PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Bacharelado em Ciência da Computação

Pedro Ribeiro Bastos Soares

PREVISÃO DE FUNÇÃO DE ENZIMAS UTILIZANDO MÁQUINAS DE VETOR DE SUPORTE

Belo Horizonte
2012

Pedro Ribeiro Bastos Soares

PREVISÃO DE FUNÇÃO DE ENZIMAS UTILIZANDO MÁQUINAS DE VETOR DE SUPORTE

Monografia apresentada ao programa de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Cristiane Neri Nobre

Belo Horizonte
2012

Pedro Ribeiro Bastos Soares

PREVISÃO DE FUNÇÃO DE ENZIMAS UTILIZANDO MÁQUINAS DE VETOR DE SUPORTE

Monografia apresentada ao programa de Bacharelado em Ciência da Computação da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Cristiane Neri Nobre

Marco Antônio da Silva Barbosa

Saulo Augusto de Paula Pinto

Belo Horizonte, 26 de Novembro de 2012

RESUMO

As proteínas são macromoléculas que são essenciais para a realização de diversas funções importantes em uma célula, como por exemplo, fazer parte da estrutura e catalizar reações químicas. O número de proteínas conhecidas tem crescido muito, mas devido ao alto custo dos processos de descoberta de função, uma parcela pequena das proteínas tem sua função conhecida. É de extrema importância conhecer a função das proteínas, para o desenvolvimento de novas drogas e para a indústria de bioquímicos. Este trabalho apresenta um método para prever funções de enzimas. Seu objetivo foi modificar um *parser* de arquivos do *Protein Data Bank* para extrair a estrutura e as características das enzimas, propor uma nova codificação para os dados, descobrir características das estruturas para adicionar às sequências e utilizar o algoritmo de máquina de vetor de suporte para classificação em classes de função. Os resultados mostram que a nova codificação apresentou baixa eficiência, com no máximo 31,78% de precisão e 18,18% de acurácia, mas a adição das características utilizando a codificação original mostrou bons resultados com a menor precisão de 81,33% e a menor acurácia de 92.67%.

Palavras-chave: Função de proteínas. SVM. Aprendizado de máquina. Validação cruzada.

ABSTRACT

Proteins are essential macromolecules for the realization of multiple functions in a cell, for example, they form the protein structure and catalyze chemical reactions. The number of known protein has grown, but due to the high cost of the function discovery processes, a small portion of proteins function is known. It's very important to know protein functions for the development of new drugs and the biochemical industry. This paper presents a method for predicting enzymes functions. It's goal was to modify a parser of Protein Data Bank files to extract the enzymes structure and characteristics, proposing a new data encoding, discover new structures features to add to the sequences, use the support vector machine algorithm for classification in function classes. The results shows that the new encoding showed a poor performance, with a maximum precision of 31,78% and maximum accuracy of 18,18%, but the addition of caracteristics in the previous encoding showed an improvement, with lower precision of 81,33% e lower accuracy of 92,67%.

Keywords: Protein function. SVM. Machine learning. Cross-Validation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DIP – *Database of Interacting Proteins*

FCANAL – *Fast Calculable Protein Function Analizer*

FFF – *Fuzzy Functional Form*

LOOCV – *Leave-One Out Cross-Validation*

PDB – *Protein Data Bank*

RBF – *Radial Basis Function*

SVM – *Support Vector Machine*

SUMÁRIO

1 INTRODUÇÃO

Proteínas são macromoléculas versáteis e são essenciais. Isto ocorre porque são moléculas que exercem diversas funções nos seres vivos, como por exemplo, elas são uma parte importante da composição das células representando a segunda maior fração do peso celular, perdendo apenas para a água. Além de ser parte da constituição das células e órgãos, elas podem também servir como catalizadoras de reações químicas do metabolismo (Enzimas) e da manutenção do meio celular (Proteína Transmembrana) (PANDEY; KUMAR; STEINBACH, 2006).

O conhecimento da função das proteínas é de extrema importância já que ele possibilita o desenvolvimento de novas drogas, tratamentos e até mesmo de bioquímicos como biocombustíveis (DING; DUBCHAK, 2000),(BOCK; GOUGH, 2001).

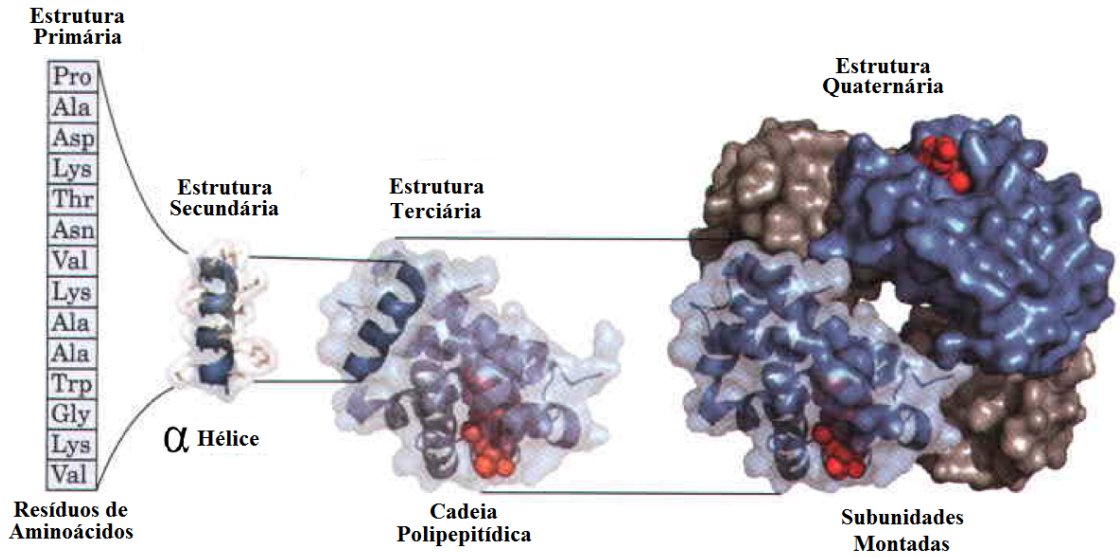
Com o avanço do projeto genoma, o número de proteínas conhecidas tem crescido muito e isso se torna um grande desafio pois os métodos tradicionais de descoberta de função, como a cristalografia, exigem grande esforço humano e experimental, fazendo com que essas metodologias fiquem com grande custo de tempo e com pouco rendimento (PANDEY; KUMAR; STEINBACH, 2006) (NOBRE, 2011). Como exemplo desse problema, o setor farmacêutico atua apenas em 500 proteínas e o PDB tem 66432 proteínas, mas apenas 3298 têm sua função conhecida (NOBRE, 2011). Uma boa alternativa para solucionar este problema reduzindo custos são soluções computacionais que podem viabilizar o processo de descoberta de funções de proteínas, diminuindo a necessidade de testes laboratoriais.

Na maioria das vezes, a função de uma proteína é dada pela possibilidade de realizar ligações com outras moléculas (Ligante), já que a interação Proteína-Ligante permite ao organismo reagir às mudanças do ambiente e metabólicas rapidamente e de forma reversível (LEHNINGER; NELSON; COX, 1999). Essa interação pode ser descrita matematicamente pela constante de associação (K_a) que representa o grau de afinidade do ligante L pela proteína P. Essa métrica não é muito precisa, pois a estrutura apresenta grande importância na interação e mesmo pequenas mudanças em sua conformação podem afetar sua função. Portanto, é de grande importância também levar em conta a estrutura da proteína (LEHNINGER; NELSON; COX, 1999).

A biologia, para entender e descrever a estrutura de uma proteína, estabeleceu um modelo hierárquico que apresenta quatro níveis de complexidade. Como definido por Lehninger, Nelson e Cox (1999), a estrutura primária é a descrição das ligações que unem os resíduos de aminoácidos em uma cadeia polipeptídica, a secundária são os padrões de conformação dos

arranjos estáveis entre os resíduos (α-Hélice em forma espiral e folha β em forma plana) e a terciária descreve a conformação da cadeia no espaço tridimensional e, por último, a quaternária é o arranjo espacial, caso a proteína possua, de duas ou mais cadeias. Um exemplo das estruturas pode ser visto na Figura 1.

Figura 1 – Estrutura das Proteínas.



Fonte: (LEHNINGER; NELSON; COX, 1999).

Lehninger, Nelson e Cox (1999) mostram que as funções das proteínas depende da sua conformação tridimensional, mas essa conformação depende da sequência dos resíduos de aminoácidos, ou seja, da sua estrutura primária. A prova deste fato é que diversas doenças genéticas ocorrem devido a defeitos na produção de proteínas, como a troca ou falta de resíduos, fazendo com que elas realizem funções diferentes.

Resende et al. (2012) descreveram um método utilizando um *parser* desenvolvido para extrair as estruturas primárias e secundárias de arquivos do PDB e usá-los no algoritmo classificador SVM (*Support Vector Machine*). Também foi desenvolvido um algoritmo genético para se descobrir os melhores parâmetros para a SVM.

Assim este trabalho tem como objetivo, melhorar o trabalho desenvolvido por Resende et al. (2012), propondo uma nova codificação para os dados e adicionando novas características da estrutura da proteína ao classificador.

1.1 Justificativa

As proteínas são moléculas extremamente importantes, pois são responsáveis pelo funcionamento do organismo de todos os seres vivos. Com o avanço do projeto genoma, muitas proteínas são identificadas e é essencial que se saiba suas funções, tanto para o estudo de processos biológicos, quanto para a indústria de bioquímicos.

A bioinformática pode ser de grande ajuda no processo de descoberta de funções de proteínas já que apresenta custos muito menores que os processos tradicionais da biologia que exigem grandes gastos com mão de obra, testes laboratoriais e tempo.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma metodologia para adicionar características da estrutura da proteína ao classificador de Resende et al. (2012) e propor uma nova codificação para os dados, a fim de melhorar o sua eficiência.

