PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Bacharelado em Ciência da Computação

Guilherme Pereira Gasparini Kingma

PREVISÃO DE FUNÇÃO DE ENZIMAS A PARTIR DA ANÁLISE DAS ESTRUTURAS DE AMINOÁCIDOS

Belo Horizonte
2013

Guilherme Pereira Gasparini Kingma

PREVISÃO DE FUNÇÃO DE ENZIMAS A PARTIR DA ANÁLISE DAS ESTRUTURAS DE AMINOÁCIDOS

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
2013

Guilherme Pereira Gasparini Kingma

PREVISÃO DE FUNÇÃO DE ENZIMAS A PARTIR DA ANÁLISE DAS ESTRUTURAS DE AMINOÁCIDOS

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

Fátima de Lima Procópio Duarte Figueiredo

Mark Alan Junho Song

Belo Horizonte, 6 de Dezembro de 2013

RESUMO

O número de proteínas conhecidas tem crescido bastante a cada dia, sendo necessário ter o conhecimento de suas funções para auxiliar, por exemplo, no desenvolvimento de fármacos e biocombustíveis. Porém, os métodos tradicionais de predição de suas funções envolvem testes laboratoriais e demandam muito tempo e custo. Este trabalho tem o objetivo de melhorar os resultados de uma metodologia já existente de predição de funções de proteínas da classe de enzimas, através da análise conjunta das estruturas primárias e secundárias, utilizando Máquinas de Vetor de Suporte (SVM). Para isso, foi utilizada a Transformada do Cosseno (TDC) para compactar as estruturas das proteínas de um *parser* de arquivos do *Protein Data Bank* (PDB). Os dados compactados servirão de entrada para a SVM. Os resultados da nova codificação apresentaram um modelo de predição mais consistente em comparação ao trabalho anterior, tendo um aumento de até 40% na precisão.

Palavras-chave: Função de proteínas. Transformada do Cosseno. SVM. Aprendizado de máquina.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DCT – *Discrete Cosine Transform*

EC – *Enzyme Commission number*

IDCT – *Inverse Discrete Cosine Transform*

IUBMB – *International Union of Biochemistry and Molecular Biology*

PDB – *Protein Data Bank*

PPI – Interações Proteína-Proteína

SVM – *Support Vector Machine*

TDC – *Transformada Discreta do Cosseno*

SUMÁRIO

1 INTRODUÇÃO

A cada ano, o número de proteínas conhecidas tem crescido muito, devido ao rápido crescimento do projeto genoma (ALVAREZ; YAN, 2010). Com isso, os métodos tradicionais para a predição de função de proteínas, envolvendo testes laboratoriais de cristalografia de Raio X apresentam uma complexidade muito grande, demorando muito tempo para obter um resultado, embora este seja muito preciso (RESENDE et al., 2012). Uma boa alternativa para solucionar este problema de elevado custo de tempo e pouco rendimento é fazer o uso de soluções computacionais que analisam a estrutura das proteínas conhecidas, para então prever as funções das desconhecidas, reduzindo a necessidade de testes laboratoriais (SOARES, 2012).

Resende et al. (2012) propuseram um método para extrair o conteúdo de estruturas primárias e secundárias de enzimas retiradas do PDB (*Protein Data Bank*), utilizando o classificador SVM (*Support Vector Machine*) para prever a função de cada amostra, podendo ser Oxidorredutases, Transferases, Hidrolases, Liases, Isomerases ou Ligases. Soares (2012) continuou o trabalho, inserindo no classificador algumas características químicas de cada aminoácido, melhorando assim a predição.

O trabalho atual apresenta uma nova metodologia utilizando o classificador SVM para prever as funções do mesmo conjunto de amostras de enzimas utilizadas nos trabalhos anteriores, tendo como objetivo obter melhores resultados. Para isso, será proposto uma redução dos vetores de entrada, de forma a treinar apenas os valores mais relevantes, esperando-se ter uma melhora na precisão do novo modelo.

1.1 Objetivos

1.1.1 Objetivos Gerais

Este trabalho tem como objetivo propor uma nova metodologia para prever funções de proteínas das classes de enzimas, continuando o trabalho de Soares (2012), de forma a melhorar os atuais resultados.

1.1.2 Objetivos Específicos

- Implementar uma TDC (Transformada Discreta do Cosseno) de modo a compactar vetores.
- Propor uma modificação no *parser* original de Resende et al. (2012) e Soares (2012), aplicando a TDC para compactar os vetores, reduzindo a entrada para o classificador SVM.

1.2 Justificativa

O estudo das estruturas e das funções de proteínas é muito utilizado tanto no aprendizado de processos biológicos, como também no desenvolvimento de novos medicamentos e biocombustíveis (DUBCHAK et al., 1995). Devido ao crescente avanço do projeto genoma, muitas proteínas ainda não tem funções conhecidas. Tais funções normalmente são descobertas através de testes laboratoriais muito lentos. Torna-se então necessário o desenvolvimento de ferramentas automatizadas, com um custo muito mais baixo, que analisam a estrutura das proteínas para então obter uma previsão de sua função.

1.3 Organização do trabalho

No Capítulo 2, será apresentado a revisão da literatura, com o referencial teórico e trabalhos relacionados ao tema. O Capítulo 3 contém a metodologia utilizada neste trabalho, enquanto os resultados foram separados no Capítulo 4. No Capítulo 5 estão as conclusões e considerações finais.

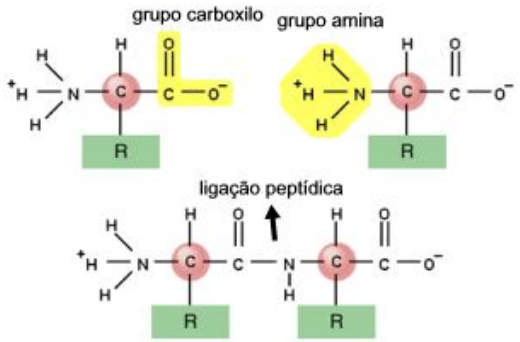
2 REVISÃO DA LITERATURA

2.1 Referencial Teórico

2.1.1 Conceitos de Biologia

As proteínas são constituintes fundamentais para a existência da vida terrestre. São macromoléculas que exercem funções estruturais e enzimáticas (catalizadora de reações químicas) na célula, além da manutenção do meio celular (PANDEY; KUMAR; STEINBACH, 2006). Cada uma é composta por uma sequência única de aminoácidos, podendo se diferenciar de outras pela quantidade, tipos, ou pela ordem dos aminoácidos na proteína (AMABIS; MARTHO, 2001). A sequência é composta por dezenas a centenas de aminoácidos, que são compostos por um carbono que fica no centro da molécula, chamado de carbono-alfa, que se liga a quatro grupos: Amina, carboxila, hidrogênio, e um radical que define a identidade do aminoácido. Ele se liga a outro através de uma ligação peptídica, ocorrendo na reação química entre o grupo amina de um aminoácido com o grupo carboxila de outro (FARAH, 2007). Nesta reação, o grupo amina perde um de seus hidrogênios, enquanto a carboxila perde a hidroxila, liberando assim, uma molécula de água (síntese por desidradação). A Figura 1 ilustra uma reação peptídica entre dois aminoácidos.

Figura 1 – Ligação Peptídica



Fonte: Penaforte (2012)

Segundo Nelson e Cox (2008), as estruturas das proteínas podem ser descritas em quatro níveis de complexidade:

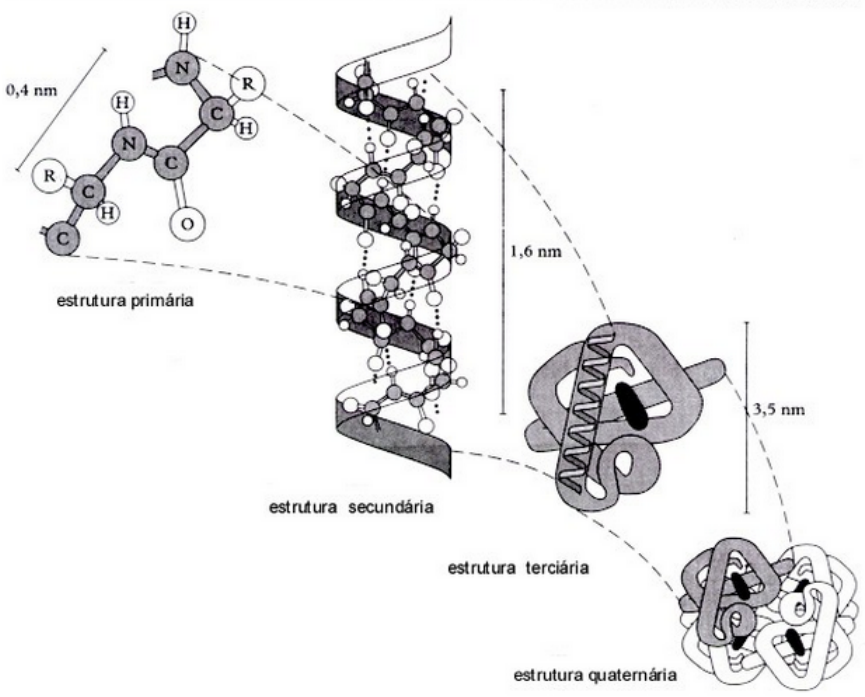
- **Estrutura Primária:** Determina a ordem dos aminoácidos em uma cadeia polipeptídica, desconsiderando sua disposição espacial;

- **Estrutura Secundária:** Representa as interações químicas entre os aminoácidos da estrutura primária da proteína, podendo se interagir formando alfa-hélices, folhas-beta, e/ou *loops*.

