

Aplicação de Multi-classificadores no Reconhecimento de Classes Estruturais de Proteínas

José Alfredo F. Costa¹, Valnaide G. Bittencourt², Marcílio C. P. de Souto³

Departamento de Engenharia Elétrica¹

Departamento de Computação e Automação²

Departamento de Informática e Matemática Aplicada³

Universidade Federal do Rio Grande do Norte - 59.072-970 - Natal - RN

E-mail: alfredo@dee.ufrn.br, valnaide@dca.ufrn.br, marcilio@dimap.ufrn.br

Resumo - A predição de classes estruturais de proteínas é um dos principais problemas em aberto da biologia molecular e uma importante abordagem para a descoberta de estruturas de proteínas desconsiderando a similaridade de suas seqüências. Este trabalho apresenta uma comparação de técnicas individuais de aprendizagem de máquina (Árvores de Decisão, K-Vizinhos Mais Próximos, *Naive Bayes*, Máquinas de Vetores Suporte e Redes Neurais) e de sistemas multi-classificadores homogêneos (*Bagging* e *Boosting*) e heterogêneos (*Stacking*, *StackingC* e *Vote*) aplicados à tarefa da predição de classes estruturais de proteína.

1. Introdução

Atualmente, com a finalização do seqüenciamento do genoma humano e de diversos outros organismos, deu-se início a uma nova fase para as pesquisas genéticas, denominada Proteômica [9]. Este termo envolve a identificação de todas as proteínas expressas pelo genoma bem como a determinação de suas funções fisiológicas e patológicas. Tal conhecimento é essencial para o desenvolvimento de novos medicamentos e métodos de diagnóstico, por exemplo.

Neste trabalho, avaliamos a aplicação de diferentes técnicas de Aprendizagem de Máquina (AM) [1] [5] [6] [17], individuais e de multi-classificação, ao problema de determinar a similaridade estrutural de proteínas sem similaridades de seqüência. Mais especificamente, consideramos o problema de reconhecimento de classes estruturais de dobras de proteína. As proteínas são ditas terem uma dobra comum, que é um padrão tridimensional, se tiverem a mesma estrutura secundária¹ principal no mesmo arranjo e com a mesma topologia, apresentando ou não uma mesma origem evolucionária [4].

Em nossa análise, como em [5], presumimos que o número de dobras é restrito. Conseqüentemente, o foco está em predições estruturais no contexto de uma classificação particular de dobras 3-D. Há diversas bases de dados de classificação, tais como

Structure Classification of Protein (SCOP) [11] e Class, Architecture, Topology, and Homologous superfamily (CATH) [13]. A base de dados SCOP (usada neste trabalho), por exemplo, é dividida em quatro níveis hierárquicos: classe, dobra, superfamília e família. Neste trabalho, concentramos nossos estudos em nível de classe, em que as proteínas podem ser rotuladas em uma das seguintes principais classes estruturais: all- α , all- β , α/β , $\alpha+\beta$ ou *small*.

O restante deste trabalho é organizado da seguinte forma: a seção 2 mostra as técnicas de AM aplicadas, a base de dados utilizada neste trabalho e o critério de avaliação empregado na análise dos resultados. Na seção 3, a metodologia de realização dos experimentos é descrita. Na seção 4, são apresentados os resultados obtidos e algumas considerações finais; e em seguida, as referências bibliográficas.

2. Materiais e Métodos

Recentemente, as ferramentas de aprendizado de máquina são usadas pela maior parte das pesquisas na classificação de estrutura de proteínas. Como não existe uma relação direta entre a seqüência e a estrutura espacial da proteína, muita atenção tem-se dado a essas técnicas.

Foram selecionados, para a predição de classes estruturais de proteínas, 5 métodos individuais de classificação de aprendizado supervisionado: Árvore de Decisão (AD), K-Vizinhos Mais Próximos (KNN), *Naive Bayes* (NB), Máquinas de Vetores Suporte (SVM) e Redes Neurais Artificiais (RN). A fim de avaliar a aplicabilidade de sistemas de multi-classificação, foram selecionados dois tipos de multi-classificadores homogêneos: *Bagging* e *Boosting* e 3 tipos de classificadores heterogêneos: *Stacking*, *StackingC* e *Vote*, empregando os 5 diferentes classificadores individuais citados anteriormente. Todos os métodos usados em nosso estudo foram obtidos do pacote de aprendizagem da máquina do WEKA [18] (http://www.cs.waikato.ac.nz/_ml/weka/) e tiveram seus resultados comparados entre si.