1.2.2 Objetivos Específicos

- Adaptar a ferramenta criada por Resende et al. (2012) para incorporar as características da estrutura das proteínas e uma codificação de uma forma mais fácil.
- Identificar características das proteínas importantes para predição de função.
- Propor novas codificações da sequência das proteínas e avaliá-las.

1.3 Organização do trabalho

Na Seção 2, são apresentadas o referencial teórico e a revisão da literatura. Na Seção 3 é apresentado a metodologia e na quarta seção estão as conclusões e as considerações finais.

2 REVISÃO DA LITERATURA

2.1 Referencial Teórico

2.1.1 Máquinas de Vetor de Suporte

Nos casos em que não existe um conhecimento prévio sobre o domínio da aplicação, uma alternativa bem sucedida na área do aprendizado de máquina é o uso da SVM, pois ela apresenta três características que possibilitam isso (RUSSELL; NORWIG, 2009). A primeira delas é que ela constrói uma margem separadora máxima, ou seja, uma fronteira com a maior distância possível entre os pontos de treinamento e isso ajuda a fazer uma boa generalização. A segunda característica é que a SVM cria um hiperplano que ajuda a colocar os dados que não sejam linearmente separáveis em maiores dimensões para que possam ser separados. Russell e Norwig (2009) fizeram as seguintes definições:

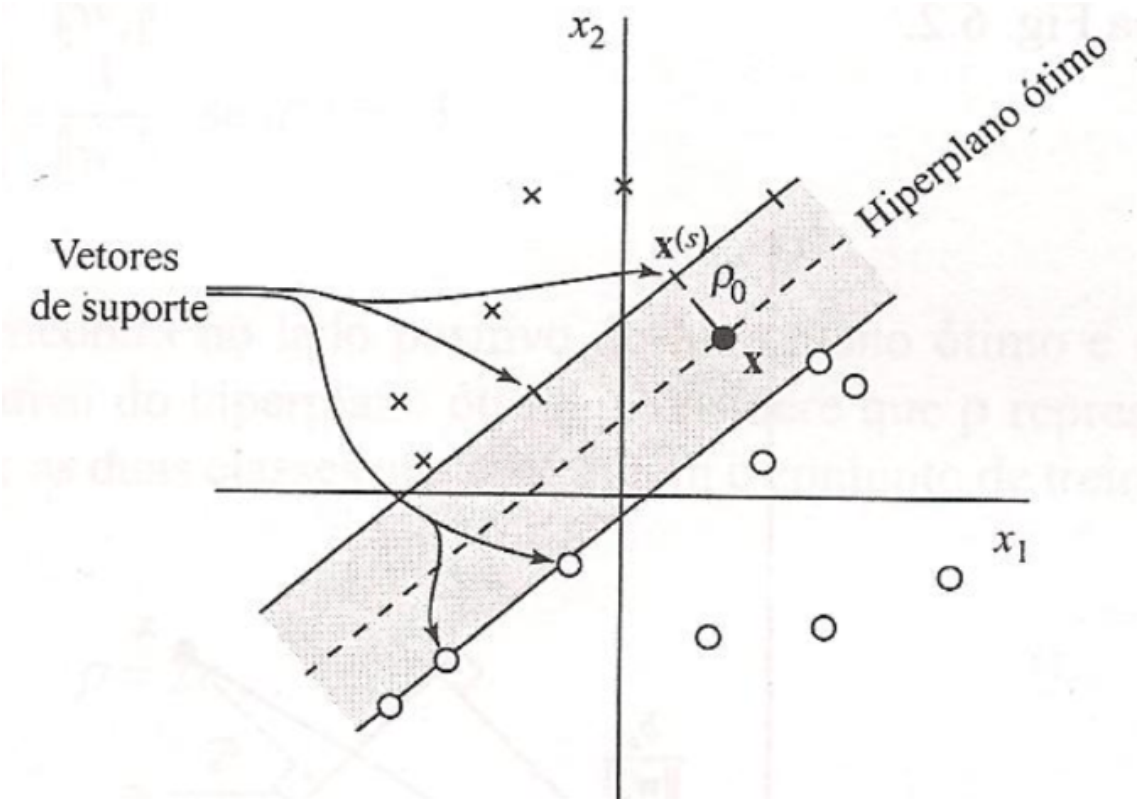
- **Métodos Paramétricos**: Métodos em que usam os dados de treinamento para extrair um conjunto fixo de parâmetros w e, em um certo momento, os treinamentos podem ser descartados já que w é uma representação do resumo dos treinos. Exemplos de métodos paramétricos são as Regressões Lineares e as Redes Neurais.

- **Métodos Não Paramétricos**: Também são conhecidos como aprendizado baseado em instância ou aprendizado baseado em memória. Simplificadamente, são métodos que guardam os dados de treinamento em uma tabela e para responder uma hipótese $h(x)$ ele verifica se x está na tabela e retorna a resposta correspondente. Um problema é que este método não faz uma boa generalização, ou seja, se x não entra na tabela ele retorna uma resposta padrão.

A terceira característica é que a SVM é um método não paramétrico que guarda uma parte do número de treinamentos, normalmente um número da ordem de uma pequena constante, multiplicado pelo número de dimensões. A SVM também agrega vantagens de métodos paramétricos e não paramétricos dando flexibilidade para representar funções complexas e é resistente ao *overfitting* (quando o modelo tende a se ajustar em demasiado às entradas de treino) (RUSSELL; NORWIG, 2009).

Segundo Haykin (1998), o objetivo da SVM é achar o hiperplano ótimo, o maior plano entre as classes a serem classificadas, e o separador máximo de margem que é o separador mais distante das amostras. Outro conceito importante são os vetores de suporte, que levam este nome pois são os pontos mais próximos do separador máximo de margem e por isso são a base para encontrar o hiperplano. Geralmente, existem muito mais pontos do que vetores de suporte e isso permite que a SVM tenha as vantagens de um método paramétrico. A Figura 2 é um exemplo desses conceitos em 2 dimensões.

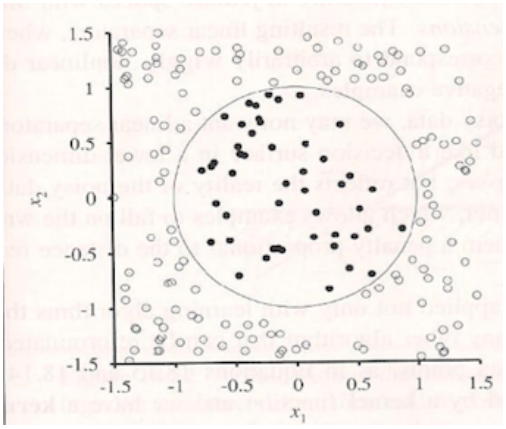
Figura 2 – Hiperplano ótimo e Vetores de Suporte.



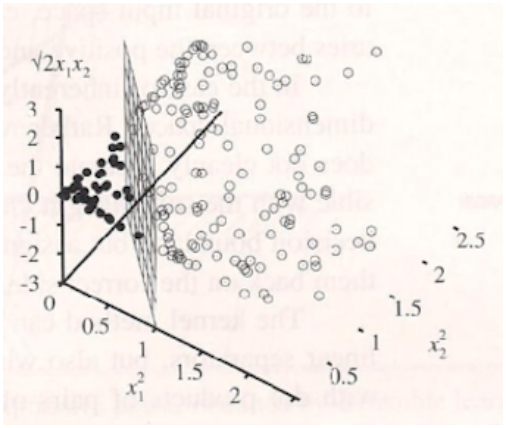
Fonte: (HAYKIN, 1998).

As classes que não são linearmente separáveis podem ser um problema já que é uma tarefa difícil separá-las. No entanto, Russell e Norwig (2009) mostram que todo dado pode ser linearmente separável caso ele seja representado com um número suficiente de dimensões. Assim, os autores definem um método que serve para achar separadores em dimensões maiores de classes não linearmente separáveis. A Figura 3 mostra um exemplo de classes não linearmente separáveis em 2 dimensões e a representação em 3 dimensões com o separador.

Figura 3 – Classes não linearmente separáveis.



(a) Exemplo de 2 classes não linearmente separáveis em 2 dimensões.



(b) Mesmas classes representadas em 3 dimensões. Como estão representadas em uma dimensão maior, podem ser separadas por um plano.

Fonte: (RUSSELL; NORWIG, 2009).

2.1.2 Validação Cruzada

A validação cruzada (*Cross-Validation*) é uma metodologia para avaliar modelos de predição muito utilizada na área da inteligência artificial, principalmente para avaliar métodos de aprendizado de máquina.

Um método de avaliação de modelos de predição deve ser capaz de saber se uma determinada hipótese vai ser suficiente para prever os dados futuros de forma eficaz. Russell e Norwig (2009) definiram que a taxa de erro de uma hipótese h é a proporção de vezes que $h(x) = y$ onde x é a entrada e y é a previsão. Deve-se tomar cuidado porque uma hipótese com baixa taxa de erro não significa que ela generaliza bem o problema porque, para se ter uma avaliação eficaz, o método deve ser avaliado com entradas que ele ainda não conhece. A solução criada por este problema foi a validação cruzada, que segundo Russell e Norwig (2009), tem 3 tipos:

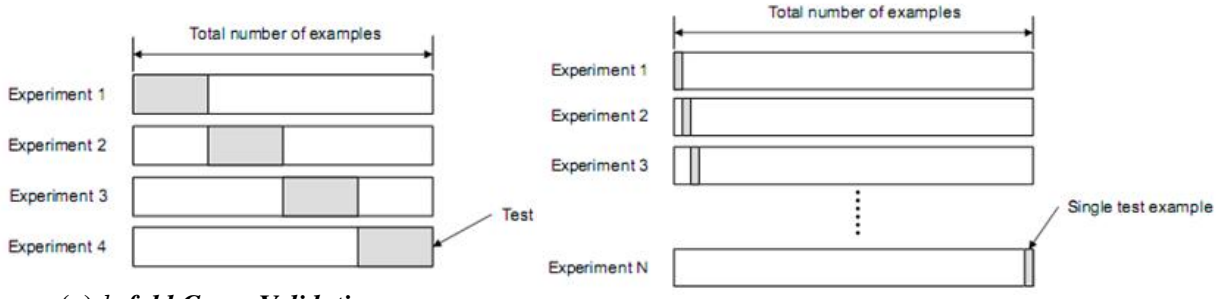
- ***Holdout Cross-Validation***: Se baseia em escolher aleatoriamente um conjunto da amostra para o método aprender (os dados de treino) e definir a hipótese h e usar o restante dos exemplos (dados de teste) para calcular a precisão de h.
- k***-fold Cross-Validation***: Cada exemplo é usado ora como treino, ora como teste. São criados k subconjuntos iguais de amostras. Executa-se k rodadas de treinamento onde $1/k$ dos elementos do conjunto é usado como teste e o restante como treinamento. Então,

a avaliação é feita calculando-se a média entre os resultados das rodadas. Normalmente, são escolhidos os valores 5 ou 10 para k pois são suficientes para um bom resultado com um custo computacional apenas 5 ou 10 vezes maior.