- **Estrutura Terciária:** Além das interações na estrutura secundária, a cadeia polipeptídica dobra-se nela mesma, formando a estrutura terciária. Isso ocorre devido a atração entre diferentes partes da proteína, bem como a atração e repulsão dos aminoácidos para com as moléculas de água no ambiente (AMABIS; MARTHO, 2001).

- **Estrutura Quaternária:** É formada quando duas ou mais cadeias polipeptídicas se interagem pelos mesmos tipos de interações químicas apresentadas na estrutura terciária.

Figura 2 – Estrutura das Proteínas



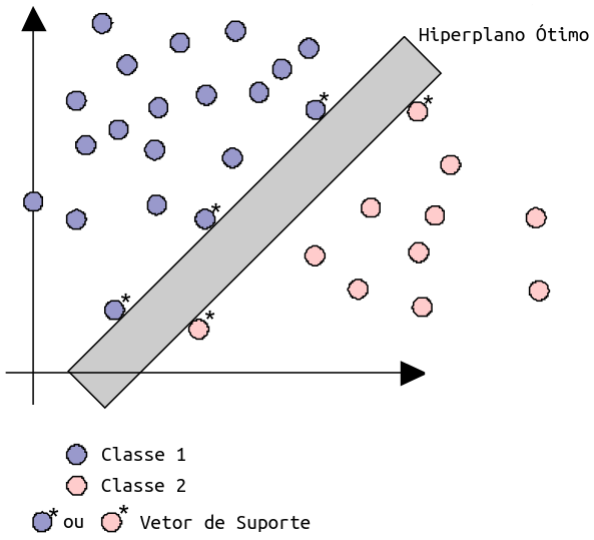
Fonte: Alten (2005)

2.1.2 Máquinas de Vetor de Suporte

A SVM tem como função analisar um conjunto de dados e reconhecer padrões, utilizando aprendizado de máquina, sendo muito usada nos casos em que não há um conhecimento prévio sobre o domínio da aplicação. Para cada entrada de um conjunto de treino, o SVM reconhece como elas se diferem entre si, construindo um modelo para poder predizer em qual classe pertence cada entrada de teste.

A SVM apresenta 3 características para que isso seja possível (RUSSEL; NORWIG, 2009). São elas: 1) a SVM constrói uma fronteira com a maior distância possível entre os pontos de treino (também chamada de margem separadora) e isto ajuda a fazer uma boa generalização; 2) a SVM cria um hiperplano que colocam dados que não são linearmente separáveis em dimensões maiores, para que possam ser facilmente separados pela margem separadora; 3) é um método flexível para representar funções complexas e é resistente sobre ajuste (quando o modelo tende a se ajustar em demasiado às entradas de treino). A Figura 3 mostra uma hiperplano ótimo de duas dimensões separando duas classes da SVM.

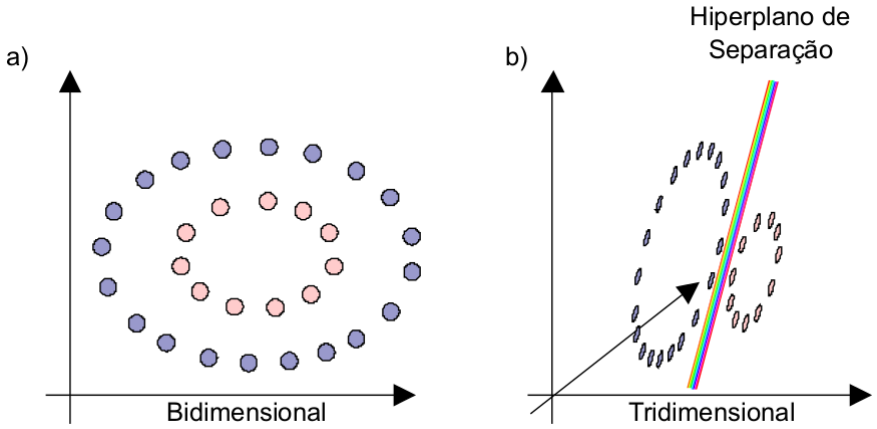
Figura 3 – Vetores de suporte



Fonte: Dias (2007)

Existem classes que não são linearmente separáveis, ou seja, não são possíveis de ser separadas pela margem separadora. Porém, isso é possível ao representar esses dados em um domínio com mais dimenções. A Figura 4 mostra como é possível obter um hiperplano ótimo com classes não linearmente separáveis em 2 dimensões.

Figura 4 – Classes não linearmente separáveis



Fonte: Dias (2007)

2.1.3 Tranformada Discreta do Cosseno

A Transformada Discreta do Cosseno (TDC) é uma técnica muito utilizada na compressão de dados, aplicada tanto em imagens e vídeos como em sequências de proteínas (DIAS, 2007). O propósito da TDC é transformar uma sequência de dados em outra, fazendo com que os coeficientes mais significativos fiquem acumulados em uma quantidade muito pequena no início do vetor de saída, enquanto o restante armazena valores irrelevantes, considerados como ruído. (AHMED; NATARAJAN; RAO, 1974). Por ser uma função $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, os n números reais são transformados em n coeficientes reais, como mostrado abaixo:

$$x_0, x_1, ..., x_n \rightarrow c_0, c_1, ..., c_n$$

A DCT possui a propriedade de ortonormalidade, ou seja, existe uma uma função inversa C^{-1} tal que $C^{-1}(c_n) = x_n$. Assim, para obter o vetor original de um vetor já transformado, basta aplicar a inversa da Transformada do Cosseno.

A DCT unidimensional de uma função $f(x)$ pode ser calculada pela Equação 2.1, e a sua inversa é dada pela Equação 2.2 (KHAYAM, 2003).

$$C(k) = \alpha(k) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)k}{2N}\right] \tag{2.1}$$

$$f(x) = \sum_{k=0}^{N-1} \alpha(k) C(k) \cos\left[\frac{\pi(2x+1)k}{2N}\right] \tag{2.2}$$

Onde N é o tamanho do vetor a ser transformado, $\alpha(k) = \frac{1}{\sqrt{N}}$ para $k = 0$ e $\alpha(k) = \sqrt{\frac{2}{N}}$ para $k \neq 0$.

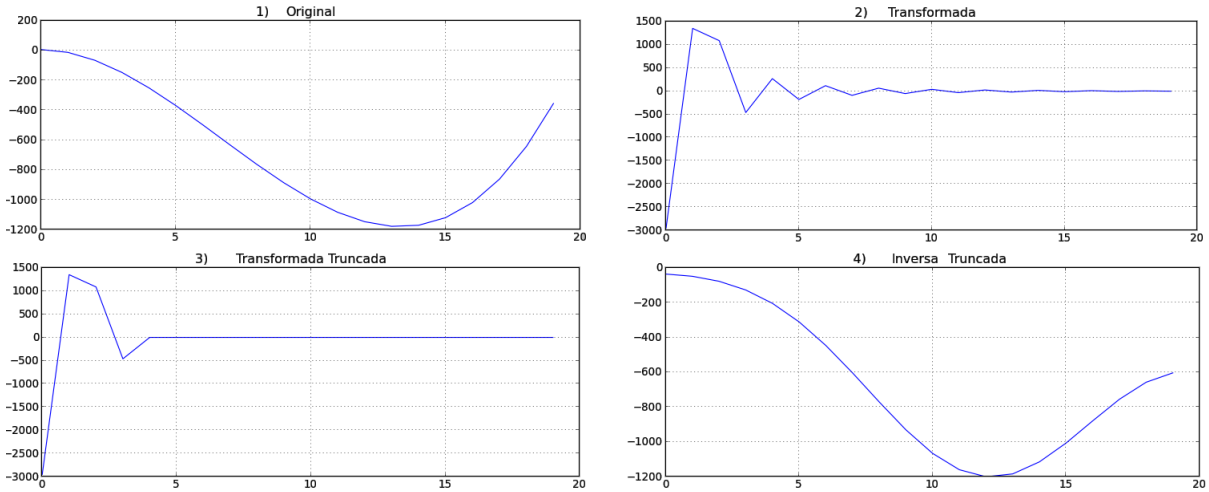
Na Equação 2.1, para $k = 0$, $C(k = 0) = \sqrt{\frac{1}{k}} \sum_{x=0}^{N-1} f(x)$. Assim, o primeiro coeficiente da transformada é o valor médio da sequência da amostra. Na literatura, este valor é chamado de coeficiente DC. Todos os outros coeficientes de transformação são chamados de coeficientes AC.

Formado o vetor transformado, é feita uma operação de quantização nos dados gerados, de forma a ser armazenado somente os dados quantizados, eliminando os valores menos significativos (DIAS, 2007) (Ahmed et al., 1974). Os tipos de quantizações que podem ser usadas nesta parte são:

- Realizar a divisão dos valores por um coeficiente de quantização fixo, utilizando menos bits para representar os valores.
- Eliminar os componentes menos significativos, determinando uma margem de valor para o truncamento dos vetores. Pode-se excluir estes componentes, ou mesmo igualá-los a zero.

O trabalho atual utilizou este último método, analisando somente os componentes mais significativos. A Figura 5 mostra os gráficos: 1) da função $f(x) = x^3 - 20x^2 + 6$; 2) de sua transformada; 3) de sua transformada truncada com o tamanho 4, e finalmente, 4) sua inversa.

Figura 5 – Gráfico - Transformada Discreta do Cosseno



Fonte: Elaborado pelo autor.

