QUALITY ASSESSMENT OF LOW FREE-ENERGY PROTEIN STRUCTURE PREDICTIONS

*Luca Cazzanti, Maya Gupta**

University of Washington
Department of Electrical Engineering
Seattle, WA 98105

Lars Malmström, David Baker

University of Washington
Department of Biochemistry
Seattle, WA 98105

ABSTRACT

Analyzing and engineering cellular signaling processes requires accurate estimation of cellular subprocesses such as protein-folding. We apply parametric and nonparametric classification to the problem of assessing three-dimensional protein domain structure predictions generated by the Rosetta ab initio structure prediction method. The assessment is based on whether the predicted structure is similar enough to a known protein structure to be classified as being in the same protein superfamily. We develop appropriate features and apply Gaussian mixture models, K-nearest-neighbors, and the recently developed linear interpolation with maximum entropy method (LIME). The proposed learning methods outperform a previous quality assessment method based on generalized linear models. Results show that the proposed methods reject the vast majority of poor structural predictions while identifying a useful number of good predictions.

1. INTRODUCTION

Analyzing and designing cellular signal processing systems requires the ability to accurately estimate cellular processes such as the folding of proteins, protein structure, and protein interactions. Machine learning has been previously applied to the problem of predicting protein structure from a given DNA sequence [1]. However, machine learning is the most effective solution only when practical causative models are lacking. In the case of protein structural predictions, molecular biochemical models make it possible to model and calculate the free-energy of protein structures, and it is known that what makes natural protein structures stable is in part their low free-energy. Rosetta [2] is one such *ab initio* method that searches for low free-energy possible foldings of protein domains. Biannually, a blind protein structure prediction competition is held called the Critical Assessment of Structure Predictions (CASP). Rosetta has consistently performed at the top in the last three CASP events [3, 4, 5]. However, it is not enough to find a molecular configuration that has low free-energy. The Rosetta process outputs several thousand low free-energy configuration for each test genetic sequence, and then these configurations are examined and analyzed by human experts to select final predictions.

In this paper we discuss recent efforts to automatically assess the quality of the low free-energy protein predictions using statistical learning approaches. A Gaussian mixture model approach (GMM) is trained and compared to two nonparametric neighborhood learning methods, K-NN and the more recent linear interpolation with maximum entropy (LIME) neighborhood method. In Section 2 we discuss particularities of the statistical learning problem. The features used are briefly described in Section 3. The chosen learning methods are reviewed and motivated in Section 4, The results presented in Section 5 are a strong improvement over previously published results, and a step forward in automating protein-structure prediction.

2. UNDERSTANDING THE STATISTICAL LEARNING PROBLEM

We assess the quality of each Rosetta low free-energy protein structure prediction by evaluating its similarity to known structures in the Protein Data Bank (PDB). Biochemically, the rationale is that if a prediction is structurally similar to a known protein then it is more likely to occur naturally, and the prediction is not thrown-out. We compare each Rosetta prediction to known protein domain structures in the PDB. We consider four features that describe how similar a Rosetta prediction is to each of the PDB structures. Using these features and a training set of known structures and Rosetta prediction/PDB structure pairs, one predicts whether the Rosetta prediction and the PDB structure are structurally similar enough to be considered in the same superfamily. A superfamily includes protein domains that share significant structural similarities and an evolu-

*The authors would like to thank the Armed Forces Communications and Electronics Association and the National Science Foundation for supporting this work. Luca Cazzanti is also with the Applied Physics Laboratory, University of Washington.

tionary link, but do not necessarily have similar genetic sequences [6]. The superfamily is determined from the SCOP (Structural Comparison of Proteins) database, a manually-curated taxonomy of domain structures. Thus estimating whether a Rosetta prediction is close enough to a known protein to be in the same superfamily is used as a proxy for deciding whether that Rosetta prediction is a reasonable structure that would occur naturally in cellular processes.

The predictions deemed accurate are kept for further consideration by human experts, while all other predictions are discarded. Since there is no ground truth in the SCOP database for newly predicted structures, we adopt the strategy in previous work [2] and use test sequences from the PDB with known structure that can be used as ground truth to determine prediction success. We hypothesize that an approach that performs well on structural predictions obtained from known sequences will generalize to predictions for newly discovered sequences with yet-undetermined structures.