- ***Leave-One Out Cross-Validation* (LOOCV)**: O mesmo que o método k-*fold* mas utilizando $k = n$, onde n é o numero de amostras, ou seja, usar um elemento para o teste e o resto como treinamento.

O método de *holdout* tem o problema que ele pode usar mal suas amostras já que depende do acaso para escolher os testes e treinos pois com poucos treinos pode ser que ele produza uma hipótese fraca, mas se for usado uma quantidade pequena de testes ele pode ter uma avaliação ruim (RUSSELL; NORWIG, 2009). Devido a esses problemas normalmente os outros métodos de *Cross-Validation* são melhores, principalmente a k-fold já que a LOOCV tem um custo computacional maior (n multiplicado pela complexidade do método).

Figura 4 – Tipos de *Cross-Validation* .



(a) k***-fold Cross-Validation*.** **(b)** ***Leave-One Out Cross-Validation***

Fonte: (RAGHAVA, 2009).

2.2 Trabalhos Relacionados

Pandey, Kumar e Steinbach (2006) identificaram as quatro abordagens mais comuns para prever funções de proteína utilizando sua estrutura:

- **Baseado em Similaridade**: Métodos que usam técnicas de alinhamento de estruturas da proteínas para achar a similaridade entre elas.

- **Baseado em *Motifs***: Métodos que tentam encontrar subestruturas das proteínas (Motifs) e achar um mapeamento entre a função e um conjunto de *Motifs*.

- **Baseado em Superfície**: Métodos que analisam a superfície da proteína encontrando concavidades e buracos para inferir a função.
- **Baseado em Aprendizado de Máquina**: Métodos que usam técnicas de aprendizado de máquina, como redes neurais e SVM, para classificar a função da proteína.

2.2.1 Similaridade

Kolodny e Linial (2004) fizeram um método de alinhamento de estrutura de proteína, chamado STRUCTAL, para achar a similaridade em estruturas e assim poder encontrar relações evolucionárias que são difíceis ou impossíveis de se encontrar. Os autores identificaram este alinhamento como um problema de otimização e desenvolveram um algoritmo polinomial aproximado para resolve-lo. Como resultado obtiveram um algoritmo de complexidade $O(\frac{n^{10}}{\epsilon^6})$ onde n é o tamanho da proteína que está no máximo a uma distancia ϵ do ótimo.

Krissinel e Henrick (2004) descreveram um método de alinhamento da estrutura secundária da proteína em três dimensões através de um algoritmo que verifica o isomorfismo entre grafos de proteínas. Os grafos são construídos de forma que os vértices são uma tupla $\{T_i, L_i\}$ onde T_i é o tipo (α-*Helix* ou β-*Sheet*) e L_i é o número de resíduos. As arestas são representações da geometria e orientações de estruturas conectadas. Como resultado os autores chegaram a um método que muitas vezes se mostrou eficiente em comparação com outros trabalhos na área de alinhamento de estrutura secundária em três dimensões.

Shatsky, Nussinov e Wolfson (2004) apresentam um método de alinhamento de estruturas de proteína chamado *MultiProt* que faz múltiplos alinhamentos simultâneos das proteínas de entrada. Os autores concluíram que o método apresenta bons resultados e tem quatro vantagens sobre outros métodos. A primeira é a sobreposição simultânea das estruturas, a segunda é que a solução é dada para um conjunto de proteínas, a terceira é que ele funciona para proteínas com mais de uma cadeia e por fim o *MultiProt* possibilita a escolha de preservar a ordem da sequência de proteína ou não.

2.2.2 Motif

Gennaro et al. (2001) propuseram o método que primeiramente acha a estrutura aproximada em três dimensões da proteína, depois usa um descritor chamado FFF que extrai informações da estrutura para achar partes que sejam relevantes para a função(*Motif*). A pesquisa também estendeu a abrangência do FFF para mais funções e como resultado tiveram uma qualidade de resposta melhor comparado com outros métodos baseados em *motif*. Eles também concluíram que a adição da estrutura terciária pode ser relevante para prever função de proteína.

PRINTS e supplement (2003) apresentaram o PRINTS e o prePRINTS. O PRINTS é um banco de dados com características e *Motifs* de proteínas que contém aproximadamente 11000 *Motifs*. Basicamente o PRINTS era mantido manualmente mas os autores desenvolveram o sistema prePRINTS para extrair *Motifs* automaticamente de proteínas coletadas do *Swiss-Prot* para alimentar o banco. Eles conseguiram um método que gera entre 30 e 50 *Motifs* por dia e apresentam 25% de dados com qualidade suficiente para ser inserida no banco.

Suzuki et al. (2005) coletaram proteínas do PDB e desenvolveu o FCANAL, um método que usa uma matriz de distâncias entre ligações de resíduos e a frequência que aparecem para extrair *Motifs*. Segundo os autores eles conseguiram aproximadamente 80% de acurácia.

2.2.3 Superfície

Binkowski, Adamian e Liang (2003) descrevem um método que usa dados coletados do PDB para achar vazios (buracos) e bolsos (concavidades) na superfície da proteína. Os autores então usaram o algoritmo de Smith-Waterman para encontrar semelhanças entre padrões de superfície de proteínas e assim poder prever suas funções.

Ferrè et al. (2004) construíram o SURFACE, um banco de dados com informações sobre a superfície de uma proteína. Os autores afirmam que esta característica pode ser usada para prever funções de proteína. Eles usaram um algoritmo para achar cavidades na estrutura representadas por *patches* (resíduos que cercam a cavidade) e assim podem inferir locais onde pode ocorrer interações com um ligante e usá-los para construir o banco.

2.2.4 Aprendizado de Máquina

Bock e Gough (2001) utilizam o algoritmo SVM para reconhecer interações proteína-proteína e gerar uma decisão binária (1 se existe e 0 se não existe interação). As proteínas foram extraídas do banco DIP e, além da estrutura primária, foi inserido para cada resíduo as propriedades de carga, hidrofobicidade e a tensão de superfície. Segundo os autores, estas características descrevem bem sítios de interação entre as proteínas. Foi utilizada a SVM criado por Joachims (1999) e o vetor foi construído concatenando a estrutura primária com o vetor de características. Como técnica de amostragem, os autores adotaram a validação cruzada e obtiveram uma acurácia média de 80%.

Cai et al. (2004) coletaram enzimas do banco *Swiss-prot* e para cada proteína foi feito um vetor com os aminoácidos, hidrofobicidade, volume de Van der Waals, polaridade, polarizabilidade, carga, tensão da superfície, estrutura secundária e acessibilidade de solvente de cada resíduo. Estes vetores foram usados em uma SVM para fazer a classificação das enzimas em sua classe de função. De 8291 enzimas coletadas, 80.03% foram classificadas corretamente (6658 enzimas).

Kunik et al. (2005) apresentam um método baseado em *Motifs*, sub-estruturas da proteína. Os autores desenvolveram um algoritmo de extração de *motifs* de proteínas coletadas do banco *Swiss-prot* e usou a SVM de Joachims (1999) para a classificação, onde as proteínas eram codificadas como um conjunto de *motifs*. Foi usado 75% das amostras para treino e o restante para teste e obtiveram resultados melhores que outros trabalhos comparados, um que faz comparações entre pares da sequência de aminoácidos e outro que usa propriedades físico-químicas.

Borgwardt et al. (2005) afirmaram que simular mecanismos atômicos e moleculares é além do conhecimento da bioquímica e da capacidade computacional atual, portanto a melhor forma de prever funções de proteínas, segundo os autores, é achando similaridades com proteínas já conhecidas. Os autores definiram um grafo para representar a estrutura, a sequência e as propriedades químicas da proteína, onde os vértices são dados da estrutura secundária (α-*Helix* ou β-*sheet*) e arestas são representações da vizinhança dos vértices na sequência de aminoácidos ou na conformação tridimensional. Foi desenvolvido um algoritmo SVM para a classificação dos grafos de proteína que foram extraídas do PDB. Os autores obtiveram 72% de acurácia para o teste de classificação de enzima/não-enzima e 90% para o teste de classes de

enzimas.

Al-Shahib, Breitling e Gilbert (2007) usaram uma SVM do conjunto de ferramentas WEKA, para classificar proteínas de sete bactérias que causam doenças sexualmente transmissíveis. O vetor da SVM foi obtido extraindo características da sequência de aminoácidos das proteínas. Os autores chegaram a conclusão de que métodos de aprendizado de máquina usando a estrutura da proteína tem um bom desempenho e também que métodos que tentam prever a função de proteínas de origens diversas tem melhor desempenho que métodos que utilizam proteínas de uma mesma espécie. Um ponto importante deste trabalho é que ele apresenta uma lista com 2579 características que podem ser extraídas da sequência de aminoácidos.

