Evolving Classifiers for Knowledge Discovery in Medical and Biological Datasets

Michael R. Peterson,[*] Travis E. Doom, Michael L. Raymer

Abstract

A key element of bioinformatics research is the extraction of meaningful information from large experimental data sets. Various approaches, including statistical and graph theoretical methods, data mining, and computational pattern recognition, have been applied to this task with varying degrees of success. Using a suite of classifiers combined with evolutionary algorithms for parameter adjustment and feature extraction, we present a set of hybrid algorithms that employ simultaneous feature selection and extraction to isolate salient features from large medical and other biological data sets.

We have previously shown that a genetic algorithm coupled with a k-nearest-neighbors classifier performs well in extracting information about protein-water binding from X-ray crystallographic protein structure data. Here, the effectiveness of several new hybrid classifiers in feature selection and classification is demonstrated for this and other bioinformatic, medical, and scientific data sets.

1 Introduction

Feature selection facilitates classification by removing non-salient features. Even features providing some useful information may reduce accuracy when there are a limited number of training points available [1]. This "curse of dimensionality", along with the expense of measuring additional features, motivates feature dimensionality reduction [2]. Though no known deterministic algorithm finds the optimal feature set for the general case in polynomial time, a wide range of feature selection algorithms may find near-optimal feature sets [3].

The main purpose of feature selection is to reduce the number of features used in classification while maintaining an acceptable classification accuracy. Less discriminatory features are eliminated, leaving a subset of the original features which retains sufficient information to discriminate well among classes. For most problems, an exhaustive approach is prohibitively expensive in terms of computation time. Cover and Van Campenhout [4] have shown that to find an optimal subset of size n from the original d features, it is necessary to evaluate all $\binom{d}{n}$ possible subsets when the statistical dependencies among the features are not known. Furthermore, when the size of the feature subset is not specified in advance, each of the (2^d) subsets of the original d features must be evaluated. In the special case where the addition of a new feature always improves performance, it is possible to significantly reduce

[*] The authors are with the Department of Computer Science and Engineering, Wright State University, 3640 Colonel Glenn Hwy., Dayton, OH 45435-0001 (email [mpeterso | doom | mraymer]@cs.wright.edu)

the number of subsets that must be evaluated using a branch and bound search technique [5]. Unfortunately, this sort of monotonic decrease in the error rate as new features are added is often not found in real-world classification problems.

Feature selection approaches can be categorized into two disjoint groups. Filter methods perform feature selection prior to classification, generally using an efficiently computable metric, such as Mahalanobis distance between class means, to evaluate the quality of a particular feature subset. In contrast, wrapper methods perform feature selection in conjunction with classification. For these methods, the classification accuracy of the actual classifier to be used provides the quality assessment of the feature subsets being evaluated.

Various heuristic methods have been proposed to search for near-optimal feature subsets. Sequential methods, including sequential forward selection [6] and sequential backward selection, involve the addition or removal of a single feature at each step, and can be used in either of the two approaches described above. "Plus l – take away r" selection combines these two methods by alternately enlarging and reducing the feature subset repeatedly. The sequential floating forward selection algorithm (SFFS) of Pudil *et al.* [7] is a further generalization of the plus l, take away r methods, where l and r are not fixed, but rather are allowed to "float" to approximate the optimal solution as much as possible. Jain and Zongker [8] have demonstrated the effectiveness of the SFFS method for multivariate Gaussian data distributions.

When classification is being performed using neural networks, node pruning techniques can be used for dimensionality reduction [9]. After training for a number of epochs, nodes are removed from the neural network in such a manner that the increase in squared error is minimized. When an input node is pruned, the feature associated with that node is no longer considered by the classifier. Similar methods have been employed in the use of fuzzy systems for pattern recognition through the generation of fuzzy if-then rules [10, 11]. Some traditional pattern classification techniques, while not specifically addressed to the problem of dimensionality reduction, can provide feature selection capability. Tree classifiers [12], for example, typically partition the training data based on a single feature at each tree node. If a particular feature is not tested at any node of the decision tree, it is effectively eliminated from classification. Additionally, simplification of the final tree can provide further feature selection [13].

Feature extraction, a superset of feature selection, involves transforming the original set of features to provide a new set of features, where the transformed feature set usually consists of fewer features than the original set. While both linear and non-linear transformations have been explored, most of the classical feature extraction techniques involve linear transformations of the original features. Formally, the objective for linear feature extraction techniques can be stated as follows:

> Given an $n \times d$ pattern matrix $\mathcal{A}$ (n points in a d-dimensional space), derive an $n \times m$ pattern matrix $\mathcal{B}$, $m < d$, where $\mathcal{B} = \mathcal{A}\mathcal{H}$ and $\mathcal{H}$ is a $d \times m$ transformation matrix.

According to this formalization, many common methods for linear feature extraction can be specified according to the method of deriving the transformation matrix, $\mathcal{H}$. For unsupervised linear feature extraction, the most common technique is principal component analysis [14]. For this method, the columns of $\mathcal{H}$ consist of the eigenvectors of the $d \times d$ covariance matrix of the given patterns. It can be shown that the new features produced by principal component analysis are uncorrelated and maximize the variance retained from the original feature set [14]. The corresponding supervised technique is linear discriminant analysis. In this case, the columns of $\mathcal{H}$ are the eigenvectors corresponding to the nonzero eigenvalues of the matrix $\mathcal{S}_W^{-1}\mathcal{S}_B$, where $\mathcal{S}_W$ is the within-class scatter matrix and $\mathcal{S}_B$ is the between-class scatter matrix for the given set of patterns. Deriving $\mathcal{H}$ in this way maximizes the separation between class means relative to the covariance of the classes [14]. In the general case, the matrix $\mathcal{H}$ is chosen to maximize some criteria, typically related to class separation or classification accuracy for a specific classifier. In this view, feature selection is a special case of linear feature extraction, where the off-diagonal entries of $\mathcal{H}$ are zero, and the diagonal entries are either zero or one.

The accuracy of some types of classification rules, such as k-nearest neighbors, improves by multiplying the value of each feature by a value proportional to its usefulness in classification. The assignment of weights to each feature as a form of feature extraction in combination with a $k$nn classifier improves classifier accuracy over the knn classifier alone, and aids in the analysis of large datasets by isolating combinations of salient features [15].