A parametric and two nonparametric neighborhood learning approaches were chosen, trained, and compared. The chosen approaches, a Gaussian mixture model, K-NN, and LIME, can all be used to form an estimate of the probability that a Rosetta prediction and a PDB structure are of the same SCOP superfamily or from different superfamilies. Estimating the probability as a precursor to classification was a requirement in order to have maximum flexibility to change the misclassification costs used to make final classification decisions. Further, the biochemistry experts preferred to specify the costs in terms of a limit (Neyman-Pearson criterion) on the percent of false positives. The goal is to reduce the set of Rosetta predictions, from a huge set of mostly unrealistic predictions, to a small set of predictions that are good in the sense that they are structurally similar to a known protein structure. If in this process a few good predictions are lost, that is not a large concern. The focus is on removing from the set the bad predictions. Ideally, one would like an automatic system that reduced the set to only predictions that were guaranteed to be good and could thus be trusted and effectively used in multi-stage cellular modeling and design. Thus the cost structure for the learning problem is highly asymmetric.

The dataset is also necessarily highly asymmetric. The dataset was developed by Lars Malmström at the Baker Laboratory of the University of Washington and is not yet public [7]. There were 185,944 Rosetta predictions for 1,005 protein domain structures. Of these, 4,462 were good predictions in the sense that the true protein corresponding to the genetic sequence that generated the Rosetta prediction belonged to the same superfamily as the PDB structure to which it was compared. These predictions are classified as class g_s. The other 181,482 Rosetta predictions in the training set do not correspond to known PDB structures in the same superfamily as the true protein corresponding to the genetic sequence that generated the Rosetta prediction. These predictions are classified as class g_d. Figure 1 shows an example of a folded protein domain for which Rosetta produces a very good structural prediction.

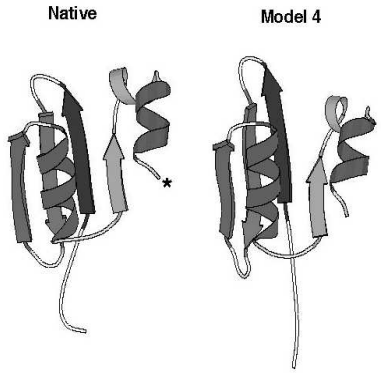

Fig. 1. Schematic visualization of the folded protein domain T116-MutS (Domain 2 - 128-198) in CASP. Note the helical structures and the sheets (denoted by arrows). The diagrams represent a true (native) structure and its Rosetta prediction (Model 4) obtained from its sequence of 68 residues. The Rosetta prediction agrees very well with the native structure [8].

3. MEASURING HOW CLOSE A PREDICTED STRUCTURE IS TO A NATURALLY OCCURRING STRUCTURE

Based on molecular biology theory and data analysis, four features were used as indicators of structural similarity between a Rosetta prediction and a PDB structure. Based on these features, the statistical learning approaches discussed in this paper assess the quality of each Rosetta prediction and produce a likelihood that the Rosetta prediction and the PDB structure to which it is compared are in the same SCOP superfamily.

The first feature is the Mammoth z-score. Mammoth is a sequence-independent structure-to-structure protein comparison algorithm which is widely used in protein structure studies [9]. The three-dimensional models of two proteins are first aligned by matching corresponding atoms in their three-dimensional carbon backbone traces. The Mammoth z-score is then computed based on the the root-mean-squared deviation (RMSD) of the distances between the corresponding elements in the two structures. Higher scores indicate a higher degree of similarity between the two protein models. Based only on the Mammoth z-score, there is still a large overlap between the pairs of proteins that are in the same SCOP superfamily and those that are not, indicating

the need for additional features to discriminate.

Much of the structure of a protein can be described as α-helices or β-sheets, and these descriptions are useful indicators of protein family membership [10]. A simple method for quantifying the contributions of the α-helices and β-sheets considers the percentage of number of amino acid residues in helices and sheets with respect to the total number of residues in the protein domain. Let (α_p, β_p) and (α_m, β_m) be the percentages of α-helices and β-sheets in the Rosetta prediction and its PDB match, respectively. Then, $\alpha_p - \alpha_m$ and $\beta_p - \beta_m$ are two of the four features used.

A notion of length of a protein is the number of residues that it contains (a residue is an amino acid sidechain plus the peptide backbone). Our data shows that predictions and matches with approximately the same residue lengths are more likely to be in the same SCOP superfamily than predictions and matches for which the lengths significantly differ. As a feature, we use the ratio of the length of the Rosetta prediction to the length of its PDB match.