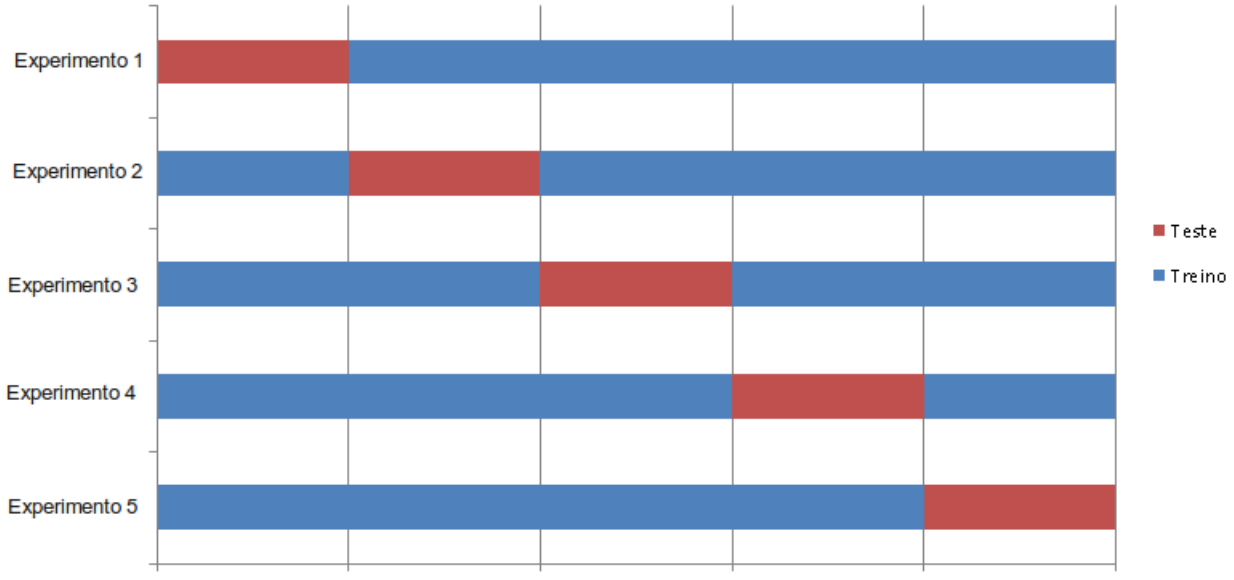
2.1.4 Técnicas de Amostragem

As técnicas de amostragem são muito utilizadas para avaliar a capacidade de generalização modelos de predição na área de Inteligência Artificial. Sua função é particionar o conjunto total dos dados em conjuntos distintos, utilizando alguns destes subconjuntos para o treino do modelo, e o restante para testes. Russel e Norwig (2009) apresentaram três formas de se particionar o conjunto de dados:

- ***Hold Out:*** Neste método, o conjunto de dados é dividido em dois subconjuntos, de tamanhos iguais ou não, um para o treino do modelo e outro para teste. Muito utilizado em um conjunto de dados muito grande, sendo que em dados pequenos, a taxa de erro pode sofrer muita variação.

- ***k-fold Cross Validation:*** Consiste em dividir o conjunto de dados em um número pré-definido (k) de subconjuntos distintos e de tamanhos iguais. Um desses subconjuntos é utilizado para o teste, e os $k - 1$ restantes são concatenados e usados como treino. Em seguida troca-se os subconjuntos e o processo é repetido k vezes, como mostra a Figura 6. A avaliação do modelo é feita calculando a média entre todos os k resultados. No presente trabalho, foi utilizado um *k-fold* de tamanho 10.

Figura 6 – *k-fold Cross Validation*



Fonte: Elaborado pelo autor.

- ***Leave-One Out Cross Validation:*** Este método é semelhante ao *k-fold Cross Validation*, porém utiliza-se um elemento do conjunto para teste enquanto o resto é usado para o treino, ou seja, o valor de k neste caso será igual a n, sendo n o número total de amostras. Possui um custo computacional elevado, sendo indicado em modelos com poucos dados.

2.2 Trabalhos Relacionados

Existem basicamente quatro abordagens utilizadas na predição de funções de proteínas:

- **Predição baseada em ferramentas de alinhamento de proteínas:** Calcula o nível de similaridade entre proteínas utilizando técnicas de alinhamento de suas estruturas.
- **Predição baseada na própria sequência:** Calcula a função das proteínas com base em suas próprias estruturas (primárias, secundárias, terciárias ou quaternárias).
- **Predição baseada em características de aminoácidos:** Calcula a função das proteínas a partir das características dos aminoácidos da proteína.
- **Abordagem Mista:** Utiliza duas ou mais abordagens citadas acima para a predição de proteínas.

2.2.1 Predição baseada em ferramentas de alinhamento de proteínas:

Kolodny e Linial (2004) apresentaram um método de alinhamento de proteínas para comparar estruturas, buscar por similaridades, e assim encontrar relações evolucionárias distantes, que normalmente são difíceis ou impossíveis de encontrar. Os autores otimizaram o alinhamento desenvolvendo um algoritmo de complexidade polinomial de $O(\frac{n^{10}}{e^6})$, onde n é o tamanho da proteína que está no máximo a uma distância e do ótimo.

Shatsky, Nussinov e Wolfson (2004) apresentaram *MultiProt*, uma ferramenta automatizada altamente eficiente na detecção de alinhamentos múltiplos nas estruturas de proteínas, que foram retiradas do PDB. O método utiliza um algoritmo que detecta fragmentos estruturalmente semelhantes de comprimento máximo, e encontra os núcleos geométricos comuns entre as moléculas de entrada. O experimento apresenta um conjunto de 10 pares de proteínas e seus

alinhamentos realizados por diferentes métodos. A solução é detectada para qualquer número de moléculas, podendo aplicar a várias cadeias de proteínas. Foi concluído que a metodologia utilizando *MultiProt* tem a habilidade de detectar semelhanças que não tem a ver com a forma estrutural da proteína.

2.2.2 Predição baseada na própria sequência:

Shen et al. (2006) desenvolveram um método para realizar a predição de PPI (Interações Proteína-Proteína), princípio importante para a maioria dos processos biológicos. Por exemplo, é medido os sinais de uma célula para seu exterior ou interior através de PPIs. Utilizando SVM, foram utilizadas somente as informações das sequências primárias das proteínas. Este método pode ser aplicado para a predição de qualquer proteína recentemente descoberta, aumentando ainda mais a precisão do modelo. Em média, o método pode produzir um modelo de previsão de PPI com uma exatidão de 83,90%.

Wang et al. (2011) apresentaram duas melhorias na previsão de função de enzimas. A primeira utilizou métodos eficientes de codificação de sequências das proteínas dadas. A segunda desenvolveu um método de previsão baseado em estrutura, com uma baixa complexidade computacional. Foi desenvolvido um método para prever funções de enzimas considerando não só a composição dos aminoácidos, como também as relações vizinhas na sequência. O resultado com estes três atributos foram comparados, e concluiu-se que a informação de todos juntos oferece melhores resultados. Para isto, foi desenvolvido um SVM, obtendo como resultado uma taxa de precisão entre 81% e 98%, dependendo da classe analisada. Com isso, foi demonstrado que o método apresentado supera os métodos que não levam em conta as relações vizinhas entre as sequências.

Resende et al. (2012) continuaram o trabalho de Nascimento, Yoshioka e Calanzans (2011). Foi utilizado o mesmo *parser*, que tinha como função pegar as estruturas primárias e secundárias das proteínas do PDB e inserir zeros para normalizar o tamanho das sequências. Desta vez, foram coletadas amostras de 6 classes de função de enzimas, as Oxidoreductases, Transferases, Hydrolases, Lyases, Isomerases e Ligases, classificadas com a SVM de Joachims (1999). Também foi utilizado algoritmos genéticos para ajustar os parâmetros do classificador SVM. Como resultado, o algoritmo obteve uma precisão média de 79,74%, sendo de 82% a

99% para 4 classes, 80% para Lyases e apenas 20% para Ligases.

Alvarez e Yan (2010) desenvolveram um sistema de distinção entre enzimas e não enzimas, utilizando a mesma base de proteínas de Dobson e Doig (2003). Foi utilizado grafos para modelar as estruturas das proteínas. Cada vértice representa um grupo de resíduos vizinho, contendo também informações sobre sequências homólogas. O trabalho apresentou três estratégias simples e diferentes de modelagem das estruturas de proteínas: *binning*, *PCA-binning* e *clustering*. Assim, utilizou-se o classificador SVM para a diferenciação entre enzimas e não enzimas, usando *k-fold Cross Validation* com 10 subconjuntos, obtendo uma acurácia de 84,31% a 86,97%.

2.2.3 Predição baseada em características de aminoácidos

Montuori, Raimondo e Pasero (2008) apresentaram os resultados selecionados para um subconjunto ótimo de características para a previsão de função de proteínas, utilizando *backward selection*, ou seja, a análise começa com um conjunto de todos as características e elimina progressivamente as menos promissoras. Esta análise é importante, porque quando aumenta o número de características de entrada para a SVM, pode aumentar até exponencialmente o tempo de execução, além de muitos dados serem irrelevantes, ou mesmo redundantes na predição. As características podem ser divididas em diferentes subgrupos, dependendo do uso. Neste trabalho, os subgrupos considerados foram interações atômicas, acessibilidade de solvente e conteúdo da estrutura secundária, num total de 109 características estruturais. Os autores utilizaram o método de filtro, ou seja, utilizam uma métrica de desempenho baseado somente nos dados de treino, e as características são filtradas antes do sistema de classificação ser treinado e testado. Os melhores desempenhos foram obtidas com o conjunto de 6 características: 1) semelhança na estrutura secundária; 2) porcentagem de ligações nitrogênio-nitrogênio com distância de 3,5A; 3) porcentagem de ligações carbono-oxigênio/enxofre com distância máxima de 5A; 4) porcentagem de ligações carbono-oxigênio/enxofre com distância abaixo de 3,5A; 5) porcentagem de ligações oxigênio/enxofre-nitrogênio com distância máxima de 7,5A e 6) porcentagem de aminoácidos de superfície, com acessibilidade de superfície inferior a 41,5%.

