton A, Mabey JE, Maudling N, McGregor L, Mitchell AL, Moulton G, Paine K, Scordis P. PRINTS and PRINTS-S shed light on protein ancestry. Nucleic Acids Res 2002;30(1):239–241.
62. Falquet L, Pagni M, Bucher P, Hulo N, Sigrist CJ, Hofmann K, Bairoch A. The PROSITE database, its status in 2002. Nucleic Acids Res 2002;30(1):235–238.
63. Sigrist CJ, Cerutti L, Hulo N, Gattiker A, Falquet L, Pagni M, Bairoch A, Bucher P. PROSITE: a documented database using patterns and profiles as motif descriptors. Brief Bioinform 2002;3(3):265–274.
64. Ponting CP, Schultz J, Milpetz F, Bork P. SMART: identification and annotation of domains from signaling and extracellular protein sequences. Nucleic Acids Res 1999;27(1):229–232.
65. Letunic I, Goodstadt L, Dickens NJ, Doerks T, Schultz J, Mott R, Ciccarelli F, Copley RR, Ponting CP, Bork P. Recent improvements to the SMART domain-based sequence annotation resource. Nucleic Acids Res 2002;30(1):242–244.
66. Gough J, Karplus K, Hughey R, Chothia C. Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. J Mol Biol 2001;313(4):903–919.
67. Apweiler R, Gateau A, Contrino S, Martin MJ, Junker V, O'Donovan C, Lang F, Mitaritonna N, Kappus S, Bairoch A. Protein sequence annotation in the genome era: the annotation concept of SWISS-PROT+TREMBL. Proc Int Conf Intell Syst Mol Biol 1997;5:33–43.
68. Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S, Schneider M. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res 2003;31(1):365–370.
69. Qu K, McCue LA, Lawrence CE. Bayesian protein family classifier. Proc Int Conf Intel Syst Mol Biol 1998;6:131–139.
70. Wang JTL, Ma Q, Shasha D, Wu CH. Application of neural networks to biological data mining: a case study in protein sequence classification. In Proceedings of the 6th ACM SIG-KDD International Conference on Knowledge Discovery and Data Mining. 2000. p 305–309.
71. Yona G, Linial N, Linial M. ProtoMap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. Proteins 1999;37(3):360–378.
72. Mitsuke H, Sugiyama Y, Shimizu T. Classification of transmembrane protein families based on topological similarity. Genome Informatics 2002;13:418-419.
73. Kim J, Moriyama EN, Warr CG, Clyne PJ, Carlson JR. Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties. Bioinformatics 2000;16(9):767–775.
74. Wu CH, Berry M, Shivakumar S, McLarty J. Neural networks for full-scale protein sequence classification: sequence encoding with singular value decomposition. Machine Learning 1995;21(1):177–193.
75. Ferran EA, Ferrara P. Clustering proteins into families using artificial neural networks. Comput Appl Biosci 1992;8(1):39–44.
76. Eskin E, Grundy WN, Singer Y. Protein family classification using sparse Markov transducers. Proc Int Conf Intell Syst Mol Biol 2000;8:134–145.
77. Eskin E, Noble WS, Singer Y. Protein family classification using sparse Markov transducers. J Comput Biol 2003;10(2):187–213.
78. Jaakkola T, Diekhans M, Haussler D. Using the Fisher kernel method to detect remote protein homologies. Proc Int Conf Intell Syst Mol Biol 1999:149–158.
79. Jaakkola T, Diekhans M, Haussler D. A discriminative framework for detecting remote protein homologies. J Comput Biol 2000;7(1-2):95–114.
80. Leslie C, Eskin E, Noble WS. The spectrum kernel: a string kernel for SVM protein classification. Pac Symp Biocomput 2002:564–575.
81. Leslie CS, Eskin E, Cohen A, Weston J, Noble WS. Mismatch string kernels for discriminative protein classification. Bioinformatics 2004;20(4):467–476.
82. Vanschoenwinkel B, Reumers J, Manderick B. A text mining and support vector machine approach to protein classification. In International Workshop in Bioinformatics, "Knowledge Discovery meets Drug Discovery." 2002.