4. REVIEW OF STATISTICAL LEARNING METHODS APPLIED

The requirements for a learning method for this application are that it produce probability estimates that could be used with flexible misclassification costs to determine a classification, that the learning method be robust to highly asymmetric costs, and that the learning method be robust to highly asymmetrically distributed classes of data. We chose to compare a robust parametric approach, GMMs, to a recent nonparametric neighborhood approach, LIME, and the long-standing simple K-NN. Due to the highly asymmetric class distribution, the need for flexible costs, and the known strong overlap of the classes in feature space, these approaches seemed better suited exploratory algorithms for this application than empirical risk minimization techniques such as neural nets or large-margin classifiers. However, in future work we hope to compare with a selection of such methods.

4.1. Gaussian mixture models

Gaussian mixture models (GMMs) are a robust but flexible parametric approach for solving classification problems. They have been successful in a variety of different statistical learning applications. GMMs form smooth approximations of arbitrary distributions by weighted sums of Gaussian functions, and are thus well-suited to model the overlapping feature space distribution of the two classes of interest. The approximation of the class-conditional probability of feature vector x is the GMM associated with g_s or g_d, and is written as a linear combination of the conditional Gaussian components,

$$\hat{P}(X = x|Y) = \sum_{k=1}^{K} w_k f(x|\mu_k, \Sigma_k), \qquad (1)$$

where $\mathcal{M}_K = \{w_k, \mu_k, \Sigma_k\}$, $k = \{1 \dots K\}$ denotes collectively the parameters of the GMM with K components and $\mathcal{M}_K$ depends on the class label $Y \in \{g_s, g_d\}$. The component weights satisfy the constraints $\sum_{k=1}^{K} w_k = 1$, $w_k \geq 0$.

Each component $f(x|\mu_k, \Sigma_k)$ of a mixture is a multivariate Gaussian function with mean vector μ_k and covariance matrix Σ_k, in a d-dimensional feature space

$$f(x|\mu_k, \Sigma_k)$$
$$= \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\}.$$

For this application there are four features as described in Section 3, so $d = 4$.

Using Bayes' rule, the a posteriori class probabilities $P(Y|X)$ can be expressed in terms of the estimated joint probabilities

$$\hat{P}(X = x, Y = g) = \hat{P}(X|Y = g)P(Y = g),$$

where $P(Y = g)$ is the a priori probability of randomly selecting class label g,and the decision rule to classify a Rosetta prediction in g_s or g_d takes the form

$$\frac{\hat{P}(X = x, Y = g_s)}{\hat{P}(X = x, Y = g_s) + \hat{P}(X = x, Y = g_d)} > t, \qquad (2)$$

where the threshold t depends on the misclassification costs.

The structure of the covariance matrices and the number of components K in each class are important design choices. In (1) one may assume a distinct covariance for each of the Gaussian components, or one covariance matrix shared by all components. Covariance matrices may be diagonal or full. Like previous investigators [11], we hypothesize that the number of training samples and feature dimensions will allow robust but sufficiently flexible modeling with distinct diagonal covariance matrices for the Gaussians in each mixture.

The choice of model order K is critical to successful classification. There are no general theoretical results to guide the choice, and one must determine an appropriate value empirically. In (1), the likelihood of the data given the model will always increase as K is increased. However, simply increasing the number of mixtures in each model does not increase the rate of correct classification, as the larger and larger models overfit the training data and generalize poorly to test data. To avoid overfitting, we use the Bayesian Information Criterion (BIC) [12] to separately select the order of the models for g_s and g_d. The BIC is a

penalized likelihood, where the penalty term is proportional to the number of model parameters. Denote by $|\mathcal{M}_K|$ the total number of parameters in the GMM for class Y. Then the BIC is expressed as

$$BIC_K = 2 \log \hat{P}(X|Y) - |\mathcal{M}_K| \log |\mathcal{T}|, \quad (3)$$

where $|\mathcal{T}|$ is the number of training data points used to estimate $\mathcal{M}_K$. For each class-conditional GMM, models are trained over a range of K, and the BIC is evaluated to determine a BIC-optimal value of K.