Dobson e Doig (2003) desenvolveram um sistema de distinção entre enzimas e não enzimas, utilizando SVM e uma base de dados com 1178 proteínas retiradas do PDB (691 enzimas e

487 não enzimas). Para isso, foram analisadas o conteúdo da estrutura secundária, bem como o número e porcentagem de aminoácidos, o número de ligações dissulfureto e o tamanho da maior fissura. Todas as características não binárias foram normalizadas em um intervalo entre 0 e 1, tal como recomendado em SVMs. Foi obtido como resultado uma precisão de 77% utilizando 52 características para descrever cada proteína. Uma pesquisa de subconjuntos possíveis de características produz um modelo simplificado de 36 características, com uma precisão de 80%. O método desenvolvido foi comparado com outros métodos que também não utilizam alinhamento de sequências, conseguindo prever um conjunto de proteínas recém descoberto.

Al-Shahib, Breitling e Gilbert (2007) mostraram que proteínas de funções desconhecidas se diferem muito mais das de funções conhecidas. Para isso, utilizaram uma SVM do conjunto de ferramentas WEKA (HALL et al., 2009) para classificar proteínas de sete bactérias que causam doenças sexualmente transmissíveis, mostrando que proteínas de espécies diferentes se diferem mais do que as de mesma espécie. As sequências de proteínas foram retiradas de *Los Alamos National Laboratory Bioscience Division STD Sequence Databases*. Para cada proteína, foi calculado o número total e a frequência de cada aminoácido. A estrutura secundária, a posição das hélices, e as regiões desordenadas foram utilizadas para a previsão, bem como algumas características como ponto isoelétrico e peso molecular. 2579 características de proteínas foram extraídas e listadas. Foi aplicado BLAST (ALTSCHUL et al., 1990) para definir quais proteínas mostraram uma maiores semelhanças entre si. Para cada par aleatório de proteínas de diferentes funções, o sistema é capaz de predizer as funções corretas em 85% dos casos. Assim, os autores concluíram que métodos de aprendizado de máquina utilizando a estrutura da proteína possuem um bom desempenho, tendo previsões melhores em algumas classes, como metabolismo intermediário, metabolismo de DNA e transporte de proteínas de ligação.

2.2.4 Abordagem Mista

Dubchak et al. (1995) utilizaram as características de hidrofobicidade, solubilidade, e estrutura secundária para servirem de parâmetros para a Rede Neural, fazendo com que esta seja capaz de determinar quando uma proteína pertence a uma determinada classe. A Rede Neural foi treinada com em base de dados com 83 classes. Testes de validação cruzada foram

realizados em 15 das maiores classes. O trabalho teve previsões de verdadeiro positivo em 71.7% dos testes, enquanto as de verdadeiro negativo variaram entre 90% a 95%.

Cai et al. (2003) fizeram seu trabalho utilizando análises nas estruturas primárias e secundárias, bem como 7 características químicas: hidrofobicidade, polaridade, polarizabilidade, carga, tensão superficial, forças de Van der Waals e acessibilidade de solvente de cada resíduo. As análises foram feitas em classes diferentes de proteínas, tais como proteínas de ligação ao RNA, homodímeros de proteínas, proteínas responsáveis pela absorção, distribuição e excressão de drogas, e enzimas metabolizadoras de fármacos. Foi utilizado o classificador SVM, e os resultados da precisão variaram entre 86,5% e 99,4%, sugerindo a aplicação do SVM para a classificação de classes funcionais de proteínas, bem como o uso na predição da função destas.

Cai et al. (2004) realizaram o trabalho anterior testando a SVM em enzimas classificadas em famílias funcionais definidas pelo IUBMB (International Union of Biochemistry and Molecular Biology), um padrão de nomenclatura de enzimas. Este trabalho realizou testes em 8291 enzimas coletadas do banco UniProt[1] provenientes de 46 familias do banco BRENDA (SCHOMBURG, 2009). Como resultado, o trabalho obteve uma precisão de 80,03%.

Soares (2012) continuou o trabalho de Resende et al. (2012). Foram alteradas as entradas do SVM de Joachims (1999), inserindo algumas características químicas da proteína como parâmetros no classificador SVM. Foi obtido resultados melhores, de, no mínimo, 84,33% na precisão, 91,59% na sensibilidade, 91,49% na especificidade e 94,15% na acurácia. Com isso, notou-se a necessidade de adicionar características da estrutura de aminoácidos para ser mais fácil definir as margens separadoras, tendo assim, resultados mais satisfatórios. Foi observado também que, ao executar a codificação de Resende et al. (2012) novamente, foi obtido resultados melhores que os antigos, pois o algoritmo genético gerou novos valores para os parâmetros da SVM que conseguiu ajustar melhor a margem separadora.

[1]Disponível em: http://www.ebi.ac.uk/uniprot

3 METODOLOGIA

Neste trabalho foi retirado do PDB, Banco de Dados de Proteínas, a mesma base de proteínas da subclasse de enzimas do trabalho de Resende et al. (2012) e Soares (2012), também analisando as informações das estruturas primárias e secundárias das proteínas. Para aumentar a precisão, foi escolhido aplicar a Transformada do Cosseno nas estruturas, reduzindo a dimensão dos vetores de suporte e diminuindo o processamento da SVM.

As classes de enzimas foram enumeradas de acordo com o seu EC *Number*, número de comissão de enzimas, que é usado para suas classificações de acordo com suas propriedades, que são: Oxidorredutases, Hidrolases, Liases, Ligases, Isomerases e Transferases. A Tabela 1 apresenta o grupo de dados utilizados neste trabalho, com suas descrições e quantidades.

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 hidrólise (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 transferência de grupos dentro da mesma molécula para formar isômeros.	51
6	Ligases	Cataliza a ligação do tipo C-C, C-O, e C-N.	17

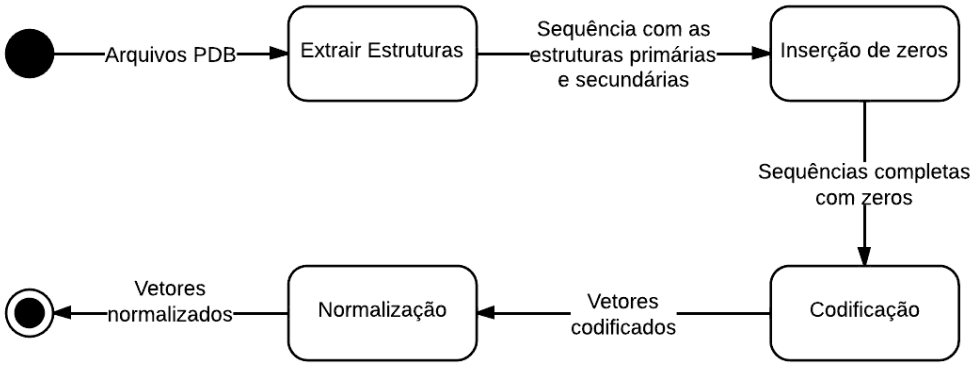
Fonte: Resende et al. (2012)

3.1 *Parser* Original

Resende et al. (2012) e Soares (2012) dividiram o *parser* em 4 etapas: 1) Extração das estruturas primárias e secundárias dos arquivos PDB; 2) Inserção de zeros para que os vetores de suporte tenham o mesmo tamanho; 3) Codificação final, implementando uma tabela

hash para codificar partes da sequência, reduzindo assim, seu tamanho e processamento. Nesta etapa, Soares (2012) concatenou algumas características das proteínas no final dos vetores; 4) Normalização do arquivo final, atribuindo todos os valores no intervalo de [0,...,1]. A última posição de cada vetor armazena a classe da proteína correspondente. A Figura 7 ilustra as fases do *parser*.

Figura 7 – Diagrama de Atividades do Parser Original



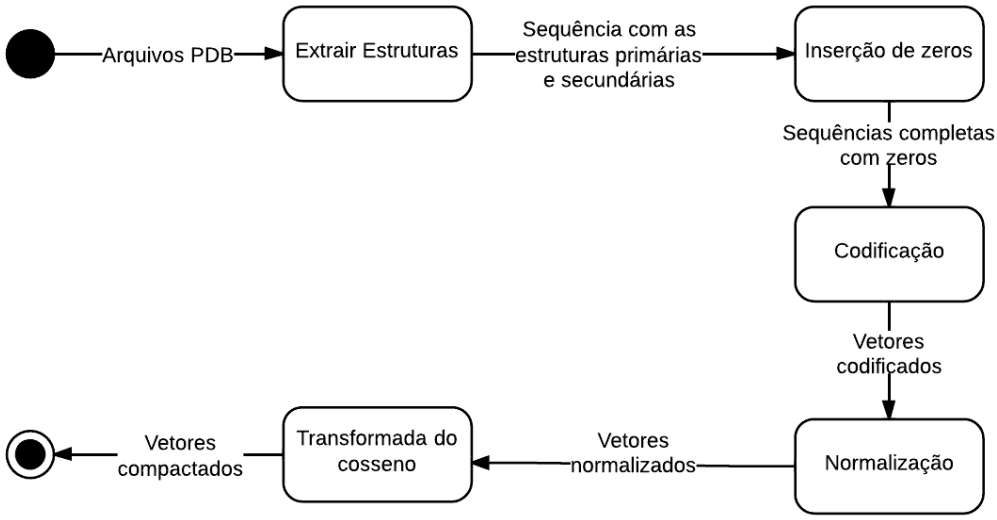
Fonte: Adaptado de Soares (2012)

A estrutura primária é codificada como sendo cada um dos 20 aminoácidos representado por um número de 1 a 20. A estrutura secundária é representada pelos números 1 (α-*Helix*), 2 (β-*Sheet*) ou 0 (Nenhuma das duas).