These feature weights can be assigned using a variety of techniques. One possibility is to optimize feature weights using evolutionary computation (EC). EC methods, generally speaking, are optimization and search algorithms based on the mechanics of Darwinian evolution and natural selection. While a number of different variants upon the general EC theme exist, they are tied by a few common traits, including the use of a population of competing solutions and a stochastic selection mechanism for culling potential solutions at each iteration of the algorithm. Popular EC techniques include Evolutionary Programming (EP) [16], Evolution Strategies (ES) [17,18], Genetic Programming (GP) [19], and Genetic Algorithms (GAs) [20,21]. In recent years, the various forms of EC have begun to merge as the methods and operators that distinguish each technique have been combined into hybrid methods. The optimization methods described here are based on Genetic Algorithms. A detailed description of the techniques and terminology of GAs can be found in [22].

Through use of a bit-masking feature vector, GAs have successfully performed feature selection in combination with a $k$nn classifier [23]. Later works expand this approach for linear feature extraction [15,24] by searching for an ideal set of feature weights. The single bit associated with each feature is expanded to a real-valued coefficient, allowing independent linear scaling of each feature, while maintaining the ability to remove features from consideration by assigning a weight of zero. Given a set of feature vectors of the form $\mathbf{X} = \{x_1, x_2, ...x_d\}$, the GA produces a transformed set of vectors of the form $\mathbf{X'} = \{w_1x_1, w_2x_2...w_dx_d\}$ where w_i is a weight associated with feature i. Each feature value is first normalized, then scaled by the associated weight prior to training, testing, and classification. This linear scaling of features prior to classification allows a classifier to discriminate more finely along feature axes with larger scale factors. A k-nearest-neighbors (knn) classifier is used to evaluate each set of feature weights. Patterns plotted in feature space are spread out along feature axes with higher weight values, and compressed along features with lower weight values. The value of k for the knn classifier is fixed and determined empirically prior to feature extraction.

In a similar approach, Yang and Honavar [25] use a simple Evolutionary Computation (EC) algorithm for feature subset selection in conjunction with DistAI, a neural network-based pattern classifier [26]. As in other EC-based feature selectors, a simple binary representation was used where each bit corresponds to a single feature. The use of the EC for feature subset selection improved the accuracy of the DistAI classifier for nearly all of the data sets explored, while simultaneously reducing the number of features considered. Their hybrid classifier, GADistAI, outperformed a number of modern classification methods on the various data sets presented.

Vafaie and De Jong [27] describe a hybrid technique in which EC methods are employed for both feature selection and extraction[1] in conjunction with the C4.5 decision tree classifier system [28]. Again, a binary representation is used for feature subset selection using traditional EC techniques. In this system, however, the features seen by the classifier are functions of the original features composed of simple arithmetic operations. For example, one such feature might be $\{(F1-F2)\times(F2-F4)\}$, where $F1, F2$, and $F4$ represent values from the original feature set.

The authors have developed a novel hybrid GA/$k$nn system that eliminates the mask vector and instead employs a population-adaptive mutation technique allowing for improved simultaneous feature selection and extraction on the weight vector [29]. Rather than maintaining separate feature weights and mask bits on the GA chromosome, a single weight value is evolved using a technique known as population-adaptive mutation that encourages low-weighted features to become masked. Additionally, a cosine similarity metric replaces the traditional Euclidean distance metric for $k$nn classification. This similarity metric allows for a novel form of GA optimization by searching for an optimal set of feature offsets used to determine the cosine of the angles between various data points considered by the knn classifier.

In this paper, we describe a method for classifier optimization and knowledge discovery that involves hybridizing a GA with a cosine-based k-nearest neighbors classifier, and compare this method with our previous work hybridizing GAs with Euclidean $k$nn and Bayes classifiers, as well as with other contemporary pattern recognition techniques. Details of the implementation and performance of the hybrid GA/Euclidean $k$nn classifier can be found in [30, 31]. Similar information regarding the hybrid GA/Bayes classifier is in [32].

The knowledge discovery properties inherent in our hybrid GA/classifier systems provide new insight into the role of water molecules during the binding of drugs or other ligands to the protein surface. Protein surface-bound water molecules often form hydrogen bonds to a docking drug or other ligand. These conserved water molecules are an essential part of the protein surface with respect to ligand screening, docking, and design [30, 33]. However, the identification of favorable protein surface sites for solvent binding has proven difficult, partially because the majority of protein surface residues are hydrophilic.

Among the various attempts to treat water molecules during ligand binding, the *Consolv* system [30] employs a (GA/Euclidean $k$nn) classifier to distinguish water molecules bound in the protein's ligand-binding site from those displaced upon ligand binding. The recently-developed hybrid GA/cosine $k$nn classifier described here improves upon both *Consolv's* and the hybrid GA/Bayes classifier's reported performance while simultaneously performing data mining to aid in understanding the physical and chemical properties governing these protein-water interactions. This new hybrid GA/classifier enables knowledge discovery by performing feature selection and extraction to improve classification accuracy and mine feature relevance. Results of this method are compared with the *Consolv* method, the EC/Bayes method, and with a number of traditional classification techniques on experimental

[1] The authors use the term "feature construction".

water-protein interaction data.

2 Methods

2.1 Cosine-based K-nearest neighbors classification

The GA feature selection and extraction techniques described here are performed in conjunction with a specific classifier. That is, they operate as wrappers rather than filters. Among a variety of classification methods evaluated, the performance of a simple k nearest neighbor ($k$nn) classification rule has been competitive. In $k$nn classification, training patterns are plotted in d-dimensional space, where d is the number of features present. These patterns are plotted according to their observed feature values and are labeled according to their known class. An unlabelled test pattern is plotted within the same space and is classified according to the most frequently occurring class among its k-most similar training patterns; its nearest neighbors. Knn classifiers often employ a Euclidean distance similarity metric, though in some cases, other similarity metrics such as cosine similarity may be used.