2.1 Base de Dados

Usamos, neste trabalho, a base de dados disponibilizada pelos autores em [17] (<http://www.brc.dcs.gla.ac.uk/~actan/eKISS/data.htm>).

¹ Estrutura secundária se refere a elementos estruturais locais, como α -hélices e β -folhas.

Esta base de dados é uma modificação da base de dados original criada e usada em [5] e [6]. Esta base de dados original (http://www.nersc.gov/_cding/protein/) é formada por um conjunto do treinamento (Ntrain) e conjunto de teste (Ntest). O conjunto de treinamento foi extraído do PDB [10] e compreende 313 proteínas de 27 dobras mais povoadas do SCOP (mais de sete exemplos para cada dobra) com todas as seqüências de proteínas com menos de 35% de similaridade entre si e representando suas principais classes estruturais (all- α , all- β , α/β e $\alpha+\beta$). O conjunto de teste foi extraído de PDB 40D [11] que contém 385 representantes (excluindo as seqüências já usadas no conjunto de treinamento) das mesmas 27 dobras de SCOP, em que, novamente, duas proteínas não têm mais do que 35% da seqüência iguais uma das outras.

As características usadas no sistema de aprendizagem são extraídas das seqüências da proteína de acordo com o método descrito em [7], onde uma proteína é representada por um conjunto de vetores baseados em várias propriedades físico-químicas e estruturais dos aminoácidos ao longo da seqüência. Estas propriedades são: hidrofobicidade, polaridade, polarizabilidade, predição da estrutura secundária, volume de Van der Waals normalizado e a composição de aminoácidos da seqüência da proteína.

As propriedades de cada seqüência são descritas por um vetor de 21 atributos contínuos, a menos da última propriedade (composição de aminoácidos), que contém 20 atributos. Assim, no total, combinando todos os 6 vetores de características, formamos um único vetor de dimensão 125 (ou 125 atributos). Além dessas propriedades, o comprimento da proteína também é considerado para cada dobra da proteína. Desse modo, a dimensionalidade das características consideradas como um todo passa para 126.

É importante ressaltar que [17] ajustaram sua base de dados em relação à base de dados original em apresentada em [5] e [6] removendo os erros tanto do conjunto de treinamento como do de teste. Os autores também aplicaram a classificação de dobra de proteína de acordo com SCOP 1.61 [11] e Astral 1.61 [3], com todas as seqüências de proteínas com menos de 40% de similaridade entre elas, removendo-se aquelas classes de dobras com menos de 8 exemplos. Feito isso, a base de dados resultante de dobra de proteína conteve 582 exemplos distribuídos em 25 classes de dobras SCOP e, em um nível hierárquico mais alto, em 5 classes estruturais (distribuídos como visto na Tabela 1). A classe acrescentada (*small*), em relação aos dados de [5] e [6], abrange as proteínas que não

se enquadravam em nenhuma das demais classes estruturais (all- α , all- β , α/β e $\alpha+\beta$).

	all- α	all- β	α/β	$\alpha+\beta$	<i>small</i>	Total
Quant. de exemplos	111	177	203	46	45	582

Tabela 1: Distribuição de proteínas em classes estruturais.

Neste trabalho, foi realizada a predição de proteínas considerando as 5 classes estruturais da classificação SCOP (com a adição da classe *small* em relação a [5] e [6]) a ser aprendidas pelas técnicas de AM empregadas.

2.2 Multi-classificadores

Neste trabalho foram usadas 2 diferentes estratégias de multi-classificação homogênea (que combina diferentes classificadores obtidos com o mesmo algoritmo de AM mas com variação dos dados de entrada):

- *Bagging*: constrói os classificadores a partir de conjuntos sucessivos e independentes de amostras de dados geradas a partir do conjunto de dados original, tendo todos eles a mesma quantidade de exemplos (há, portanto, replicação e ausência de certos exemplos), criando classificadores diferentes devido à variação de exemplos nas amostras, sendo combinados através de um método de votação [2].
- *Boosting*: altera a distribuição do conjunto de treinamento baseando-se na performance das classificações anteriores. Isto se deve à característica básica do seu funcionamento, onde os classificadores são gerados seqüencialmente. A cada passagem os pesos dos exemplos são alterados em função do sucesso de sua classificação. As saídas também são combinadas por um esquema de votação [8].