3.2 Modificação do *Parser*

Para alterar o *parser* de modo a reduzir o tamanho dos vetores, foi realizada a Transformada do Cosseno com os vetores gerados pela fase de Normalização. Foi escolhido este método a fim de armazenar apenas os atributos mais relevantes, fazendo não só com que os vetores fossem compactados, como também mantendo-os do mesmo tamanho, compatíveis com a SVM. Como a TDC elimina grande parte das informações que não são relevantes, espera-se uma melhoria na predição do novo modelo. A Figura 8 ilustra o novo *parser*, seguido pelas descrições de cada etapa:

Figura 8 – Diagrama de Atividades do Parser Modificado



Fonte: Adaptado de Soares (2012)

3.2.1 Extração das Estruturas Primárias e Secundárias

Os arquivos retirados do PDB contêm várias informações das proteínas, como estrutura primária, estrutura secundária, fatores de temperatura, coordenadas atômicas, registros de conectividade, data de revisão e outras informações relacionadas (BERMAN et al., 2000). Como este trabalho analisa somente as estruturas primárias e secundárias, esta etapa faz com que o *parser* leia somente essas informações. As linhas que se iniciam com a *tag* SEQRES representam as estruturas primárias, enquanto as que se iniciam com HELIX e SHEET, secundárias (α-*Helix* e β-*Sheet*, respectivamente). Um trecho das estruturas primárias e secundárias no formato PDB pode ser visto na Figura 9.

Figura 9 – Estruturas Primárias e Secundárias do PDB

```
HEADER    COMPLEX (ONCOGENE PROTEIN/PEPTIDE)      30-SEP-96   1YCR
TITLE     MDM2 BOUND TO THE TRANSACTIVATION DOMAIN OF P53
SEQRES   1 A  109  SER GLN ILE PRO ALA SER GLU GLN GLU THR LEU VAL ARG
SEQRES   2 A  109  PRO LYS PRO LEU LEU LEU LYS LEU LEU LYS SER VAL GLY
SEQRES   3 A  109  ALA GLN LYS ASP THR TYR THR MET LYS GLU VAL LEU PHE
SEQRES   4 A  109  TYR LEU GLY GLN TYR ILE MET THR LYS ARG LEU TYR ASP
SEQRES   5 A  109  GLU LYS GLN GLN HIS ILE VAL TYR CYS SER ASN ASP LEU
SEQRES   6 A  109  LEU GLY ASP LEU PHE GLY VAL PRO SER PHE SER VAL LYS
SEQRES   7 A  109  GLU HIS ARG LYS ILE TYR THR MET ILE TYR ARG ASN LEU
SEQRES   8 A  109  VAL VAL VAL ASN GLN GLN GLU SER SER ASP SER GLY THR
SEQRES   9 A  109  SER VAL SER GLU ASN
SEQRES   1 B   15  SER GLN GLU THR PHE SER ASP LEU TRP LYS LEU LEU PRO
SEQRES   2 B   15  GLU ASN
HELIX    1   1 PRO A   32  VAL A   41  1                                  10
HELIX    2   2 MET A   50  THR A   63  1                                  14
HELIX    3   3 LEU A   81  PHE A   86  1                                   6
HELIX    4   4 HIS A   96  ASN A  106  1                                  11
HELIX    5   5 PHE B   19  LEU B   25  1                                   7
SHEET    1   A 2 ILE A  74  TYR A  76  0
SHEET    2   A 2 SER A  90  SER A  92 -1  N  PHE A  91   O  VAL A  75
```

Fonte: Elaborado pelo autor.

O próximo passo nesta etapa é a extração das informações dos arquivos gerados. O primeiro número da sequência primária indica o índice da linha em cada sequência. O caractere seguinte indica em qual sequência a linha pertence. Podem haver mais de uma sequência, porém, como não foi considerado a estrutura quaternária da proteína, é tratado apenas a primeira. Em seguida, um número que identifica a quantidade total de resíduos nesta sequência, seguido por no máximo 13 resíduos da proteína.

Na estrutura secundária, a linha referente à α-*Helix* possui os seguintes atributos: um número com o índice da hélice, um alfanumérico que também identifica a hélice, um resíduo indicando o início da formação secundária, um caractere indicando a sequência a que ela pertence, seguido pela posição do resíduo na sequência. Os três valores seguintes representam o resíduo final da hélice, a sequência que pertence e a última posição do resíduo. Os últimos valores são: um inteiro representando a classe da hélice, um comentário, e um inteiro com o tamanho da hélice.

Na linha 18 da Figura 9, a α-*Helix* tem o índice e o identificador iguais a 5, começa no resíduo PHE da sequência B que está na posição 19 e termina na posição 25 com o resíduo LEU da sequência B, não tendo nenhum comentário e possuindo tamanho igual a 7 resíduos $(25 - 19 + 1)$. A estrutura é semelhante na linha referente à β-*Sheet*, porém os dados adicionais são desconsiderados no *parser*. Na linha 19, a folha começa no resíduo ILE da sequência A que fica na posição 74 e termina no resíduo TYR da sequência A, na posição 76.

Com os valores de cada aminoácido referente às estruturas primárias e secundárias, a saída desta fase na codificação de Resende et al. (2012) ficará como mostrada abaixo:

$$\underbrace{1 \quad 4 \quad 16 \quad 2 \quad 13 \quad 8 \quad 1 \quad 4 \quad 16}_{\textit{Primária}} \quad \underbrace{2 \quad 0 \quad 2 \quad 2 \quad 0 \quad 2 \quad 1 \quad 1 \quad 0}_{\textit{Secundária}}$$

O primeiro conjunto representa a estrutura primária com a codificação transformada e o segundo, a secundária.

3.2.2 Inserção de Zeros

A Máquina de Vetor de Suporte de Joachims (1999), utilizada no trabalho, exige que os vetores de entrada sejam do mesmo tamanho. Por causa disso, nesta fase do *parser*, as sequências são completadas com zeros para que fiquem do mesmo tamanho da maior sequência

encontrada, compatíveis com o SVM.

3.2.3 Codificação

O aprendizado de máquina com o vetor gerado na etapa anterior possui um elevado custo de tempo devido ao tamanho da sequência. Com isso, Nascimento, Yoshioka e Calanzans (2011) propuseram uma implementação de uma tabela *hash* que poderia ser usada para codificar partes iguais dentro da sequência, diminuindo seu tamanho e, com isso, reduzindo o tempo de processamento. Resende et al. (2012) e Soares (2012) utilizaram um bloco de tamanho 7 para referenciar os trechos da sequência, sendo mantido esse valor no presente trabalho. Nesta fase, todas as proteínas são gravadas em um arquivo, com o nome de suas classes concatenadas no final de cada linha. A expressão a seguir mostra a sequência anterior codificada nesta fase.

$$\underbrace{1 \quad 4 \quad 16}_{1} \quad \underbrace{2 \quad 13 \quad 8}_{2} \quad \underbrace{1 \quad 4 \quad 16}_{1} \quad \underbrace{2 \quad 0 \quad 2}_{3} \quad \underbrace{2 \quad 0 \quad 2}_{3} \quad \underbrace{1 \quad 1 \quad 0}_{4}$$

3.2.4 Normalização

Os atributos gerados na fase de codificação possuem valores muito diferentes entre si, que podem saturar a função de ativação do SVM, prejudicando sua eficiência (LIMA, 2010). Assim, é necessário normalizar os dados, enquadrando-os no intervalo [0,...,1]. Desta forma, o valor de uma variável não se torna mais significativo que o de outra. Para isso, utilizou-se a seguinte função retirada de Goldschmidt e Passos (2005):

$$MinMax = \frac{(ValueX - Minimum)}{(Maximum - Minimum)} \tag{3.1}$$

3.2.5 Compactação das sequências

Depois de normalizado, os vetores do arquivo gerado são compactados um a um. Isto é necessário não só para reduzir o tamanho dos vetores e diminuir o processamento, como também garantir que a maior parte das informações desnecessárias não sejam treinadas pelo SVM. Para isso, foi utilizada a Transformada Discreta do Cosseno, aplicada em todos os

vetores, transformando cada um separadamente. Foi escolhido este método, porque é uma transformação que preserva as normas e os ângulos dos vetores (OLIVEIRA et al., 2006). Em seguida, foi necessário truncar o número de coeficientes de expansão para um tamanho fixo, para efetuar a compactação, excluindo grande parte do ruído. Após a aplicação da TDC, foi gerado um arquivo compactado do mesmo formato do original.