The classifier here employs the following cosine similarity metric. If x_i and x_j are feature vectors representing two patterns, then the cosine of the angle between them is defined as:

$$cos(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|}$$

where "$\cdot$" represents the dot product between the vectors, and $\|x\|$ represents the vector length. Larger cosine values represent a greater similarity between vectors. Classification occurs after similarity-based neighbor identification. The algorithm here assigns class labels using a weighted scheme based on each neighbor's similarity to the query point. If the data contains only two classes, the positive and the negative class, then the query point, x, is classified by the value of the measure q [34]:

$$q = \sum_{i=1}^{n} cos(x_i, x) c(x_i)$$

where

$$c(x_i) = \begin{cases} 1 & : \text{ if } x_i \in \text{the positive class} \\ -1 & : \text{ otherwise} \end{cases}$$

The query point is assigned to the positive class if q is positive, otherwise it is assigned to the negative class.

There are multiple opportunities for cosine-based $k$nn classifier optimization. First, these classifiers are sensitive to feature weighting. Figure 1 illustrates this point for a 5-nearest neighbor classifier within 2-dimensional feature space. Within the plots shown, the triangle represents an unclassified test pattern. In **(a)**, among the test pattern's five nearest neighbors, three belong to class 2, and two belong to class 1. The test pattern is labeled as belonging to class 2, since the sum of the cosine of the angles between the test pattern and its class 2 neighbors is larger than that of its class 1 neighbors. In plot **(b)**, feature 2 has an increased weight relative to feature 1, as shown by the extension of the axis representing feature 2. After applying this scaling factor, the classification of the test pattern changes from class 2 to class 1, because among the nearest neighbors, three now represent class 1 points. The sum of the cosine of angles between class 1 points and the test pattern is now greater than that of class 2 points. Because of the sensitivity of cosine-based $k$nn classifiers to feature weights an evolutionary algorithm performing feature weight extraction may significantly improve the classification rate accuracy.

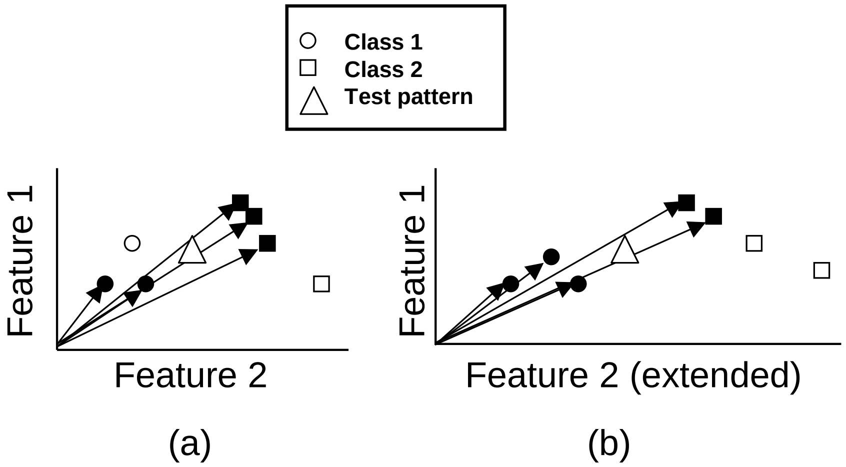

Figure 1: Effect of feature weighting on cosine-based $k$nn classification.

While feature weighting affects cosine-based $k$nn classification, feature offset shifting provides an additional opportunity for optimization. By shifting the origin, a GA affects the classification of unseen data. For cosine similarity the angle between two vectors is taken relative to the origin; a GA may perform feature extraction by shifting the position of the origin relative to the data. Shifting all feature values by a specific offset changes the point of reference used to compute the angle between two vectors, and the subsequent selection of near neighbors. In Figure 2 (a), the cosine values are determined using the default origin. Taking $k = 5$, the test pattern is assigned to class 1, since 3 of its 5 nearest neighbors belong to class 1. In Figure 2 (b), the origin is shifted. From the new perspective, the 5 nearest neighbors belong to class 2, so the test point's label is now 2. A GA increases cosine-based $k$nn classifier performance by simultaneously evolving feature weights and offsets.

Often one cannot obtain an equal number of training patterns for each class. Like some other approaches, $k$nn classifiers demonstrate a bias towards choosing the most frequent class when the training patterns are not balanced, since a large number of neighbors may belong to the majority class regardless of feature space position. To avoid bias, data training, test, and bootstrap subsets are created for each GA run. The number of points in the least-represented class determines the size of the class-balanced training and test sets. Additionally, the GA's fitness function favors optimizations leading to balanced performance. The resulting GA/$k$nn classifier reduces the voting bias.

2.2 The genetic algorithm

The GA used for cosine classifier optimization simultaneously evolves feature weights, offsets, and a k-value for classifier optimization. Figure 3 shows the GA chromosome employed for this process. $M_1 \ldots M_n$, represent real-valued weights for each

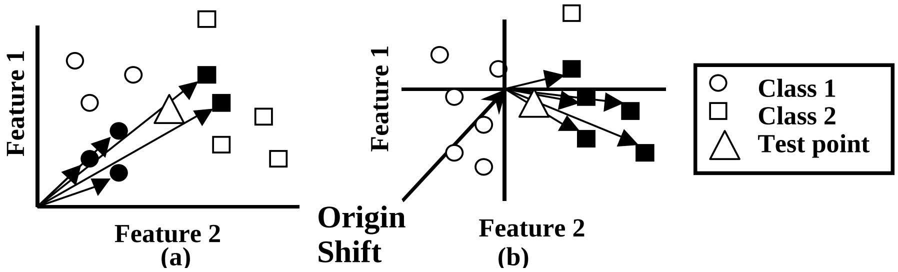

Figure 2: Effect of the origin position on cosine-based $k$nn classification.

of the n features, while $O_1 \ldots O_n$, represent real-valued offsets for each feature. K represents the k-value for classification. To avoid implicit weighting of features with different ranges of values, datasets are normalized in the range of [1.0 ... 10.0] prior to classification. On the GA chromosome, weights range from 0.0 to 100.0, offsets from -15.0 to 25.0, and the k-value ranges over integers from 1 to 100. Prior to classification, the weights are normalized to sum to one to represent a feature's relative importance, though the unnormalized values remain on the chromosome. The offset range is chosen in a manner that allows the origin for a given feature to be shifted to a range that extends past both the minimum and maximum feature values to a width of approximately twice the normalized range. For smaller datasets, the k-value may be restricted to a smaller range so that possible number of neighbors considered does not represent too large a portion of the overall dataset.

W_1	W_2	...	W_n	O_1	O_2	...	O_n	k

Figure 3: Structure of the GA chromosome.