Também foram usados 3 sistemas de multi-classificação heterogênea (que combinam distintos algoritmos de meta conjunto de dados, originado da saída dos classificadores aprendizado):

- *Vote*: Combina classificadores usando a média não ponderada das probabilidades estimadas, através de um esquema de votação por maioria simples.
- *Stacking*: Constrói um meta conjunto de dados, originado da saída dos classificadores base, para ser usado no treinamento do meta classificador, responsável pela predição final do sistema [19].
- *StackingC (Stacking with Confidences)*: Variação do *Stacking* baseada na remoção prévia de características irrelevantes e na redução da dimensionalidade do meta conjunto de dados, de tal modo a ser independente do número de classes [15].

2.3 Avaliação

A comparação de dois métodos de aprendizagem supervisionada é realizada, tradicionalmente, analisando o

significado estatístico da diferença entre a média da taxa do erro global de classificação, em conjuntos independentes de teste, dos métodos avaliados [14]. A fim de avaliar a média da taxa do erro de classificação, diversos conjuntos (distintos) de dados são necessários. Entretanto, a quantidade de dados disponíveis é normalmente limitada. Uma forma de superar este problema é dividir a base de dados em conjuntos de treinamento e de teste pelo uso do procedimento de *k-fold cross validation* [12]. Neste trabalho, é usado o *10-fold stratified cross validation*, garantindo que cada um dos 10 *fold*s apresentem a mesma proporção das diferentes classes.

3. Experimentos

Os melhores parâmetros de cada uma das técnicas foram escolhidos de acordo com o seguinte procedimento: para um algoritmo, por exemplo, com somente um parâmetro a ser definido, um valor inicial para tal parâmetro é escolhido e o algoritmo executado. Então, experimentos com valor maior e menor que ele são também realizados. Se com o valor inicialmente escolhido o classificador obteve o melhores resultado (em termos da média de erro de classificação), então outros experimentos não precisam mais ser executados. Caso contrário, o mesmo processo é repetido para o valor do parâmetro com o melhor resultado até então, e assim por diante. Naturalmente, este procedimento consome mais tempo com o aumento do número dos parâmetros a serem investigados.

Usando tal procedimento, os seguintes valores dos parâmetros de cada uma das técnicas de AM empregadas foram obtidos (os parâmetros não citados foram definidos para os seus próprios valores *default*):

- NB: *KernelEstimator* foi definido para *true*;
- KNN: *distance Weighting* = $1/distance$;
- AD: todos os parâmetros foram definidos para os seus valores *default*;
- SVM: $c = 2 \cdot 6$ e expoente = 2;
- RN: número de neurônios na camada escondida = 10; taxa de aprendizado = 0.001; máximo número de iterações = 1000; momento = 0.9; tamanho do conjunto de validação = 10%.

Para a formação dos multi-classificadores homogêneos, os classificadores usados como base usaram os mesmos parâmetros definidos acima. Tanto para o processo de *Bagging* como para o de *Boosting*, o número de iterações foi definido para 100 e todos os outros parâmetros tiveram seus valores *default* mantidos.

Por representarem diferentes paradigmas de aprendizado, os métodos individuais (citados

anteriormente) foram escolhidos para compor os classificadores base do *Stacking*, *StackingC* e *Vote*, [16]. Assim como para os classificadores homogêneos, esses métodos individuais foram empregados com os mesmos parâmetros definidos acima. Foram avaliadas algumas variações quanto ao uso do meta classificador no *Stacking* e *StackingC*. Para o *Stacking*, foram utilizados como meta os métodos NB e KNN (métodos simples e de rápido aprendizado); no *StackingC*, o *Linear Regression* (LR) e KNN (o NB não foi utilizado porque o meta classificador desse método necessita ter como saída um valor numérico, e na implementação do WEKA, o NB é categórico). O LR foi usado com os valores *default* para seus parâmetros; o KNN, com *distance Weighting* = $1/distance$; e o NB com *KernelEstimator* = *true*.

