PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Programa de Pós-Graduação em Informática

METODOLOGIA PARA PREVISÃO DE SÍTIO DE INÍCIO DE TRADUÇÃO DE PROTEÍNAS EM SEQUÊNCIAS DE mRNA

Lívia Márcia Silva

Belo Horizonte

2010

Lívia Márcia Silva

METODOLOGIA PARA PREVISÃO DE SÍTIO DE INÍCIO DE TRADUÇÃO DE PROTEÍNAS EM SEQUÊNCIAS DE mRNA

Dissertação apresentada ao Programa de Pós-Graduação em Informática como requisito parcial para qualificação ao Grau de Mestre em Informática pela Pontifícia Universidade Católica de Minas Gerais.

Orientador: Luis Enrique Zárate Gálvez

Co-orientadora: Cristiane Neri Nobre

Belo Horizonte

2010

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

Silva, Lívia Márcia
S586m Metodologia para previsão de sítio de início de tradução de proteínas em sequencias de mRNA / Lívia Márcia Silva. – Belo Horizonte, 2010.
74f. : il.

Orientador: Luis Enrique Zárate Gálvez
Co-orientadora: Cristiane Neri Nobre
Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-graduação em Informática.
Bibliografia.

1. Bioinformática – Teses. 2. Algoritmos de computador. I. Zárate Gálvez, Luis Enrique. II. Nobre, Cristiane Neri III. Pontifícia Universidade Católica de Minas Gerais. IV. Título.

CDU: 681.3.03.056:576.34

Bibliotecário: Fernando A. Dias – CRB6/1084

PUC Minas
Programa de Pós-graduação em Informática

FOLHA DE APROVAÇÃO

Metodologia para previsão de sítio de início de tradução de proteínas em sequências de mRNA

LÍVIA MÁRCIA SILVA

Dissertação defendida e aprovada pela seguinte banca examinadora:

Prof. Luis Enrique Zárate Gálvez - Orientador (PUC Minas)
Doutor em Engenharia Metalúrgica e de Minas – UFMG

Profª. Cristiane Neri Nobre - Co-orientadora (UFSJ)
Doutora em Bioinformática pela UFMG

Prof. Zenilton Kleber Gonçalves do Patrocínio Júnior (PUC Minas)
Doutor em Ciências da Computação - UFMG

Prof. Antônio de Pádua Braga (UFMG)
Doutor em História - UFF

Belo Horizonte, 10 de dezembro de 2010.

Anel Rodoviário – KM 23,5, Rua Walter Ianni, 255 – São Gabriel / Belo Horizonte – MG / CEP: 31980-110
Fone: (31) 3439-5204 – Fax: (31) 3439-5204 – e-mail: mestradoinfsec@pucminas.br

Aos meus pais, Maria do Carmo e Geraldo, pela paciência
e ao Victor, pela dedicação e carinho.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por ter me dado essa oportunidade. Pela força e coragem durante toda essa caminhada.

Aos meus pais, Geraldo e Maria do Carmo, pela paciência e apoio de forma incondicional e pelo incentivo constante.

Ao meu orientador Zárate e à minha co-orientadora Cristiane, pela orientação nestes anos, por incentivar idéias, pelo tempo despendido ao longo deste trabalho, e pela amizade.

A todos meus amigos e familiares que direta ou indiretamente contribuíram para a realização deste trabalho e que estiveram presentes durante todo esse tempo.

Ao meu irmão Ricardo, que tanto sofreu com minha falta de paciência e disponibilidade.

Muito obrigada, Victor, pelo companheirismo, paciência, apoio e carinho me ofertado de forma incondicional. Pelas tantas e tantas viagens realizadas, sem seu apoio teria sido muito mais penoso...

"Uma jornada de duzentos quilômetros
começa com um simples passo."

Provérbio Chinês

"Para cada esforço disciplinado há
uma retribuição múltipla."

Jim Rohn

RESUMO

A previsão correta do início de tradução em sequências de mRNA é uma atividade importante para a anotação genômica. No entanto, realizar uma previsão correta nem sempre é uma tarefa trivial e, dessa forma, pode ser modelada como um problema de classificação entre sequências positivas (codificadoras de proteínas) e negativas (não codificadoras). Por ser um problema desbalanceado, já que cada molécula de mRNA possui um único início de tradução e vários outros não são iniciadores, esse trabalho focou em métodos de balanceamento que resolve o problema proposto com eficácia e eficiência. Para isso, está sendo proposto um método de balanceamento do tipo *undersampling* baseado em Clusterização, M-Clus, além de uma nova metodologia que adiciona características às sequências e que melhora o desempenho do classificador a partir da inclusão do conhecimento obtido pelo modelo. Por meio dessa metodologia, as taxas de desempenho utilizadas, acurácia, sensibilidade, especificidade e acurácia ajustada são superiores a 93% (*Mus musculus*). A precisão aumenta significativamente, de 43,05% para 82,05% (*Mus musculus*) e de 13,54% para 35,63% (*Rattus novergicus*), adotando a inclusão do conhecimento obtido pelo modelo. Para resolução do problema, faz-se necessário o investimento em técnicas de balanceamento de classes, além de uma metodologia criteriosa que melhora visivelmente os resultados. Ao utilizar o método de balanceamento M-Clus há um aumento significativo na taxa de sensibilidade, de 51,39% para 91,55% e de 47,45% para 88,09%, para os organismos *Mus musculus* e *Rattus novergicus*, respectivamente. A inclusão de algumas características durante o treinamento, tais como a presença de ATG na região *upstream* do Sítio de Início de Tradução, melhora a taxa de sensibilidade em aproximadamente 9% para o organismo *Mus musculus* e 6% para o *Rattus novergicus*.

Palavras-chave: Sítio de Início de Tradução de proteínas. Support Vector Machine. Balanceamento de classes.

ABSTRACT

The accurate prediction of the initiation of translation in sequences of mRNA is an important activity for genome annotation. However, obtaining an accurate prediction is not always a simple task and can be modelled as a problem of classification between positive sequences (protein codifiers) and negative sequences (non-codifiers). The problem is imbalanced because each molecule of mRNA has a unique translation initiation site and various others that are not initiators. Therefore, this study focuses on balancing methods which resolve the proposed problem effectively and efficiently. M-Clus, an undersampling balancing method based on Clustering is proposed, in addition to a new methodology that adds features to sequences and that improves the performance of the classifier through the inclusion of knowledge obtained by the model. Through this methodology, the measures of performance used (accuracy, sensitivity, specificity and adjusted accuracy) are greater than 93% (*Mus musculus*). The precision increases significantly from 43.05% to 82.05% (*Mus musculus*) and 13,54% to 35,63% (*Rattus novergicus*) when the knowledge obtained by the model is included. In order to resolve the problem, it is necessary to invest in class balancing techniques in addition to a judicious methodology which visibly improves the results. Using the M-Clus balancing method generates a significant increase in the rate of sensitivity from 51.39% to 91.55% and 47,45% to 88,09%, for organisms *Mus musculus* and *Rattus novergicus*, respectively. The inclusion of certain features during training, for example, the presence of ATG in the upstream region of the Translation Initiation Site, improves the rate of sensitivity by approximately 9% for organism *Mus musculus* and 6% for *Rattus novergicus*.

Keywords: Translation initiation sites. Support Vector Machines. Balancing class.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

CDS	CoDing Sequence
DNA	Ácido Desoxiribonucleico
NCBI	National Center for Biotechnology Information
mRNA	Ácido Ribonucléiclo Mensageiro
RefSeq	Reference Sequence - Sequências de referência
RN	Redes Neurais Artificiais
RSM	Ribosome Scanning Model
SIT	Sítio de Início de Tradução
SVM	Support Vector Machines

SUMÁRIO

1 INTRODUÇÃO

A transcrição e a tradução são os meios pelos quais as células interpretam e expressam suas informações genéticas (TZANIS; BERBERIDIS; VLAHAVAS, 2006). Somente parte das sequências transcritas carrega informação para codificar proteínas (CDS - CoDing Sequence), ou seja, mesmo o mRNA podendo ser traduzido em sua totalidade, apenas um trecho desse mRNA é traduzido em aminoácido (NOBRE; ORTEGA; BRAGA, 2007). Portanto, dada uma molécula de mRNA, um problema central da biologia molecular é determinar se a mesma contém CDS; e a partir daí, descobrir qual proteína será codificada.

A região da sequência de mRNA onde se dá o início do processo de síntese de proteínas denomina-se Sítio de Início de Tradução (SIT). O controle do início da tradução é um dos mais importantes processos na regulação da expressão genética (NAKAGAWA et al., 2008).

A Figura 1 ilustra a região codificadora de uma sequência de aminoácidos, supondo que todo o "retângulo" é uma sequência de RNA mensageiro, a tradução da proteína referente a essa sequência se inicia no primeiro traço e termina no segundo. A região delimitada pelos traços é a região codificadora da sequência em questão.

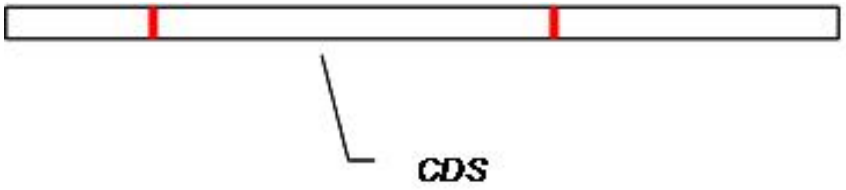

Figura 1: Exemplo de região codificadora (CDS).

Dessa forma a determinação do SIT não é uma tarefa trivial; ao mesmo tempo é de grande relevância para inferência genética. Alta acurácia na predição pode ser útil para um melhor entendimento da proteína obtida a partir das sequências de nucleotídeos (LIU et al., 2004).

Em eucariotos, o modelo de escaneamento supõe que os ribosomos se ligam primeiro à região 5' do mRNA e percorre em direção à região 3' (KOZAK, 1999), exemplificado pela Figura 2.

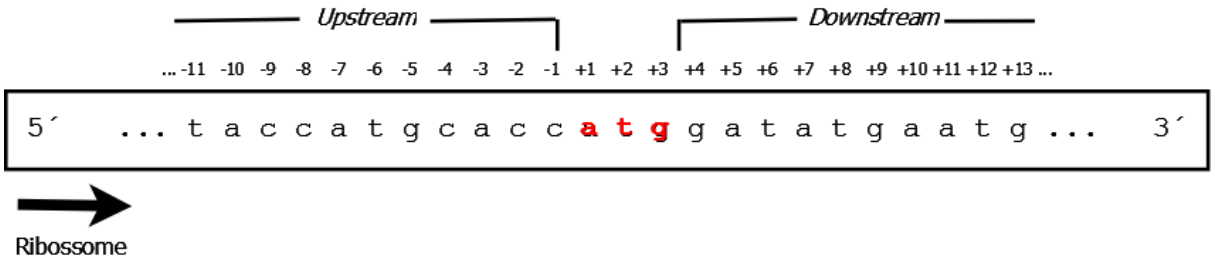

Figura 2: Exemplo de região *upstream* e *downstream*. Neste caso o início da tradução é marcado pelo ATG indicado pela posição +1, +2 e +3, respectivamente.

Normalmente, o sítio de início de tradução começa em um códon[1] ATG. Kozak (1984) detectou que na maioria dos casos, a previsão do SIT é determinada pela presença do primeiro ATG na sequência de mRNA; porém segundo Pedersen e Nielsen (1997), o reconhecimento do SIT em eucariotos nem sempre começa na primeira metionina (ATG), e a escolha pelo códon inicial da CDS depende da posição e também do contexto da sequência. Sendo assim, a correta determinação do SIT é uma importante etapa na análise genômica para determinar a proteína resultante da codificação da sequência de nucleotídeos.

A tradução, normalmente, inicia-se no primeiro ATG da molécula de mRNA que tem um contexto apropriado (KOZAK, 1984; PEDERSEN; NIELSEN, 1997), mas pode iniciar em um códon diferente (HATZIGEORGIOU, 2002).

Ao realizar previsão de SIT, podem-se identificar também duas regiões, ilustradas na Figura 2:

- Região *upstream*: região que antecede o SIT, ou seja, inicia-se juntamente com a sequência e termina exatamente no nucleotídeo anterior ao SIT.
- Região *downstream*: região que sucede o SIT, ou seja, inicia-se no nucleotídeo subsequente ao SIT e termina juntamente com a sequência.

Dependendo da posição de início da síntese na fita de mRNA, o trio de nucleotídeos selecionado para a síntese poderá variar, alterando-se também os aminoácidos que serão traduzidos. O desconhecimento de características conservativas no processo de identificação do início da tradução faz da predição de SIT uma tarefa complexa. Por essa razão, métodos computacionais de busca de padrões podem ser utilizados a fim de extrair o conhecimento implícito envolvido nesse processo (NOBRE; ORTEGA; BRAGA, 2007).

[1] Trinca de nucleotídeos, componente do mRNA, resultantes em um aminoácido.

O trabalho proposto visa o desenvolvimento de uma metodologia para previsão de sítio de início de tradução de proteínas em sequências de mRNA, ou seja, através de fragmentos de sequências de mRNA determinar se o mesmo contém um SIT ou não. Para tal será utilizado o classificador Support Vector Machines (SVM); métodos de balanceamento, devido ao problema ser desbalanceado; inclusão de características juntamente às sequências de mRNA; e desenvolvimento de uma metodologia para inclusão de conhecimento adquirido pelo classificador (InAKnow).

De acordo com Semolini (2002), SVM implementa um mapeamento não-linear (executado por um produto interno *kernel* escolhido a priori) dos dados de entrada para um espaço característico de alta-dimensão, em que um hiperplano ótimo é construído para separar os dados linearmente em duas classes. Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço característico é aquele que apresenta a máxima margem de separação. Essas características são importantes para esse trabalho, visto que é um problema de classificação binária, ou seja, as sequências serão classificadas como positivas ou negativas; e é um problema de alta-dimensão, sendo que essa dimensão será o tamanho das janelas, neste caso, tamanho dos fragmentos de sequências de mRNA.

O tamanho das janelas dos fragmentos de sequências de mRNA também é objeto de estudo desse trabalho. Inicialmente, janelas com tamanho simétrico, por exemplo -10+10 (10 nucleotídeos nas regiões *upstream* e *downstream*, respectivamente), foram experimentalmente avaliados, e posteriormente experimentos com janelas de tamanho assimétrico, por exemplo -10+30, se mostraram mais eficientes.

Uma vez identificado que o problema de previsão de SIT é desbalanceado e que os métodos de *oversampling*[2], já utilizados no presente contexto, (NOBRE; ORTEGA; BRAGA, 2007), aumentam significativamente a complexidade computacional do mesmo, esse trabalho propõe um método *undersampling*[3] de balanceamento de classes, M-Clus. Isso é particularmente importante em grandes bases de dados, onde técnicas de *oversampling* são inviáveis, já que aumentam ainda mais o tamanho das bases envolvidas.

Somado ao método de balanceamento, esse trabalho também investiga a integração de características às sequências positivas e negativas, buscando aumentar as medidas de desempenho.

Finalmente, é proposta uma metodologia para inclusão de conhecimento adquirido pelo classificador, onde, a partir do modelo obtido do treinamento com as sequências

[2]Técnica de replicação de casos da classe minoritária.
[3]Técnica de eliminação de casos da classe majoritária.

da região *upstream* e o SIT, as sequências da região *downstream* são classificadas, num primeiro momento, e posteriormente são incluídas em um novo treinamento.

1.1 Objetivos

O presente trabalho tem por objetivo desenvolver e propor nova metodologia para previsão de sítio de início de tradução em sequências de mRNA, visando obtenção de maior compreensão dos fenômenos biológicos relacionados a tradução de proteínas. Especificando os objetivos:

- Desenvolver uma ferramenta, PredicTIS, para preparação e análise das sequências de mRNA extraídas do banco de dados público *RefSeq*[4], para que as sequências possam então ser utilizadas pelo classificador SVM. A preparação das sequências envolve extração das sequências e características, codificação, particionamento e formatação. O Apêndice A apresenta a descrição e objetivos da ferramenta PredicTIS.
- Estudar, analisar, propor e implementar método para balanceamento das sequências de mRNA, bem como utilizar outros métodos de balanceamento com a finalidade de avaliar o método proposto. Esse passo se deve ao fato de que para cada sequência de mRNA há um ATG correspondente ao SIT, e todos os outros não iniciam a tradução de proteínas, dessa forma há o problema de classes desbalanceadas; e para resolvê-lo faz-se necessário a utilização de métodos balanceamento de dados.
- Propor uma metodologia para inclusão de conhecimento adquirido pelo classificador.
- Realizar experimentos e análises sobre as bases de dados já preparadas.
- Analisar a qualidade dos métodos de balanceamento de dados.

1.2 Justificativa

O controle do início da tradução é um dos mais importantes processos na regulação da expressão genética (NAKAGAWA et al., 2008). Dessa forma, a determinação do SIT não é uma tarefa trivial, ao mesmo tempo sendo de grande relevância para inferência genética. Segundo Liu et al. (2004), alta acurácia na predição pode ser útil para um melhor entendimento da proteína das sequências de nucleotídeos. Portanto, o estudo proposto permitirá:

[4]Disponível em www.ncbi.nlm.nih.gov/RefSeq

1. Classificação de fragmentos de sequência de mRNA, determinando se o mesmo possui ou não um sítio de início de tradução;

2. Avaliação do método de balanceamento *undersampling* proposto;

3. Maior compreensão dos fenômenos biológicos relacionados ao início da tradução de proteínas em sequências de mRNA, através da metodologia de inclusão de conhecimento adquirido pelo classificador.

1.3 Motivação

Numa sequência de mRNA há trincas de nucleotídeos com funções especiais. Sabe-se que o início da tradução quase sempre inicia-se em um códon ATG (*start codon*) e termina em um dos *stop codons* (TAA, TAG, TGA), características encontradas no processo de escaneamento. No entanto, a tradução em eucariotos pode não se iniciar no primeiro ATG da sequência (PEDERSEN; NIELSEN, 1997; HATZIGEORGIOU, 2002).

Dependendo da posição de início da síntese na fita de mRNA, a trinca de nucleotídeos selecionado para a síntese poderá variar, variando também os aminoácidos que serão gerados. Essa falta de padrão no início da tradução faz da predição de SIT uma tarefa complexa, por isso métodos computacionais de busca de um conjunto de características devem ser utilizados a fim de extrair o conhecimento implícito envolvido nesse processo.

A proposta de método de balanceamento de dados do tipo *undersampling* é particularmente importante em grandes bases de dados, onde técnicas de *oversampling* são inviáveis, já que aumentam ainda mais o tamanho das bases envolvidas.

1.4 Contribuições

Em virtude do que foi mencionado, esse trabalho traz as seguintes contribuições:

Inicialmente, é analisado e proposto o tamanho ideal de janela das sequências, sugerindo o uso de janelas de tamanho assimétrico, ou seja, tamanhos diferentes para região *downstream* e *upstream*.