3.2.6 Geração de arquivos de testes e treinos

Com o arquivo compactado, foi necessário realizar as seguintes etapas para gerar os arquivos de testes e treinos:

1. Separar as sequências dos vetores compactados gerados pelo *parser* em seis arquivos referentes à cada classe;
2. Efetuar a Validação Cruzada de cada classe gerando dez arquivos de testes e 10 arquivos de treinos para cada uma. Para gerar os arquivos de testes e treinos que servirão de entrada para o SVM, foi utilizado um *k-fold Cross Validation* com $k = 10$;
3. Os arquivos de testes e treinos são convertidos para o formato da SVM de Joachims (1999);
4. Junta-se cada arquivo de teste de todas as classes em dez arquivos finais de teste. É feito o mesmo para os arquivos de treino;

3.3 SVM

A SVM multiclasse utilizada foi desenvolvida por Joachims (1999). Cada linha do arquivo SVM representa um vetor. A primeira posição de cada vetor representa a classe da proteína. Em seguida, cada elemento representa o seu índice e o seu valor, concatenado com dois pontos. Os índices devem aparecer ordenadamente e, na falta de um deles, significa que o seu valor é igual a 0, como mostra na expressão a seguir: o vetor pertence à classe 2, a primeira posição tem valor 9, a segunda tem valor 8, a terceira tem valor 3, a quarta, implícita, tem valor 0, e a quinta, 13.

$$2 \quad 1:9 \quad 2:8 \quad 3:3 \quad 5:13$$

A SVM de Joachims (1999) possui 2 chamadas. Na primeira, é lido os arquivos de treino gerados na etapa anterior junto com os parâmetros de entrada, tendo como saída arquivos de modelo para ser usado pela próxima chamada, a etapa de classificação. Para descobrir os melhores valores dos parâmetros da SVM para cada teste, foi utilizado o Algoritmo Genético de Resende et al. (2012). Na etapa de classificação, é lido como entrada os arquivos de modelo e os de testes, tendo como saída arquivos de predição das classes.

3.4 Cálculo de Desempenho

Para realizar os cálculos de desempenho, deve-se considerar os conceitos de Verdadeiro Positivo, Verdadeiro Negativo, Falso Positivo e Falso Negativo (SOARES, 2002):

- **Verdadeiro Positivo (*VP*)**: Corresponde ao número de instâncias corretamente classificadas na classe em questão;
- **Verdadeiro Negativo (*VN*)**: Corresponde ao número de instâncias de outras classes classificadas como pertencentes às outras classes;
- **Falso Positivo (*FP*)**: Corresponde ao número de instâncias de outras classes classificadas como pertencentes à classe analisada;
- **Falso Negativo (*FN*)**: Corresponde ao número de instâncias da classe analisada que foram classificadas erroneamente, ou seja, pertencentes à outra classe da base de dados;

Foram utilizadas quatro métricas para o cálculo do desempenho: precisão, sensibilidade, especificidade e acurácia (SOARES, 2002).

- **Precisão**

  Proporção de verdadeiros positivos em relação ao número total de proteínas classificadas como pertencentes a uma determinada classe. Definida pela Equação 3.2.

$$P = \frac{VP}{VP + FP} \tag{3.2}$$

- **Sensibilidade**

  Capacidade de um teste de predição identificar os verdadeiros positivos em proteínas que realmente pertencem à classe analisada. Definida pela Equação 3.3.

$$S = \frac{VP}{VP + FN} \tag{3.3}$$

- **Especificidade**

  Capacidade de um teste de predição identificar os verdadeiros negativos em proteínas que não pertencem à classe analisada. Definida pela equação 3.4.

$$E = \frac{VN}{VN + FP} \tag{3.4}$$

- **Acurácia**

  Proporção de acertos, ou seja, número total de verdadeiros positivos e verdadeiros negativos em relação a amostra total estudada. Definida pela Equação 3.5.

$$A = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.5}$$

4 RESULTADOS

Foram realizados dois testes neste trabalho. No primeiro teste, foi excluída a etapa do *parser* referente à inserção de zeros. Como a Transformada do Cosseno mantém todos os vetores com a mesma dimensão, esta etapa seria desnecessária. Porém, como os vetores foram compactados em um tamanho muito pequeno, muitos dados relevantes foram perdidos, tornando os resultados deste teste piores do que os de Resende et al. (2012).

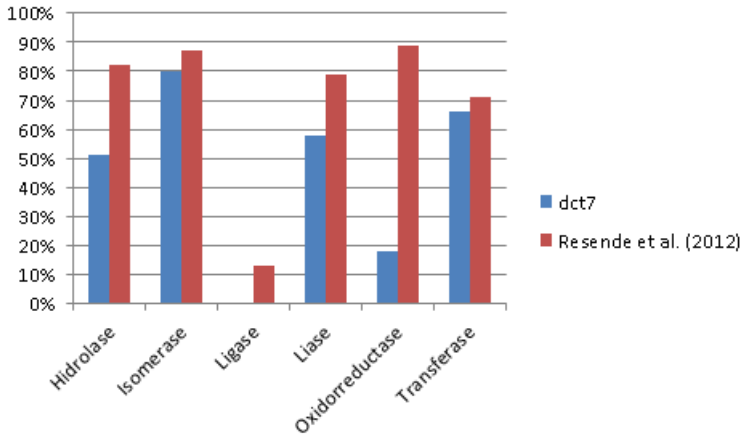
Com isso, o segundo teste manteve a etapa de inserção de zeros, fazendo com que nao sejam perdidas muitas informações das proteínas maiores. Neste teste, foram comparadas as predições em diversos tamanhos de compactação realizados pela TDC, obtendo resultados melhores do que o trabalho anterior.

4.1 Sem inserção de zeros

Nesta seção de testes, a etapa de inserção de zeros, que tem a função de igualar os tamanhos dos vetores, foi retirada do *parser*. Os resultados foram piores do que os de Resende et al. (2012), por causa da etapa de compactação que fez um truncamento de dados com base na dimensão do menor vetor de suporte, que possui tamanho 7. Com isso, foram perdidas muitas informações importantes referentes aos vetores de maior tamanho, que aparecem em maior número.

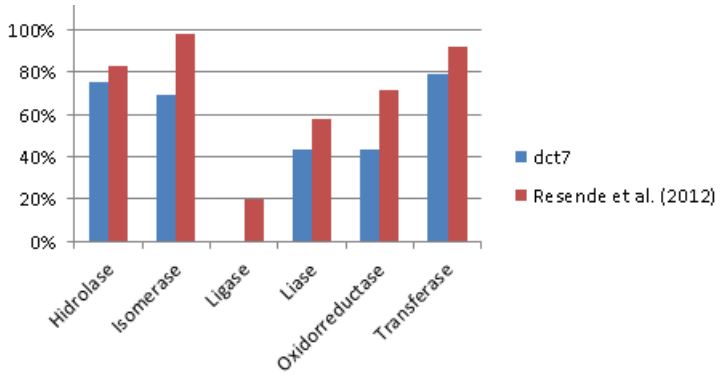
As Figuras 10, 11, 12 e 13 mostram os resultados desta codificação, em comparação com os de Resende et al. (2012), separadas por métricas.

Figura 10 – Precisão do teste sem inserção de zeros



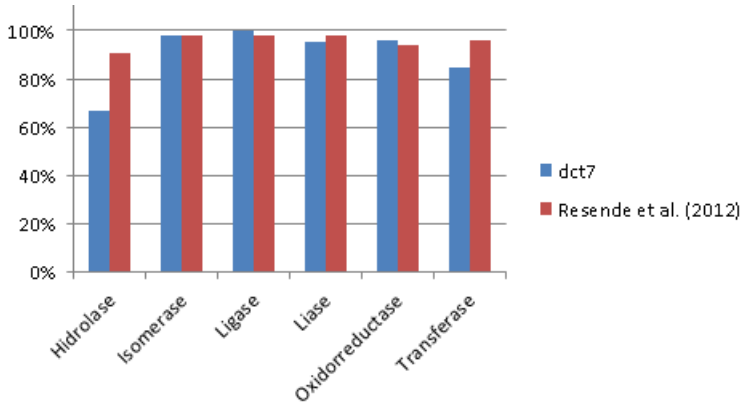
Fonte: Elaborado pelo autor.

Figura 11 – Sensibilidade do teste sem inserção de zeros



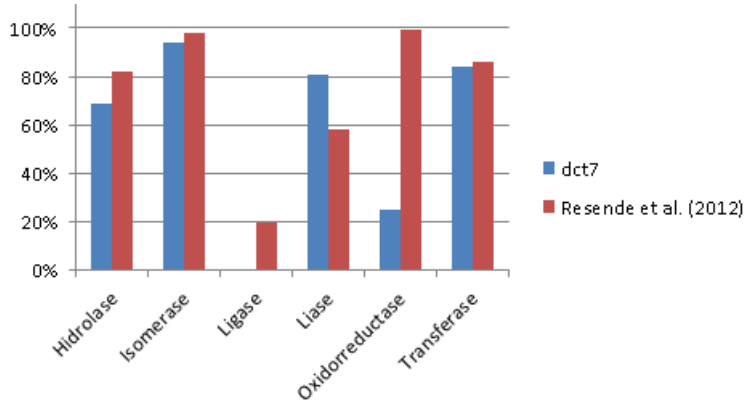
Fonte: Elaborado pelo autor.

Figura 12 – Especificidade do teste sem inserção de zeros



Fonte: Elaborado pelo autor.

Figura 13 – Acurácia do teste sem inserção de zeros



Fonte: Elaborado pelo autor.

Na maioria dos casos, os dados foram classificados como pertencentes à classe Hidrolase, a que tem mais amostras de proteínas. Com isso, os valores de precisão, sensibilidade e acurácia de todas as classes neste trabalho ficaram piores dos que os de Resende et al. (2012), com excessão apenas na acurácia da Liase. Na Ligase, classe com o menor número amostras, a precisão chegou a 0%.