For a given value of K, the parameters w_k, μ_k, and Σ_k for each of the two Gaussian mixtures are learned from training data (X_i, Y_i) using the maximum likelihood (ML) method and the well-known expectation-maximization (EM) algorithm [12]. The EM algorithm iteratively estimates the model parameters and maximizes a function of the likelihood of the data, based on the parameters estimated at the previous iteration. To obtain good ML estimates of the model parameters with the EM algorithm, it is important to properly initialize the parameters prior to the first iteration. Like model order selection, there are no generally applicable theoretical results to help us. In this work the final model is obtained by averaging R randomly-initialized models:

$$\hat{P}(X = x|Y) = \frac{1}{R} \sum_{r=1}^{R} \hat{P}(X = x|Y) \quad (4)$$

This approach is robust and generalizes well to test data. In this work $R = 5$.

4.2. Nearest-Neighbor

Nonparametric neighborhood methods are known to perform well in practice, are intuitive, and can achieve optimal error rates asymptotically [12]. Nonparametric neighborhood classifiers weight the training samples in a neighborhood around the test point x, and then classify x as the class with minimum weighted expected cost. Various neighborhood definitions can be used with these methods; in this work we use the standard definition of the K nearest neighbors, measured by Euclidean distance. For each test point x, let the sample pairs be re-indexed by their distance to x, so that X_k is the kth nearest-neighbor to x. Given a neighborhood, a weighted nearest-neighbor classifier assigns a weight w_k to each neighbor, usually with the constraints $\sum_{k=1}^{K} w_k = 1$ and $w_k \geq 0$. From the weights and the neighborhood sample pairs, it is standard to form a maximum likelihood estimate of the probability of each class,

$$\hat{P}(Y = g|x) = \sum_{k=1}^{K} w_k I_{(Y_k = g)} \quad (5)$$

for $g \in \{g_s, g_d\}$ and where $I_{(\cdot)}$ is an indicator function that is one when its argument is true and zero otherwise. The classification decision rule is then simply

$$\hat{P}(Y = g|x) > t, \quad (6)$$

where the threshold t depends on the costs [12].

The neighborhood size K and the weight vector w are important parameters to achieve low classification error. As for the case of model order selection for GMMs, there is no theoretical rule to help decide the best value of K, and usually it is determined by cross-validation [12]. Many weighting kernels have been proposed in the literature [12, 13]. Most kernels are symmetric and decay the weight with distance from the test point. In this work, we consider two weighting schemes for weighted nearest-neighbors: K-NN and LIME [14, 15]. For K-NN, the weights are uniform, $w_k = 1/K$. For LIME, the weights solve

$$\arg\min_{w} \left(\| \sum_{k=1}^{K} w_k X_k - x \|_2^2 - \lambda H(w) \right) \quad (7)$$

where $H(w) = -\sum_k w_k \log w_k$ (the Shannon entropy).

The LIME objective (7) trades-off between two goals: satisfying the linear interpolation equations, and maximizing the entropy of the weights. If the only goal were to maximize the entropy of the weights, the weights would all be equal, and solving the LIME objective would result in the K-NN classifier. In general, maximizing the entropy of the weights forces the weights to be as uniform as possible [16]. The parameter λ controls the trade off between maximizing the entropy and solving the linear interpolation equations. For a test point x, the linear interpolation equations require that the weights on the neighbors have x as their center of mass, that is, so $\sum_k w_k X_k = x$. These linear interpolation equations are not always solvable, and thus the LIME objective is to minimize the squared l_2 error between $\sum_k w_k X_k$ and x. Jointly determining the weights in this way is helpful particularly when the distribution of neighbors is asymmetric. If two neighbors are too similar, they are each less informative, and they each get less weight due to the linear interpolation equations. If a neighbor occupies a region of the neighborhood that is more sparse, then the linear interpolation equations ensure that this neighbor receives relatively more weight. We conjecture that maximizing the entropy of the weights helps keep the estimation variance down, while solving the linear interpolation equations helps reduce the estimation bias. The parameters K and λ were determined using leave-one-out cross-validation.

5. RESULTS

For the GMM approach to prediction assessment, we used the BIC to determine that the best model for g_s contains $K = 10$ mixtures, and the best model for g_d contains $K = 70$ mixtures. The performance of the GMMs is computed

using 10-way cross-validation after determining the number of components in each mixture [1]. For K-NN and LIME, we determined the neighborhood size K and the parameter λ with leave-one-out cross-validation, ranging over several values of the parameters. We varied K from 1 to 50, and tested $\lambda = \{10^{-6}, 10^{-3}, 0.01, 0.05, 0.1, 0.2, \ldots 1\}$. No single value of K performed best for all the costs we consider; however, we determined that the most useful range for λ is $\{0.01 \ldots 0.1\}$.