Propõe-se também a inclusão de características juntamente às sequências extraídas das bases de dados, melhorando efetivamente o desempenho do classificador.

Analisa-se o desempenho dos métodos de balanceamento utilizados, bem como propõe-se um novo método de *undersampling* para balanceamento de classes, obtendo um

bom desempenho em relação aos métodos utilizados.

Realizam-se análises sobre a taxa de precisão avaliada, indicando que há um grande número de falso-positivos quando se utiliza todas as sequências negativas, tanto as que estão antes do SIT quanto as que estão depois, na criação do modelo; problema resolvido através da proposta de inclusão de conhecimento adquirido pelo classificador.

Dessa forma, esse trabalho vem contribuir de forma significativa com a sugestão de nova metodologia para previsão de SIT.

1.5 Visão geral desse trabalho

Esse trabalho está organizado em capítulos, a saber:

O Capítulo 2 traz uma descrição dos trabalhos relacionados ao problema de identificação do sítio de início de tradução de proteínas.

O Capítulo 3 descreve os materiais e métodos propostos, descrevendo as bases de dados, a forma de extração dos dados, a codificação, os métodos de balanceamento, a inclusão de características junto às sequências, a incorporação de conhecimento adquirido pelo classificador e o classificador utilizados, dentre outros critérios sugeridos por esse trabalho.

O Capítulo 4 apresenta os resultados obtidos e as principais discussões a respeito do SIT. Apresenta também a análise dos métodos de balanceamento utilizados.

O Capítulo 5 apresenta as conclusões e propostas de continuidade desse trabalho.

O Apêndice A apresenta as características da ferramenta implementada PredicTIS, bem como suas funcionalidades e objetivos.

2 ESTADO DA ARTE

Esse capítulo destina-se à apresentação dos principais métodos de previsão de SIT. Para melhor entendimento, foi dividido de acordo com a abordagem adotada: Análise Estatística, Redes Neurais Artificiais (RNA), *Support Vector Machines* (SVM) e Avaliação de Características.

2.1 Reconhecimento através de Análise Estatística

Desde 1982 a previsão de SIT tem sido extensivamente estudada utilizando-se métodos biológicos, estatísticos e técnicas computacionais (TZANIS; BERBERIDIS; VLAHAVAS, 2006).

Inicialmente, métodos estatísticos foram explorados a fim de descobrir padrões nas sequências positivas. O trabalho pioneiro de Kozak (1984), uma análise estatística sobre as sequências de 211 mRNAs de células eucarióticas, revelou que algumas posições das sequências de mRNAs, relativas ao SIT, são muito conservadas. A posição -3, ou seja, três nucleotídeos *upstream* do SIT, apresenta uma purina, nucleotídeo A (Adenina) ou G (Guanina), sendo em 79% das sequências, o nucleotídeo A. Há um predomínio do nucleotídeo C nas posições -1, -2, -4 e -5. Seguindo este raciocínio, determinou-se um consenso, denominado consenso de Kozak, ilustrado pela Figura 3.

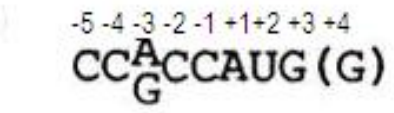

Figura 3: Consenso de Kozak.
Fonte: Kozak (1984)

Relatou também que em 95% dos casos, por ela selecionados, o SIT inicia-se no primeiro ATG da sequência.

Uma outra análise estatística foi realizada por Cavener e Ray (1991) sobre o *start* códon (códon inicializador da tradução) e sobre os *stop* códons (códons finalizadores da tradução), sendo desenvolvido um algoritmo para analisar as frequências dos nucleotídeos, e realizar uma análise de múltiplas posições de nucleotídeos. As sequências foram extraídas

do *GenBank*[1] e analisadas as posições -23 até +6 para início de tradução, e -9 até +13 para término de tradução.

No trabalho desenvolvido por Kozak (1984) foi identificada uma proporção de 79% de Adenina (A) na posição -3 (e 18% de Guanina (G)); já Cavener e Ray (1991), utilizando 2.595 sequências, constataram 58% de probabilidade de ser A na referida posição.

Nakagawa et al. (2008) realizaram análises comparativas entre 47 espécies, incluindo animais, fungos, plantas e protistas, revelando a existência de consensos para diferentes espécies. Com base nessa análise, foram identificadas as seguintes regiões de consenso: presença de uma purina (A ou G) na posição -3, presença de A ou C na posição -2, presença de C na posição +5. A posição -3 já havia sido descoberta por Kozak (1984), sendo, portanto confirmada nesse estudo.

2.2 Reconhecimento através de Redes Neurais Artificiais

Redes neurais artificiais (RN) têm se mostrado de grande utilidade em diversas tarefas, sendo também bastante utilizada para predição de sítio de início de tradução.

Em um dos trabalhos pioneiros de previsão de SIT, Stormo, Schneider e Gold (1982) utilizaram o algoritmo *perceptron* como classificador, e utilizaram *Escherichia coli* como organismo a ser realizada a previsão de SIT, um organismo procarioto, diferentemente da grande maioria dos trabalhos que são relativos a eucariotos. Utilizaram codificação de quatro bits (A = 1000, C = 0100, G = 0010 e T = 0001) e janelas de 51, 71 e 101 nucleotídeos centradas no ATG. Sendo obtidos os melhores resultados com janelas de 101 nucleotídeos.

Pedersen e Nielsen (1997) treinaram redes neurais artificiais com base de dados de vertebrados e obtiveram excelentes resultados, sendo um trabalho de grande expressão e relevância. Sendo assim, a base de dados criada por esses autores é até hoje explorada por outros trabalhos, a fim de comparar os resultados obtidos e validar a metodologia utilizada.

A base de dados de Pedersen e Nielsen (1997) é composta pelos vertebrados *Bos taurus* (touro), *Gallus gallus* (galinha), *Homo sapiens* (homem), *Mus musculus* (camundongo), *Oryctolagus cuniculus* (coelho), *Rattus norcevigus* (rato), *Sus scrofa* (porco) e *Xenopus laevis* (rã), sendo processada para obter as sequências de mRNA correspondentes. Dessas sequências, somente aquelas com o SIT anotado e com pelo menos 10 nucleotídeos

[1]Disponível em http://www.ncbi.nlm.nih.gov/genbank/

na região *upstream* e 150 nucleotídeos na região *downstream* foram selecionadas. Essas sequências foram então filtradas para remover aquelas pertencentes a uma mesma família gênica, genes homólogos de diferentes organismos e sequências repetidas. Nessa base de dados existem 13502 ATGs, sendo que 3312 (24,5%) são SITs; enquanto que os outros 10190 (75,4%) são não-SITs. Essa base de dados, bem como uma descrição detalhada da mesma, encontra-se disponível em `http://www.cbs.dtu.dk/databases/NetStart/`. Vale ressaltar que essa base de dados foi construída em 1997, época que ainda não existiam muitas sequências disponíveis nos bancos de dados públicos. Hoje é possível criar bases muito maiores e melhores a partir do *GenBank*, *RefSeq* ou outras bases de dados existentes (NOBRE; ORTEGA; BRAGA, 2007).

Pedersen e Nielsen (1997) utilizaram janelas de tamanhos de 13, 33, 53, 73, 93, 113, 133, 153, 173 e 203 nucleotídeos centradas no ATG. A codificação utilizada foi a mesma utilizada por Stormo, Schneider e Gold (1982), binária de quatro bits. Obtiveram sensibilidade (porcentagem de acertos dentro da classe positiva), especificidade (porcentagem de acertos dentro da classe negativa) e acurácia de 78%, 87% e 85%, respectivamente. O sistema desenvolvido por eles, denominado NetSart, também encontra-se disponível em `http://www.cbs.dtu.dk/services/NetStart/`.

Os autores realizaram ainda uma análise das sequências para revelar que características são importantes para distinguir SIT de não-SIT. Eles testaram base a base (da janela de 203 nucleotídeos), retirando uma a uma para descobrir o efeito dessa eliminação no desempenho do classificador. Eles descobriram que a posição -3 é crucial na identificação do SIT, o que corrobora com os estudos já citados.

Hatzigeorgiou (2002) também utilizou redes neurais em sequências de mRNA humano, alcançando acurácia de 94%. A base de dados utilizada é composta por sequências de cDNA completas. A autora utilizou dois módulos: *consensus-ANN* (analisa a vizinhança imediata do candidato a SIT) e *coding-ANN* (avalia as regiões *upstream* e *downstream* do candidato). O método final é a integração dos módulos.

O módulo *consensus-RN* avalia o SIT candidato e sua vizinhança mais imediata por meio de uma janela de 12 nucleotídeos. As sequências foram extraídas a partir das posições -7 a +5 e a codificação de quatro bits foi utilizada.

O módulo *coding-RN* avalia as regiões *upstream* e *downstream* do SIT candidato e trabalha com janelas de 54 nucleotídeos. Como cada três nucleotídeos formam um códon que se traduz em aminoácido, existem 64 códons possíveis. Assim, para avaliar as sequências dos 54 nucleotídeos, estas são transformadas em um vetor de 64 unidades

correspondendo à frequência de determinado códon na sequência.

Os dois módulos propostos são integrados da seguinte forma: dados um ATG candidato, o *consensus-RN* é aplicado para calcular um consenso S1. O *coding-RN* é aplicado então para obter um score S2 da região *upstream*. Isso se repete também para a região *downstream* e o score S3 é calculado. O score final para o candidato a SIT é então obtido por

$$S1 \times (S3 - S2)$$

Esse cálculo é realizado para todos os ATGs da molécula e o primeiro ATG que oferecer um score acima de 0.2 é considerado o SIT da sequência.

Note que no modelo de escaneamento do ribossomo, uma sequência de mRNA é escaneada do início para o final (da região 5' para região 3') testando cada ATG até encontrar aquele que será classificado como SIT; todos os outros ATGs posteriores a esse ATG são considerados como não-SITs. Ou seja, exatamente uma única previsão é feita por molécula nesta abordagem. Assim, a acurácia (94%) obtida por este modelo não pode ser comparada àquela obtida por outros modelos que avaliam todos os ATGs da molécula.

Tikole e Sankararamakrishnan (2008) utilizaram redes neurais artificiais, com duas camadas ocultas, para previsão de SIT em sequências de mRNA humano em que há um contexto de Kozak pobre. Eles definiram que um sítio de início de tradução tem contexto de Kozak (ver Figura 3 na Seção 2.1) pobre se purina e guanina são ausentes, respectivamente, nas posições -3 e +4. Dessa forma, desenvolveram redes neurais com o objetivo de previsão do início de tradução nessas sequências, que possuem um contexto de Kozak pobre. Identificaram que 731 sequências de mRNA humano tinham SIT com contexto de Kozak pobre. Avaliaram também vários tamanhos de janela, -5+8, -10+4, -15+4, -10+13 e -20+4, sendo a de melhor resultado a de tamanho -5+8, obtendo sensibilidade de 83% e especificidade de 73%.

2.3 Reconhecimento através de SVM

Utilizando SVM, Zien et al. (2000) alcançaram acurácia de 88,1% para a mesma base de dados de Pedersen e Nielsen (1997). Eles também utilizaram o mesmo tamanho de janela (203 nucleotídeos) e a mesma codificação. Os autores mostraram como obter melhorias usando uma nova função de kernel, chamada *locality-improved kernel*, com uma pequena janela em cada posição. O locality-improved kernel enfatiza correlações entre as posições da sequência que são próximas entre si, e um tamanho de 3 nucleotídeos

upstream e *downstream* foi empiricamente determinado como ótimo. Ou seja, a modificação consistiu em privilegiar correlações locais entre nucleotídeos, enquanto dependências entre nucleotídeos de posições distantes foram consideradas de pouca importância ou inexistentes. Com esta função de kernel, eles obtiveram sensibilidade, especificidade e acurácia de 69,9%, 94,1% e 88,1%, respectivamente.

Posteriormente, Zien et al. (2000) melhoraram estes resultados através de uma função de kernel mais sofisticada, também conhecida de *kernel de Salzberg*. O *kernel de Salzberg* é essencialmente um modelo probabilístico condicional das posições de dinucleotídeos. Por meio desse kernel, eles obtiveram uma acurácia de 88,6% para a mesma base de dados.

Li e Jiang (2004) utilizaram duas novas propostas para identificação do SIT. Primeiro, eles introduziram uma classe de novos kernels baseados em *string edit distance*, chamados de *edit kernels*, para serem utilizados com SVM. Segundo eles, os edit kernels são simples e possuem interpretações biológicas significativas e probabilísticas. Em um segundo momento, eles converteram a região *downstream* de um ATG em uma sequência de aminoácidos antes de aplicar o SVM. Eles mostraram que a abordagem adotada por eles é significativamente melhor (sensibilidade = 99,92%, especificidade = 99,82% e acurácia de 99,9% para a base de dados utilizadas por Pedersen e Nielsen (1997)).

Nobre, Ortega e Braga (2007) realizaram experimentos para descoberta de SIT utilizando-se 12 nucleotídeos na região *upstream* e *downstream*, além de SVM com funções simples de kernel. Inspirados por um estudo realizado a partir da frequência de trincas das sequências positivas e negativas, eles apresentaram uma nova metodologia para codificação. Assim, ao invés de codificar individualmente cada nucleotídeo, a codificação foi feita por trinca, com janela deslizante de tamanho 3. Com isso, eles obtiveram uma redução de 50% do número de entradas. Para balanceamento dos dados, utilizaram o algoritmo Smote (CHAWLA et al., 2002) para replicação das amostras da classe minoritária. Os autores utilizaram a base de dados de Pedersen e Nielsen, obtendo 95,63% de acurácia, e também de sequências extraídas do *RefSeq* (PRUITT; MAGLOTT, 2001) de cinco organismos: *Danio rerio*, *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus* e *Rattus norgegicus*, sob seis níveis de inspeção (*reviewed, provisional, predicted, validated, model* e *inferred*), onde, de acordo com seus resultados, as sequências com maior curadoria manual (*reviewed* e *validated*) tendem a oferecer os melhores resultados.

2.4 Reconhecimento através de avaliação de características

Até o momento, os métodos apresentados extraem informações para previsão do SIT basicamente através do conhecimento contido na própria sequência. No entanto, a partir de 2002, surgem métodos que utilizam características extraídas *a priori* como entradas para os classificadores.

Zeng, Yap e Wong (2002) empregam a técnica chamada de *k-grams* e poucos refinamentos para produzir características candidatas. Um k-gram é um padrão de k letras consecutivas, que podem ser aminoácidos ou nucleotídeos. Utilizando-se os nucleotídeos, existem 4^k possíveis *k-grams* para cada valor de k. Alguns valores típicos para k são 1, 2, 3, 4 ou 5.

Partindo-se do fato de que o processo biológico de traduzir nucleotídeos em aminoácidos a partir de 3 nucleotídeos (também chamado códon) inicia-se no SIT, Zeng, Yap e Wong (2002) utilizam *3-grams* , tanto da região *upstream* quanto da região *downstream* da molécula. Utilizam também 1-gram para considerarem cada posição especificamente.

Pelo fato de que o número de características geradas ser muito grande, Zeng, Yap e Wong (2002) propõe metodologia para selecionar as características mais importantes, utilizando o método de seleção de características baseado em correlação.

Esses autores selecionaram 9 (nove) características, descritas abaixo:

1. Posição -3;
2. ATG *upstream* em fase;
3. TAA *downstream* em fase;
4. TAG *downstream* em fase;
5. TGA *downstream* em fase;
6. CTG *downstream* em fase;
7. GAC *downstream* em fase;
8. GAG *downstream* em fase;
9. GCC *downstream* em fase.

Zeng, Yap e Wong (2002) mostraram também que essas 9 características são fundamentais para a previsão de SIT; e apresentaram explicações biológicas para algumas características, a saber:

- A posição -3 pode ser explicada pelo já conhecido consenso de Kozak, onde Kozak (1984) mostra que a posição -3 é altamente conservada para identificação do SIT;
- A característica ATG *upstream* em fase pode ser explicado pelo modelo de escaneamento do ribossomo (descrito na Figura 5 seção 3.2), onde o ribossomo escaneia o mRNA da região 5' para a região 3' até encontrar o primeiro ATG que contenha um contexto apropriado para a iniciar a tradução. Dessa forma, um ATG mais próximo da região 5' tem alta probabilidade de ser SIT. Consequentemente, a presença de um ATG na região *upstream* em fase com o SIT pode indicar que o SIT (previsto inicialmente) tem menos probabilidade de ser o SIT.
- As características TAA, TAG e TGA *downstream* em fase podem ser explicadas pelo fato de corresponderem aos *stop-códons* que estão em fase com o SIT. Assim, a presença de qualquer uma dessas características em fase nos próximos 100 nucleotídeos da região *downstream* indica que a proteína produzida não deverá possuir mais do que 33 aminoácidos; sendo esse quantitativo menor que a maioria das proteínas existentes, indicando que o ATG validado pode não ser o SIT (NOBRE; ORTEGA; BRAGA, 2007).

Já para as características CTG, GAC, GAG e GCC *downstream* em fase os autores não apresentaram explicações biológicas.

Utilizando-se as características descritas acima, Zeng, Yap e Wong (2002) obtiveram sensibilidade de 84,3%, especificidade de 86,1%, precisão de 66,3% e acurácia de 85,7% utilizando classificador Baysiano. Já por meio de SVM, encontraram sensibilidade 73,9%, especifidade de 93,2%, precisão de 77,9% e acurácia de 88,5%; utilizando redes neurais obtiveram sensibilidade de 77,6%, especifidade de 93,2%, precisão de 78,8% e acurácia de 89,4%; e sensibilidade de 74%, especificidade de 94,4%, precisão de 81,1% e acurácia de 89,4% ao utilizar árvores de decisão.

Li, See-Kiong e Wong (2002), Liu et al. (2004) apresentaram uma abordagem alternativa de geração de características baseada em aminoácidos. Nessa nova abordagem, utilizaram mesma metodologia apresentada por Zeng, Yap e Wong (2002), considerando 3-*grams* em fase com o SIT a ser previsto. A partir daí, todas possibilidades foram

convertidas em suas letras de aminoácidos correspondentes. Seguindo este raciocínio os seguintes *k-grams* foram gerados Liu et al. (2004):

1. X-up, número de vezes que o aminoácido X aparece na região *upstream*.
2. X-down, número de vezes que o aminoácido X aparece na região *downstream*.
3. XY-up, número de vezes que os dois aminoácidos XY aparecem como uma *substring* na região *upstream*.
4. XY-down, número de vezes que os dois aminoácidos XY aparecem como uma *substring* na região *downstream*.