As previously discussed, the inclusion of irrelevant features may degrade the performance of pattern recognition systems. Knn classifiers are especially sensitive to the inclusion of spurious features, since each feature affects the neighborhood of each test pattern. While feature weighting reduces the impact of spurious features, the complete removal of irrelevant features is desirable, since the cost of feature measurement decreases with the feature set size. In our previous EC/classifier systems [31, 32], the GAs employ bit-wise masks to explicitly perform feature selection. The separation of feature weight extraction and feature selection on the chromosome introduces the possibility of prematurely masking a relevant feature. If the GA gives preference to chromosomes masking a larger number of features, then a prematurely masked feature may never be unmasked, and the classifier performance may suffer.

To avoid this problem, the GA now employs population-adaptive mutation within the GA to implicitly perform feature selection while optimizing feature weights. In the absence of an explicit feature mask, the weight of a feature must be reduced to zero (or below some prespecified level) in order to remove the feature from consideration. When applied to feature weights, population-adaptive mutation increases the likelihood that the weights of spurious features will be reduced to zero. When a gene on the chromosome is selected for mutation, its value is randomly shifted either above or below its current value within a given range. This range depends upon the current values of the same gene across the entire GA population. The variance of a gene across the entire population is computed, and the standard deviation is used to set the mutation range. The gene's new value is instead randomly chosen from a Gaussian distribution with a mean equal to the gene's current value and a standard deviation based on the population's variance for that gene.

The effect of population adaptive mutation during the course of a GA run is demonstrated in Figure 4. **(a)** shows the probable range of a gene's mutation (represented by the shaded triangle) early in the GA run. Initially, the gene's variance across the population is higher, leading to a larger range of probable mutation. This behavior allows the GA to quickly explore diverse regions of the search space. In later generations, as the population begins to converge, the probable mutation range reduces, allowing the GA to fine-tune solutions **(b)**. Note that a gene may mutate to a value either above the maximum or below the minimum allowable value **(c)**. In this case, the gene is set to the maximum or minimum value. Thus, mutation frequently causes feature weights to be set to zero due to this boundary effect, effectively removing that feature from consideration. As the value of a feature weight gradually decreases across the population during a GA run, the probability the that feature will remain masked increases. Conversely, features with generally high weights are unlikely to become masked under population-adaptive mutation, thus preventing the premature masking problem.

The major drawback of population-adaptive mutation is that it becomes difficult for the GA to escape a locally-optimal solution in search of a globally-optimal solution, since the probability of a large mutation decreases as the population variance decreases. Nevertheless, each GA run thoroughly explores the area within the search space that the population converges to, due to the large number of small mutations late in the GA run. This allows for a more complete search within the local solution space than random value mutation allows. Repeated GA runs are required to throughly search various localities within the search space.

Chromosomes are evaluated by applying the weight and offset vectors to the feature set, and performing classification on a set of patterns of known class with a feature weighted and offset shifted cosine-based $k$nn classifier. The fitness function has several parts. The two most important parts measure the classification accuracy on known test data and the balance between accuracies of each class within the dataset. This allows the GA to simultaneously seek higher overall classification accuracy while maintaining accuracy balance among the different classes. Additionally, the GA gives a small preference to solutions employing fewer features, since it is desirable to achieve high accuracy without the expense of measuring many

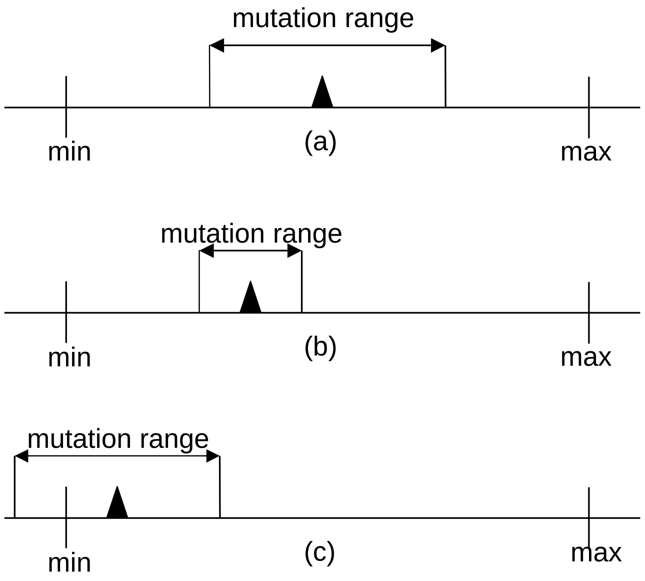

Figure 4: The effects of population-adaptive mutation.

features. The minimized GA cost function is:

$$\begin{aligned} cost(\vec{w}, k) &= C_{pred} \times \% \text{ of incorrect predictions} \\ &+ C_{mask} \times \# \text{ of unmasked features} \\ &+ C_{bal} \times \text{class accuracy difference} \end{aligned}$$

where C_{pred}, C_{mask}, and C_{bal}, are the cost function coefficients. For our experiments the empirically derived values: $C_{pred} = 25.0$, $C_{bal} = 10.0$, and $C_{mask} = 1.0$ are used. The GA places maximum emphasis upon achieving high classification accuracy; maintaining class balance and reducing the overall feature set are secondary goals. GA runs are typically executed for a maximum of 100 generations, with a population size of either 50 or 100. The probability of mutation of any position on the chromosome (weight, offset, or k-value) is 0.1 for each individual at each generation. Because this mutation rate is fairly high for a genetic algorithm, mutation drives the GA search as much as other genetic operators in these experiments.

For each GA experiment, datasets are split into class-balanced training and test sets, with remaining data patterns reserved for bootstrap validation upon GA run completion. At the completion of each GA run, the optimized classifier's performance is assessed over the previously unused remaining data patterns using a variant of the bootstrap test method [32, 35] in order to obtain both an unbiased accuracy estimate and a simple measure of the variance of this estimate. This bootstrap test helps ensure that the reported results are not the result of GA overfitting of the test data.

3 Experimental Results

3.1 Classification of Coordinated Water Molecules

One of the more challenging problems in structure-based rational drug design is modeling the interactions between a protein surface and the water molecules that surround the protein.