Resende et al. (2012) propuseram modificações no trabalho de Nascimento, Yoshioka e Calanzans (2011). O mesmo *parser* foi utilizado para extrair os dados da estrutura primária e secundária mas, desta vez, foi utilizada a SVM multi-classes desenvolvido por Joachims (1999) para a classificação e foram coletadas 6 classes de função de enzima, *Oxidoreductases*, *Transferases*, *Hydrolases*, *Lyases*, *Isomerases* e *Ligases*. Também foi desenvolvido um algoritmo genético para gerar os melhores parâmetros para a SVM. Os autores tiveram acurácia entre 82% e 99% para 4 classes, mas as *Ligases* e *Lyases* tiveram acurácia de 20% e 80%, respectivamente.

3 METODOLOGIA

Este trabalho é uma tentativa de melhorar os resultados de Resende et al. (2012), e por isso, foram usadas as mesmas proteínas da subclasse das enzimas (proteínas que catalizam reações químicas), coletadas pelos autores. As enzimas foram coletadas de acordo com seu número de comissão de enzimas (EC), que é um código para a classificação de acordo com suas propriedades. As funções consideradas foram de Oxidorredutases, Hidrolases, Liases, Ligases, Isomerases e Transferases. A Tabela 1 apresenta as classes consideradas, suas descrições e a quantidade de amostras.

Tabela 1 –
Tabela de Classes de Enzimas.

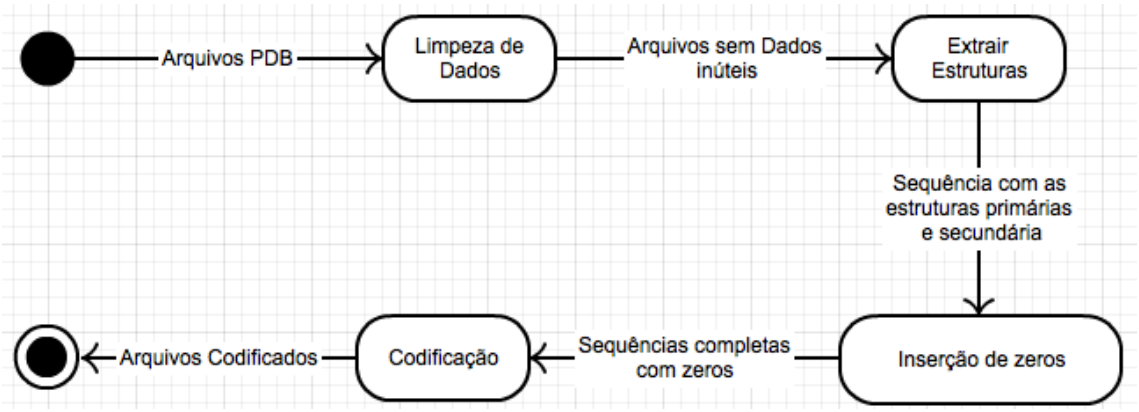
EC Number	Classe	Descrição	Quantidade
1	Oxidorredutases	Cataliza reações de redução e oxidação.	79
2	Transferases	Cataliza reações de transferências de grupos funcionais.	120
3	Hidrolases	Cataliza reações de hidrolise (transferência de grupos funcionais para água).	148
4	Liases	Cataliza a quebra de ligações C-C, C-O e C-N.	58
5	Isomerases	Cataliza a transferências de grupos dentro da mesma molécula para formar isômeros.	51
6	Ligases	Cataliza a ligação do tipo C-C, C-S, C-O e C-N.	17

Fonte: (LEHNINGER; NELSON; COX, 1999), (RESENDE et al., 2012).

3.1 Parser

As enzimas utilizadas foram extraídas do PDB, pois é um banco de dados de proteínas muito utilizado. Resende et al. (2012) desenvolveram um *parser* de arquivos do PDB e este trabalho propõe alterações para melhorá-lo. O *parser* foi dividido em 4 partes. A primeira é a limpeza dos dados; a segunda é a extração das estruturas primárias, secundárias e características da sequência; a terceira é a inserção de zeros e, por fim, a quarta é a codificação final. A Figura 5 apresenta um diagrama de atividades que ilustra as fases do *parser*.

Figura 5 – Diagrama de Atividades do Parser de Arquivos PDB.



Fonte: Elaborado pelo Autor.

3.1.1 Limpeza dos Dados

O objetivo desta parte é remover dos arquivos do PDB toda a informação que não interessa para a construção da sequência da estrutura da proteína. No PDB, a estrutura primária é descrita por linhas que começam com a *tag* SEQRES e, a secundária, pelas *tags* HELIX para α-*Helix* e SHEET para β-*Sheet*. Este passo remove todas as linhas que não começam com as *tags* citadas anteriormente. A Figura 6 ilustra um exemplo de um arquivo PDB e a Figura 7 sua respectiva saída após a limpeza

Figura 6 – Arquivo do PDB.

```
HEADER    LIGASE                                  29-AUG-02   1MKH
TITLE     C-TERMINAL DOMAIN OF METHIONYL-TRNA SYNTHETASE FROM
TITLE    2 PYROCOCCUS ABYSSI
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: C-TERMINAL DOMAIN OF METHIONYL-TRNA SYNTHETASE;
COMPND   3 CHAIN: A;
COMPND   4 FRAGMENT: C-TERMINAL DOMAIN, RESIDUES 616-722 OF SWS
COMPND   5 Q9V011;
COMPND   6 SYNONYM: METHIONINE--TRNA LIGASE; METRS;
COMPND   7 EC: 6.1.1.10;
COMPND   8 ENGINEERED: YES
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: PYROCOCCUS ABYSSI;
SOURCE   3 ORGANISM_TAXID: 29292;
SOURCE   4 GENE: METG;
SOURCE   5 EXPRESSION_SYSTEM: ESCHERICHIA COLI BL21(DE3);
SOURCE   6 EXPRESSION_SYSTEM_TAXID: 469008;
SOURCE   7 EXPRESSION_SYSTEM_STRAIN: BL21-DE3;
SOURCE   8 EXPRESSION_SYSTEM_VECTOR_TYPE: PLASMID;
SOURCE   9 EXPRESSION_SYSTEM_PLASMID: PET3A
KEYWDS    BETA BARREL, DIMERIZATION DOMAIN, LIGASE
EXPDTA    X-RAY DIFFRACTION
```

Fonte: Elaborado pelo Autor.

Figura 7 – Arquivo do PDB após limpeza dos dados.

```
SEQRES   1 A  107  MET TYR VAL LYS PHE ASP ASP PHE ALA LYS LEU ASP LEU
SEQRES   2 A  107  ARG VAL GLY LYS ILE ILE GLU VAL LYS ASP HIS PRO ASN
SEQRES   3 A  107  ALA ASP LYS LEU TYR VAL VAL LYS VAL ASP LEU GLY ASP
SEQRES   4 A  107  GLU VAL ARG THR LEU VAL ALA GLY LEU LYS LYS TYR TYR
SEQRES   5 A  107  LYS PRO GLU GLU LEU LEU ASN ARG TYR VAL VAL VAL VAL
SEQRES   6 A  107  ALA ASN LEU GLU PRO LYS LYS LEU ARG GLY ILE GLY SER
SEQRES   7 A  107  GLN GLY MET LEU LEU ALA ALA ASP ASP GLY GLU ARG VAL
SEQRES   8 A  107  ALA LEU LEU MET PRO ASP LYS GLU VAL LYS LEU GLY ALA
SEQRES   9 A  107  LYS VAL ARG
HELIX    1   1 LYS A    4  LYS A   10  1                                   7
HELIX    2   2 LYS A   53  LEU A   58  1                                   6
SHEET    1   A 4 VAL A  41  ALA A  46  0
SHEET    2   A 4 TYR A  31  ASP A  36 -1  N  VAL A  35   O  ARG A  42
SHEET    3   A 4 LEU A  13  ASP A  23 -1  N  LYS A  22   O  VAL A  32
SHEET    4   A 4 TYR A  61  VAL A  65 -1  O  VAL A  62   N  GLY A  16
SHEET    1   B 4 VAL A  41  ALA A  46  0
SHEET    2   B 4 TYR A  31  ASP A  36 -1  N  VAL A  35   O  ARG A  42
SHEET    3   B 4 LEU A  13  ASP A  23 -1  N  LYS A  22   O  VAL A  32
SHEET    4   B 4 LYS A 105  VAL A 106 -1  O  VAL A 106   N  LEU A  13
SHEET    1   C 2 LYS A  71  LYS A  72  0
SHEET    2   C 2 GLY A  77  SER A  78 -1  O  SER A  78   N  LYS A  71
SHEET    1   D 2 ALA A  84  ASP A  86  0
SHEET    2   D 2 VAL A  91  LEU A  93 -1  O  ALA A  92   N  ALA A  85
```

Fonte: Elaborado pelo Autor.

3.1.2 Extração das Estruturas

Esta parte do *parser* é responsável por extrair as estruturas primárias, secundárias e características da sequência dos dados dos arquivos de saída da fase anterior.

A estrutura primária é representada por linhas que começam com a *tag* SEQRES. A Figura 8 mostra um exemplo de uma linha da estrutura primária.

Figura 8 – Estrutura Primária.

```
SEQRES   5 A  107  LYS PRO GLU GLU LEU LEU ASN ARG TYR VAL VAL VAL VAL
```

Fonte: Elaborado pelo Autor.

O primeiro número da *tag* SEQRES é um índice da linha dentro da sequência em que ele está inserido. Em seguida, existe um caractere para identificar em qual sequência a linha pertence. É comum ter apenas uma sequência, mas, caso haja mais, o *parser* trata como apenas uma, pois não foi considerado a estrutura quaternária da proteína. Assim, um número identifica a quantidade total de resíduos na sequência seguido de até, no máximo, treze resíduos que pertencem a sequência. A Figura 8 mostra que essa linha é a quinta parte da sequência A, que tem, no total, 107 resíduos e os resíduos que compõem esta parte da sequência.