Liu et al. (2004) também geraram características booleanas a partir dos segmentos de sequências extraídas da base de Pedersen e Nielsen (1997): presença ou ausência de um ATG na região *upstream*; presença ou ausência do nucleotídeo "A" ou "G" na posição -3; presença ou ausência de "G" na posição +4.

Para selecionar as características mais relevantes, utilizaram medida de entropia usando validação cruzada com 3 dobras (KOHAVI, 1995) e selecionaram 100 características. As nove principais características se repetem nas 3 dobras, e interessantemente, sete delas correspondem às características selecionadas por Zeng, Yap e Wong (2002):

- ATG-up corresponde a ATG *upstream* em fase;
- STOP-down corresponde a TAA, TAG e TGA *downstream* em fase;
- Pos-3AouG corresponde à posição -3;
- L-down corresponde à CTG *downstream* em fase;
- D-down corresponde à GAC *downstream* em fase;
- E-down corresponde à GAG *downstream* em fase;
- A-down corresponde à GCC *downstream* em fase.

Somente as características A-up e V-down não correspondem a nenhuma característica de estudos anteriores.

Tzanis, Berberidis e Vlahavas (2007) desenvolveram uma metodologia para predição de SIT, chamada MANTIS, com três componentes principais: *Consensus*, *Coding*

Region classification, e *ATG Location*. O componente *Coding Region Classification* envolve o treinamento de um modelo para classificar se o ATG de uma sequência é ou não SIT.

Eles utilizaram características selecionadas de estudos anteriores Liu et al. (2004), Tzanis, Berberidis e Vlahavas (2006) e utilizaram PCA (Análise de Componentes Principais) para obter o menor número de características não correlacionadas, visto que muitas dessas são correlacionadas entre si. Já o componente *Consensus* utiliza regras de Markov, que captura não somente a probabilidade de ocorrência de um nucleotídeo em determinada posição, mas também como a ocorrência de uma base interfere na ocorrência de alguma outra, na região próxima ao ATG (posições entre -7 e +5). O componente *ATG location* é considerado um novo modelo, sendo baseado na localização do ATG na sequência, de acordo com o *Ribossome Scanning Model* (RSM), descrito por Kozak (1984, 1999). A etapa final em MANTIS é a fusão da decisão dos componentes, sendo a saída a probabilidade estimada de um ATG ser um SIT, em vez de uma simples decisão verdadeiro/falso. Para predição foram utilizados quatro algoritmos de classificação: Naive Bayes, C4.5, K-vizinhos mais próximos e SVM, obtendo uma acurácia média e acurácia ajustada de 98,03% e 94,28%, respectivamente.

3 MATERIAIS E MÉTODOS

Esse capítulo trata-se da descrição de toda a metodologia utilizada para o desenvolvimento do trabalho, a saber: (1) descrição da base de dados utilizada; (2) a forma de extração de sequências positivas (que codificam proteínas) e negativas (que não codificam proteínas) do mRNA; (3) os métodos de balanceamento; (4) o classificador utilizado; (5) a inclusão de características; (6) incorporação do conhecimento adquirido pelo classificador; (7) as medidas de desempenho e o processo de validação utilizados.

As seções seguintes descrevem cada uma destas fases.

3.1 Descrição das bases de dados

A base de dados utilizada nesse trabalho foi extraída do banco de dados público *RefSeq* (PRUITT; MAGLOTT, 2001) do NCBI[1], e são referentes aos seguintes organismos: *Mus musculus* (camundongo) e *Rattus norvegicus* (rato).

As sequências do *RefSeq* fornecem uma coleção não redundante de sequências de DNA, RNA e proteínas. As principais características dessas sequências são (NOBRE; ORTEGA; BRAGA, 2007):

- Não redundância;
- Sequências de nucleotídeos e proteínas explicitamente indexados;
- Validação dos dados e consistência no formato dos arquivos (normalmente dois caracteres, seguido pelo undescore e seis dígitos);
- Curado pelo *staff* do NCBI e colaboradores, com status de revisão, indicados em cada registro.

O *RefSeq* existe sob seis níveis de confiança: *reviewed*, *validated*, *inferred*, *provisional*, *predicted* e *model*, correspondendo à diminuição do grau de inspeção, respectivamente. As sequências *reviewed*, por serem revisadas por membros do *staff* do NCBI e

[1] Órgão mantido pelo governo americano que disponibiliza, para acesso público, dados de pesquisas na área de biotecnologia. Disponível em http://www.ncbi.nlm.nih.gov/

seus colaboradores, são, de maneira geral, as melhores sequências disponíveis de um determinado organismo (NOBRE; ORTEGA; BRAGA, 2007). Somente sequências sob o nível de inspeção *reviewed*, já avaliado por Nobre, Ortega e Braga (2007), serão abordadas nesse trabalho.

A Figura 4 representa um fragmento de um arquivo *RefSeq* original extraído do NCBI, onde pode-se observar que o organismo é o *Mus musculus*, o nível de inspeção é *reviewed*, a região codificadora (CDS) tem início no nucleotídeo 44, ou seja, o início da tradução dessa molécula começa na posição 44 do mRNA; e tem término no nucleotídeo 1483.

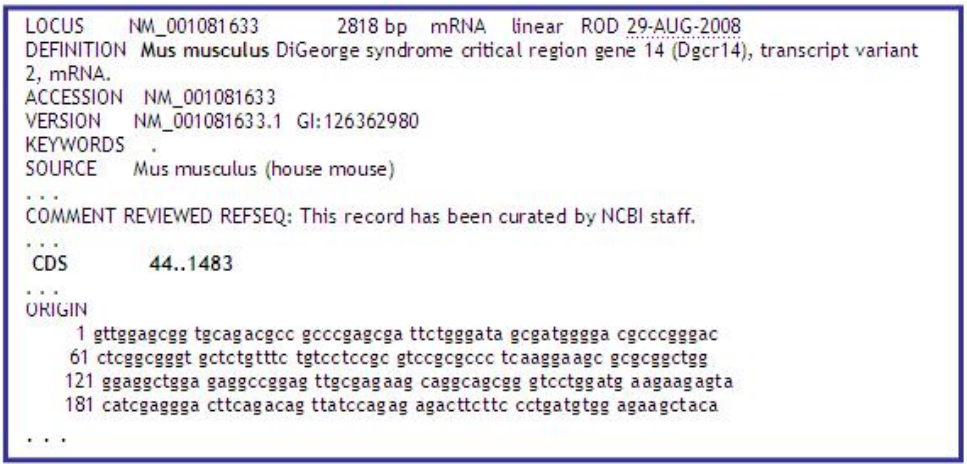

Figura 4: Fragmento de um arquivo do banco de dados *RefSeq* em seu formato original.

3.2 Extração das sequências positivas e negativas

Para se utilizar o classificador SVM, sequências positivas (SIT) e negativas (não-SIT) foram extraídas através de uma ferramenta implementada, PredicTIS, com variação de janelas dos seguintes tamanhos: -8+8 (8 nucleotídeos nas regiões *upstream* e *downstream*, respectivamente), -12+12, -20+20, -30+30, -40+40, -50+50, -60+60, -10+50, -50+10, -10+30, -10+20, -8+30, -12+20 e -12+30. Inicialmente, foram realizados testes experimentais com janelas de tamanho simétrico, por exemplo -12+12. No entanto, testes com janelas assimétricas mostraram-se mais eficientes.

As sequências foram extraídas apenas de arquivos contendo a quantidade mínima de nucleotídeos da região *upstream* da janela considerada. Dessa forma, todas as sequências que não continham esse número foram desconsideradas.

Ao extrair fragmentos de sequências negativas a partir das bases de dados, há duas classificações possíveis, de acordo com o alinhamento do ATG com o SIT: mesma fase de

leitura (MFL) ou fora de fase de leitura (FFL). Dizer que uma sequência está em fase com o SIT, significa que ela está alinhada com ele, ou seja, o início do ATG é uma posição múltipla de 3 das regiões *upstream* ou *downstream* referente ao SIT.

A Figura 5 apresenta exemplos de extração de sequências positivas e negativas dada uma molécula de mRNA. O SIT é determinado pelo ATG destacado em vermelho, e é representado pelas posições +1, +2 e +3. A Figura 5 (a) apresenta um exemplo de uma sequência positiva. As partes (b) e (c) da Figura 5 apresentam, respectivamente, um exemplo de sequência negativa fora de fase e na mesma fase de leitura.

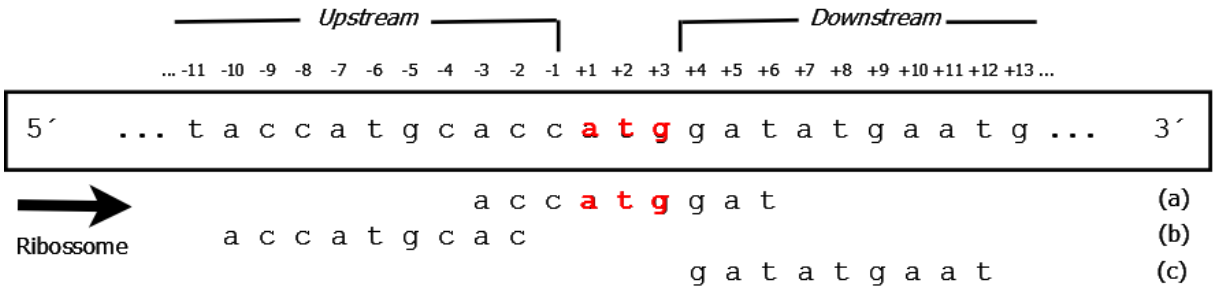

Figura 5: O ribossomo escaneia a sequência de mRNA da região 5' para a região 3' até encontrar um códon de ATG com contexto apropriado. A parte (a) da figura apresenta um exemplo de extração de sequências positivas (TIS), partes (b) e (c) apresentam sequências negativas fora de fase de leitura e negativas na mesma fase de leitura, respectivamente.

De acordo com os principais autores da literatura, as sequências foram codificadas utilizando-se o esquema de codificação de 4 bits já mencionado no estado da arte. Dessa forma, sem considerar a inclusão de características (abordado na Seção 3.4) e considerando-se, por exemplo, janela de tamanho -10+30, o classificador SVM possui 160 entradas (40 *bases* $\times$ 4 *bits* = 160), isto porque o tamanho das janelas é 10 nucleotídeos *upstream* e 30 nucleotídeos *downstream* centrados no ATG; e uma saída (0 ou 1), ou seja, a saída 1 representa que a sequência contém o códon ATG inicializador da tradução, e a saída 0 representa que a sequência não contém esse inicializador.

Nesse trabalho foram consideradas duas abordagens para a extração de sequências positivas e negativas. Na primeira, todas as sequências que possuíam ATG e que não eram SIT foram consideradas negativas. A segunda abordagem, chamada de inclusão de conhecimento adquirido (InAKnow), discutida posteriormente na Seção 3.6, considerou que todas as sequências da região *downstream* que continham ATG não possuíam classificação, uma vez que pelo modelo de escaneamento do ribossomo esses ATGs não são avaliados.

A Tabela 1 apresenta o número total de moléculas de mRNA para os organismos analisados (*Mus musculus* e *Rattus norvegicus*), além do número de sequências positivas

(POS), negativas fora de fase de leitura (FFL) e sem classificação, para as duas abordagens utilizadas (com ou sem InAKnow), com janela de tamanho de -10+30, visto que foi a janela que apresentou os melhores resultados.

Tabela 1: Quantidade total de sequências, número de sequências extraídas: positivas e negativas fora de fase de leitura (FFL), com janela de tamanho -10+30. Considerando as duas abordagens: com ou sem a metodologia de inclusão de conhecimento adquirido (InAKnow).

Organismo: Mus musculus		
	Sem InAKnow	Com InAKnow
Total moléculas	309	309
Sequências Positivas	269	269
Sequências Negativas	6256	327
Sequências sem classificação	-	5929
Organismo: Rattus novergicus		
Total moléculas	1317	1317
Sequências Positivas	101	101
Sequências Negativas	13245	305
Sequências sem classificação	-	12940

*Download in 01/03/2010.

Percebe-se claramente, por essa tabela, que esse problema é desbalanceado, justificando investimentos em métodos de balanceamento, objeto abordado por esse trabalho. Para o organismo *Mus musculus*, não considerando a metodologia de inclusão de conhecimento adquirido, o desbalanceamento é de 1:23, um exemplo de sequência positiva para 23 sequências negativas. Já para o organismo *Rattus norvegicus* a proporção é de 1:131.

Nota-se também que o número de sequências positivas extraídas não é igual ao número de moléculas de mRNA, visto que foram extraídas somente as sequências que tivessem CDS maior ou igual a 10 nucleotídeos (tamanho da região *upstream*). Em 1203 sequências do organismo *Rattus norvegicus* (que representa 91% das moléculas de mRNA) o CDS inicia-se em posição anterior ao décimo nucleotídeo da sequência. Essas sequências foram descartadas para as simulações com janela de tamanho de -10+30, e seguindo o mesmo raciocínio todas as sequências com tamanho da região *upstream* inferior ao tamanho da janela foram descartadas.

Além disso, foram descartadas também algumas moléculas, por essas não se inicializarem com o códon ATG, representando 7,44% para o organismo *Mus musculus* e 0,99% para o organismo *Rattus norvegicus*.

Da mesma forma, o problema de desbalanceamento de classes permanece para todos os outros tamanhos de janela analisados.

3.3 Balanceamento de classes

Uma base de dados é dita desbalanceada, no domínio de classificação, quando existem muito menos casos de algumas classes do que de outras (CHAWLA; JAPKOWICZ; KOTCZ, 2004), com proporções de, por exemplo, 1:100, 1:1.000, 1:10.000 e até 1:100.000. Este tipo de problema é de grande importância uma vez que conjuntos de dados com essa característica podem ser encontrados em diversos domínios. Muitos sistemas de aprendizado assumem que as classes estão balanceadas e, dessa forma, esses sistemas falham em induzir um classificador que seja capaz de predizer a classe minoritária com precisão na presença de dados com classes desbalanceadas (BATISTA; PRATI; MONARD, 2004). Muito frequentemente os classificadores tendem a valorizar as classes (casos) predominantes e a ignorar as classes (casos) de menor representação (MACHADO; LADEIRA, 2007).

O problema de previsão do SIT é inerentemente desbalanceado, visto que uma molécula de mRNA tem apenas um ATG que codifica proteína, enquanto todos os outros são não-SIT. Para o organismo *Mus musculus*, temos uma desproporção média de 1:23, e para o *Rattus novergicus* de 1:131.

Os métodos de amostragem para balanceamento de classes visam alterar a distribuição dos dados de treinamento, de modo a aumentar a acurácia de seus modelos (MACHADO; LADEIRA, 2007). Isto é alcançado com a eliminação de casos da classe majoritária (denominados *undersampling*) ou replicação de casos da classe minoritária (denominados *oversampling*). Na literatura são conhecidos os métodos *undersampling* e *oversampling* aleatórios, que não utilizam heurísticas na eliminação/replicação de casos, e aqueles que utilizam (MACHADO; LADEIRA, 2007; BATISTA; PRATI; MONARD, 2004; CHAWLA et al., 2002; YEN; LEE, 2009).

Segundo Batista, Prati e Monard (2004), diversos autores concordam que os métodos de amostragem que não utilizam heurísticas podem provocar distúrbios indesejados nos modelos gerados. A simples replicação de casos da classe minoritária pode causar *overfitting*, ao passo que a eliminação aleatória de casos da classe majoritária pode remover informação importante para o processo de aprendizagem.

No presente trabalho, foram comparados os seguintes métodos de balanceamento: *Random undersampling* (BATISTA; PRATI; MONARD, 2004; MACHADO; LADEIRA, 2007), SBC (*Sampling Based on Clustering*) (YEN; LEE, 2009), Smote(*Synthetic minority oversampling technique*) (CHAWLA et al., 2002) e M-Clus (*Majority class undersampling based in Clustering*), proposto nesse trabalho.

3.3.1 *Random undersampling*

Método que elimina aleatoriamente casos da classe majoritária, a fim de igualar à quantidade da classe minoritária. Essa abordagem não utiliza heurística alguma para realizar a eliminação. Assim, pode remover informação importante para o processo de aprendizagem (BATISTA; PRATI; MONARD, 2004). O método *Random undersampling* foi utilizado nesse trabalho a fim de avaliar e validar os outros métodos utilizados e propostos.

3.3.2 *SBC (Sampling Based on Clustering)*

O método SBC (*Sampling Based on Clustering*), proposto por Yen e Lee (2009), considera que em uma base de dados há diferentes *clusters* com diferentes características. Nesse método, a base de dados completa, composta pelas classes minoritária e majoritária, é agrupada em k *clusters*. A partir desses *clusters* são selecionados exemplos da classe majoritária, de acordo com a proporção de exemplos dessa classe ($Size_{MA}$), e classe minoritária ($Size_{MI}$), em cada cluster i. O número de exemplos selecionados da classe majoritária no cluster i, representado por $SSize^i_{MA}$, é calculado pela Equação 3.1.

$$SSize^i_{MA} = (m \times Size_{MI}) \times \frac{Size^i_{MA}/Size^i_{MI}}{\sum_{i=1}^{k} Size^i_{MA}/Size^i_{MI}} \tag{3.1}$$

Na Equação 3.1, a expressão $m \times SizeMI$ indica o número total de exemplos selecionados da classe majoritária que deve haver no arquivo de treinamento final e m indica a proporção entre as classes majoritária e minoritária; nesse caso 1:1.

A expressão $\sum_{i=1}^{k} Size^i_{MA}/Size^i_{MI}$ indica a proporção do número total de exemplos da classe majoritária pelo número de exemplos da classe minoritária em todos os *clusters*. Dessa forma, a Equação 3.1 determina que a maioria dos exemplos da classe majoritária serão selecionados nos *clusters* que houver maior predomínio de exemplos da classe majoritária. Nesse trabalho, foi utilizado o método de clusterização k-means, com 4 *clusters* e função de distância euclidiana, quantidade já avaliada por Yen e Lee (2009).

O pseudocódigo deste método é descrito no Algoritmo 1.

Algoritmo 1: SBC (*Sampling Based on Clustering*)

1: **Entrada:** Conjunto de exemplos da classe majoritária (MA) e minoritária (MI); Número de exemplos da classe majoritária ($SizeMA$) e minoritária ($SizeMI$);
2: **Saída:** Base de dados de treinamento balanceada
3: Determinar a proporção entre $SizeMA$ e $SizeMI$ na base de dados de treinamento
4: Clusterizar todos os exemplos da base de dados em alguns *clusters*
5: Determinar o número de exemplos da classe majoritária a serem selecionados em cada *cluster* utilizando a Equação 3.1, e selecionar aleatoriamente exemplos da classe majoritária em cada cluster
6: Combinar os exemplos selecionados da classe majoritária e todos os elementos da classe minoritária para obter a base de dados de treinamento

Fonte: Yen e Lee (2009), adaptado pela autora.