Na especificidade, foi obtido resultados semelhantes ao trabalho anterior, porque ambos os testes obteve uma alta taxa de verdadeiros negativos em proteínas não pertencentes à classe analisada.

4.2 Com inserção de zeros

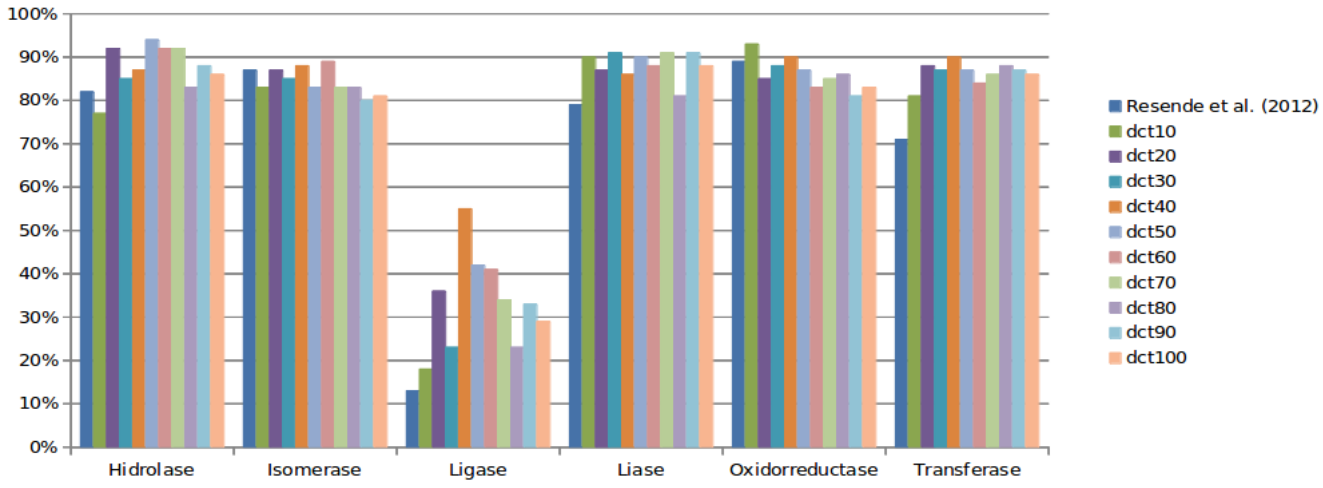
A fim de tentar melhorar os resultados, decidiu-se voltar com a etapa de inserção de zeros, proposta por Resende et al. (2012). Com isto, os vetores continuaram do mesmo tamanho entre si, porém mais informações relevantes serão consideradas na SVM, já que não haverá um corte muito grande nas proteínas de maior tamanho.

Como os vetores ficaram maiores, nesta etapa foram comparados os resultados de diversos valores de compactação. Foi obtido um resultado ainda melhor com o valor de corte na compactação igual a 40, comparando com o trabalho anterior. Para o pior resultado, pertencente à classe das Ligases, a precisão subiu para 58%, em comparação com 13% no trabalho de Resende et al. (2012).

As Figuras 15, 16, 17 e 18 mostram os novos resultados, inserindo zeros e compactando

as estruturas, comparados com Resende et al. (2012). As primeiras colunas de cada classe representam os resultados do trabalho anterior. O restante das colunas representam os resultados com diferentes valores na compactação utilizando a Transformada do Cosseno.

Figura 14 – Precisão do teste com inserção de zeros

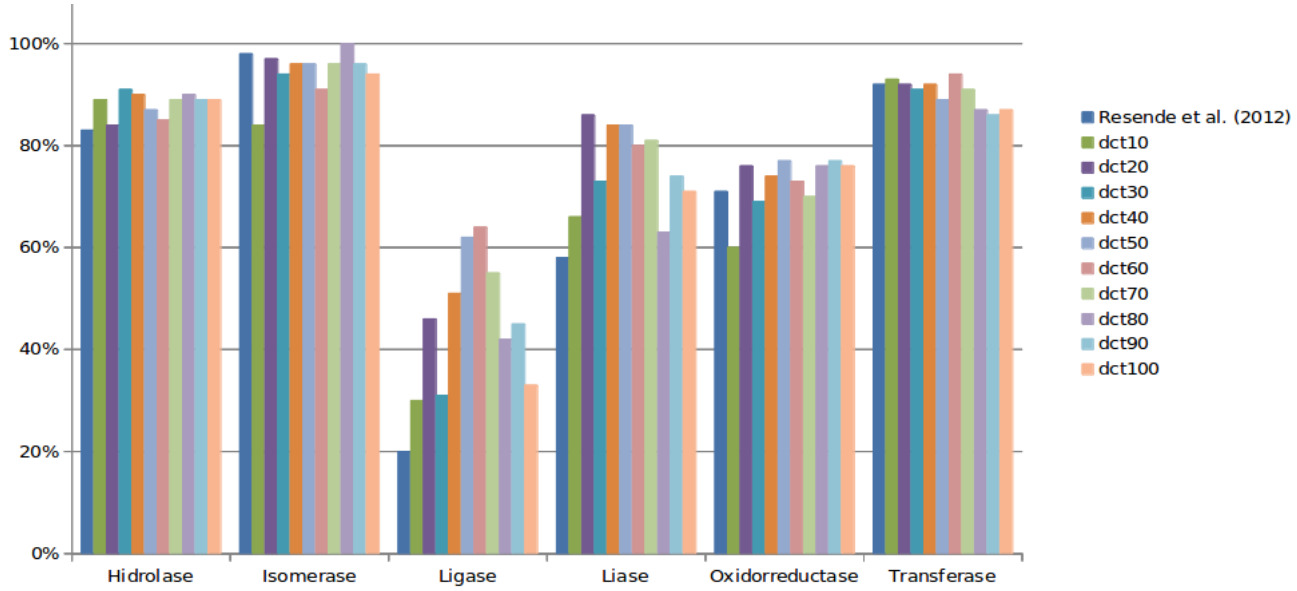


Fonte: Elaborado pelo autor.

Nos resultados de Precisão, a classe das Ligases apresentou um pior resultado, demonstrando que de todos os testes classificados como Ligases, o número de Verdadeiros Positivos foi muito baixo. O pior valor para esta classe foi de 15%, com o vetor de tamanho 10, e o melhor valor foi de 55%, com o tamanho 40. As demais classes apresentaram bons resultados para esta métrica, de 77% a 95%.

Na classe das Ligases, é importante notar que o valor da precisão aumenta quando o tamanho do vetor de suporte compactado é aumentado. Porém, depois do tamanho 40, os resultados pioram. Isso ocorre devido à quantidade de ruídos que são inseridos no classificador, que prejudicam a eficiência do modelo de predição.

Figura 15 – Sensibilidade do teste com inserção de zeros

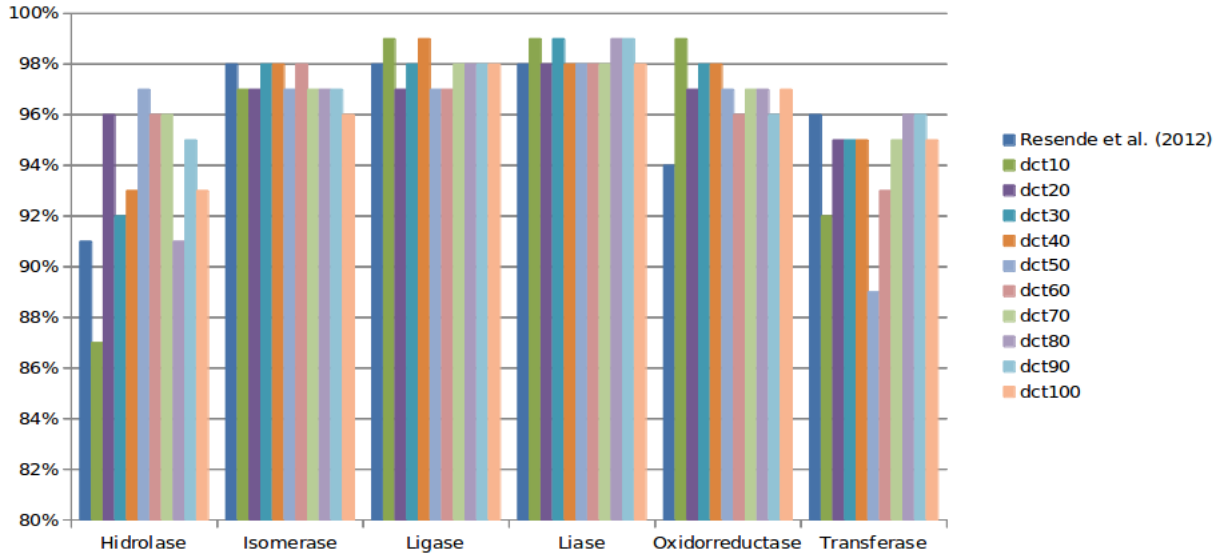


Fonte: Elaborado pelo autor.

Como a Sensibilidade indica a capacidade de se indicar os Verdadeiros Positivos em proteínas que pertencem à classe analisada, seu resultado também foi baixo nas Ligases, por causa da baixa quantidade de proteínas corretamente classificadas nesta classe, chegando a 64% no melhor caso. Para a Isomerase, classe com melhores valores de Sensibilidade, o melhor valor foi com o vetor de tamanho 80, chegando a 99%.

Na maioria das classes, o valor de sensibilidade diminui a partir do tamanho de compactação igual a 60, por causa da quantidade de ruídos que são inseridos no classificador.