A estrutura secundária é representada por linhas que começam pelas *tags* HELIX ou SHEET. A Figura 9 é um exemplo das duas linhas.

Figura 9 – Estrutura Secundária.

```
HELIX    2   2 LYS A   53  LEU A   58  1                                  6
SHEET    1   A 4 VAL A  41  ALA A  46  0
```

Fonte: Elaborado pelo Autor.

Para a *tag* HELIX, o primeiro número é um índice da hélice e o segundo valor é um alfa numérico que também serve como identificador da hélice. Em seguida, tem um resíduo, um caractere e um número que representam, respectivamente, o primeiro resíduo da hélice, a sequência a que ela pertence e a posição do resíduo na sequência. Depois existem mais três valores estruturados iguais aos anteriores mas, dessa vez representando o resíduo final da hélice. Por último, a linha apresenta um inteiro representando a classe da hélice, um comentário (na maioria dos casos não existe) e um inteiro com o tamanho da hélice. No exemplo da Figura 9, a hélice tem o índice e o identificador igual a 2, começa no resíduo LYS da sequência A que está

na posição 53 e acaba na posição 58 onde existe o resíduo LEU da sequência A. Ela não possui comentário e tem o tamanho de 6 resíduos $(58 - 53 + 1)$.

A *tag* SHEET é muito semelhante à estrutura de HELIX e os dados a mais não são considerados no *parser* e, por isso, não serão detalhados. Na Figura 9, a folha começa no resíduo VAL da sequência A que fica na posição 41 e termina no resíduo ALA da sequência A, que fica na posição 46.

Resende et al. (2012) estruturaram a sequência concatenando a estrutura primária com a secundária como exemplificado pela expressão abaixo:

$$\underbrace{1 \quad 2 \quad 3 \quad 4 \quad 3 \quad 4 \quad 5 \quad 6 \quad 2}_{\text{Primária}} \quad \underbrace{10 \quad 10 \quad 10 \quad 11 \quad 11 \quad 01 \quad 01 \quad 01 \quad 01}_{\text{Secundária}}$$

Bock e Gough (2001) e Lehninger, Nelson e Cox (1999) mencionam a importância da polaridade, da interação com a água e da carga dos resíduos para ocorrer interações com outras moléculas. Tendo isto em vista, foi adicionado à sequência, seis características baseadas no trabalho de Al-Shahib, Breitling e Gilbert (2007). As características são:

- Número de resíduos hidrofílicos (tendência de uma molécula realizar interações com a água).
- Número de resíduos hidrofóbicos (tendência de uma molécula repelir a água).
- Número de resíduos com carga total negativa.
- Número de resíduos com carga total positiva.
- Número de resíduos polares (característica de uma molécula, em que os elétrons não ficam igualmente distribuídos, criando pólos elétricos).
- Tamanho da sequência.

Outro teste realizado por este trabalho, foi uma nova codificação onde os resíduos da estrutura primária são agrupados com sua estrutura secundária a que ele está inserido com o propósito de que na fase de codificação, tenha valores que descrevem melhor a estrutura, e assim tentar melhorar os resultados obtidos. A seguinte expressão mostra como ficou esta codificação:

$$1 \quad 10 \quad 2 \quad 10 \quad 3 \quad 10 \quad 4 \quad 11 \quad 3 \quad 11 \quad 4 \quad 01 \quad 5 \quad 01 \quad 6 \quad 01 \quad 2 \quad 01$$

3.1.3 Inserção de zeros

A SVM desenvolvido por Joachims (1999), que foi utilizado nesse trabalho, exige que os vetores de características inseridos sejam do mesmo tamanho. Então, nessa fase do *parser*, as sequências são completadas com zeros para que fiquem do mesmo tamanho da maior sequência encontrada. O maior tamanho é fornecido pela fase anterior.

3.1.4 Codificação Final

O processo de aprendizagem tem um custo de tempo alto devido ao grande tamanho da sequência. Nascimento, Yoshioka e Calanzans (2011) propuseram que uma tabela *hash* poderia ser usada para codificar partes da sequência, reduzindo seu tamanho e diminuindo o tempo de processamento. Resende et al. (2012) utilizaram um tamanho de bloco igual a 7 para mapear trechos da sequência, mas o presente trabalho também usou os tamanhos de 6, 8 e 14 para os testes com a nova codificação, já que agora que as estruturas primárias e secundárias estão juntas, faz mais sentido agrupa-las com máscaras de tamanho par, para que os resíduos sejam agrupados com sua estrutura secundária equivalente. Além disso, neste passo todas as proteínas são gravadas em um arquivo e com a sua classe concatenada no final da sequência. A expressão abaixo mostra um exemplo de uma sequência antes e depois da codificação com máscara 6.

$$\underbrace{1\ \ 10\ \ 2\ \ 10\ \ 3\ \ 10}_{1}\ \ \underbrace{4\ \ 11\ \ 5\ \ 11\ \ 2\ \ 01}_{2}\ \ \underbrace{1\ \ 10\ \ 2\ \ 10\ \ 3\ \ 10}_{1}\ \ \underbrace{0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0}_{3}$$

3.2 Testes e Treinos

Para fazer os arquivos de testes e treinos, foram usados *scripts* desenvolvidos por Resende et al. (2012) que fizeram um k*-fold Cross-Validation* com k igual a 10. O primeiro *script* tem a função de separar as sequências das proteínas contidas no *parser* em arquivos de acordo com suas classes.

O segundo *script* faz a validação cruzada de cada classe gerando dez arquivos de testes

e dez de treinos para cada classe. O terceiro converte os arquivos de testes e de treinos para o formato da SVM de Joachims (1999). O quarto *script* percorre cada arquivo de teste de todas as classes, juntando-os em dez arquivos finais de teste. O quinto faz o mesmo mas para os arquivos de treino. O sexto e último remove o nome da classe no final de cada proteína dos testes e treinos.

3.3 SVM

Este trabalho utilizou a SVM multi-classe desenvolvido por Joachims (1999). Os arquivos da SVM são compostos por uma lista de vetores em que cada vetor é uma linha. Os vetores são construídos com um número que identifica a sua classe e depois cada elemento é o seu índice concatenado com dois pontos e o seu valor. Os índices devem ser necessariamente ordenados mas podem faltar números (nesse caso são tratados como zero). A expressão, a seguir, mostra um exemplo de um arquivo da SVM em que apresenta um vetor da classe 3 em que o índice 1 tem valor 10.0, o 2 tem valor de 5.5, o 3 tem valor 0 e o 4 tem valor 11.0.

$$3 \quad 1:10.0 \quad 2:5.5 \quad 4:11.0$$

A SVM usado tem 2 partes, a primeira é a de aprendizado que recebe os arquivos de treino como entrada e alguns parâmetros opcionais e tem como saída um arquivo de modelo para ser usado pela etapa de classificação. A etapa de classificação recebe o modelo e os testes e tem como saída um arquivo de predição das classes.

Assim como usado por Resende et al. (2012), com relação aos parâmetros da etapa de aprendizagem, foram usados o parâmetro C, que determina um balanceamento entre erro e margem; o parâmetro T igual a 2, para usar a função RBF e o parâmetro G que é um parâmetro gamma desta função (JOACHIMS, 1999). Foi utilizado o algoritmo genético de Resende et al. (2012) para descobrir os melhores valores dos parâmetros para cada teste.

O arquivo de predição contem um número de linhas equivalente aos arquivos de testes e cada linha tem o número de classes mais um. A primeira coluna é a classe em que a proteína foi classificada e as outras são os valores calculados para cada classe. A Figura 10 mostra um exemplo de um arquivo de predição.

Figura 10 – Arquivo de Predição.

```
5 -0.017850 0.000000 0.000000 0.000000 0.017850 0.000000
5 -0.017850 0.000000 0.000000 0.000000 0.017850 0.000000
1 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
1 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
6 -0.017850 0.000000 0.000000 0.000000 0.000000 0.017850
6 -0.017850 0.000000 0.000000 0.000000 0.000000 0.017850
```

Fonte: Elaborado pelo Autor.

3.4 Cálculo de Desempenho

Para avaliar o desempenho do método foi utilizado quatro métricas, a precisão, a sensibilidade, a acurácia e a especificidade. Para fazer esta análise para cada classe é necessário os seguintes conceitos:

- **Verdadeiro Positivo (VP)**: quantidade de proteínas da classe analisada, corretamente classificadas.
- **Falso Negativo (FN)**: quantidade de proteínas da classe analisada, erroneamente classificadas.
- **Falso Positivo (FP)**: quantidade de proteínas de outras classes, classificadas como a classe analisada.
- **Verdadeiro Negativo (VN)**: quantidade de proteínas de outras classes, classificada como outras classes.

A precisão é a taxa de proteínas da classe analisada que foram classificadas corretamente sobre todas as que foram classificadas nesta classe. A Equação 1 define como a precisão é calculada.

$$P = \frac{VP}{VP + FP} \tag{1}$$

A sensibilidade é a taxa de proteínas da classe analisada que foram corretamente classificadas sobre todas as proteínas desta classe. A Equação 2 define como a sensibilidade é calculada.

$$S = \frac{VP}{VP + FN} \tag{2}$$

A especificidade é a taxa das proteínas de outras classes classificadas corretamente sobre o número de classes classificadas em outras classes. A Equação 3 define como a especificidade é calculada.

$$E = \frac{VN}{VN + FP} \tag{3}$$

A acurácia é a taxa total de proteínas corretamente classificadas sobre todos os dados analisados. A Equação 4 define como a especificidade é calculada.

$$A = \frac{VP + VN}{VP + VN + FP + FN} \tag{4}$$

3.5 Resultados

Os resultados deste trabalho para o caso da nova codificação, foram piores do que o trabalho de Resende et al. (2012). Provavelmente a nova codificação gerou dados mais juntos, tornando mais difícil para a SVM achar uma boa margem separadora. Na maioria dos casos os dados foram classificados como pertencentes da classe Hidrolase, a mais numerosa.