Water molecules bound in ligand-binding sites of proteins often form hydrogen bonds with ligands, making them an essential part of the protein surface with respect to ligand design, docking, and screening [30, 33]. Furthermore, surface-bound water molecules contribute to the formation and stabilization of surface grooves [36, 37], while internal water molecules can make a significant contribution to the overall structural stability of the protein [38]. While water is clearly important to protein structure and function, the identification of protein surface sites favorable for solvent binding (as opposed to bulk solvent interactions) has proven difficult.

Empirical methods for determining the favorability of a solvation site do so by analogy with known sites. A site is evaluated and compared to a database of known solvation and non-solvation sites, and predicted as being more similar to one than the other. A set of features to observe and compare between the solvation sites and non-solvated sites must be selected prior to classification. We use our novel classifier to distinguish the features of the data set that are the most relevant to water binding and to weigh these features appropriately to aid in the classification of possible water coordination sites.

For analyzing conserved solvation, the Brookhaven Protein Databank (PDB) [39, 40] was screened for high-resolution crystallographic protein structures to provide a knowledge base of crystallographically observed solvation sites. All proteins included in the database were non-homologous (had $\leq 25\%$ sequence identity, based on the PDB_Select list [41]), to avoid redundant structural information. Proteins with a resolution of $\leq\sim 2.0$ Å and low residual R-values were preferred. Table 1 lists the 30 proteins selected.

The first hydration shell was defined as the set of water molecules within 3.6 Å of any protein surface atom, and thus capable of making a van der Waals' contact or hydrogen bond with the protein. The environment of each of first-shell water molecules from each structure was characterized according to the six features listed in Table 2. The resulting database consisted of these six feature measurements for each of 5325 first-shell water molecules.

Water molecules from the 30 proteins in Table 1 were labeled as either conserved upon ligand binding or displaced, based upon superposition of ligand-bound and unbound structures. Water molecules in the unbound structure with corresponding water sites within 1.2 Å in the superimposed ligand-bound structure were identified as conserved. All other water molecules were identified as displaced.

In addition to predicting solvation site conservation, it was desirable to identify likely water binding sites for unsolvated protein models, such as those that might be generated by homology modeling. To distinguish protein solvation sites from non-solvated sites, it was first necessary to generate a set of randomly spaced probe sites about the protein surface where no bound water was observed in the crystal structure. To generate the probe site positions, the solvent accessible molecular surface was computed for each of the structures in Table 1 using MS [43]; a probe radius of 1.2 Å was used, with a surface density of 1 dot/Å^2 and unit normals generated at each surface dot. For each protein, the minimum distance of any

Table 1: Proteins included in the solvation knowledge base.

PDB	**Protein**	**Res(Å)**	**# Wats**
1ahc	α-momorcharin	2.0	163
1apm	cAMP-dependent protein kinase	2.0	207
1bia	bira bifunctional protein	2.3	43
1bsa	barnase	2.0	258
1ca2	carbonic anhydrase II	2.0	167
1cgf	fibroblast collagenase	2.1	181
1cgt	cyclodextrin glycosyltransferase	2.0	588
1chp	cholera toxin β pentamer	2.0	248
1dr2	dihydrofolate reductase	2.3	73
1gta	glutathione S-transferase	2.4	118
1hel	hen egg-white lysozyme	1.7	185
1lib	adipocyte lipid-binding protein	1.7	89
1nsb	neuraminidase	2.2	446
1poa	phospholipase A2	1.5	151
1syc	staphylococcal nuclease	1.8	69
1thm	thermitase	1.37	193
1udg	uracil-DNA glycosylase	1.75	121
2act	actinidin	1.7	272
2apr	acid proteinase	1.8	373
2cla	chloramphenicol acetyltransferase	2.35	104
2ctv	concanavalin A	1.95	146
2sga	proteinase A	1.5	220
2wrp	Trp repressor	1.65	170
3cox	cholesterol oxidase	1.8	453
3dni	deoxyribonuclease I	2.0	375
3enl	enolase	2.25	353
3grs	glutathione reductase	1.54	523
3tln	thermolysin	1.6	173
5cpa	carboxypeptidase A	1.54	315
See note 1	RTEM-1 β-lactamase	1.7	182

[1] Provided by Drs. Natalie Strynadka and Michael James, University of Alberta, Edmonton.

Table columns are PDB code, protein name, resolution of the crystallographic structure in Ångstroms, and the number of water molecules resolved in the structure, respectively.

Table 2: The physical and chemical features used to represent protein-bound water molecules and protein surface sites.

Tag	Feature	Description
ADN	Atomic density	The number of protein atom neighbors within 3.6 Å of the water molecule. This feature correlates with the local protein topography. Water molecules bound in deep grooves will have high ADN values, while those bound to protrusions will have low ADN values [36].
AHP	Atomic hydrophilicity	The hydrophilicity of the neighborhood of the water molecule is based on the frequency of hydration for each atom type in 56 high-resolution protein structures [42]. Each water molecule is assigned an AHP value equal to the sum of the atomic hydrophilicity values of all atom neighbors within 3.6 Å of the water molecule.
HBDP	Hydrogen bonds to protein	The number of hydrogen bonds between the water molecule and neighboring protein atoms. Each donor or acceptor atom within 3.5 Å is considered a potential hydrogen bond.
HBDW	Hydrogen bonds to water	The number of hydrogen bonds between the water molecule and other water molecules in the ligand-free protein structure, based on $\leq$ 3.5 Å distance between oxygen atoms in the two water molecules.
ABVAL	Average B-value of protein atom neighbors	The average (mean) temperature factor of all protein atoms within 3.6 Å of the water molecule.
NBVAL	Net B-value of protein atom neighbors	The sum of the B-values of all protein atoms within 3.6 Å of the water molecule.

crystallographically observed water molecule from the protein surface, $d_{\min}$, and the maximum distance, $d_{\max}$ were determined. For each surface point, a probe site was generated and placed at a distance along the surface normal, selected at random over the range of distances from $d_{\min}$ to $d_{\max}$. Any probes overlapping crystallographically observed water sites or other probes (with positions within 3.2 Å) were removed. Finally, the same number of probe sites were selected for each protein as there were crystallographic water sites. This selection was done so that the distribution of distances of probes from the protein surface matched the distribution of distances for observed water molecules for that protein.