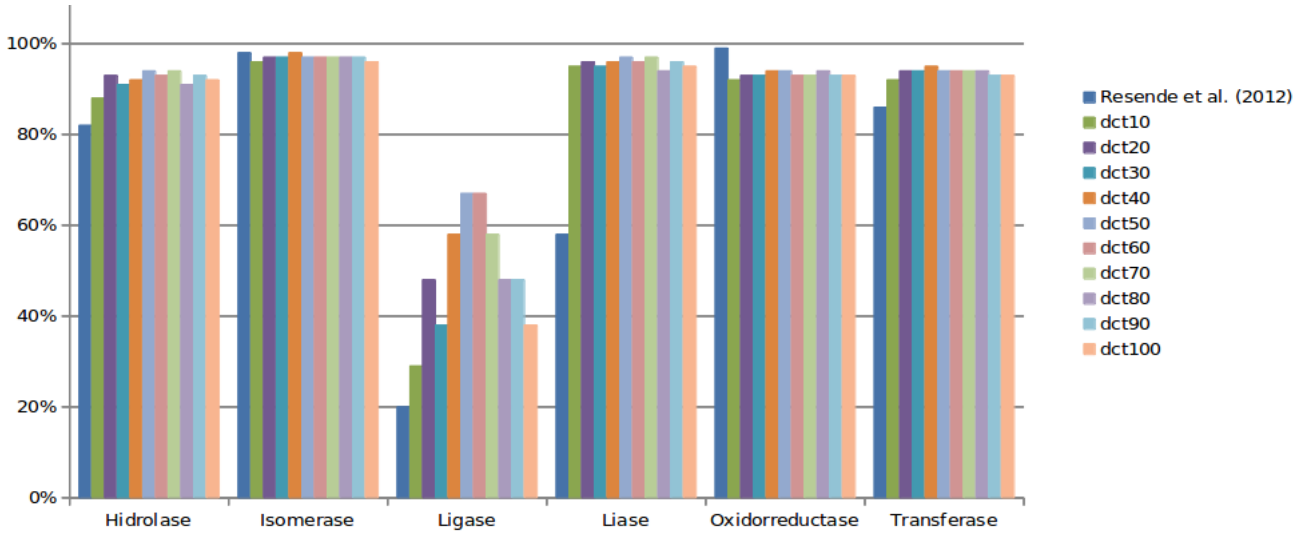
Figura 16 – Especificidade do teste com inserção de zeros



Fonte: Elaborado pelo autor.

A especificidade apresentou bons resultados para todas as classes, já que é muito alto o número de verdadeiros negativos que não pertencem à classe analisada. O valor desta métrica foi superior a 90% na maioria dos testes e 78% no pior teste.

Figura 17 – Acurácia do teste com inserção de zeros



Fonte: Elaborado pelo autor.

A acurácia na maioria das classes foi superior à 80% porém, como a classe das Ligases possui uma menor proporção de acertos, seu resultado foi baixo, com valores entre 28% a 67%.

5 CONCLUSÃO

Neste trabalho foi apresentada um método para realizar a predição de funções de enzimas utilizando o classificador SVM. Para isso, foi alterado um *parser* de forma a compactar as informações das proteínas. Foram realizados dois testes: no primeiro retira-se a etapa do *parser* referente a adicionar zeros nas sequências, utilizado para manter os vetores do mesmo tamanho. No segundo, voltou-se com a inserção de zeros, sendo compactadas as sequências com zeros, comparando-se cada tamanho de compactação.

No primeiro teste, foi obtido resultados piores do que os trabalhos anteriores, por causa da grande quantidade de informações desconsideradas pela DCT. Foi obtido 0% de acerto para a classe de enzimas com piores resultados, enquanto a classe com os melhores resultados obteve 86% na precisão, 71% na sensibilidade, 98% na especificidade e 95% na acurácia.

No segundo teste, no entanto, foi obtido resultados melhores do que os dos trabalhos anteriores. Os valores da precisão para a classe de enzimas com menos amostras passou de 13% para 55%, com o tamanho dos vetores compactados igual a 40. Essa grande diferença com os resultados do primeiro teste ocorre pela maior quantidade de informações que foram consideradas durante o aprendizado de máquina, já que, ao compactar as proteínas maiores, menos informações foram perdidas.

Como a DCT fez com que os valores mais relevantes fossem treinados pela SVM, foi concluído que a quantidade de informações a serem treinadas não é importante para se ter um bom resultado com o SVM, e sim a qualidade. Também foi concluído que, para as amostras do presente trabalho, o melhor tamanho dos vetores depois de compactados com a TDC é 40, levando em conta a grande melhoria dos resultados para a Ligase, classe que apresentou os piores resultados.

Para trabalhos futuros, pretende-se analisar outras metodologias de compactação de sequências, além de pesquisar características estruturais das proteínas que podem ser inseridas no classificador, de modo a tornar os resultados ainda mais consistentes. Como os piores resultados deste trabalho estão relacionados à baixa quantidade de amostras na classe das Ligases em relação às outras classes, pode ser proposto também um método para balancear as quantidades de todas as amostras em cada classe.

REFERÊNCIAS

AHMED, N.; NATARAJAN, T.; RAO, K. R. **Discrete Cosine Transform**. IEEE, 1974.

AL-SHAHIB, A.; BREITLING, R.; GILBERT, D. R. **Predicting protein function by machine learning on amino acid sequences - a critical evaluation**. BioMed Central, 2007.

ALTEN, D. v. ***Reações Bioquímicas***. 2005. Disponível em: <http://daanvanalten.nl/quimica>.

ALTSCHUL, S. et al. **Basic local alignment search tool.** NCBI, 1990.

ALVAREZ, M. A.; YAN, C. **Exploring Structural Modeling of Proteins for Kernel-Based Enzyme Discrimination**. IEEE, 2010.

AMABIS, J.; MARTHO, G. ***Conceitos de Biologia***. [S.l.]: Editora Moderna, 2001.

BERMAN, H. et al. ***The Protein Data Bank***. 2000. Acessado em: 2013-09-30. Disponível em: <http://www.brenda-enzymes.org>.

CAI, C. et al. **Enzyme Family Classification by Support Vector Machines**. WILEY-LISS, 2004.

CAI, C. Z. et al. **Protein function classification via support vector machine approach**. Mathematical Biosciences 185, 2003.

DIAS, U. M. **Predição da Função das Proteínas Sem Alinhamentos Usando Máquinas de Vetor de Suporte**. UFAL - Universidade Federal de Alagoas, 2007.

DOBSON, P. D.; DOIG, A. J. **Distinguishing Enzyme Structures from Non-enzymes Without Alignments**. Elsevier, 2003.

DUBCHAK, I. et al. **Prediction of protein folding class using global description of amino acid sequence**. Biophysics, 1995.

FARAH, S. C. ***Ligação peptídica entre aminoácidos***. 2007. Disponível em: <http://www.iq.usp.br/chsfarah>.

GOLDSCHMIDT, R.; PASSOS, E. ***Data Mining - Um Guia Prático***. [S.l.]: CAMPUS, 2005.

HALL, M. et al. **The WEKA Data Mining Software: An Update**. SIGKDD Explorations, 2009.

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

KHAYAM, S. A. **The Discrete Cosine Transform (DCT): Theory and Application**. Michigan State University, 2003.

KOLODNY, R.; LINIAL, N. **Approximate Protein Structural Alignment in Polynomial Time**. 2004.

LIMA, V. R. de. **Desenvolvimento e Avaliação de Sistema Neural para Redução de Tempo de Ensaio de Desempenho de Compressões**. UFSC - Universidade Federal de Santa Catarina, 2010.

MONTUORI, A.; RAIMONDO, G.; PASERO, E. **An information theoretic approach for improving data driven prediction of protein model quality**. Elsevier, 2008.

NASCIMENTO, R.; YOSHIOKA, S.; CALANZANS, T. **Predição de Funções Proteicas através da Análise Conjunta das Estruturas Primárias e Secundárias**. PUC-MG, 2011.

NELSON, D. L.; COX, M. M. ***Lehninger's Principles of Biochemistry***. [S.l.]: W. H. Freeman and Company, 2008.

OLIVEIRA, S. R. de M. et al. **Uma Metodologia para Seleção de Parâmetros em Modelos de Classificação de Proteínas**. IEEE, 2006.

PANDEY, G.; KUMAR, V.; STEINBACH, M. **Computational Approaches for Protein Function Prediction: A Survey**. National Science Foundation, 2006.

PENAFORTE, A. ***Ligacao Peptidica***. 2012. Disponível em: <http://profpenafortehiperlinks.blogspot.com.br/2012/04>.

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.

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

SCHOMBURG, D. ***BRENDA: The Comprehensive Enzyme Information System***. 2009. Acessado em: 2013-10-30. Disponível em: <http://www.rcsb.org>.

SHATSKY, M.; NUSSINOV, R.; WOLFSON, H. J. **A Method for Simultaneous Alignment of Multiple Protein Structures**. WILEY-LISS, INC, 2004.

SHEN, J. et al. **Predicting protein-protein interactions based only on sequences information**. PNAS, 2006.

SOARES, J. ***Métodos diagnósticos. Consulta rápida.*** [S.l.]: ARTMED, 2002.

SOARES, P. R. B. **Previsão de Função de Enzimas Utilizando Máquinas de Vetor de Suporte**. PUC-MG, 2012.

WANG, Y.-C. et al. **Support vector machine prediction of enzyme function with conjoint triad feature and hierarchical context**. BMC, 2011.