3.3.3 M-Clus (Majority class undersampling based in Clustering)

Método proposto por esse trabalho, tem como principal característica a realização de uma clusterização com as sequências da classe majoritária. A partir dessa clusterização, sequências mais significativas de cada *cluster* são selecionadas para o treinamento. Intuitivamente, cada grupo é composto por padrões que são similares entre si e dissimilares em relação aos padrões de outros grupos (JAIN; MURTY; FLYNN, 1999).

Sumariamente, o método M-Clus funciona da seguinte maneira: a partir da base de dados desbalanceada (Figura 6), a classe majoritária é agrupada em k *clusters*, onde k é a quantidade de exemplos da classe minoritária (Figura 7); para cada cluster é selecionado um exemplo da classe majoritária. O exemplo selecionado é o que está mais próximo do centróide do cluster, sendo a distância utilizada a Euclidiana.

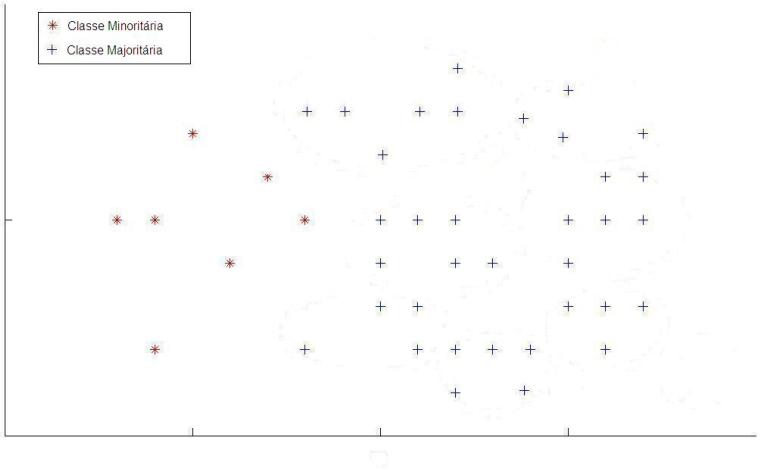

Figura 6: Exemplo de base de dados desbalanceada.

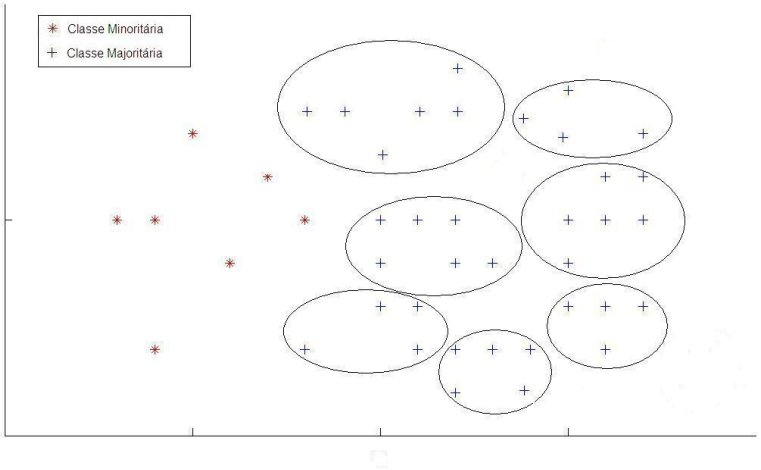

Figura 7: Funcionamento do método M-Clus. Os exemplos da classe majoritária são divididos em k *clusters*, onde k é a quantidade de elementos da classe minoritária. Posteriormente, são selecionados um exemplo por *cluster*, igualando, assim, a quantidade de elementos das classes majoritária e minoritária.

Por se tratar de um método de *undersampling*, o método M-Clus é particularmente interessante em grandes bases de dados, onde técnicas de *oversampling* são inviáveis, já que aumenta significativamente o tamanho das bases de dados envolvidas.

A abrangência dos exemplos selecionados da classe majoritária é outro ponto importante. Isso se deve pelo fato de que ao realizar clusterização com todos os exemplos da classe majoritária e extrair a sequências mais significativa de cada cluster, os exemplos de cada cluster são representados pelo exemplo selecionado. Assim não há perda de muita informação, já que os exemplos estão próximos do exemplo selecionado.

O pseudocódigo do M-Clus é apresentado no Algoritmo 2.

Algoritmo 2: M-Clus (*Majority class undersampling based in Clustering*)

1: **Entrada:** Conjunto de exemplos da classe majoritária (MA) e minoritária (MI); Número de exemplos da classe minoritária (SizeMI);
2: **Saída:** Seleção de exemplos da classe majoritária ($SSMI$)
3: Realizar clusterização com os exemplos da classe majoritária em k *clusters*, onde k= SizeMI
4: **for** $i = 1$ to k **do**
5: Calcular distância dos exemplos do *cluster* i ao centróide do *cluster* i
6: Selecionar o exemplo do cluster i com menor distância ao centróide
7: Retirar o exemplo selecionado e inclui-lo ao conjunto $SSMI$
8: **end for**

Para a realização da clusterização, foi utilizado o algoritmo k-means, proposto por Macqueen (MACQUEEN, 1967), e voltado para aplicações em que todas as variáveis são quantitativas e as dissimilaridades entre elas podem ser medidas em um espaço Euclidiano (BERKHIN, 2002).

O algoritmo inicia com a escolha dos k elementos que formaram as sementes iniciais. Essa escolha pode ser feita, dentre outras formas, selecionando-se as k primeiras observações, ou de maneira completamente aleatória ou ainda de forma que os seus valores sejam bastante diferentes.

Escolhidas as sementes iniciais, é calculada a distância de cada elemento em relação às sementes, agrupando o elemento ao grupo que possuir a menor distância (mais similar) e recalculando o centróide do mesmo. O processo é repetido até que todos os elementos façam parte de um dos *clusters*. Após agrupar todos os elementos, procura-se encontrar uma partição melhor do que a gerada arbitrariamente. Para isto, calcula-se o grau de homogeneidade interna dos grupos através da Soma de Quadrados Residual (SQRes), que é a medida usada para avaliar o quão boa é uma partição. Após o cálculo, move-se o primeiro objeto para os demais grupos e verifica-se se existe ganho na Soma de Quadrados Residual, ou seja, se ocorre uma diminuição no valor da SQRes. Existindo, o objeto é movido para o grupo que produzir o maior ganho, a SQRes dos grupos é recalculada e passa-se ao objeto seguinte. Depois de um certo número de iterações ou, não havendo mais mudanças, o processo é interrompido (MACQUEEN, 1967).

Para efeito de avaliação da abrangência das sequências selecionadas, a quantidade de *clusters* (k) variou entre o total ($kSizeMI$), metade ($kSizeMI/2$) e um terço ($kSizeMI/3$) das sequências da classe minoritária; e para cada cluster é retirada uma, duas e três sequências, respectivamente. Para escolha das sequências, retiraram-se aquelas

com menores distâncias até o centróide do *cluster*. Os melhores resultados foram quando se utilizar a quantidade de cluster igual a quantidade de elementos da classe minoritária, podendo ser explicado pela abrangência dos exemplos selecionados, quanto mais *clusters* temos menos exemplos em cada cluster e consequentemente os exemplos são mais próximos entre si, assim ao escolher um elemento por cluster esse representa de forma melhor os exemplos do cluster.

3.3.4 Smote (Synthetic minority over-sampling technique)

Smote é um método de *oversampling* que consiste na geração de casos sintéticos (casos artificiais) para a classe de interesse a partir dos casos já existentes. Este algoritmo, desenvolvido por Chawla et al. (2002), gera novos exemplos da classe minoritária através da interpolação entre vários exemplos da amostra. Esses novos casos são gerados na vizinhança de cada caso da classe minoritária, de forma a fazer crescer a região de decisão e, assim, aumentar o poder de generalização dos classificadores gerados para estes dados (MACHADO, 2007). O Algoritmo 3 apresenta o pseudocódigo do Smote.

Algoritmo 3: Smote (*Synthetic minority over-sampling technique*)

Entrada: Conjunto de exemplos da classe minoritária (S); Número de exemplos da classe minoritária ($SizeMI$); Número de novos casos desejado para cada exemplo da classe minoritária (qtd); Número de vizinhos mais próximos considerados (k);
Saída: Conjunto de exemplos da classe minoritária acrescido dos novos exemplos criados
for $i = 1$ to $SizeMI$ **do**
    Calcular k vizinhos mais próximos do caso i
    for $j = 1$ to qtd **do**
        for $attr = 1$ to $qtdAttb$ **do**
            Crie um valor entre i e um dos k vizinhos
        end for
    end for
end for

Fonte: Chawla et al. (2002).

3.4 Inclusão de características

Nesse trabalho, além da própria sequência utilizada, foram incluídas também algumas características, relatadas em estudos anteriores (LIU et al., 2004; TZANIS; BERBERIDIS; VLAHAVAS, 2007; ZENG; YAP; WONG, 2002; LI; LEONG, 2005; TZANIS; BERBERIDIS; VLA-

HAVAS, 2006). Assim, ao gerar os conjuntos de treinamento e de teste, formou-se uma combinação entre as sequências extraídas e as características selecionadas: presença ou ausência de um ATG *upstream* em fase com o SIT (ZENG; YAP; WONG, 2002; TZANIS; BERBERIDIS; VLAHAVAS, 2006; LIU et al., 2004; LI; LEONG, 2005; TZANIS; BERBERIDIS; VLAHAVAS, 2007), presença ou ausência de um stop códon nos próximos 100 nucleotídeos (ZENG; YAP; WONG, 2002; TZANIS; BERBERIDIS; VLAHAVAS, 2007; LIU et al., 2004; LI; LEONG, 2005; TZANIS; BERBERIDIS; VLAHAVAS, 2006), presença ou ausência dos códons CTG, GAC, GAG e GCC na região *downstream* em fase com o SIT (ZENG; YAP; WONG, 2002; LIU et al., 2004; TZANIS; BERBERIDIS; VLAHAVAS, 2007).

O ATG *upstream* em fase pode ser explicado pelo modelo de escaneamento do ribossomo, da região 5' para a região 3', até encontrar o primeiro ATG que contenha um contexto de tradução. Assim, um ATG mais próximo da região 5' tem uma alta probabilidade de ser SIT. Consequentemente, a presença de um ATG na região *upstream* em fase com o SIT pode indicar que o SIT (previsto inicialmente) tem menos chances de ser o SIT. Esse fato foi relatado também por Rogozin et al. (2001), que mostraram que existe um correlação negativa entre a qualidade do contexto do início de tradução e o número de ATGs na região *upstream*. Essa característica mostrou-se de grande relevância nesse trabalho, visto que os melhores resultados foram obtidos com a inclusão dessa característica combinada a outras.

A presença ou não de *stop-códons* (TAA, TAG e TGA) em fase na região *downstream* nos próximos 100 nucleotídeos pode ser explicada pelo processo biológico de tradução. Esse processo de traduzir os códons, que estão em fase, em aminoácidos termina quando um *stop-códon* em fase é encontrado. Dessa forma, a presença de qualquer um *stop-códon* em fase nos próximos 100 nucleotídeos sinaliza que a proteína não deverá ter mais de 33 aminoácidos. Isto é menor do que a maioria das proteínas existentes, indicando que o ATG pode não ser o SIT (NOBRE; ORTEGA; BRAGA, 2007).

Algumas características apresentadas em trabalhos anteriores, como as posições -3, +4, o tamanho das regiões *upstream* e *downstream* não foram consideradas já que estão implícitas nas sequências extraídas.

3.5 Support Vector Machine - SVM

Support Vector Machines (SVM) é uma técnica de aprendizado de máquina, fundamentada nos princípios indutivos da Minimização do Risco Estrutural. Estes princípios

são provenientes da Teoria do Aprendizado Estatístico (SEMOLINI, 2002).

Originalmente denominado "classificador de margem ótima", ela foi introduzida em (BOSER; GUYON; VAPNIK, 1992) para aplicação em problemas de classificação binária. Em Cortes e Vapnik (1995), sendo chamado de "máquina de vetores de suporte", foi proposta uma maneira de se lidar eficientemente com os exemplos que são notadamente incorretos, isto é, que estão fora da região de sua classe. A denominação "máquina de vetores de suporte", ou Support Vector Machine (SVM), enfatiza a importância que os vetores mais próximos da margem de separação representam, uma vez que eles determinam a complexidade da SVM (BURBIDGE; BUXTON, 2001).

SVM implementa um mapeamento não-linear (executado por um produto interno *kernel* escolhido a priori) dos dados de entrada para um espaço característico de alta-dimensão, em que um hiperplano ótimo é construído para separar os dados linearmente em duas classes (SEMOLINI, 2002).

Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço característico é aquele que apresenta a máxima margem de separação. A Figura 8 apresenta uma visão geométrica da construção do hiperplano ótimo para um espaço bi-dimensional, além da interpretação dos vetores-suporte.

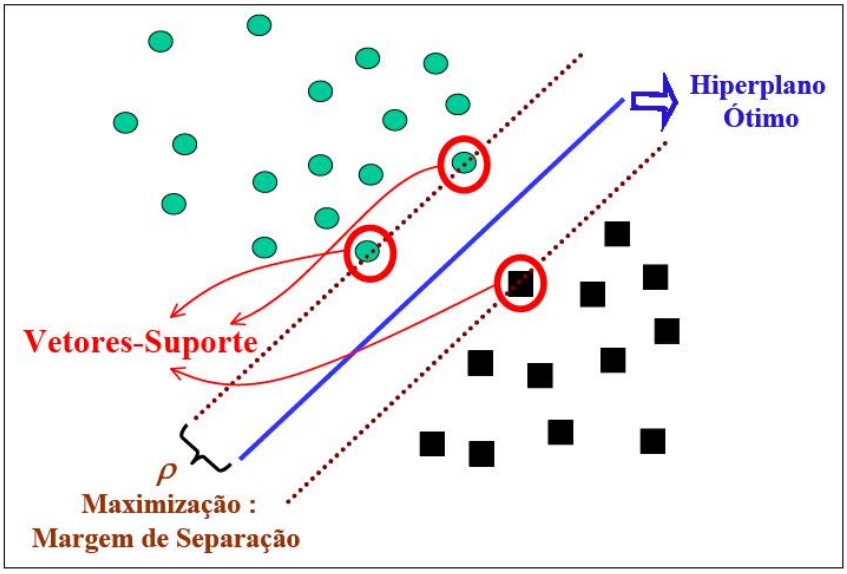

Figura 8: Hiperplano ótimo separando os dados com a máxima margem ρ, os vetores-suporte (support vectors) e uma distribuição dos dados no R^2.
Fonte: Semolini (2002)

SVM pode ser caracterizado como um algoritmo de aprendizado de máquina capaz de resolver problemas de classificação lineares e não lineares. A ideia principal da classificação por vetor de suporte é separar exemplos com uma superfície de decisão linear e

maximizar a margem de separação entre as classes a serem classificadas (NOBRE; ORTEGA; BRAGA, 2007).

Seja o conjunto de treinamento $\{x_i, y_i\}_i^N = 1$, com cada vetor de entrada $x_i \in \Re_n$ e saída binária correspondente $y_i \in \{-1, +1\}$. Dado um vetor de entrada x, a saída da SVM é representada por $f(x)$.

A SVM realiza um mapeamento não linear dos dados em um espaço de dimensão mais elevada. Nessa nova dimensão, os pontos que representam os dados das duas classes são considerados linearmente separáveis (SEMOLINI, 2002). Um hiperplano ótimo (com a maior margem de separação possível) é construído para separar os vetores da classe -1 dos da classe +1. A superfície de decisão $f(x) = 0$ criada pela SVM é representada por:

$$\omega^t \varphi(x) + b = 0$$

onde $\omega \in \Re_n$ é o vetor de pesos, b é o termo de polarização e $\varphi(\cdot)$ é o mapeamento realizado em um espaço de dimensão elevada, conhecido como espaço de características (ALMEIDA, 2002).

Se o espaço de características, obtido após o mapeamento, possuir dimensão 2, a superfície de separação é uma reta. Se esse espaço for de ordem n, a superfície é um hiperplano $(n-1)$-dimensional. A superfície de decisão divide o espaço de características em dois sub-espaços, um para cada classe.

A classificação de cada padrão de treinamento é dada em relação a sua proximidade em relação às margens, quer à positiva $\omega^t \varphi(x) + b = +1$ quer à negativa $\omega^t \varphi(x) + b = -1$, de acordo com sua classe. O padrão i é considerado corretamente classificado se ele se encontra fora da margem de separação de sua classe, ou seja, quando

$$\begin{cases} \omega^t \varphi(x_i) + b \geq +1, & \text{se } y_i = +1 \\ \omega^t \varphi(x_i) + b \leq -1, & \text{se } y_i = -1 \end{cases}$$

A equação acima pode ser expressa de uma forma mais compacta como

$$y_i \left[\omega^t \varphi(x_i) + b\right] \geq 1$$

para o padrão de entrada i.

O processo de treinamento de uma SVM consiste na obtenção de valores para os

pesos ω e para o termo de polarização b, de forma a minimizar uma certa função de custo $J_P(\omega, b)$.

Nesse trabalho, utilizou-se o algoritmo SVM^{light}, implementado por Joachims (1999) e disponível em `http://svmlight.joachims.org/`.

O motivo da escolha do algoritmo SVM^{light} para implementar o treinamento do SVM foi baseado nos seguintes fatores (NOBRE; ORTEGA; BRAGA, 2007):

- É projetado para operar com grande número de dados de treinamento não tendo problema com quantidade de dados armazenados na memória;
- O tempo de processamento para grandes tarefas é muito satisfatório;
- Trabalha com problemas de todos os tipos: classes separáveis, classes não separáveis e ainda problemas com muita interseção (ruído) entre as classes.

3.6 Inclusão de conhecimento adquirido

Conforme apresentado na Figura 5, o modelo de escaneamento supõe que o ribossomo se liga primeiro à região 5' do mRNA e percorre em direção à região 3' até encontrar o primeiro ATG da sequência (KOZAK, 1984). No entanto, existem exceções: devido a um contexto pobre (com ruídos, por exemplo), esse primeiro ATG pode ser ignorado. Considerando esse modelo de escaneamento do ribossomo, temos que apenas os ATGs que estão na região *upstream* do SIT e o próprio SIT possuem classificação. A Figura 5 apresenta esse modelo: o ribossomo não identifica o(s) primeiro(s) ATGs da sequência como sendo SIT, ele segue até o segundo ou o terceiro ou mais ATGs, classificando-os como não-SIT até encontrar o ATG que ele classifica como SIT. Neste sentido, não se tem um classificação exata para nenhum dos ATGs que estão na região *downstream* do SIT (NOBRE; ORTEGA; BRAGA, 2007).

Neste sentido, esse trabalho apresenta uma metodologia para classificar essas sequências e posteriormente incluí-las no processo de treinamento e classificação. Este processo de inclusão de conhecimento adquirido é apresentado na Figura 9.