Cada um dos métodos, como já mencionado, foi treinado com o *Stratified 10-fold Cross-Validation* da base de dados, de acordo com os melhores parâmetros encontrados. Então, considerando todos os experimentos, a média da porcentagem de classificação incorreta nos conjuntos de testes independentes foi calculada. Em seguida, essas médias foram comparadas duas a duas pelo teste de hipótese [12] com o nível de significância (α) igual a 0,05.

4. Resultados e Discussões

Inicialmente, analisamos o desempenho dos métodos individuais aplicados ao problema de acordo com a melhor configuração para cada um deles, considerando a média da taxa de erro de classificação global.

A Tabela 2 a seguir apresenta a média da porcentagem incorreta de classificação de cada um dos métodos de AM empregados, de acordo com a melhor configuração para cada um deles, e o desvio padrão verificado em cada caso. De acordo com essa tabela e com o teste de hipótese aplicado, pode-se verificar que o SVM apresenta um desempenho superior aos outros métodos aplicados (taxa de erro igual a 17,60%), com exceção a RN, cujo teste de hipótese indicou que não há evidência de diferença estatisticamente significativa entre os resultados desses dois métodos.

Algoritmo	Média	Desvio Padrão
AD	25,21%	5,74%
KNN	24,34%	4,82%
NB	22,04%	5,27%
SVM	17,60%	4,54%
RN	18,79%	2,62%

Tabela 2: Taxa de erro dos classificadores individuais.

A Tabela 3 mostra a média e o desvio padrão da taxa de erro de classificação para as técnicas de multi-classificação homogênea: *Bagging* e *Boosting*. Os multi-classificadores gerados com essas técnicas usaram como

classificadores base os métodos individuais previamente apresentados, com exceção do KNN e do NB, pois, para eles, os resultados preliminares não demonstraram que os sistemas multi-classificadores fossem apropriadas tais métodos.

Em relação aos resultados obtidos com a técnica *Bagging*, não é verificada nenhuma diferença estatisticamente significativa entre eles. Em relação aos resultados obtidos com a técnica *Boosting*, verifica-se que esta técnica aplicada à AD apresenta uma menor taxa de classificação incorreta (15,81%) do que os sistemas multi-classificadores obtidos tanto com o SVM como com a RN, não apresentando estes dois últimos uma diferença significativa entre eles.

	Média	Desvio Padrão
<i>Bagging</i> AD	18,55%	5,17%
<i>Bagging</i> SVM	17,60%	4,54%
<i>Bagging</i> RN	18,91%	2,53%
<i>Boosting</i> AD	15,80%	5,04%
<i>Boosting</i> SVM	19,32%	4,60%
<i>Boosting</i> RN	18,78%	4,24%

Tabela 3: Taxa de erro dos sistemas de multi-classificação homogênea.

De acordo com a Figura 1, pode-se verificar uma considerável melhoria nos resultados obtidos com os sistemas de multi-classificação que usaram as Árvores de Decisão como classificador base. Observa-se que o resultado obtido com a AD individual apresentou uma taxa de erro de 25,21%. Então, com o uso do *Bagging*, essa taxa de erro caiu para 18,56%. E, finalmente, aplicando-se o *Boosting*, foi obtida uma taxa de erro ainda mais baixa (15,81%).

Em relação ao SVM e a RN, os resultados obtidos com seus multi-classificadores não mostraram a mesma melhoria que com a AD. Por exemplo, nenhuma diferença estatística foi detectada entre o *Bagging* SVM e o SVM individual. No caso do *Boosting* SVM, ele mostrou um desempenho estatisticamente inferior quando comparado ao SVM individual. Para a RN, nenhuma diferença estatística foi detectada em relação ao *Bagging* RN e *Boosting* RN.

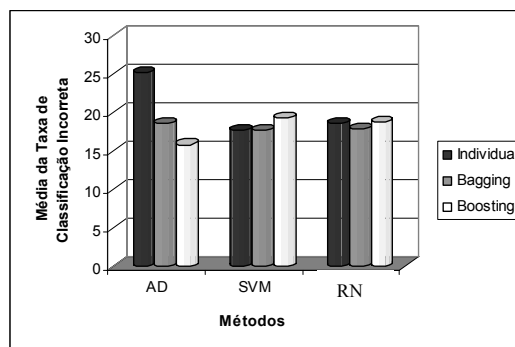


Figura 1: Resumo dos resultados para os sistemas individuais e de multi-classificação homogênea.