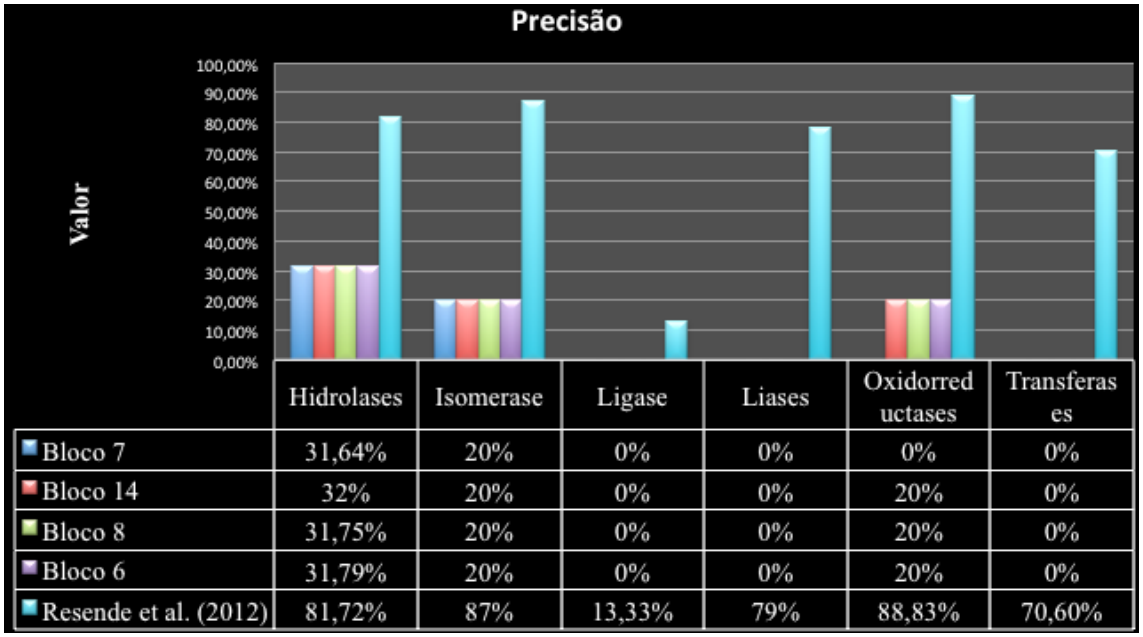
A Figura 11 mostra o resultado da precisão em que a hidrolase teve melhor eficiência, já que a maioria dos elementos desta classe foi classificada corretamente, mas não teve bom resultado no geral porque grande parte das outras classes também foram classificadas como hidrolases.

A Figura 12 mostra os resultados da sensibilidade. Como esta métrica analisa somente os elementos de cada classe, as hidrolases tiveram alta sensibilidade e as outras classes tiveram valores muito baixos.

A especificidade representa a habilidade de encontrar negativos, ou seja, o contrário da sensibilidade e por isso as hidrolases tiveram baixa eficiência e as outras classes tiveram valores altos. A Figura 13 ilustra esses resultados.

A Figura 14 mostra os baixos resultados da acurácia, que evidencia que a nova codificação não ajudou na separação dos pontos pela SVM.

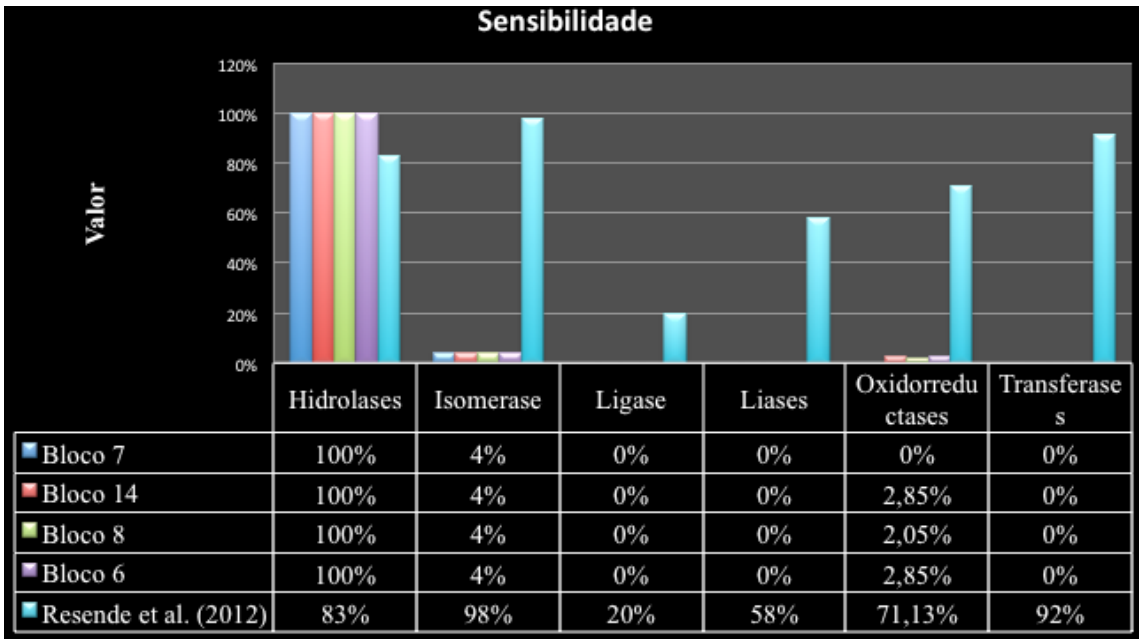
Figura 11 – Resultados da precisão para nova codificação.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Bloco 7	31,64%	20%	0%	0%	0%	0%
Bloco 14	32%	20%	0%	0%	20%	0%
Bloco 8	31,75%	20%	0%	0%	20%	0%
Bloco 6	31,79%	20%	0%	0%	20%	0%
Resende et al. (2012)	81,72%	87%	13,33%	79%	88,83%	70,60%

Fonte: Elaborada pelo autor.

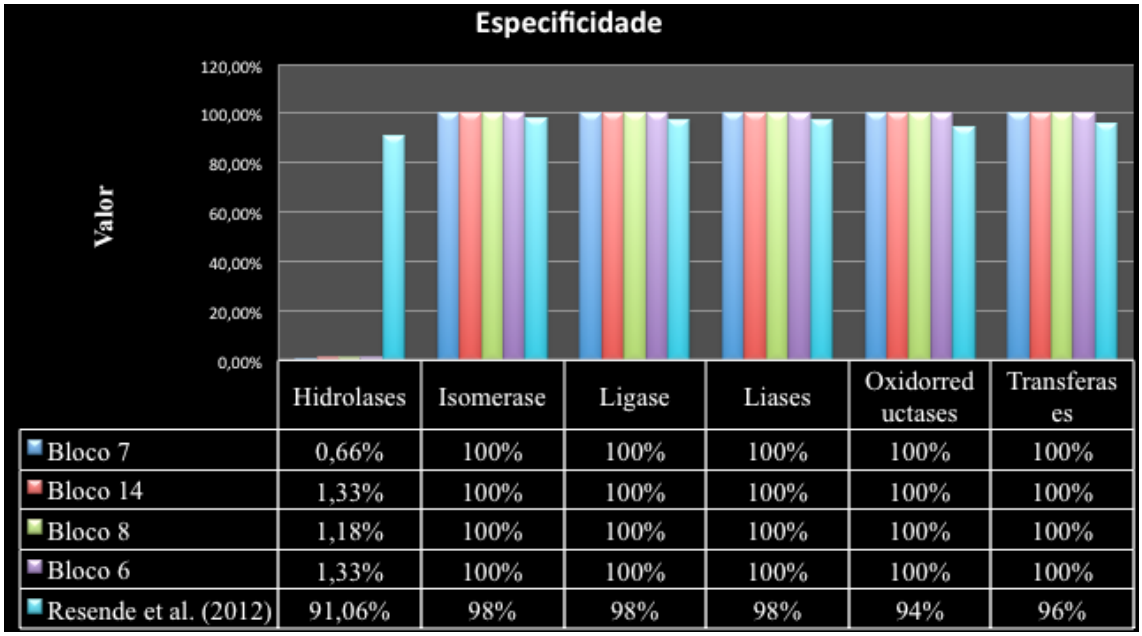
Figura 12 – Resultados da sensibilidade para nova codificação.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Bloco 7	100%	4%	0%	0%	0%	0%
Bloco 14	100%	4%	0%	0%	2,85%	0%
Bloco 8	100%	4%	0%	0%	2,05%	0%
Bloco 6	100%	4%	0%	0%	2,85%	0%
Resende et al. (2012)	83%	98%	20%	58%	71,13%	92%

Fonte: Elaborada pelo autor.