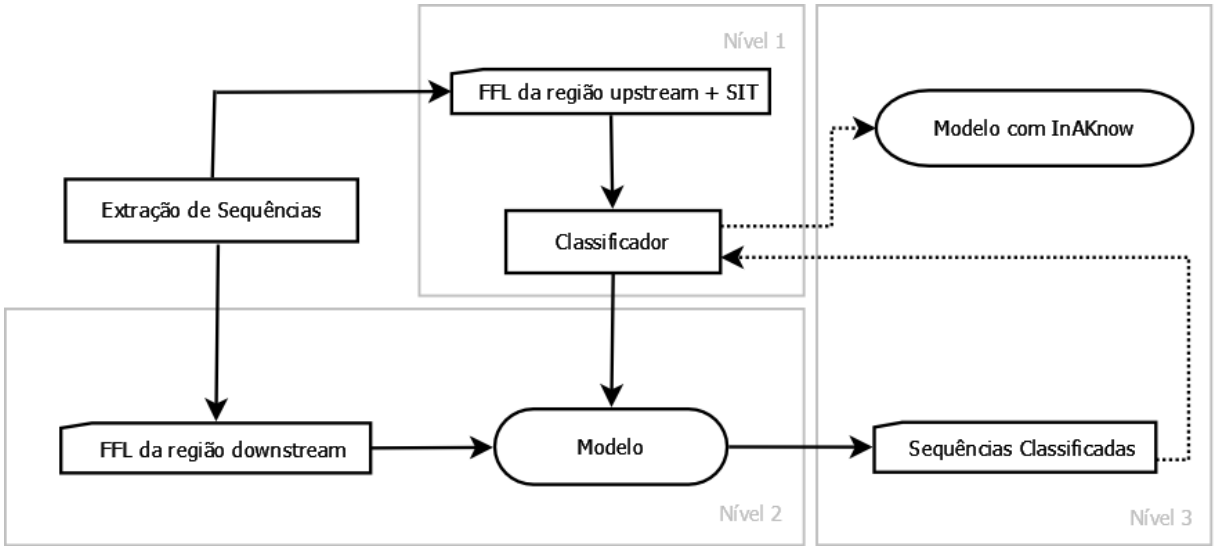

Figura 9: Seguindo o modelo de escaneamento do ribossomo, as sequências extraídas da região *downstream* do SIT não possuem classificação. Assim, no Nível 1 dessa metodologia proposta, as sequências da região *upstream* (positivas e negativas) são treinadas pelo classificador, e no Nível 2, o modelo criado por esse treinamento é aplicado às sequências da região *downstream* (sem classificação). Posteriormente, as sequências da região *downstream* (já classificadas) juntamente com as da região *upstream* são inseridas em um novo treinamento, gerando assim um novo modelo (Nível 3).

De acordo com essa metodologia, os seguintes passos são executados:

1. Obtenção de um modelo, considerando-se apenas as sequências positivas (SIT) e as negativas (não-SIT) fora de fase de leitura (FFL) contidas na região *upstream* (Figura 9 - Nível 1);

2. Classificação dos ATGs pertencentes à região *downstream*, por meio do modelo gerado no passo anterior (Figura 9 - Nível 2);

3. Novo treinamento com todas as sequências, inclusive com aquelas classificadas pelo passo anterior (Figura 9 - Nível 3). Nessa etapa, há uma diminuição no desbalanceamento de classes, devido à inclusão das sequências classificadas como positivas pelo modelo. Dessa forma, a proporção de 1:23 passou a ser de 1:5, para o organismo *Mus musculus*, e de 1:131 para 1:34 para o *Rattus novergicus*.

3.7 Medidas de desempenho

Foram utilizadas cinco medidas para se avaliar o desempenho do classificador: acurácia (Ac), precisão(Pr), sensibilidade(Se), especificidade(Ep) e a acurácia ajustada (Adj) (NOBRE; ORTEGA; BRAGA, 2007; TZANIS; BERBERIDIS; VLAHAVAS, 2007; ZIEN et al.,

2000; ZENG; YAP; WONG, 2002; LIU et al., 2004).

A acurácia mede a proporção de predições corretas, conforme descrito na Equação 3.2.

$$Ac = 100 * \frac{TP + TN}{TP + TN + FN + FP} \tag{3.2}$$

onde TP, TN, FP e FN denotam o número de verdadeiros positivos, verdadeiros negativos, falso positivos e falso negativos, respectivamente.

A precisão mede a proporção dos possíveis SIT que são certamente SIT (Equação 3.3).

$$Pr = 100 * \frac{TP}{TP + FP} \tag{3.3}$$

A sensibilidade, também conhecida como taxa de verdadeiro-positivo, refere-se à porcentagem de acertos dentro da classe positiva, ou seja, mede a proporção de SIT que foi corretamente classificada como SIT (Equação 3.4).

$$Se = 100 * \frac{TP}{TP + FN} \tag{3.4}$$

A especificidade, também conhecida como taxa de verdadeiro-negativo, refere-se à porcentagem de acertos dentro da classe negativa, ou seja, mede a proporção de não-SIT que foi corretamente reconhecida como não-SIT (Equação 3.5).

$$Sp = 100 * \frac{TN}{TN + FP} \tag{3.5}$$

Já a acurácia ajustada é obtida como sendo a média das medidas de sensibilidade e especificidade (Equação 3.6).

$$Adj = \frac{Sensibilidade + Especificidade}{2} \tag{3.6}$$

3.8 Validação

Utilizou-se a validação cruzada com dez dobras (10-*fold cross validation*), indicada por Kohavi (1995) como a forma de avaliação mais eficiente para seleção de modelos.

No método de validação cruzada com k-dobras o conjunto D, de tamanho N, é dividido em k subconjuntos (dobras) mutuamente excludentes de tamanhos aproximadamente iguais. Onde $1 < k <$ N. O treinamento e o teste são realizados k vezes, sempre utilizando k - 1 subconjuntos para treinamento e o subconjunto que restou para teste.

A principal vantagem do método de validação cruzada com k-dobras é que todos os exemplos do conjunto de dados são eventualmente usados para treinamento e teste. A Figura 10 mostra um esquema ilustrando o método de validação cruzada com k-dobras, mais precisamente uma validação cruzada com 10-dobras.

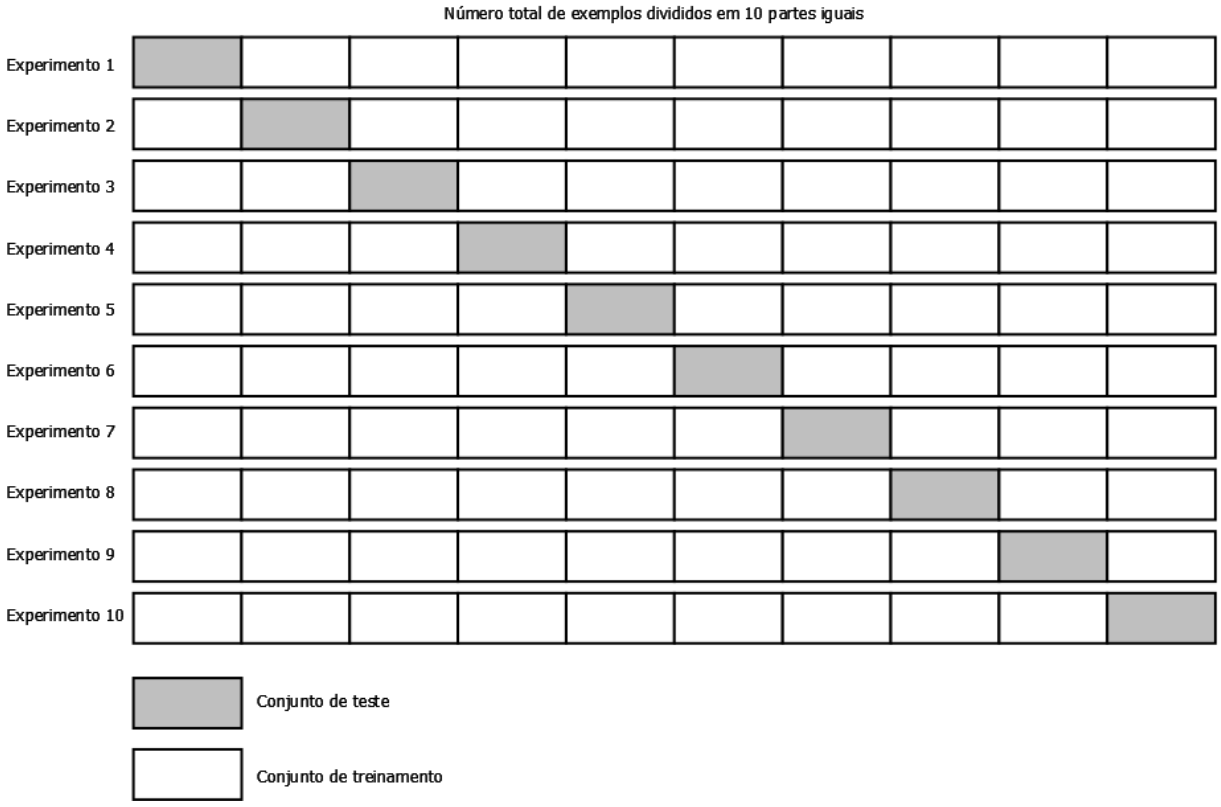

Figura 10: Esquema mostrando o exemplo do método de validação cruzada com 10-dobras.

O processo de validação cruzada utilizada nesse trabalho seguiu a metodologia sugerida por Machado e Ladeira (2007), onde inicialmente a base de dados desbalanceada é dividida em dez subconjuntos. Em seguida, nove subconjuntos são reservados para o treinamento e apenas um é destinado ao teste. O conjunto de treinamento é, então, balanceado com a aplicação dos métodos de balanceamento supracitados. Esses dados são utilizados no SVM durante o treinamento e testado com o subconjunto reservado. Esse processo é repetido dez vezes e ao final das dez dobras é calculada a média e o desvio padrão dos desempenhos obtidos.

4 RESULTADOS E DISCUSSÕES

Nesse capítulo são apresentados os resultados obtidos, utilizando-se a metodologia descrita no Capítulo 3.

4.1 Avaliação do tamanho da janela

Foram realizados extensivos experimentos a fim de se avaliar o tamanho de janela que oferecesse o melhor desempenho. Testes iniciais com tamanhos de janelas simétricos, por exemplo -10+10 (dez nucleotídeos nas regiões *upstream* e *downstream*, respectivamente), foram realizados, aumentando assim tanto o tamanho da janela da região *downstream* quanto da região *upstream*. As Tabelas 2 e 3 apresentam os resultados desses experimentos para o organismo *Mus musculus*, apresentando a média das 10 execuções geradas pela validação cruzada, e o desvio padrão, entre parênteses. Percebe-se que, com o aumento do tamanho da janela simétrico, há um aumento nas taxas de acurácia e especificidade. Em contrapartida, há uma queda no índice da sensibilidade (sendo essa uma importante taxa). Sendo essa queda interpretada pela perda de aprendizado do classificador SVM quanto às sequências positivas. Dessa forma, ao aumentar o tamanho da janela, de forma simétrica, o classificador tem o desempenho pior em relação ao acerto da classe positiva, indicando assim que a sensibilidade está relacionada com os nucleotídeos da região próxima ao SIT. A diferença de desempenho do classificador entre as Tabelas 2 e 3 deve-se à inclusão das características ATG+STOP+GAG na Tabela 3, sendo a avaliação das características apresentada na Seção 4.2.

Tabela 2: Comparação de desempenho em função do tamanho da janela para o organismo *Mus musculus*, sem a inclusão de características.

Organismo: Mus musculus					
Tamanho Janela	**Ac**	**Pr**	**Se**	**Sp**	**Adj**
Avaliação de janelas de tamanho simétrico					
-8+8	87,77 (1,50)	22,46 (2,41)	79,82 (9,16)	88,11 (1,69)	83,97 (4,25)
-12+12	91,35 (1,15)	29,91 (3,87)	81,13 (7,56)	91,77 (1,11)	86,45 (3,90)
-20+20	94,03 (0,69)	39,08 (4,15)	81,06 (5,96)	94,58 (0,66)	87,82 (3,03)
-30+30	96,42 (0,61)	54,08 (5,68)	81,99 (9,10)	97,03 (0,63)	89,51 (4,48)
-40+40	97,49 (0,55)	66,18 (7,68)	77,06 (6,66)	98,32 (0,63)	87,69 (3,22)
-50+50	98,21 (0,41)	77,95 (6,78)	74,17 (8,29)	99,16 (0,30)	86,66 (4,15)
-60+60	98,27 (0,53)	82,74 (4,66)	68,20 (11,75)	99,45 (0,18)	83,83 (5,86)
Avaliação de janelas de tamanho assimétrico					
Avaliação região upstream					
-8+30	94,35 (0,89)	41,07 (5,44)	82,03 (8,65)	94,87 (0,96)	88,45 (4,22)
-10+30	**95,23** (1,09)	**46,44** (9,42)	**82,13** (8,94)	**95,79** (1,12)	**88,96** (4,48)
-12+30	94,77 (0,89)	43,99 (10,75)	81,28 (9,13)	95,29 (1,19)	88,28 (4,44)
-30+30	96,42 (0,61)	54,08 (5,68)	81,99 (9,10)	97,03 (0,63)	89,51 (4,48)
-50+30	97,39 (0,60)	63,65 (10,54)	78,45 (8,34)	98,14 (0,67)	88,30 (4,08)
Avaliação região downstream					
-10+10	89,96 (1,35)	27,01 (3,19)	82,80 (7,96)	90,26 (1,60)	86,53 (3,51)
-10+20	92,44 (1,25)	33,49 (4,60)	81,85 (8,79)	92,90 (1,43)	87,37 (4,13)
-10+30	**95,23** (1,09)	**46,44** (9,42)	**82,13** (8,94)	**95,79** (1,12)	**88,96** (4,48)
-10+50	96,29 (0,98)	54,67 (10,78)	78,77 (7,13)	97,04 (1,09)	87,90 (3,47)

Estes resultados foram obtidos utilizando-se balanceamento de classes pelo método M-Clus, não foi considerada nenhuma característica, e não foi utilizado o método de inclusão de conhecimento adquirido.

Tabela 3: Comparação de desempenho em função do tamanho da janela para o organismo *Mus musculus*, com a inclusão das características ATG+STOP+GAG.

Organismo: Mus musculus					
Tamanho Janela	**Ac**	**Pr**	**Se**	**Sp**	**Adj**
Avaliação de janelas de tamanho simétrico					
-8+8	89,82 (0,77)	27,68 (2,82)	90,88 (4,80)	89,78 (0,89)	90,33 (2,21)
-12+12	91,73 (0,84)	32,12 (3,18)	90,18 (5,32)	91,79 (0,92)	90,98 (2,56)
-20+20	93,36 (1,28)	37,52 (4,72)	88,29 (4,76)	93,57 (1,44)	90,93 (2,05)
-30+30	95,98 (0,48)	50,05 (3,75)	84,30 (8,82)	96,47 (0,42)	90,38 (4,37)
-40+40	97,55 (0,57)	65,81 (9,15)	83,33 (6,98)	98,13 (0,71)	90,73 (3,30)
-50+50	97,51 (0,59)	64,09 (7,61)	83,27 (5,32)	98,08 (0,73)	90,67 (2,44)
-60+60	98,24 (0,51)	78,09 (6,85)	74,39 (9,56)	99,17 (0,33)	86,78 (4,76)
Avaliação de janelas de tamanho assimétrico					
Avaliação região upstream					
-8+30	94,19 (0,85)	41,11 (5,76)	89,72 (5,08)	94,39 (1,01)	92,06 (2,29)
-10+30	**94,54** (1,15)	**43,05** (6,18)	**91,55** (3,76)	**94,68** (1,14)	**93,11** (2,13)
-12+30	94,84 (1,09)	44,30 (6,41)	89,41 (6,50)	95,07 (1,11)	92,24 (3,29)
-30+30	95,98 (0,48)	50,05 (3,75)	84,30 (8,82)	96,47 (0,42)	90,38 (4,37)
-50+30	97,67 (0,33)	66,41 (6,28)	80,89 (6,14)	98,34 (0,52)	89,61 (2,87)
Avaliação região downstream					
-10+10	90,86 (1,85)	30,85 (5,06)	92,60 (4,03)	90,79 (2,00)	91,70 (1,78)
-10+20	92,55 (1,66)	35,49 (5,51)	92,32 (3,83)	92,57 (1,72)	92,44 (2,14)
-10+30	**94,54** (1,15)	**43,05** (6,18)	**91,55** (3,76)	**94,68** (1,14)	**93,11** (2,13)
-10+50	96,17 (0,74)	52,72 (6,43)	87,03 (4,57)	96,56 (0,86)	91,80 (2,10)

Estes resultados foram obtidos utilizando-se balanceamento de classes pelo método M-Clus, e considerando-se as características ATG+STOP+GAG, e não foi utilizado o método de inclusão de conhecimento adquirido.

Analogamente, para o organismo *Rattus novergicus* foram realizados também experimentos a fim de avaliar o desempenho do classificador quanto ao tamanho das janelas. A Tabela 4 apresenta os resultados desses experimentos. Da mesma forma, ao aumentar o tamanho da janela simétrico há um aumento nas taxas de acurácia e especificidade e queda no índice de sensibilidade.

Tabela 4: Comparação de desempenho em função do tamanho da janela para o organismo *Rattus novergicus*, sem a inclusão de características.

Organismo: Rattus novergicus					
Tamanho Janela	**Ac**	**Pr**	**Se**	**Sp**	**Adj**
Avaliação de janelas de tamanho simétrico					
-8+8	92,28 (0,85)	7,77 (1,69)	84,17 (15,04)	92,34 (0,88)	88,25 (7,39)
-12+12	94,27 (1,09)	9,75 (1,67)	79,09 (15,84)	94,39 (1,16)	86,74 (7,58)
-20+20	96,99 (0,97)	19,06 (9,96)	76,11 (21,38)	97,16 (1,10)	86,63 (10,39)
-30+30	98,95 (0,28)	39,69 (10,19)	72,05 (17,56)	99,15 (0,32)	85,60 (8,72)
-40+40	99,56 (0,20)	72,17 (16,36)	71,21 (12,16)	99,76 (0,18)	85,49 (6,07)
-50+50	99,67 (0,18)	89,40 (11,72)	62,75 (17,34)	99,94 (0,06)	81,35 (8,66)
-60+60	99,70 (0,12)	94,67 (8,19)	59,54 (11,39)	99,97 (0,04)	79,76 (5,70)
Avaliação de janelas de tamanho assimétrico					
Avaliação região upstream					
-8+30	97,07 (0,97)	20,40 (6,78)	83,17 (11,21)	95,15 (0,98)	89,16 (5,56)
-10+30	**97,14** (0,76)	**19,35** (4,36)	**82,09** (14,05)	**97,26** (0,79)	**89,67** (6,91)
-12+30	98,70 (0,52)	24,72 (7,45)	76,47 (16,50)	98,70 (0,43)	87,58 (8,05)
-30+30	98,95 (0,28)	39,69 (10,19)	72,05 (17,56)	99,15 (0,32)	85,60 (8,72)
-50+30	99,65 (0,18)	84,63 (11,14)	64,00 (16,81)	99,91 (0,08)	81,95 (8,39)
Avaliação região downstream					
-10+10	93,76 (1,16)	8,94 (1,45)	78,09 (14,08)	93,88 (1,22)	85,99 (6,73)
-10+20	95,21 (0,85)	11,75 (2,11)	80,09 (11,04)	95,32 (0,87)	87,70 (5,46)
-10+30	**97,14** (0,76)	**19,35** (4,36)	**82,09** (14,05)	**97,26** (0,79)	**89,67** (6,91)
-10+50	98,80 (0,27)	37,76 (7,45)	78,09 (17,84)	98,97 (0,24)	88,53 (8,92)

Estes resultados foram obtidos utilizando-se balanceamento de classes pelo método M-Clus, não considerando nenhuma característica, e não foi utilizado o método de inclusão de conhecimento adquirido.