A Tabela 4 mostra a média e o desvio padrão da taxa de erro de classificação para as técnicas de multi-classificação heterogênea: *Stacking*, *StackingC* e *Vote*. O *Stacking* e o *StackingC* foram treinados com diferentes meta classificadores, como mencionado na seção 3.

Em relação aos resultados obtidos com a técnica *Stacking*, verifica-se que o emprego do KNN como meta classificador apresenta uma maior taxa de classificação incorreta (22,15%) do que o sistema multi-classificador obtido com o NB usado como meta. Em relação aos resultados obtidos com a técnica *StackingC*, não é verificada nenhuma diferença estatisticamente significativa com a variação do meta classificador (LR e KNN).

Entre as diferentes técnicas de multi-classificação apresentadas acima, podemos concluir, com a aplicação do teste de hipótese, que o *StackingC* apresenta um menor erro de classificação que os outros multi-classificadores, seguido do *Vote* e do *Stacking*.

	Média	Desvio Padrão
<i>Stacking</i> com NB	19,18	4,48
<i>Stacking</i> com KNN	22,15	4,75
<i>StackingC</i> com LR	16,31	4,07
<i>StackingC</i> com KNN	16,52	4,74
Vote	17,47	4,29

Tabela 4: Taxa de erro dos sistemas de multi-classificação homogênea.

A Figura 2 a seguir apresenta um gráfico que resume os resultados obtidos com os classificadores individuais e os melhores multi-classificadores encontrados (embora não haja diferença estatística entre o *StackingC* com LR e com KNN usados como meta classificadores, o *StackingC* com LR é considerado por ter apresentado um menor custo computacional – tempo – na sua construção do que com o KNN).

Observando a Figura 1 pode-se verificar a melhoria nos resultados obtidos com método *StackingC* tanto em relação aos outros multi-classificadores quanto a todos os

classificadores base (16,31%). Já o *Stacking*, mostrou-se melhor do que os métodos AD, NB e KNN, mas não apresentou melhor desempenho que a RN (nenhuma diferença estatística foi detectada); e quando comparado ao SVM, mostrou um desempenho estatisticamente inferior a ele. Daí pode-se verificar a maior eficiência do *StackingC*, sendo capaz de melhorar o desempenho do *Stacking*, reduzir o custo computacional e ser mais estável do que outros sistemas de multi-classificação heterogênea [16]. O método *Vote*, por sua vez, apesar de simples, apresentou uma boa performance quando comparado com os classificadores base utilizados, com uma menor taxa de classificação incorreta (17,47%), com exceção do resultado obtido com o SVM, que através da aplicação do teste de hipótese, nenhuma diferença estatística é detectada entre eles.

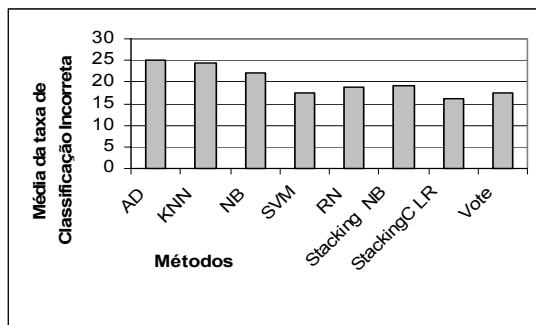


Figura 2: Resumo dos resultados para os sistemas individuais e de multi-classificação heterogênea.

A Figura 3 a seguir mostra um gráfico dos melhores resultados obtidos com os sistemas de multi-classificação homogêneos e heterogêneos (em termos da menor taxa de erro e do custo computacional). Com a utilização do teste de hipótese, não é verificada nenhuma diferença estatisticamente significativa entre os multi-classificadores, como intuitivamente dá para perceber ao observar a Figura 3.

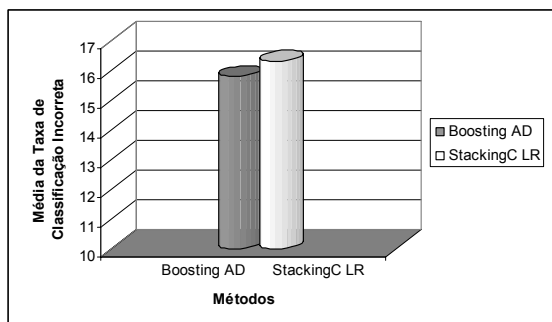


Figura 3: Melhores resultados para os sistemas multi-classificadores homogêneos e heterogêneos.