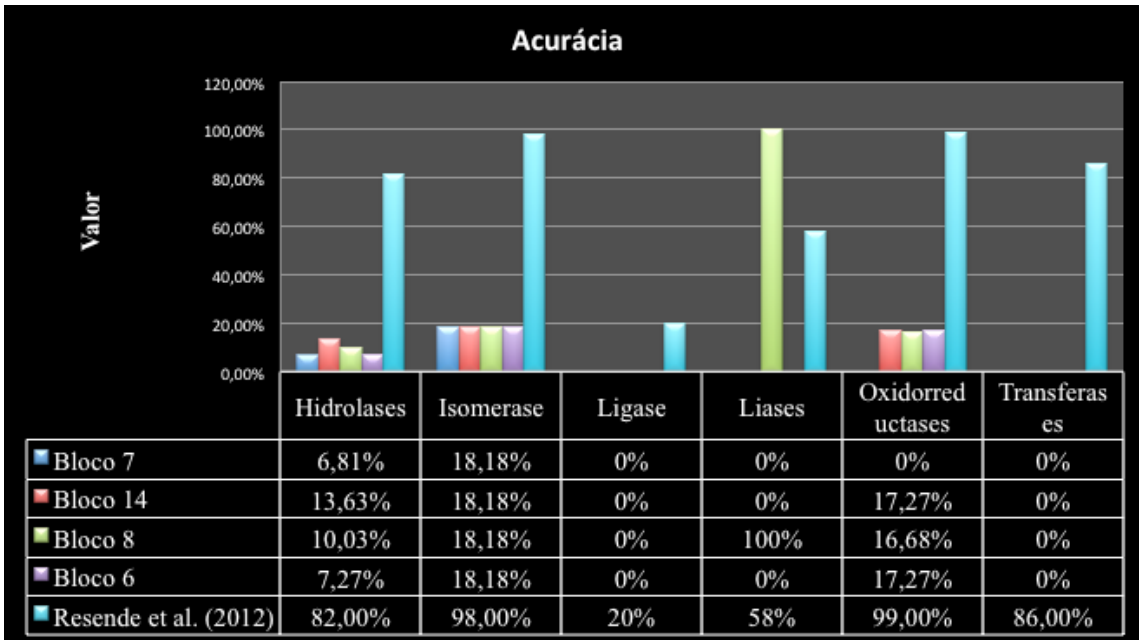
Figura 13 – Resultados da especificidade para nova codificação.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Bloco 7	0,66%	100%	100%	100%	100%	100%
Bloco 14	1,33%	100%	100%	100%	100%	100%
Bloco 8	1,18%	100%	100%	100%	100%	100%
Bloco 6	1,33%	100%	100%	100%	100%	100%
Resende et al. (2012)	91,06%	98%	98%	98%	94%	96%

Fonte: Elaborada pelo autor.

Figura 14 – Resultados da acurácia para nova codificação.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Bloco 7	6,81%	18,18%	0%	0%	0%	0%
Bloco 14	13,63%	18,18%	0%	0%	17,27%	0%
Bloco 8	10,03%	18,18%	0%	100%	16,68%	0%
Bloco 6	7,27%	18,18%	0%	0%	17,27%	0%
Resende et al. (2012)	82,00%	98,00%	20%	58%	99,00%	86,00%

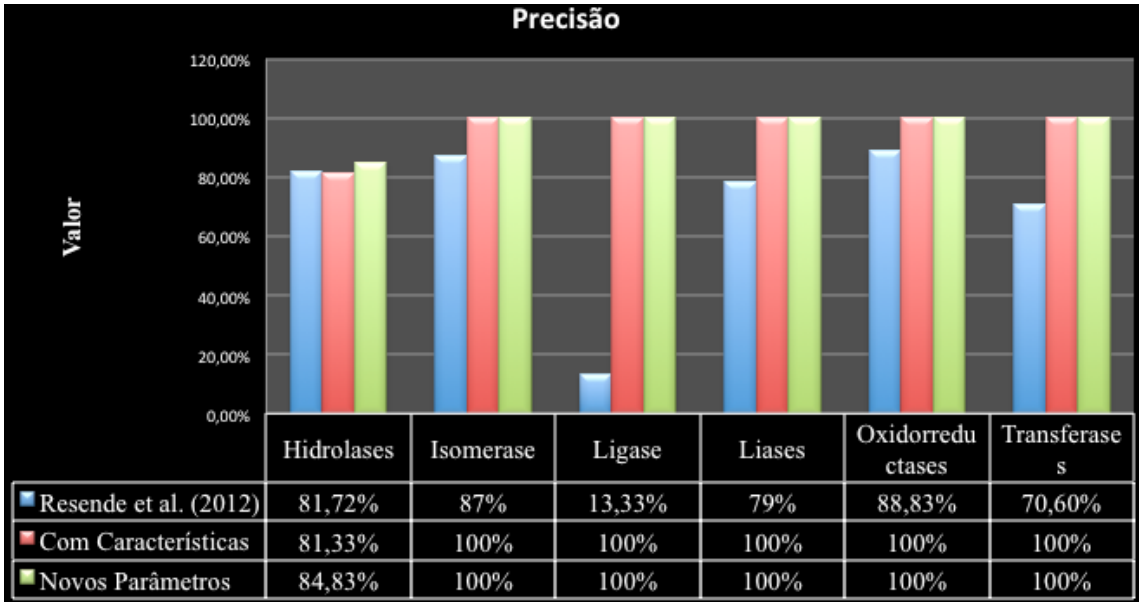
Fonte: Elaborada pelo autor.

Devido ao baixo desempenho da nova codificação, foi escolhido voltar a codificação antiga de Resende et al. (2012), adicionando as características. Obteve-se um resultado melhor do que os obtidos por Resende et al. (2012), principalmente para a classe das Ligases, que no trabalho original teve um desempenho ruim provavelmente por ser a classe com menos amostras.

Outro caso que vale mencionar é que testes iguais seguindo a mesma codificação de Resende et al. (2012), o algoritmo genético encontrou novos valores para os parâmetros da SVM (C = 15.244136 e gamma = 0.986328), que geraram resultados melhores. Provavelmente isso ocorreu porque o novo parâmetro C conseguiu ajustar melhor a margem separadora. Observando-se os resultados obtidos, percebe-se que os parâmetros da SVM são muito sensíveis, principalmente o C, que segundo Joachims (2002), se for muito pequeno, leva a muitos erros no treino e se for grande demais deixa a margem separadora difícil de ser encontrada.

As Figuras 16 e 17 mostram os resultados da sensibilidade e da especificidade, mostrando resultados próximos para todas as classes, evidenciando que a maioria foi classificada corretamente, diferente dos resultados da nova codificação. Os valores altos da precisão e da acurácia, ilustrados nas Figuras 15 e 18, também evidenciam isso mostrando a melhora em comparação com os resultados de Resende et al. (2012), principalmente para a classe das Ligases. Por outro lado, os resultados para o caso da adição de características e dos novos parâmetros da SVM foram muito próximos, mostrando que os parâmetros foram mais importantes para a classificação.

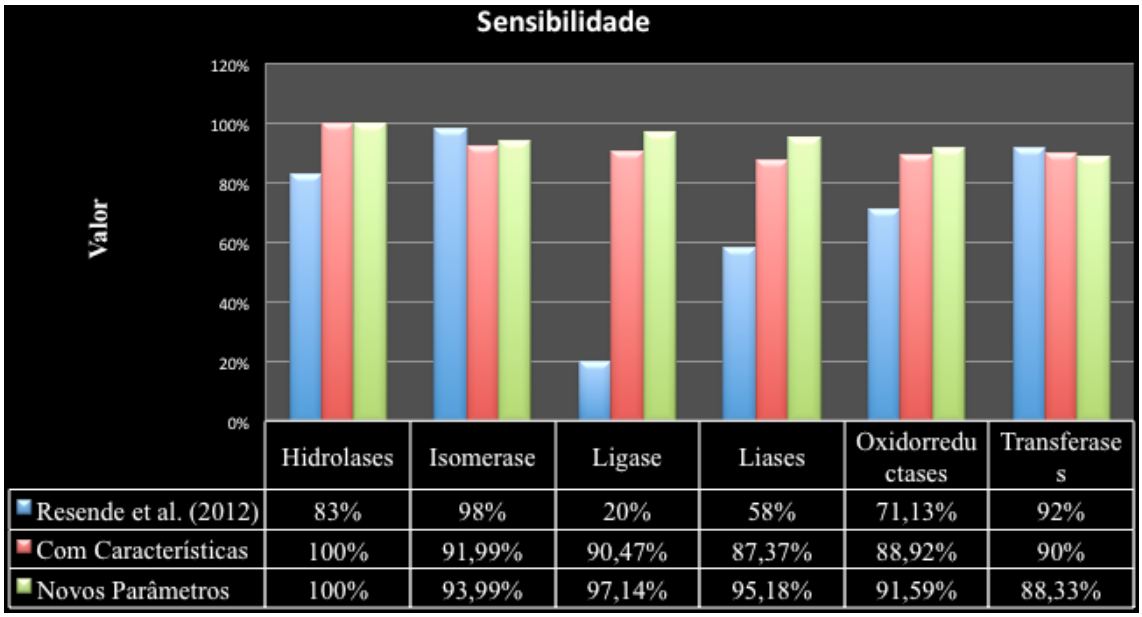
Figura 15 – Resultados da precisão adicionando características.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Resende et al. (2012)	81,72%	87%	13,33%	79%	88,83%	70,60%
Com Características	81,33%	100%	100%	100%	100%	100%
Novos Parâmetros	84,83%	100%	100%	100%	100%	100%

Fonte: Elaborada pelo autor.

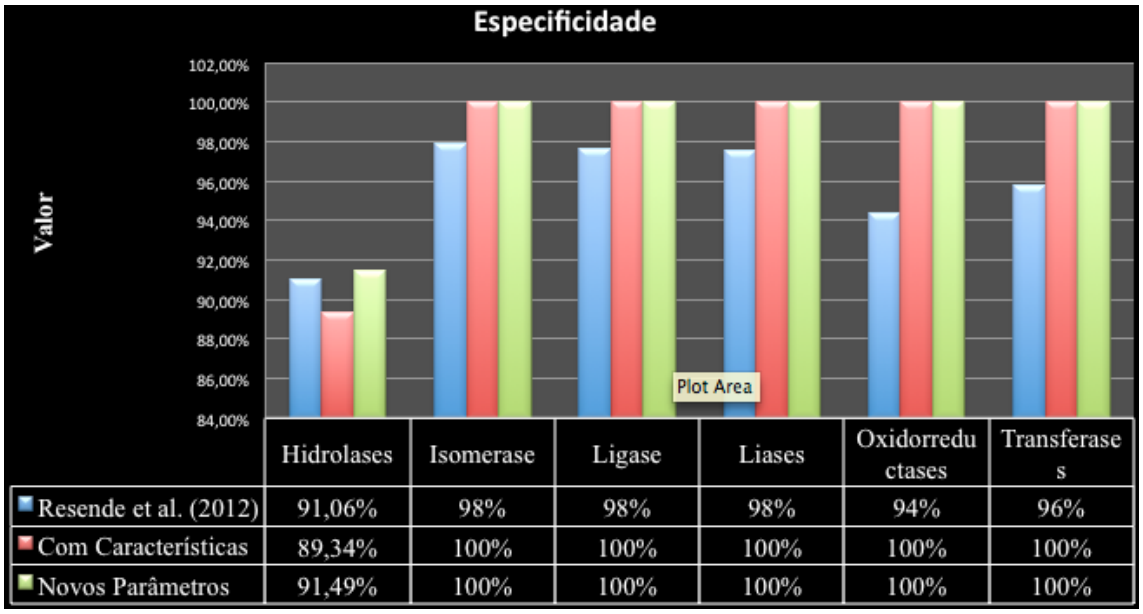
Figura 16 – Resultados da sensibilidade adicionando características.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Resende et al. (2012)	83%	98%	20%	58%	71,13%	92%
Com Características	100%	91,99%	90,47%	87,37%	88,92%	90%
Novos Parâmetros	100%	93,99%	97,14%	95,18%	91,59%	88,33%

Fonte: Elaborada pelo autor.