Dessa forma, ao aumentar o tamanho da janela na região *upstream* e *downstream*, ao mesmo tempo, as taxas de sensibilidade e especificidade se contrapõem, ou seja, quando há um aumento em uma há queda na outra, conforme apresentado nas Figuras 11 (*Mus musculus*) e 12 (*Rattus novergicus*).

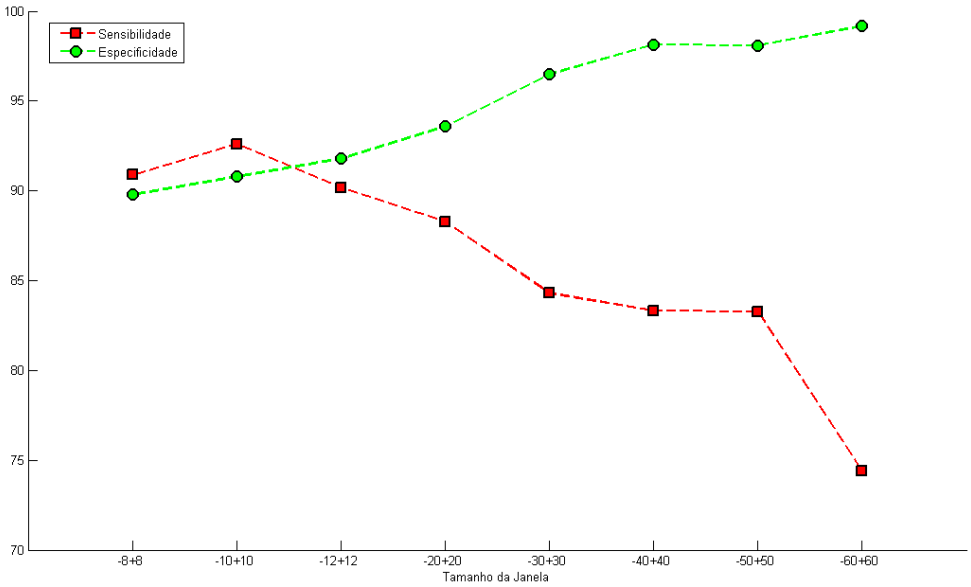

Figura 11: Gráfico - Avaliação de tamanhos simétricos de janelas para o organismo *Mus musculus*. Apresenta os resultados, em especial as taxas de sensibilidade e especificidade, para variados tamanhos simétricos de janelas. Estes resultados foram obtidos utilizando-se o método de balanceamento M-Clus, as características ATG + STOP + GAG e não foi utilizada a inclusão de conhecimento adquirido.

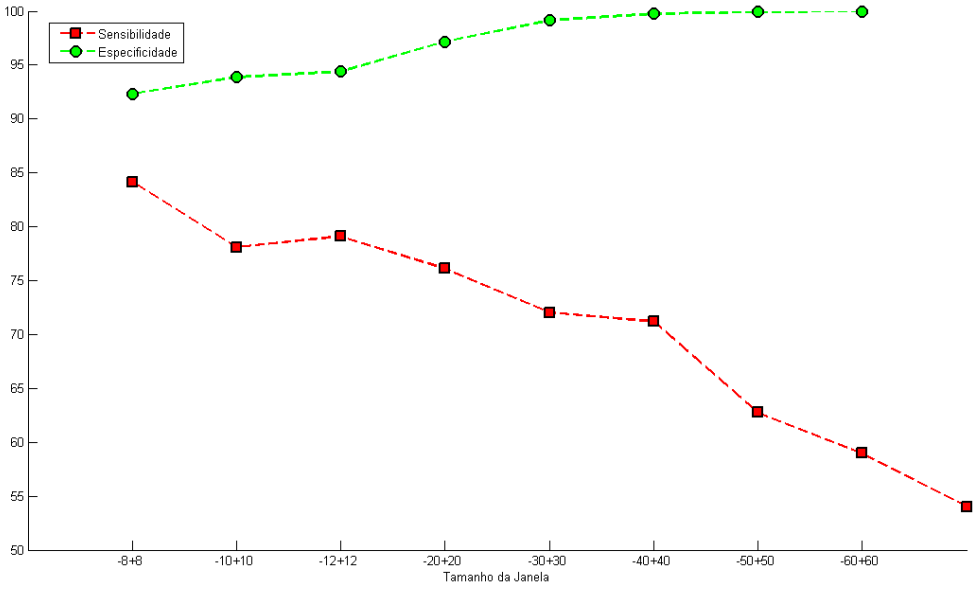

Figura 12: Gráfico - Avaliação de tamanhos simétricos de janelas para o organismo *Rattus novergicus*. Apresenta os resultados, em especial as taxas de sensibilidade e especificidade, para variados tamanhos simétricos de janelas. Estes resultados foram obtidos utilizando-se o método de balanceamento M-Clus, não foram consideradas características e não foi utilizada a inclusão de conhecimento adquirido.

Na tentativa de evitar esse efeito, queda da taxa de sensibilidade, e melhorar o desempenho do classificador, experimentos com janelas de tamanhos assimétricos foram exploradas. Através das Figuras 13 (Mus musculus) e 14 (*Rattus novergicus*), percebe-se que ao aumentar o tamanho da região *downstream* há um aumento da especificidade e diminuição da sensibilidade; dessa forma o tamanho dessa região não pode ser muito grande para não interferir na taxa de sensibilidade, mas também não pode ser muito pequeno para se garantir uma boa taxa de especificidade.

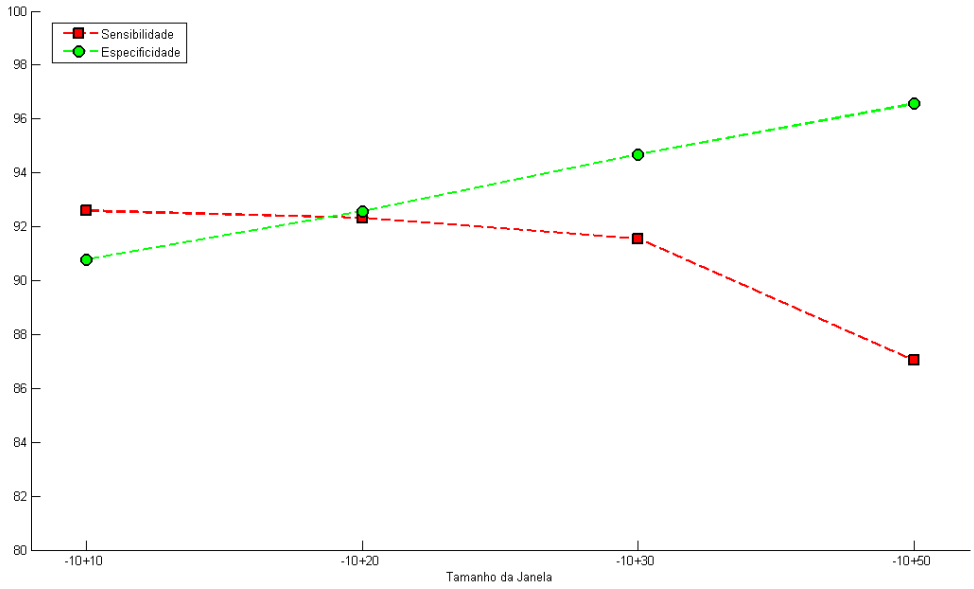

Figura 13: Gráfico - Avaliação de tamanhos assimétricos de janelas para o organismo *Mus musculus* - Região *downstream*. Apresenta os resultados, em especial as taxas de sensibilidade e especificidade, para variados tamanhos assimétricos de janelas, variando-se a região *downstream*. Esses resultados foram obtidos utilizando-se o método de balanceamento M-Clus, as características ATG + STOP + GAG e não foi utilizada a inclusão de conhecimento adquirido.

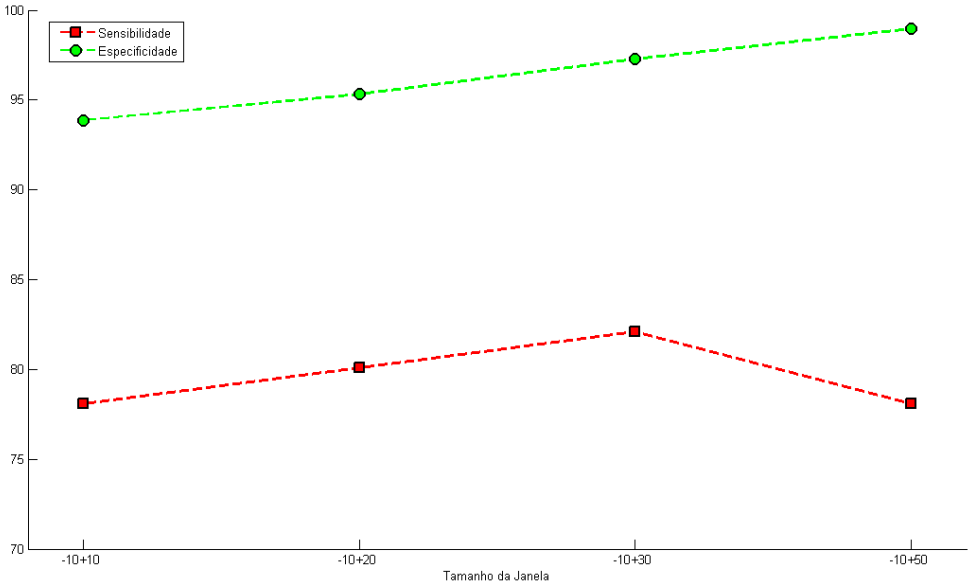

Figura 14: Gráfico - Avaliação de tamanhos assimétricos de janelas para o organismo *Rattus novergicus* - Região *downstream*. Apresenta os resultados, em especial as taxas de sensibilidade e especificidade, para variados tamanhos simétricos de janelas. Estes resultados foram obtidos utilizando-se o método de balanceamento M-Clus, as características ATG + STOP + GAG e não foi utilizada a inclusão de conhecimento adquirido.

Por outro lado, quando há um aumento da região *upstream* há uma queda bastante significativa na sensibilidade, de acordo com as Figuras 15 (*Mus musculus*) e 16 (*Rattus novergicus*). Dessa forma, há indícios de que a sensibilidade está relacionada com os nucleotídeos das posições próximas ao SIT. Ou seja, o contexto para que o ribossomo inicie a tradução em um determinado ATG são os nucleotídeos anteriores e próximos ao ATG que está sendo validado. Isso corrobora com o estudo de Hatzigeorgiou (2002), que utilizou o módulo Consensus-ANN, com janela de tamanho de 12 nucleotídeos, de -7 a +5, e verificou que esse módulo era sensível à região conservada do SIT. Tzanis, Berberidis e Vlahavas (2007) também utilizaram um componente que analisou a região em torno do SIT, de -7 a +5, utilizando cadeias de Markov para capturar o padrão de consenso, indicando que para a identificação da SIT é importante examinar uma área restrita em torno dele.

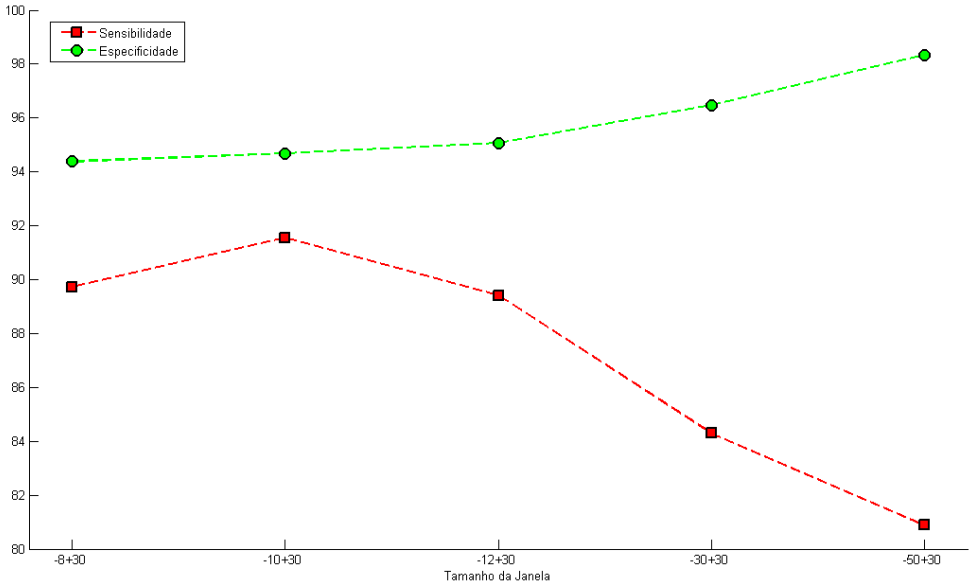

Figura 15: Gráfico - Avaliação de tamanhos assimétricos de janelas para o organismo *Mus musculus* - Região *upstream*. Apresenta os resultados, em especial as taxas de sensibilidade e especificidade, para variados tamanhos assimétricos de janelas, variando a região *upstream*. Esses resultados foram obtidos utilizando-se o método de balanceamento M-Clus, as características ATG + STOP + GAG e não foi utilizada a inclusão de conhecimento adquirido.

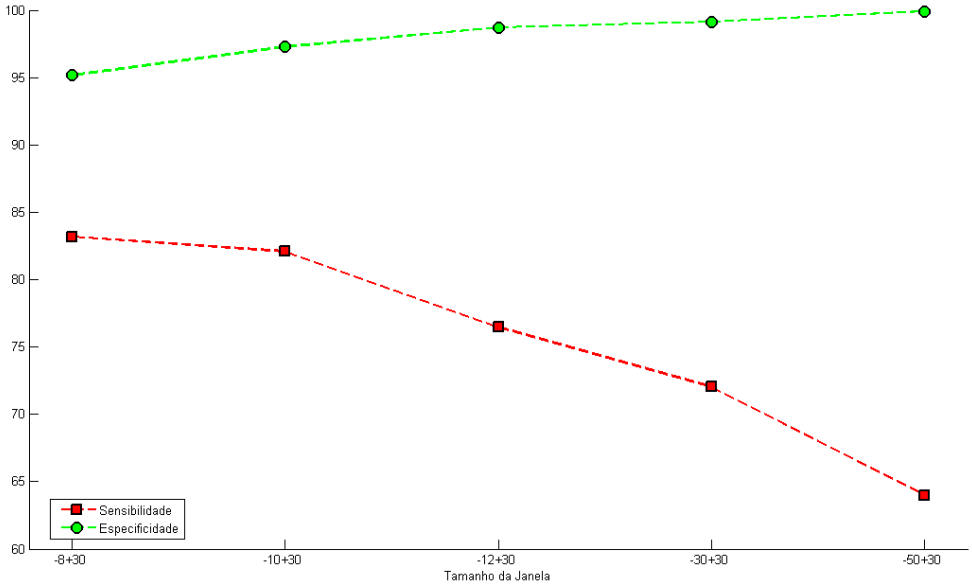

Figura 16: Gráfico - Avaliação de tamanhos assimétricos de janelas para o organismo *Rattus novergicus* - Região *upstream*. Apresenta os resultados, em especial as taxas de sensibilidade e especificidade, para variados tamanhos simétricos de janelas. Estes resultados foram obtidos utilizando-se o método de balanceamento M-Clus, as características ATG + STOP + GAG e não foi utilizada a inclusão de conhecimento adquirido.

Assim, dentre os testes realizados, a janela de tamanho -10+30 proporcionou os melhores resultados, melhorando assim a acurácia ajustada, representada pela média entre a sensibilidade e a especificidade.

4.2 Avaliação das características incluídas

Percebe-se, através das Tabelas 5 (*Mus musculus*) e 6 (*Rattus novergicus*), que a inclusão de características aumenta o desempenho do classificador. Considerando-se os testes sem nenhuma característica e a inclusão de ATG + STOP + GAG, observa-se um aumento de aproximadamente 9% na taxa de sensibilidade para o organismo *Mus musculus* e 6% para o *Rattus novergicus*.

Os resultados das Tabelas 5 e 6 foram gerados a partir do método de balanceamento proposto nesse trabalho, M-Clus, e tamanho da janela -10+30. A taxa de especificidade não variou muito, em torno de 1,2% para *Mus musculus* e 2,24% para *Rattus novergicus*; no entanto, houve um aumento na taxa de sensibilidade de 9% e 6%, para os organismos *Mus musculus* e *Rattus novergicus*, respectivamente. Estes testes foram aplicados também a outros tamanhos de janelas -50+50, -12+12 e -10+20, e esse comportamento foi observado em todas as situações. Dessa forma, a inclusão de características é relevante para o aumento da sensibilidade do classificador.