Non-solvent sites were generated using this technique for each of the 30 protein structures in Table 1. Each probe site was characterized using the same physical and chemical that were used to characterize observed solvation sites (Table 2). For observed solvation sites, the temperature factor (B-value) and occupancy value from the crystallographic data can be used to estimate the thermal mobility of the water molecule in the context of the protein crystal. However, since temperature factor and occupancy are undefined for probe sites, two new features were computed to approximate the local thermal mobility of a probe site. ABVAL is the average (arithmetic mean) B-value of all protein atoms within 3.6 Å of the probe position. NBVAL is a non-normalized version of the same feature. That is, the simple sum of the B-values of all neighboring (within 3.6 Å) protein atoms.

In measuring the number of hydrogen bonds to other water molecules (HBDW) for probe sites, potential hydrogen bonds to neighboring probe sites were included in the calculation. In other words, probe sites and crystallographically observed water molecules were treated equally for purposes of feature value calculation. The final database consisted of the environments of 5325 crystallographically observed water molecules, and 5325 non-solvated sites.

Our primary goal for experimental research on protein-bound water molecules is to classify whether specific water molecules on the protein surface are conserved or displaced upon ligand binding. A secondary goal remains elucidation of the determinants of water conservation. Examining the final relative feature weights of the various features evolved during classifier optimization provides insight into these determinants. The inclusion of feature weights on the GA chromosome and the use of population-adaptive mutation for feature selection and extraction successfully yields combinations of features that provide improved distinction between conserved and displaced water molecules.

The left side of Table 3 presents the bootstrap results of the three best optimizations for the water conservation dataset. Over all GA runs, the average bootstrap accuracy is 62.74% with a standard deviation of 0.75%. The average balance between class accuracies averaged over all runs is 4.23% with 1.03% standard deviation. For comparison, the three best classifier results, in terms of both accuracy and balance, from the WEKA classification and machine learning package [44] are presented on the right side of the table. While the most accurate WEKA classifiers achieve slightly higher accuracy, they all exhibit a notable bias towards the conserved class. In contrast, the GA-trained cosine $k$nn classifier achieves similar accuracy without significant bias toward either class, using as few as 4 of the 8 available features. The utility of using a GA favoring class-balanced results during optimization is clear.

Table 3: Water conservation results, GA (left) and WEKA classifiers (right)

Bootstrap Accuracy (%)					Feature Weights, Offsets			Top 5 WEKA Classifiers by Accuracy				
Total	**Disp**	**Cons**	**Avg Bal**	**K**	**ADN**	**AHP**	**BVAL**	**Classifier**	**Total (%)**	**Disp**	**Cons**	**Bal**
65.29	66.57	64.00	4.10	48	-	-	.426, 4.70	NeuralNetwork	66.62	44.17	80.70	36.53
64.76	63.91	65.61	3.89	29	-	.096, 4.321	.281, 2.29	j48.J48	66.02	37.06	84.20	47.14
64.31	63.52	65.10	3.61	26	-	-	.207, 1.08	ADTree	65.97	44.27	79.59	35.32
			Feature Weights, Offsets					**Top 5 WEKA Classifiers by Balance**				
Total		**HBDP**	**NBVAL**		**HBDW**	**MOB**	**ABVAL**	**Classifier**	**Total (%)**	**Disp**	**Cons**	**Bal**
65.286		-	.101, 3.68		-	.406, -.51	.067, -12.94	IB1	61.35	48.62	69.34	20.72
64.762		-	.078, 2.63		.115, 4.85	.238, 2.88	.192, -13.67	KernelDensity	61.30	47.73	69.81	22.08
64.309		.193, -14.96	-		.281, -9.48	.227, -1.95	.093, -12.51	NaiveBayesSimple	64.06	49.93	72.92	22.99

The goals for experiments investigating the physical and chemical determinants of solvation sites on the protein surface are two-fold. The first goal is to train a classifier to accurately identify favored solvation sites given the properties of a protein surface at varying localities. The second goal is to determine the relative importance of the various chemical and physical factors governing solvation. Examination of the selected features and their evolved weights within a trained classifier leads to biological insights into the properties governing protein-water binding.

The left side of Table 4 presents the best three results obtained for the solvation dataset, while the right side presents the best three WEKA classifiers in terms of both classification accuracy and class balance. The best GA-trained classifier achieves a mean bootstrap accuracy of 69.91% using five of the six available features. In contrast, the best WEKA classifiers achieve similar though slightly lower accuracy than the best optimized cosine $k$nn classifier while maintaining a similar level of prediction balance. Over all runs, the GA obtains an average bootstrap accuracy of 68.21% with 0.70% standard deviation. The average balance is 3.74% with standard deviation of 1.65%. The main benefit of employing a hybrid GA/classifier system for the solvation dataset is the ability to elucidate the biological relevance of each feature through feature selection and extraction in order to form a more complete understanding of protein-water interactions.

Table 4: Water solvation results, GA (left) and WEKA classifiers (right)

Bootstrap Accuracy (%)					Weights, Offsets		Top 5 WEKA Classifiers by Accuracy				
Total	**non**	**site**	**Avg Bal**	**K**	**ADN**	**AHP**	**Classifier**	**Total(%)**	**Non**	**Site**	**Bal**
69.91	67.78	72.04	4.36	80	.252, -11.84	.177, -7.70	Logistic	69.33	65.50	73.16	7.67
69.48	65.79	73.16	7.38	67	.271, -8.42	.253, -15.00	NeuralNetwork	69.29	66.00	72.58	6.58
69.42	64.38	74.47	10.08	83	.242, -5.36	.222, 9.70	VotedPerceptron	69.25	66.75	71.74	4.98
		Feature Weights, Offsets					Top 5 WEKA Classifiers by Balance				
Total		**HBDP**	**HBDW**		**ABVAL**	**NBVAL**	**Classifier**	**Total(%)**	**Non**	**Site**	**Bal**
69.91		.352, -11.77	.105, -1.31		-	.113, 2.75	IB1	63.56	63.45	63.68	0.23
69.48		.154, -14.90	.113, -15.0		-	.209, -12.89	j48.J48	68.99	68.75	69.24	0.49
69.31		.205, -5.86	.165, 15.00		-	.166, 9.64	j48.PART	68.03	68.56	67.49	1.06

The features selected by the GA and their corresponding relative weights provide an opportunity to extract biologically relevant information from the optimized classifier systems. The most intriguing result arises from the features selected by the GA during the best run on the conserved water dataset. All four selected features (BVAL, MOB, ABVAL, and NBVAL) relate to the thermal mobility of the given water molecule or of the atoms surrounding it. In previous research, the classifier optimization techniques employed by the *Consolv* system often favored BVAL and MOB, but almost always removed ABVAL and NBVAL from classifier consideration. Other measures, such as AHP, HBDP, and HBDW, which relate to the atomic environment surrounding the water molecule, are more frequently considered. The GA-facilitated cosine classification method increases classifier accuracy over previously published results using variations on thermal mobility alone. This suggests that most of the information necessary to determine conservation of bound water molecules upon ligand binding can be extracted from the thermal mobility (and occupancy values) of the water molecule and its neighbors. While other features may be related to conservation, they are correlated with temperature factor in such a way that they bring no additional information to the classification problem.