Figure 2 shows FP and FN error curves for GMM, K-NN and LIME, as a function of the classification threshold t. It is straightforward to determine the threshold t and the FN error rate based on a given upper limit for the FP error. From the graphs, one simply finds the smallest value $\hat{t}$ that guarantees that the FP error does not exceed the upper limit. The intersection of the vertical line at $t = \hat{t}$ with the FN error curve determines the corresponding FN error rate. Thus, the FP error limit, the threshold $\hat{t}$, and the corresponding FN error rate completely describe the performance of each classifier.

Table 1 shows the classification errors for the GMM, K-NN and LIME classifiers. We considered two different operating conditions of interest to the Rosetta experts, corresponding to the Neyman-Pearson FP upper limits of 5% and 0.5%. The LIME classifier yielded lower FN error rates than the GMM or K-NN. Due to the large size of the data set, the small differences in the percentages in the table correspond to a sizable difference in the number of predicted structures correctly classified.

For the 5% rejection rate, the near-neighbor classifiers are on the edge of the training parameters ($K = 50$) and training with larger neighborhoods is needed to verify that the reported performance is the optimal.

Table 1. Classification error rates for GMM, K-NN, and LIME.

GMM	% FP	% FN
	5.00	45.41
	0.50	81.67
K-NN	% FP	% FN
$K = 50$	5.00	45.67
$K = 28$	0.50	82.16
LIME	% FP	% FN
$K = 50, \lambda = 0.1$	5.00	44.53
$K = 40, \lambda = 0.1$	0.50	81.26

The presented results compare favorably with the only previously published assessment method for Rosetta protein structure predictions [2]. In the previous assessment, a generalized linear model (GLM) [12] estimates the probability that a prediction and its PDB match belong to the same SCOP superfamily, based on four features: the Mammoth z-score, the ratio of amino acid sequence lengths, the length of the prediction, and a measure of whether the Monte Carlo

[1] We used a subset of the LNKnet software [17], adapted for Matlab by Dr. Jack McLaughlin, of the Applied Physics Laboratory, University of Washington.