Tabela 5: Comparação de desempenho em função da inclusão de características para o organismo *Mus musculus*

Organismo: Mus musculus					
Características	**Ac**	**Pr**	**Se**	**Sp**	**Adj**
Nenhuma	95,23 (1,09)	46,44 (9,42)	82,13 (8,94)	95,79 (1,12)	88,96 (4,48)
ATG + STOP + GAG	94,54 (1,15)	43,05 (6,18)	**91,55** (3,76)	94,68 (1,14)	93,11 (2,13)
ATG+STOP+CTG+GAC+GAG	94,48 (1,11)	42,69 (4,85)	**90,88** (4,80)	94,64 (1,20)	92,76 (2,27)
ATG + STOP	94,54 (0,89)	42,83 (5,44)	**90,59** (4,56)	94,72 (0,95)	92,66 (2,26)
ATG + STOP + CTG + GAC	94,54 (1,40)	43,35 (6,77)	**90,30** (4,91)	94,72 (1,45)	92,51 (2,53)
ATG+STOP+CTG+GAC+GAG+GCC	94,54 (0,99)	42,72 (4,76)	89,44 (5,08)	94,77 (0,98)	92,11 (2,66)
ATG	94,85 (1,28)	45,02 (8,66)	89,34 (4,38)	95,09 (1,37)	92,21 (2,18)
ATG + STOP + GCC	94,54 (1,18)	43,02 (6,56)	89,34 (5,01)	94,77 (1,23)	92,06 (2,55)
ATG + STOP + GAC	94,66 (1,34)	43,89 (7,62)	88,86 (4,37)	94,91 (1,41)	91,89 (2,20)
ATG + STOP + CTG	94,70 (1,17)	43,68 (5,90)	88,57 (4,12)	94,96 (1,20)	91,77 (2,18)
ATG + CTG	94,86 (1,18)	44,65 (6,69)	88,47 (3,44)	95,14 (1,23)	91,81 (1,79)
ATG + GAG	94,70 (1,41)	44,25 (8,95)	88,09 (4,69)	94,98 (1,45)	91,53 (2,49)
ATG + GCC	95,12 (1,11)	46,03 (6,78)	88,00 (5,92)	95,44 (1,14)	91,72 (3,02)
ATG + GAC	95,26 (0,78)	46,42 (5,63)	87,42 (5,41)	95,60 (0,81)	91,51 (2,70)
STOP + GAC	94,48 (1,31)	42,31 (7,73)	85,01 (8,51)	94,88 (1,19)	89,95 (4,52)
STOP + CTG	94,25 (1,35)	41,52 (7,51)	84,72 (5,71)	94,66 (1,42)	89,69 (2,88)
STOP	94,08 (1,23)	40,13 (7,01)	82,61 (8,31)	94,58 (1,19)	88,60 (4,30)
STOP + GAG	94,48 (1,49)	42,61 (8,69)	82,33 (8,27)	95,01 (1,48)	88,67 (4,30)
GCC	94,74 (1,14)	43,52 (7,86)	82,13 (8,77)	95,28 (1,22)	88,71 (4,31)
CTG + GAG	94,80 (1,13)	44,05 (8,56)	81,94 (7,69)	95,36 (1,21)	88,65 (3,80)
CTG + GAC + GAG	94,53 (1,11)	42,20 (6,34)	81,94 (9,26)	95,08 (1,14)	88,51 (4,60)
STOP + GCC	93,97 (1,45)	40,05 (7,72)	81,94 (7,69)	94,50 (1,60)	88,22 (3,70)
GAG	94,78 (1,14)	43,85 (8,45)	81,56 (9,38)	95,36 (1,23)	88,46 (4,63)
CTG	94,74 (1,36)	43,96 (9,04)	81,56 (9,22)	95,31 (1,49)	88,44 (4,50)
GAC	94,85 (1,13)	44,03 (7,79)	81,46 (8,75)	95,42 (1,10)	88,44 (4,45)
GAG + GCC	94,88 (1,34)	44,33 (8,70)	81,26 (8,89)	95,46 (1,23)	88,36 (4,70)
GAC + GAG	94,95 (1,10)	44,77 (8,06)	81,17 (9,00)	95,55 (1,21)	88,36 (4,39)
CTG + GAC	95,00 (1,11)	45,00 (8,98)	81,08 (8,16)	95,60 (1,15)	88,43 (4,11)
CTG + GAC + GAG + GCC	94,77 (1,22)	43,62 (8,15)	80,98 (8,27)	95,36 (1,18)	88,17 (4,28)
CTG + GCC	94,39 (1,34)	41,82 (8,03)	80,98 (8,45)	94,96 (1,41)	87,97 (4,18)
GAC + GCC	94,34 (1,06)	41,07 (6,78)	80,21 (9,11)	94,95 (1,11)	87,58 (4,50)

Esses resultados foram obtidos utilizando-se janelas de tamanho -10+30 e balanceamento por M-Clus.

Tabela 6: Comparação de desempenho em função da inclusão de características para o organismo *Rattus novergicus*

Organismo: Rattus novergicus					
Características	**Ac**	**Pr**	**Se**	**Sp**	**Adj**
Nenhuma	97,14 (0,76)	19,35 (4,36)	82,09 (14,05)	97,26 (0,79)	89,67 (6,91)
ATG + STOP + GAG	95,38 (1,09)	13,54 (3,65)	**88,09** (9,82)	96,54 (1,13)	92,32 (4,61)
ATG + STOP	95,64 (0,82)	13,83 (2,18)	**88,09** (8,74)	95,69 (0,87)	91,89 (4,08)
ATG+STOP+CTG+GAC+GAG	96,48 (0,78)	16,93 (3,81)	**88,09** (9,88)	95,44 (0,89)	91,76 (4,35)
ATG + STOP + CTG + GAC	95,03 (1,19)	12,68 (2,94)	**88,09** (9,44)	95,08 (1,20)	91,58 (4,75)
ATG+STOP+CTG+GAC+GAG+GCC	96,38 (0,63)	16,06 (2,34)	87,18 (8,87)	96,45 (0,64)	91,82 (4,40)
ATG + GAC	96,57 (0,74)	17,09 (3,55)	87,09 (10,08)	96,64 (0,76)	91,86 (4,93)
ATG + CTG	96,42 (0,53)	16,22 (1,72)	87,09 (9,82)	96,49 (0,58)	91,79 (4,76)
STOP + GAC	96,30 (0,82)	16,08 (3,59)	87,09 (10,08)	96,37 (0,84)	91,73 (4,97)
ATG + STOP + GCC	96,11 (0,61)	15,52 (2,19)	87,09 (8,30)	96,15 (0,63)	91,62 (4,04)
ATG + STOP + GAC	95,68 (1,24)	14,55 (3,72)	87,09 (8,74)	95,73 (1,28)	91,41 (4,03)
ATG + STOP + CTG	94,97 (1,16)	12,54 (2,90)	87,09 (9,45)	95,02 (1,18)	91,05 (4,70)
ATG + GAG	96,70 (0,61)	17,54 (3,11)	86,09 (10,79)	96,76 (0,62)	91,42 (5,35)
ATG	96,32 (0,83)	15,97 (3,29)	86,09 (9,21)	96,40 (0,84)	91,24 (4,54)
STOP	95,99 (0,81)	14,62 (2,50)	86,09 (12,03)	96,07 (0,84)	91,08 (5,88)
ATG + GCC	96,93 (0,56)	18,05 (2,71)	84,09 (12,85)	97,02 (0,61)	90,56 (6,25)
STOP + CTG	95,86 (0,86)	14,04 (2,97)	84,09 (13,61)	95,95 (0,89)	90,02 (6,70)
STOP + GAG	96,36 (0,65)	15,51 (2,66)	84,09 (14,32)	96,45 (0,70)	90,27 (7,00)
CTG + GAC	96,84 (0,85)	17,90 (3,56)	83,09 (11,06)	96,94 (0,88)	90,01 (5,40)
GAG	97,35 (0,47)	20,13 (4,17)	82,09 (14,05)	97,46 (0,48)	89,78 (6,99)
GAC	96,99 (0,37)	17,82 (2,22)	82,09 (14,05)	97,10 (0,43)	89,59 (6,90)
GAG + GCC	91,65 (0,39)	22,00 (2,82)	81,18 (14,45)	97,77 (0,46)	89,48 (7,07)
CTG + GAG	97,25 (0,51)	19,32 (3,60)	81,18 (15,78)	97,37 (0,55)	89,28 (7,79)
CTG + GCC	97,47 (0,53)	21,04 (5,05)	81,09 (14,51)	97,59 (0,54)	89,34 (7,24)
GAC + GCC	97,54 (0,71)	21,73 (3,83)	80,09 (14,90)	97,67 (0,77)	88,88 (7,23)
GCC	97,52 (0,41)	20,74 (3,73)	80,09 (18,49)	97,65 (0,47)	88,87 (9,13)
GAC + GAG	97,50 (0,38)	20,79 (4,97)	80,09 (15,55)	97,63 (0,35)	88,86 (7,80)
CTG + GAC + GAG	97,05 (0,64)	18,25 (4,06)	80,09 (14,90)	97,18 (0,64)	88,64 (7,43)
CTG + GAC + GAG + GCC	97,52 (0,45)	20,75 (4,44)	79,09 (17,06)	97,66 (0,46)	88,37 (8,49)
STOP + GCC	96,91 (0,54)	17,15 (3,47)	79,09 (17,06)	97,05 (0,59)	88,07 (8,41)
CTG	96,69 (0,72)	16,20 (3,51)	78,18 (17,80)	96,83 (0,78)	87,50 (8,71)

Esses resultados foram obtidos utilizando-se janelas de tamanho -10+30 e balanceamento por M-Clus.

Contudo, há também características que, ao serem acrescentadas às sequências, diminuíram ligeiramente (de 0,19% a 1,92%, para *Mus musculus* e de 0,91% a 3,91%, para *Rattus novergicus*) o desempenho do classificador. Por exemplo, para *Mus musculus*, acrescentando-se as características CTG ou GAC ou GAG ou CTG + GAC ou GAC + GCC, há uma diminuição de, aproximadamente, 1,9% na taxa de sensibilidade. Já para o organismo *Rattus novergicus*, ao acrescentar as características CTG + GAC + GAC + GCC ou STOP + GCC há diminuição de, aproximadamente, 3% na referida taxa. No entanto, essa variação é pouco expressiva para se considerar uma queda no desempenho do classificador.

Interessantemente, as características que apresentaram melhor desempenho do classificador para o organismo *Mus musculus*, repetem seu desempenho quando aplicadas ao organismo *Rattus novergicus*. A Tabela 7 apresenta as dezesseis características mais importantes, observando-se a sensibilidade, para cada organismo analisado.

Tabela 7: As dezesseis características mais importantes para cada organismo analisado

Características	*M. musculus*	*R. novergicus*
ATG + STOP + GAG	1	1
ATG+STOP+CTG+GAC+GAG	2	3
ATG + STOP	3	2
ATG + STOP + CTG + GAC	4	4
ATG+STOP+CTG+GAC+GAG+GCC	5	5
ATG	6	13
ATG + STOP + GCC	7	9
ATG + STOP + GAC	8	10
ATG + STOP + CTG	9	11
ATG + CTG	10	7
ATG + GAG	11	12
ATG + GCC	12	15
ATG + GAC	13	6
STOP + GAC	14	8
STOP + CTG	15	16
STOP	16	14

As cinco características mais importantes, ATG + STOP + GAG, ATG + STOP + CTG + GAC + GAG, ATG + STOP, ATG + STOP + CTG + GAC e ATG + STOP + CTG + GAC + GAG + GCC, repetem-se, não nessa mesma ordem, para os organismos analisados. E a característica com melhor desempenho, ATG + STOP + GAG, para o organismo *Mus musculus*, também é para o *Rattus novergicus*. Indicando, assim, que essa combinação de características é relevante para o desempenho do classificador.

Além disso, fica evidente que a característica ATG *upstream* em fase é bastante relevante, visto que os melhores resultados foram obtidos pela combinação dessa somada a outras características. Vale ressaltar que há um aumento bastante significativo na sensibilidade, de 7,21% (*Mus musculus*) e 4% (*Rattus novergicus*), quando se considera

apenas a característica ATG *upstream*.

Dessa forma, os testes realizados mostram que o classificador consegue boas taxas de desempenho considerando-se apenas as sequências positivas e negativas (sequência linear de bases). No entanto, comprova também que é possível ter um aumento no desempenho por meio da inclusão de características consideradas relevantes para o contexto considerado.

4.3 Avaliação dos métodos de balanceamento

A Figura 17 apresenta os resultados, para o organismo *Mus musculus*, do método de balanceamento M-Clus, variando-se o número de *clusters*. Percebe-se que, ao diminuir esse número, as taxas de especificidade e precisão diminuem. Isso pode ser compreendido pelo fato de pouca representatividade da classe negativa (majoritária), pois quando se diminui pela metade a quantidade de *clusters* extrai-se duas sequências por *cluster*. Dessa forma, as sequências extraídas são as mais próximas ao centróide, e consequentemente são próximas entre si. Assim, não há boa representatividade de todas as sequências negativas. Nesse sentido, para se obter maior representatividade da classe majoritária, considerou-se a quantidade de clusters igual à quantidade de elementos da classe minoritária ($SizeMI$).

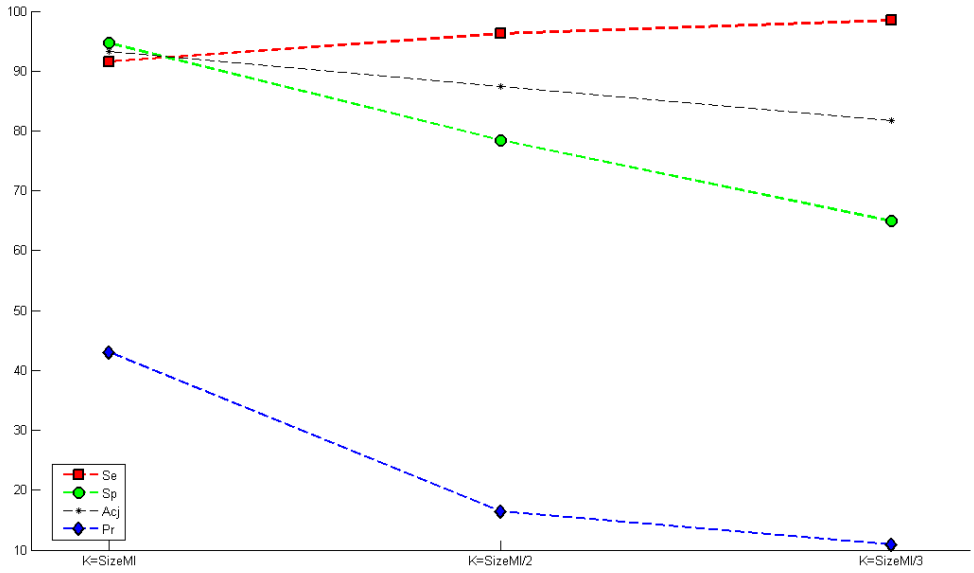

Figura 17: Gráfico - Avaliação da quantidade de *clusters* para o método de balanceamento M-Clus, organismo *Mus musculus*. Apresenta os resultados para o método M-Clus, variando-se a quantidade de *clusters* criados (k).

Ao comparar os resultados obtidos pelos outros métodos de balanceamento anali-

sados, apresentados na Tabela 8, percebe-se que quando não se utiliza método de balanceamento a taxa de sensibilidade é muito baixa, 51,39% para o organismo *Mus musculus* e 47,45% para o *Rattus novergicus*. Isto ocorre porque, como a base de dados está desbalanceada, o classificador aprende muito sobre a classe negativa (majoritária) e pouco sobre a positiva (minoritária). Assim, o sistema tenderá a encontrar um número grande de falsos negativos, e, pela Equação 3.4 (Se = 100 * TP / (TP + FN)) , percebe-se que a sensibilidade diminui com o aumento de FN (falsos negativos).

Esse fato vai de encontro ao que os autores Machado e Ladeira (2007) dizem, quando afirmam que os classificadores gerados a partir de bases de treinamento desbalanceadas apresentam altas taxas de falsos negativos para as classes raras, o que é problemático quando a classe de interesse é justamente essa classe.

Tabela 8: Comparação de desempenho em função do método de balanceamento

Organismo: Mus musculus					
Balanceamento	**Ac**	**Pr**	**Se**	**Sp**	**Adj**
Sem Balanceamento	97,96 (0,37)	98,50 (3,02)	51,39 (6,97)	99,97 (0,06)	75,68 (3,49)
Rand undersampling	93,70 (0,83)	38,95 (3,94)	91,06 (3,85)	93,81 (0,88)	92,44 (1,90)
M-Clus	94,54 (1,15)	43,05 (6,18)	**91,55** (3,76)	**94,68** (1,14)	**93,22** (2,13)
SBC	92,23 (1,70)	34,12 (4,96)	89,63 (3,46)	92,34 (1,81)	90,98 (1,74)
Smote	98,45 (0,36)	84,95 (7,14)	76,56 (9,80)	99,39 (0,34)	87,98 (4,82)
Organismo: Rattus norvegicus					
Balanceamento	**Ac**	**Pr**	**Se**	**Sp**	**Adj**
Sem Balanceamento	99,59 (0,08)	96,90 (6,21)	47,45 (11,21)	99,98 (0,03)	73,72 (5,60)
Rand undersampling	95,90 (2,07)	13,89 (4,57)	83,18 (10,75)	96,00 (2,36)	89,59 (4,89)
M-Clus	95,38 (1,09)	13,55 (3,65)	88,09 (9,82)	95,44 (1,14)	91,76 (4,61)
SBC	88,23 (6,09)	6,73 (2,57)	91,00 (11,35)	88,20 (6,20)	89,60 (4,09)
Smote	99,25 (0,94)	79,69 (28,18)	56,45 (7,77)	99,58 (0,96)	78,01 (3,81)

Esses resultados foram obtidos utilizando-se janelas de tamanho -10+30, as características ATG + STOP + GAG e não foi utilizada a inclusão de conhecimento adquirido.

Ao se utilizar qualquer um dos métodos de balanceamento do tipo *undersampling* (*Rand undersampling*, SBC e M-Clus), a sensibilidade aumenta em torno de 40%, tanto para o organismo *Mus musculus* quanto para o *Rattus novergicus*.

O método de balanceamento proposto, M-Clus, possui um desempenho melhor que todos os outros métodos, principalmente quanto à taxa de acurácia ajustada. Para o organismo *Mus musculus*, o desempenho melhor em relação à taxa de sensibilidade é referente ao método M-Clus. Já para o organismo *Rattus novergicus* o melhor valor para a taxa de sensibilidade é dada ao se utilizar o método SBC; porém, há uma queda nas taxas de precisão e especifidade e, consequentemente, na acurácia ajustada.

Ao analisar os resultados para os dois organismos estudados neste trabalho, o método *Random undersampling* é ligeiramente melhor que o método SBC, sendo interessante sua utilização, já que é um método bastante simples de ser implementado.

Foi observado também que o método de balanceamento Smote, recomendado pela comunidade científica, não apresentou um bom desempenho na avaliação experimental deste trabalho. Isso pode ser explicado pelo fato da metodologia utilizada, já que é realizado um balanceamento para cada dobra da validação cruzada. Como o algoritmo Smote realiza interpolação entre vizinhos e cria novos casos a partir disso, há um aprendizado maior por parte do classificador apenas do conjunto de treino, ficando o classificador bastante específico para aquela dobra.