For the solvation site dataset, the trained classifier consistently employs all measured features except ABVAL. The features ADN, AHP, and HBDP typically had higher weights than other features. Each of these features depends upon the amount and type of atoms neighboring the probe site. This suggests that the atomic environment of the site is more relevant than the thermal mobility of atoms surrounding the site in determining the favorability for solvation at the given site.

The process of fitting water molecules to electron density data can be imprecise and somewhat interpretive when density is smeared or blurred, when coordinated water molecules exhibit significant thermal mobility, and when the lattice structure of the protein crystal induces solvent binding sites that would not otherwise be favorable in a globular protein structure. Because solvent site prediction is based on training from and comparison with crystallographic structures, these and other factors make accurate prediction of water binding a difficult problem. Current algorithms for predicting the locations of bound water molecules can be divided into two classes. Theoretical approaches, such as GRID [45], use a potential energy function to evaluate the favorability of a probe site. For GRID, the potential energy function includes terms for Lennard-Jones interactions, electrostatic interactions, and a detailed evaluation of potential hydrogen bonds. For all theoretical approaches, the relative contribution of the terms in the potential energy function must be determined before solvation sites can be predicted.

In contrast, empirical methods determine the favorability of a solvation site by analogy with known sites. A site is evaluated and compared to a database of known solvation sites and non-solvated sites, and predicted as being more similar to one than the other. A set of features to observe and compare between solvation sites and non-solvated sites must be selected prior to classification. Several empirical methods for prediction of protein-bound water molecule locations have been developed. AQUARIUS2 [46] uses a knowledge base of the distributions of water molecules around polar atoms at the protein surface. A "likelihood" value is assigned to a putative water molecule location based on its geometric relationship to nearby polar protein atoms. If the site lies in a region highly occupied by water molecules in the knowledge base and has significant electron density, as determined by X-ray crystallography, it receives a higher score. The highest scoring positions in a 3-dimensional matrix surrounding the protein are selected as likely water molecule locations.

AUTO-SOL [47] predicts water sites based on the directionality of hydrogen bonds. A database of small-molecule crystal structures was analyzed to find the distribution of hydrogen-bond lengths and directions, and possible solvent sites are evaluated by AUTO-SOL according to how well their hydrogen-bond geometry matches this database. Current methods can reproduce ~70% of crystallographically observed solvent molecules within 1.5 Å of the experimental locations [47]. There remains, however, a tendency for many current methods to produce false positives, predicting solvation sites where none are observed in the crystal structure.

The EC-cosine *k*nn classifier obtains similar accuracy to these methods, while maintaining tight balance between the number of false-positives and false-negatives.

4 Conclusion

A key advantage of the GA-optimized cosine method is the ability to optimize the classifier's point of reference in feature space in addition to simple feature weight enhancement. Evolution of both weights and offsets provides the GA an opportunity for classifier enhancement that cannot be leveraged for either Euclidean *k*nn optimization or Bayes discriminant function optimization. When compared to WEKA classifiers, results indicate that the hybrid GA/cosine classifier compares favorably with other tested methods in terms of simultaneously increasing classification accuracy while maintaining class balance and reducing the feature set size.

While the aggregate bootstrap accuracy of the optimized classifiers over all runs may not seem overly impressive, one must remember that because GAs are stochastic optimization processes, multiple GA experiments are required to search a sufficiently large portion of the potential solution space. It is highly probable that a few experiments will converge to more favorable areas of the solution space than others. This reasoning justifies reporting the best results in addition to the average results over all GA runs.

Various EC-hybrid classifiers have been demonstrated to be effective techniques for identifying the physical and chemical determinants of protein-water binding [48]. With its combined classification and feature selection capability, the EC/cosine $k$nn classifier proves to be an effective tool for mining biologically relevant information for this and other large structural biology data sets. While maintaining a balanced accuracy competitive with contemporary classification techniques, the feature selection and extraction capabilities of the EC/classifier system described here provides the ability to provide insight into the relative importance of various features provided for a given classification problem. This property enables the GA/cosine classifier system to be employed in situations requiring knowledge discovery in addition to traditional pattern recognition, where traditional techniques that do not maintain feature independence would not be well-suited.

References

[1] G. V. Trunk, "A problem of dimensionality: A simple example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 306–307, 1979.

[2] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition in practice," in *Handbook of Statistics*, P. R. Krishnaiah and L. N. Kanal, Eds., vol. 2. North-Holland, 1982, pp. 835–855.

[3] H. Liu and H. Motodata, *Feature Selection for Knowledge Discovery and Data Mining.* Boston, MA: Kulwer Academic Publishers, 1998, pp. 73–95.

[4] T. M. Cover and J. M. V. Campenhout, "On the possible orderings in the measurement selection problem," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 7, pp. 657–661, 1977.

[5] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Transactions on Computers*, vol. C-26, pp. 917–922, 1977.

[6] A. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. 20, pp. 1100–1103, 1971.

[7] P. Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1,119–1,125, Nov. 1994.

[8] A. K. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, February 1997.

[9] J. Mao, K. Mohiuddin, and A. K. Jain, "Parsimonious network design and feature selection through node pruning," in *Proc. of the Intl. Conf. on Pattern Recognition*, Jerusalem, October 1994, pp. 622–624.

[10] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Adaptive fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 238–250, August 1996.

[11] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 260–270, August 1995.