De um modo geral, como pode ser visto, sistemas multiclassificadores, sejam eles homogêneos ou heterogêneos, apresentam globalmente um desempenho superior ou similar aos obtidos com os classificadores individuais. Uma seleção adequada dos classificadores usados como base, ou como meta, é primordial para o bom desempenho dos sistemas multi-classificadores. O método *Vote*, por exemplo, apesar de ser o multi-classificador utilizado mais simples e de mais baixo custo computacional, em termos de tempo e de complexidade, apresentou um desempenho surpreendente diante dos métodos individuais e, inclusive, dos demais multi-classificadores.

IV- Referências Bibliográficas

- [1] P. Baldi and S. Brunak, *Bioinformatics: the Machine Learning Approach*, second edition ed. MIT Press, 1998.
- [2] L. Breiman (1996a). Bagging predictors. *Machine Learning*, 24:123–40.
- [3] J.-M. Chandonia, N. S. Walker, L. L. Conte, P. Koehl, M. Levitt, and S. E. Brenner, “ASTRAL compendium enhancements,” *Nucleic Acids Research*, vol. 30, no. 1, pp. 260–263, 2002.
- [4] M. W. Craven, R. J. Mural, L. J. Hauser, and E. C. Uberbacher, “Predicting protein folding classes without overly relying on homology,” in *Proc. of ISBM*, vol. 3, 1995, pp. 98–106.
- [5] C. H. Q. Ding and I. Dubchak, “Multiclass protein fold recognition using support vector machines and neural networks,” *Bioinformatics*, vol. 17, no. 4, pp. 349–358, 2001.
- [6] I. Dubchak, I. Muchnik, S. R. Holdbrook, and S. H. Kim, “Prediction of protein folding class using global description of amino acid sequence,” *Proc. Natl. Acad. Sci.*, vol. 92, pp. 8700–8704, 1995.
- [7] I. Dubchak, I. Muchnik, and S. H. Kim, “Protein folding class predictor for SCOP: approach based on global descriptors,” in *Proc. of 5th ISBM*, 1997, pp. 104–107.
- [8] Y. Freund R. Schapire. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*. Bari, Italy. p 148–56.
- [9] K. Guimarães e J. Melo, “Uma Introdução à Análise de Sequências e Estruturas Biológicas.” Cap. 1. In: III Jornada de Mini-Curso de Inteligência Artificial – Livro Texto, Editora SBC.
- [10] U. Hobohm and C. Sander, “Enlarged representative set of proteins,” *Protein Sci.*, vol. 3, pp. 522–524, 1994.

- [11] L. LoConte, B. Ailey, T. J. P. Hubbard, S. E. Brenner, A. G. Murzin, and C. Chothia, "Scop: a structural classification of proteins database." *Nucleic Acids Research*, vol. 28, no. 1, pp. 257–259, 2000.
- [12] T. Mitchell, *Machine Learning*. New York: McGraw Hill, 1997.
- [13] F. M. G. Pearl, D. Lee, J. E. Bray, I. Sillitoe, A. E. Todd, A. P. Harrison, J. M. Thornton, and C. A. Orengo, "Assigning genomic sequences to CATH." *Nucleic Acids Research*, vol. 28, no. 1, pp. 277–282, 2000.
- [14] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions." In *proceedings of the Third international conference on Knowledge discovery and data mining*, Menlo park, CS. AAAI Press, 43–48, 1997. Slovenia, 1999.
- [15] A. K. Seewald, "How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness", in *Proceedings of the Nineteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers, 2002, pp. 554–561.
- [16] A. K. Seewald, "Towards Understanding Stacking - Studies of a General Ensemble Learning Scheme." PhD thesis, Institute for Med. Cybernetics and Artificial Intelligence, University of Vienna, 2003.
- [17] A. C. Tan, D. Gilbert, and Y. Deville, "Multiclass protein fold classification using a new ensemble machine learning approach," *Genome Informatics*, vol. 14, pp. 206–217, 2003.
- [18] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementation*. USA: Morgan Kaufman Publishers, 2000.
- [19] D. H. Wolpert, "Stacked generalization". *Neural Networks* (1992), 5:241–259, Pergamon Press.