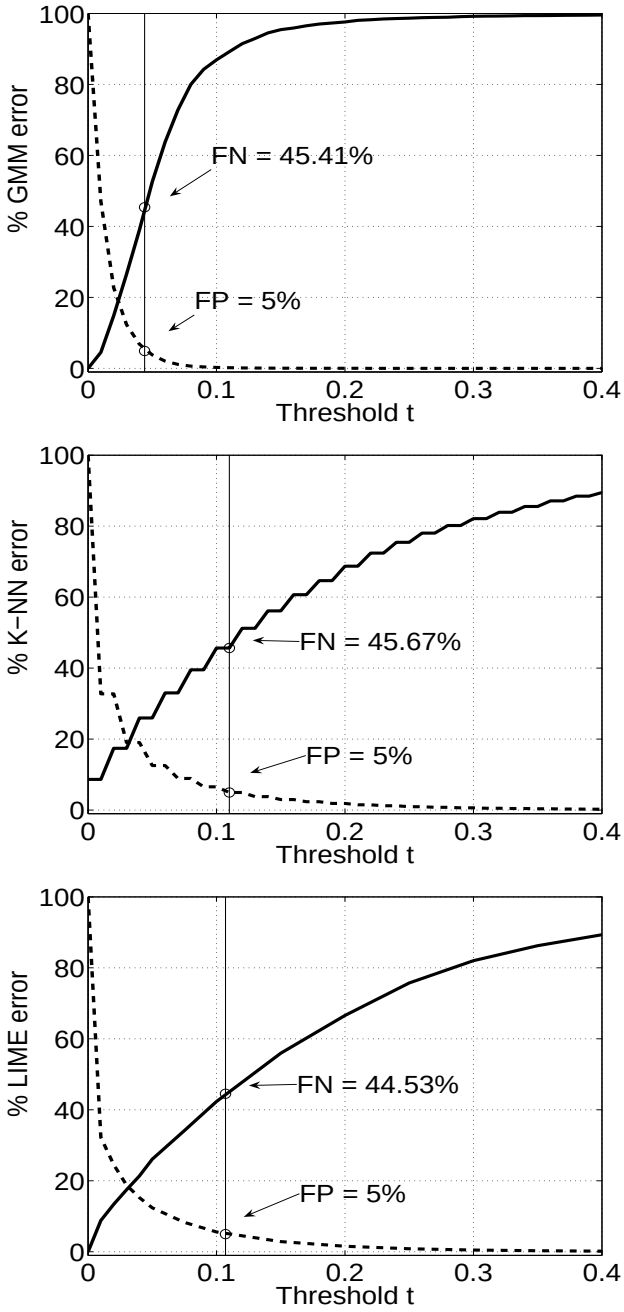

Fig. 2. FN (solid) and FP (dashed) error rates as a function of the threshold t for top: GMM, middle: K-NN, and bottom: LIME. For the FP upper limit of 5%, the corresponding FN error rates are indicated. For K-NN, K = 50; for LIME, $K = 50$ and λ = 0.01. These results are also listed in Table 1.

search at the core of Rosetta has converged to a viable prediction. The FN error for this GLM ranges from 75.35% to 97.24% for the range of operating conditions we consider. All the methods evaluated here outperformed the GLM.

6. CONCLUSION AND FUTURE WORK

Proven and new statistical learning methods were applied to the automatic assessment of the quality of computational protein domain structural predictions obtained with Rosetta. The presented classifiers performed better than a previously-published assessment method based on a GLM. The proposed features, GMM and LIME method will be incorporated into the Rosetta toolbox to help automate protein structure prediction. Further, the proposed methods are helpful in estimating protein function. Further work to improve the automated learning is planned, including the investigation of hybrid classification methods, comparisons to empirical risk minimization techniques, and improvements in the choice of neighborhood used with LIME.

7. REFERENCES

[1] P. Baldi and S. Brunak, *Bioinformatics: the machine learning approach*, MIT Press, Boston, MA, 2001.

[2] R. Bonneau, C. E. M. Strauss, C. A. Rohl, D. Chivian, P. Bradley, L. Malmström, T. Robertson, and D. Baker, "De novo prediction of three-dimensional structures for major protein families," *Journal of Molecular Biology*, vol. 322, no. 1, pp. 65–78, 2002.

[3] K. T. Simons, R. Bonneau, I. Ruczinski, and D. Baker, "Ab initio protein structure prediction of CASP III targets using ROSETTA," *Proteins: Struct. Funct. Genet.*, vol. 37, no. S3, pp. 171–176, 1999.

[4] R. Bonneau, J. Tsai, I. Ruczinski C. Rohl, C. E. M. Strauss, and D. Baker, "Rosetta in CASP4: Progress in ab inition protein structure prediction," *Proteins: Struct. Funct. Genet.*, vol. 45, no. S5, pp. 119–126, 2001.

[5] P. Bradley, D. Chivian, J. Meiler, K. M. Misura, C. Rohl, W. Schief, J. W. Wedemeyer, O. Schueler-Furman, P. Murphy, J. Schonbrun, C. Strauss, and D. Baker, "Rosetta predictions in CASP5: Successes, failures, and prospects for complete automation," *Proteins*, vol. 53, no. S5, pp. 457–468, 2003.

[6] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, vol. 247, pp. 563–540, 1995.

[7] L. Malmström, "Unpublished collected data," The Baker Laboratory, Department of Biochemistry, University of Washington, 2005.

[8] The Baker Laboratory, "Using Rosetta for ab initio structure prediction in the fourth community wide experiment on the critical assessment of techniques for protein structure prediction (CASP4)," `http://depts.washington.edu/bakerpg/casp4/casp4.html`.

[9] A. R. Ortiz, C. E. M. Strauss, and O. Olmea, "MAMMOTH (matching molecular models obtained from theory): An automated method for model comparison," *Protein Science*, vol. 1, pp. 2606–2621, 2002.

[10] A. Lesk, *Introduction to Protein Architecture*, Oxford University Press, London, 2001.

[11] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker-identification using Gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 1, 1995.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer-Verlag, New York, 2001.

[13] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer-Verlag Inc., New York, 1996.

[14] M. R. Gupta and R. M. Gray, "Reducing bias in supervised learning," *Proceedings of the IEEE Workshop on Statistical Signal Processing*, pp. 482–485, 2003.

[15] M. R. Gupta, R. M. Gray, and R. A. Olshen, "Nonparametric supervised learning by linear interpolation with maximum entropy," *Submitted for journal publication*, 2004.

[16] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, United States of America, 1991.

[17] "LNKnet Pattern Classification Software," `http://www.ll.mit.edu/IST/lnknet/index.html`.