[12] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[13] ——, "Simplifying decision trees," *International Journal of Man-Machine Studies*, pp. 221–234, 1987.

[14] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[15] J. D. Kelly and L. Davis, "Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm," in *Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications*, 1991, pp. 377–383.

[16] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution.* NY: John Wiley, 1966.

[17] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Stuttgart: Fromman-Holzboog, 1973.

[18] H.-P. Schwefel, *Numerische Optimierung von Computermodellen mittels der Evolutionsstrategie.* Basel: Birkhäuser, 1977.

[19] J. R. Koza, *Genetic Programming: On the Programming of a Computer by Means of Natural Selection.* MIT Press, 1992.

[20] J. H. Holland, *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press, 1975.

[21] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* San Mateo, CA: Addison-Wesley, 1989.

[22] M. Mitchell, *An Introduction to Genetic Algorithms.* MIT Press, 1996.

[23] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters*, vol. 10, pp. 335–347, 1989.

[24] W. F. Punch, E. D. Goodman, M. Pei, L. Chia-Shun, P. Hovland, and R. Enbody, "Further research on feature selection and classification using genetic algorithms," in *Proc. International Conference on Genetic Algorithms 93*, 1993, pp. 557–564.

[25] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, Eds. Norwell, MA: Kluwer Academic Publishers, 1998, ch. 8, pp. 117–136.

[26] J. Yang, R. Parekh, and V. Honavar, "DistAl: an inter-pattern distance-based constructive learning algorithm." in *Proceedings of the International Joint Conference on Neural Networks*, Anchorage, Alaska, 1998.

[27] H. Vafaie and K. De Jong, "Evolutionary feature space transformation," in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, Eds. Norwell, MA: Kluwer Academic Publishers, 1998, ch. 19, pp. 307–323.

[28] J. R. Quinlan, "The effect of noise on concept learning," in *Machine Learning: an Artificial Intelligence Approach*, R. Michalski, J. Carbonnell, and T. Mitchell, Eds. Morgan Kaufmann, 1986, pp. 149–166.

[29] M. R. Peterson, T. E. Doom, and M. L. Raymer, "Ga-facilitated knowledge discovery and pattern recognition optimization applied to the biochemistry of protein solvation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2004.

[30] M. L. Raymer, P. C. Sanschagrin, W. F. Punch, S. Venkataraman, E. D. Goodman, and L. A. Kuhn, "Predicting conserved water-mediated and polar ligand interactions in proteins using a k-nearest-neighbors genetic algorithm," *J. Mol. Biol.*, vol. 265, pp. 445–464, 1997.

[31] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computing*, vol. 4, no. 2, pp. 164–171, 2000.

[32] M. L. Raymer, T. E. Doom, and L. A. K. W. F. Punch, "Knowledge discovery in medical and biological datasets using a hybrid bayes classifier/evolutionary algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, vol. B*, vol. 33, no. 5, pp. 802–813, 2003.

[33] C. S. Poornima and P. M. Dean, "Hydration in drug design. 1. Multiple hydrogen-bonding features of water molecules in mediating protein-ligand interactions," *Journal of Computer-Aided Molecular Design*, vol. 9, pp. 500–512, 1995.

[34] M. Kuramochi and G. Karypis, "Gene classification using expression profiles: a feasibility study," in *Proceedings of the Second Annual IEEE International Symposium on Bioinformatics and Bioengineering*, 2001, pp. 191–200.

[35] A. K. Jain, R. C. Dubes, and C. C. Chen, "Bootstrap techniques for error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 628–633, Sept. 1987.

[36] L. A. Kuhn, M. A. Siani, M. E. Pique, C. L. Fisher, E. D. Getzoff, and J. A. Tainer, "The interdependence of protein surface topography and bound water molecules revealed by surface accessibility and fractal density measures," *J. Mol. Biol.*, vol. 228, pp. 13–22, 1992.

[37] C. S. Poornima and P. M. Dean, "Hydration in drug design. 2. Influence of local site surface shape on water binding," *Journal of Computer-Aided Molecular Design*, vol. 9, pp. 513–520, 1995.

[38] E. N. Baker and R. E. Hubbard, "Hydrogen bonding in globular proteins," *Prog. Biophys. Mol. Biol.*, vol. 44, pp. 97–179, 1984.

[39] E. E. Abola, F. C. Bernstein, S. H. Bryant, T. F. Koetzle, and J. Weng, *Protein Data Bank*. Bonn/Cambridge/Chester: Data Commission of the International Union of Crystallography, 1987, pp. 107–132.

[40] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer, Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, "The Protein Data Bank: A computer-based archival file for macromolecular structures," *J. Mol. Biol.*, vol. 112, pp. 535–542, 1977.

[41] U. Hobohm, M. Scharf, R. Schneider, and C. Sander, "Selection of representative protein data sets," *Protein Sci.*, vol. 1, pp. 409–417, 1992.

[42] L. A. Kuhn, C. A. Swanson, M. E. Pique, J. A. Tainer, and E. D. Getzoff, "Atomic and residue hydrophilicity in the context of folded protein structures," *Proteins: Str. Funct. Genet.*, vol. 23, pp. 536–547, 1995.

[43] M. L. Connolly, "Solvent-accessible surfaces of proteins and nucleic acids," *Science*, vol. 221, pp. 709–713, 1983.

[44] I. H. Witten and E. Frank, *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations.* San Francisco, GA: Morgan Kaufmann, 2000, pp. 265–319.

[45] R. C. Wade, K. J. Clark, and P. J. Goodford, "Further developments of hydrogen bond functions for use in determining energetically favorable binding sites on molecules of known structure," *J. Med. Chem.*, vol. 36, pp. 140–147, 1993.

[46] W. R. Pitt, J. Murray-Rust, and J. M. Goodfellow, "AQUARIUS2: Knowledge-based modeling of solvent sites around proteins," *J. Comp. Chem.*, vol. 14, no. 9, pp. 1007–1018, 1993.

[47] A. Vedani and D. W. Huhta, "An algorithm for the systematic solvation of proteins based on the directionality of hydrogen bonds," *J. Am. Chem. Soc.*, vol. 113, pp. 5860–5862, 1991.

[48] M. L. Raymer, D. Holstius, and L. A. Kuhn, "Identifying the determinants of favorable solvation sites," *Protein Engng.*, 2001, *submitted.*