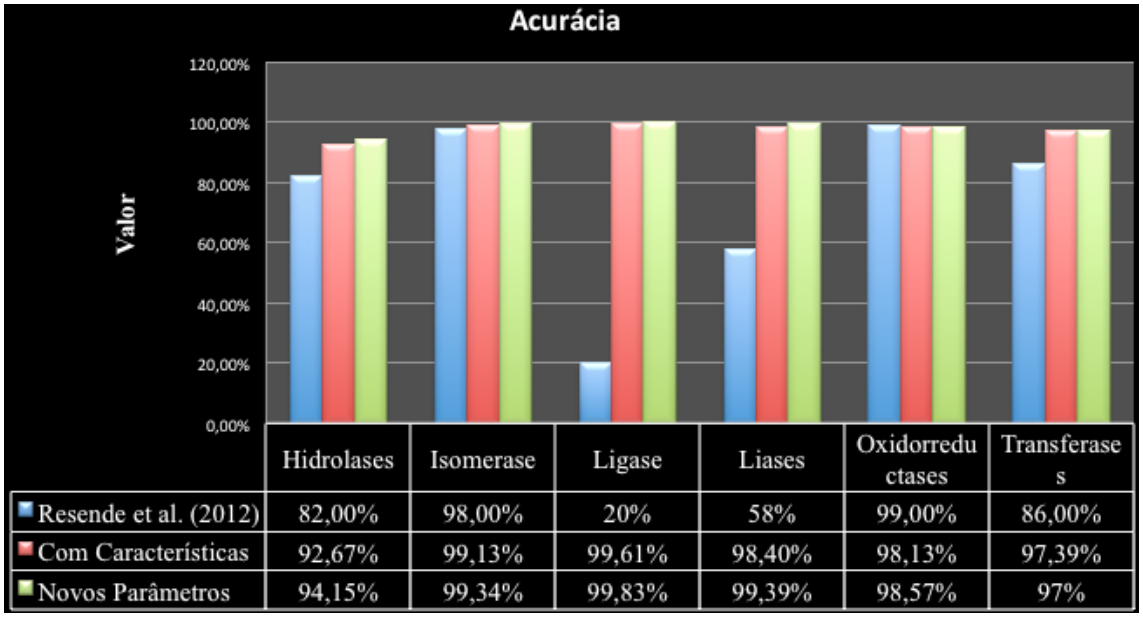
Figura 17 – Resultados da especificidade adicionando características.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Resende et al. (2012)	91,06%	98%	98%	98%	94%	96%
Com Características	89,34%	100%	100%	100%	100%	100%
Novos Parâmetros	91,49%	100%	100%	100%	100%	100%

Fonte: Elaborada pelo autor.

Figura 18 – Resultados da acurácia adicionando características.



	Hidrolases	Isomerase	Ligase	Liases	Oxidorreductases	Transferases
Resende et al. (2012)	82,00%	98,00%	20%	58%	99,00%	86,00%
Com Características	92,67%	99,13%	99,61%	98,40%	98,13%	97,39%
Novos Parâmetros	94,15%	99,34%	99,83%	99,39%	98,57%	97%

Fonte: Elaborada pelo autor.

4 CONCLUSÃO

Neste trabalho foi descrito um método para previsão de proteínas utilizando o algoritmo de máquina de vetor de suporte e dados da estrutura primária e secundária de enzimas extraídas do PDB. Foram realizadas alterações no *parser* de Resende et al. (2012) para extrair as características da estrutura (Tamanho da sequência, Número de resíduos hidrofóbicos, hidrofílicos, polares, com carga negativa e positiva). Também foi proposto uma nova codificação para sequência, intercalando a estrutura primária e secundária.

Como resultado, a nova codificação ficou com resultados ruins, provavelmente porque deixou os dados mais juntos, dificultando a possibilidade de se encontrar margens separadoras eficientes para a SVM. Por outro lado, a adição de características resultou em bons resultados melhorando muito o desempenho para no minimo 84,33% na precisão, 91,59% para sensibilidade, 91,49% na especificidade e 94,15% de acurácia.

Pode-se concluir com esse trabalho, que uma boa opção para se codificar a estrutura de proteínas que serão usadas por uma SVM é separar a estrutura primária e secundária. Também foi demonstrado que apenas as estruturas primárias e secundárias são suficientes para se ter um bom desempenho para prever funções, pelo menos para as classes de enzimas consideradas. Um ponto crítico é que os parâmetros da SVM são muito sensíveis e isso torna difícil a descoberta de valores que vão gerar resultados satisfatórios e por isso a adição de características pode ser um meio de contornar este problema, pois como demonstrado, pode ser mais fácil definir melhores margens separadoras.

Como trabalhos futuros, pretende-se estudar mais as características da estrutura de aminoácidos, descritas por Al-Shahib, Breitling e Gilbert (2007), para tentar melhorar os resultados. Também deve-se estudar se o método tem bom desempenho para proteínas de outras classes além das enzimas e novos métodos para descoberta de parâmetros da SVM.

REFERÊNCIAS

AL-SHAHIB, A.; BREITLING, R.; GILBERT, D. R. Predicting protein function by machine learning on amino acid sequences - a critical evaluation. *BMC Genomics*, v. 8, p. 78, March 2007.

BINKOWSKI, T. A.; ADAMIAN, L.; LIANG, J. Inferring functional relationships of proteins from local sequence and spatial surface patterns. *Mol. Biol.*, v. 332, p. 505–526, 2003.

BOCK, J. R.; GOUGH, D. A. Predicting protein–protein interactions from primary structure. *Bioinformatics*, January 2001.

BORGWARDT, K. M. et al. Protein function prediction via graph kernels. *Bioinformatics*, March 2005.

CAI, C. et al. Enzyme family classification by support vector machines. 2004.

DING, C. H. Q.; DUBCHAK, I. Multi-class protein recognition using support vector machines and neural networks. *Bioinformatics*, Novembro 2000.

FERRÈ, F. et al. Surface: a database of protein surface regions for functional annotation. *Nucleic Acids Researsh*, v. 32, p. 240–244, 2004.

GENNARO, J. A. D. et al. Enhanced functional annotation of protein sequences via the use of structural descriptors. *Journal of Structural Biology*, v. 134, p. 232–245, 2001.

HAYKIN, S. *Redes Neurais Princípios e Prática*. 2. ed. [S.l.]: Prentice Hall, 1998.

JOACHIMS, T. Multi-class support vector machine. 1999. Disponível em: <svmlight.joachims.org>.

JOACHIMS, T. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms (The Springer International Series in Engineering and Computer Science)*. [S.l.]: Springer, 2002.

KOLODNY, R.; LINIAL, N. Approximate protein structural alignment in polynomial time. *Proc. Natl Acad. Sci. USA*, v. 101, p. 12201, 12206 2004.

KRISSINEL, E.; HENRICK, K. Secondary-structure matching (ssm), a new tool for fast protein structure alignment in three dimensions. p. 2256–2268, 2004.

KUNIK, V. et al. Motif extraction and protein classification. *ACM*, Agosto 2005.

LEHNINGER, A. L.; NELSON, D. L.; COX, M. M. *Princípios de Bioquímica*. [S.l.]: Worth Publisher, 1999.

NASCIMENTO, R. A. S.; YOSHIOKA, S. R. I.; CALANZANS, T. C. Predição de funções protéicas através da análise conjunta das estruturas primárias e secundárias. *PUC-MG*, 2011.

NOBRE, C. N. Predição de função de proteínas a partir da análise conjunta das estruturas primárias e secundárias. *UFSJ*, 2011.

PANDEY, G.; KUMAR, V.; STEINBACH, M. Computational approaches for protein function prediction: A survey. *ACM*, October 2006.

PRINTS; SUPPLEMENT, p. its automatic. T k attwood and p bradley and d r flower and a gaulton and n maudling and a l mitchell and g moulton and a nordle and k paine and p taylor and a uddin and c zygouri. *Nucleic Acids Researsh*, v. 31, n. 1, p. 400–402, 2003.

RAGHAVA, G. *Evaluation of Bioinformatics Methods*. [S.l.], 2009.

RESENDE, W. K. et al. The use of support vector machine and genetic algorithms to predict protein function. *IEEE International Conference on Systems, Man, and Cybernetics*, 2012.

RUSSELL, S.; NORWIG, P. *Artificial Intelligence A Modern Approach*. 3. ed. [S.l.]: Prentice Hall, 2009.

SHATSKY, M.; NUSSINOV, R.; WOLFSON, H. J. A method for simultaneous alignment of multiple protein structures. *PROTEINS: Structure, Function, and Bioinformatics*, v. 56, p. 143–156, 2004.

SUZUKI, A. et al. Fcanal: Structure based protein function prediction method. application to enzymes and binding proteins. *Chem-Bio Informatics Journal*, v. 5, n. 3, p. 39–55, 2005.