De forma contrária, grande parte dos pesquisadores tem utilizado o algoritmo Smote da seguinte forma: aplica-se o método para balancear todo o conjunto das bases de dados, depois separa-se os conjuntos de treinamento e de teste (ou em dobras, em alguns casos). Utilizando essa metodologia o desempenho do classificador é notoriamente superior, porém uma questão pode ser levantada: se ao separar os conjuntos de treino e teste algum exemplo que foi criado sinteticamente a partir de outros ficar no conjunto de teste e os exemplos que serviram para criação dele ficarem no conjunto de treino, há uma probabilidade muito grande do exemplo ser classificado corretamente. Porém ficam as seguintes questões: o exemplo não foi criado a partir de outros que estão no conjunto de treino? Realmente há aprendizado nesta metodologia? A fim de análise, o presente trabalho também utilizou essa metodologia e os resultados foram, como esperados, notoriamente superiores, todas as taxas analisadas ficaram acima de 99%. Entretanto, há indícios de que utilizar balanceamento antes de separar os conjuntos de treino e teste faz com que o desempenho do classificador seja alto, porém os exemplos utilizados para avaliá-lo não foram separados de forma adequada.

O alto valor da taxa de especificidade quando não se utiliza método de balanceamento e ao utilizar o método Smote pode ser entendida pelo fato de que todas as sequências negativas são utilizadas para treinamento, não havendo perda de informação alguma. Em contrapartida o desempenho dessas metodologias ao visualizar a taxa de sensibilidade é bem inferior ao comparar com os outros métodos utilizados.

Ao se analisar a taxa de precisão percebe-se que o desempenho é notoriamente superior, 98,50% para o organismo *Mus musculus* e 96,90% para o *Rattus novergicus*, sem utilizar nenhuma técnica de balanceamento. Essa taxa reduz significativamente quando se utiliza algum método para realizar o balanceamento. Esse fato pode ser explicado porque, quando não se utiliza método de balanceamento, o classificador pouco aprende sobre a classe positiva, fazendo com que poucos exemplos do conjunto de teste sejam classificados como positivos, gerando assim, poucos falsos positivos. Como a taxa de precisão, dada por

Pr = TP/(TP+FP), avalia quantos possíveis SIT (classificados como SIT) são realmente SIT, essa taxa tende a assumir um valor alto, já que a quantidade de falso positivos (FP) é bem pequena.

Finalmente, é importante ressaltar a necessidade de se apresentar todas essas medidas de desempenho, uma vez que podemos ter um sistema com uma acurácia altíssima, porém não apresentando praticamente nenhum conhecimento a respeito da classe que seja realmente de interesse.

Com o objetivo de melhorar o desempenho da precisão foi planejada uma nova metodologia de inclusão de conhecimento adquirido (InAKnow). Os resultados estão apresentados a seguir.

4.4 Avaliação do método de inclusão de conhecimento

Com a finalidade de melhorar o desempenho do classificador, em especial da taxa de precisão, este trabalho propos uma metodologia para inclusão de conhecimento adquirido (InAKnow), descrita no Capítulo 3. Os resultados dos experimentos utilizando essa metodologia são apresentados na Tabela 9. Há um aumento em todas as taxas avaliadas, em especial a taxa de precisão, que aumenta em 39% para o organismo *Mus musculus*.

Já para o organismo *Rattus novergicus* o aumento é de 12,55% para 35,63%, representado um aumento de 2,84 vezes. Porém o valor da precisão ainda é baixo. Isso pode ser explicado pela quantidade pequena de sequências positivas (somente 101), já que 91% das moléculas de mRNA foram desconsideradas por iniciarem a tradução em posições anteriores aos 10 nucleotídeos (tamanho da janela da região *upstream*).

Tabela 9: Comparação de desempenho em função do método InAKnow

Organismo: Mus musculus		
	Sem InAKnow	**Com InAKnow**
Acurácia	94,54 (1,15)	95,56 (0,78)
Precisão	43,05 (6,18)	82,05 (2,82)
Sensibilidade	91,55 (3,76)	93,23 (2,69)
Especificidade	94,68 (1,14)	96,01 (0,74)
Acurácia Ajustada	93,22 (2,13)	94,62 (1,42)
Organismo: Rattus norvegicus		
	Sem InAKnow	**Com InAKnow**
Acurácia	95,38 (1,09)	94,90 (0,80)
Precisão	13,54 (3,65)	35,63 (3,36)
Sensibilidade	88,09 (9,82)	95,24 (3,78)
Especificidade	96,54 (1,13)	94,89 (0,89)
Acurácia Ajustada	92,32 (4,61)	95,07 (1,67)

Esses resultados foram obtidos utilizando-se janelas de tamanho -10+30, as características ATG + STOP + GAG e o método de balanceamento M-Clus.

As taxas de sensibilidade, especificidade e acurácia ajustada aumentam em 1,68%, 1,33% e 1,4%, respectivamente, para o organismo *Mus musculus*.

Ao analisar os resultados dos experimentos com o organismo *Rattus norvegicus*, percebe-se que houve uma melhora significativa na taxa de sensibilidade (8,15%) ao utilizar a metodologia InAKnow, significando que houve uma aprendizagem melhor do classificador por parte das sequências positivas. Já para as taxas de acurácia e especifidade há uma pequena queda, de 0,31% e 0,38%, respectivamente.

Graficamente, os resultados da Tabela 9 são apresentados nas Figuras 18 (*Mus musculus*) e 19 (*Rattus novergicus*)

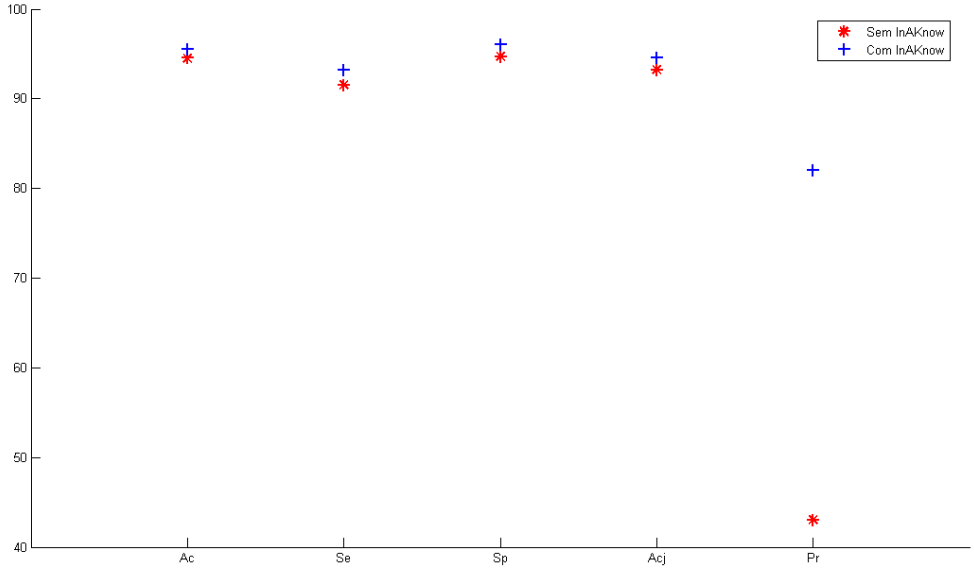

Figura 18: Gráfico - Avaliação da metodologia de inclusão de conhecimento adquirido para o organismo *Mus musculus*. Apresenta os resultados obtidos com a metodologia de inclusão de conhecimento adquirido. Percebe-se que há um aumento de 39% na taxa de precisão ao se utilizar a metodologia proposta. Estes resultados foram obtidos utilizando-se o método de balanceamento M-Clus, e as características ATG + STOP + GAG.

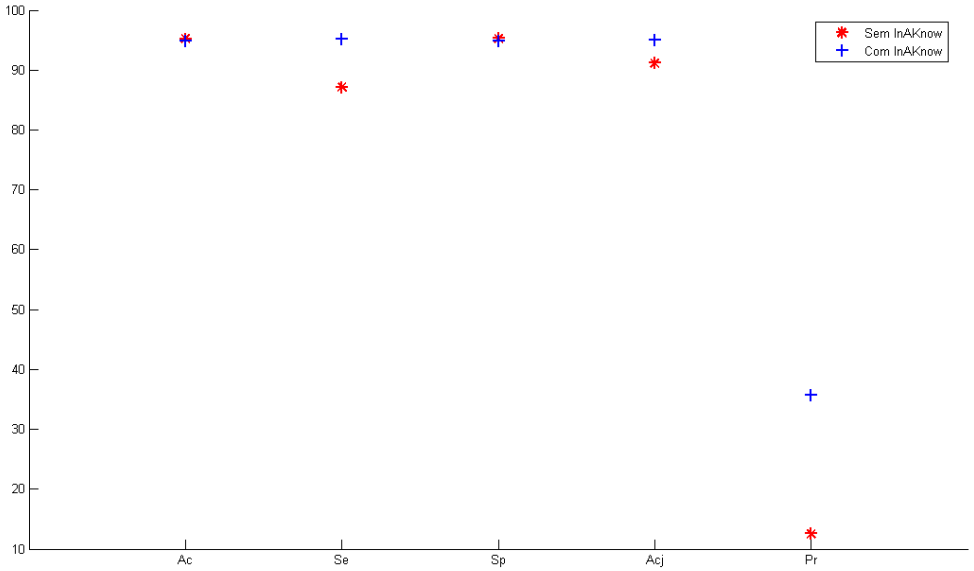

Figura 19: Gráfico - Avaliação da metodologia de inclusão de conhecimento adquirido para o organismo *Rattus novergicus*. Apresenta os resultados obtidos com a metodologia de inclusão de conhecimento adquirido. Percebe-se que há um aumento de 22,09% na taxa de precisão ao se utilizar a metodologia proposta. Estes resultados foram obtidos utilizando-se o método de balanceamento M-Clus, e as características ATG + STOP + GAG.

A melhora significativa na taxa de precisão é devida à diminuição na quantidade de exemplos classificados como falsos positivos. Essa diminuição ocorre em conformidade com o modelo de escaneamento do ribossomo (KOZAK, 1984), que não avalia os ATGs da região *downstream* do SIT, podendo existir nessa região sequências de ATG com contexto apropriado para serem SIT.

Assim, ao se utilizar a metodologia proposta, as sequências da região *downstream* do SIT são inicialmente classificadas através de um modelo gerado previamente, utilizando-se apenas as sequências conhecidas como SIT e as negativas da região *upstream* e que estão fora de fase de leitura. Somente após essa primeira classificação é que essas sequências farão parte dos conjuntos finais de treinamento e teste.

Nesse sentido, as sequências que estavam na região *downstream*, mas que possuíam características para serem SIT fizeram parte do conjunto de sequências positivas, já que pelo modelo do ribossomo essas sequências poderiam vir a ser SIT, se não já tivesse sido encontrado um ATG com contexto apropriado antes delas. Utilizando-se essa metodologia, 14% das sequências que estavam na região *downstream* foram classificadas como positivas, para o organismo *Mus musculus*, e 2,15% para o *Rattus novergicus*.

Essa metodologia, portanto, foi de fundamental importância para se obter um classificador com uma alta taxa de precisão e mostra como o conhecimento adquirido pelo classificador é relevante para classificar sequências com classificação desconhecida a priori.

4.5 Parâmetros utilizados e recursos computacionais

Para a classificação utilizando o SVM, a função de *kernel* utilizada foi a polinomial de grau 4, função já avaliada por Nobre, Ortega e Braga (2007).

Ao utilizar o método de clusterização K-means o número máximo de iterações foi 15, utilizando-se a implementação do *toolbox* do Matlab.

As simulações foram executadas em dois computadores com as seguintes configurações:

1. Acer Extensa 4620, 1.83GHz Core Duo, 2GB DDR
2. Semp Toshiba 1.86GHz Core Duo, 2GB DDR

A Tabela 10 apresenta o tempo médio de execução dos experimentos para cada um dos métodos de balanceamento utilizados, considerando para cada método de balanceamento o tempo de execução do método de balanceamento, bem como o tempo de execução do classificador SVM com as bases de dados já preparadas.

Tabela 10: Comparação dos tempos de execução, dados em minutos

Organismo: Mus musculus		
	Balanceamento das 10 dobras	**Execução do classificador SVM**
Sem Balanceamento	-	5
Rand undersampling	2	2
SBC	40	1
M-Clus	50	1
Smote	25	12
Organismo: Rattus norvegicus		
	Balanceamento das 10 dobras	**Execução do classificador SVM**
Sem Balanceamento	-	50
Rand undersampling	2	2
SBC	56	1
M-Clus	19	1
Smote	21	84

*Tempos em minutos. Esses tempos médios de execução são referentes a testes com janelas de tamanho -10+30.

Percebe-se que ao utilizar o método de balanceamento Smote o tempo de execução do classificador SVM aumenta significativamente.

5 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

Este capítulo apresenta as principais conclusões obtidas com o desenvolvimento desse trabalho, indicando quais estratégias atenderam aos objetivos introduzidos no Capítulo 1. Além disso, são apresentadas as propostas de continuidade desse trabalho, de modo a se tornar possível o desenvolvimento de novas estratégias ou a modificação das existentes.

5.1 Conclusões

Como visto nesse trabalho, a tarefa de previsão de SIT não é um problema trivial de ser resolvido. Inúmeros métodos têm sido avaliados ao longo da literatura e esse trabalho apresentou uma nova metodologia para se encontrar o SIT com eficácia e eficiência, objetivos expostos no início desse trabalho.

Assim, o que os autores buscaram ao longo de todo o desenvolvimento foi apresentar métodos que encontrassem o SIT que efetivamente eram SITs, e que, além disso, isso fosse feito com uma preocupação quanto ao número de sequências trabalhadas.

Nesse sentido, uma vez que esse problema é intrinsecamente desbalanceado, métodos de balanceamento de classes *undersampling* foram avaliados, além de ser proposto também o método M-Clus, também de *undersampling*. Os métodos de *undersampling*, o contrário dos métodos de *oversampling* que replicam o número de sequências, possuem a vantagem de trabalhar com um número muito menor de sequências, o que diminui sensivelmente o processamento computacional. Isso é particularmente importante em grandes bases de dados, como a do Homo sapiens, por exemplo, que possui 15482 moléculas de mRNA no Refseq revisado, contra as 309 moléculas do Mus musculus analisadas nesse trabalho.

Considerando-se as taxas de desempenho avaliadas, o método de balanceamento proposto mostrou-se bastante promissor, oferecendo os melhores resultados, se comparados ao método de balanceamento *undersampling* aleatório, o SBC e a classificação sem balanceamento. Com o M-Clus houve um aumento de 40,16% e 40,64%, para *Mus musculus* e *Rattus novergicus*, respectivamente, na taxa de sensibilidade e de 17,54% (*Mus*

musculus) e 18,04% (*Rattus novergicus*) na acurácia ajustada, indicando que investimentos em métodos de balanceamento são necessários para resolução deste problema. No entanto, a precisão foi reduzida em 55,45% (*Mus musculus*) e 88,35% (*Rattus novergicus*), o que foi resolvido pelo método de inclusão de conhecimento adquirido.

Houve um aumento de 39% (*Mus musculus*) e 22,09% (*Rattus novergicus*) na precisão quando o conhecimento adquirido pelo classificador, InAKnow, foi incluído no novo conjunto de treinamento. Isso se justifica devido à diminuição na quantidade de exemplos classificados como falsos positivos, estando, portanto, em conformidade com o modelo de escaneamento do ribossomo (KOZAK, 1984), que não avalia os ATGs da região *downstream* do SIT, podendo existir nessa região sequências de ATG com contexto apropriado para serem SIT.

Analisou-se também a inclusão de algumas características junto às sequências extraídas, concluindo-se que a inclusão, de maneira geral, favorece o desempenho do classificador. A inclusão de características tais como a presença de ATG na região *upstream* do SIT, melhorou a taxa de sensibilidade em aproximadamente 9% para o organismo *Mus musculus* e 6% para o *Rattus novergicus*.

Finalmente, de acordo com os testes realizados quanto ao tamanho da janela, há indícios de que a sensibilidade esteja relacionada aos nucleotídeos próximos ao SIT. Ou seja, o contexto para que o ribossomo inicie a tradução em um determinado ATG são os nucleotídeos anteriores e próximos ao ATG que está sendo validado. A janela que ofereceu os melhores resultados foi aquela que possuía 10 nucleotídeos na região *upstream* e 30 na *downstream* (-10+30), para os dois organismos analisados.

Em vista de todos os argumentos apresentados, conclui-se que a metodologia proposta contribui de maneira significativa para a previsão de SIT.

5.2 Propostas de continuidade

Quanto às recomendações de trabalhos futuros, pode-se sugerir alguns temas, a saber:

- Aplicar a metodologia apresentada à outras bases de dados, como por exemplo *Homo sapiens*, a fim de validá-la;
- Extrair regras do classificador SVM para que o conhecimento adquirido seja de fácil entendimento. Uma grande contribuição seria obter uma base de conhecimento a

partir dos resultados obtidos. Este conhecimento, que pode ser representado por meio de regras *if-then*, por exemplo, poderá ser utilizado para classificar sequências positivas e negativas arbitrárias. Elas também poderão ser utilizadas para melhorar o conhecimento dos especialistas, uma vez que as regras geradas podem criar novas relações a partir dos dados;

- Aplicar o método de balanceamento proposto à outras bases de dados e em outros problemas de balanceamento, a fim de avaliar seu comportamento e desempenho.
- Realizar comparações do M-Clus com outros métodos não utilizados neste trabalho, como por exemplo: CNN, Tomek links, C-Clear, entre outros.

REFERÊNCIAS

ALMEIDA, M. B. ***SVMs training using re-sampling based on error and a priori strategies for sample selection***. Tese (Dissertacao de Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2002.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. **A study of the behavior of several methods for balancing machine learning training data**. *SIGKDD Explorations*, ACM, v. 6, n. 1, p. 20–29, 2004.

BERKHIN, P. ***Survey of Clustering Data Mining Techniques***. San Jose, CA, 2002.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. **A training algorithm for optimal margin classifiers**. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, n. 144-152, 1992.

BURBIDGE, R.; BUXTON, B. **An introduction to support vector machines for data mining**. *Operational Research Society*, 2001.

CAVENER, D. R.; RAY, S. C. **Eukaryotic start and stop translation sites**. *Nucleic Acids Research*, v. 19, n. 12, p. 3185–3192, 1991.

CHAWLA, N. V. et al. **Smote: Synthetic minority over-sampling technique**. *Journal of Artificial Intelligence and Research*, v. 16, p. 321–357, 2002.

CHAWLA, N. V.; JAPKOWICZ, N.; KOTCZ, A. **Editorial: special issue on learning from imbalanced data sets**. *SIGKDD Explorations*, ACM, v. 6, n. 1, p. 1–6, 2004.

CORTES, C.; VAPNIK, V. **Support-vector networks**. *Machine Learning*, v. 3, n. 20, p. 273–297, 1995.

HATZIGEORGIOU, A. G. **Translation initiation start prediction in human cDNAs with high accuracy**. *Bioinformatics*, v. 18, p. 343–350, 2002.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. **Data clustering: a review**. *ACM Comput. Surv.*, ACM, v. 31, n. 3, p. 264–323, 1999.

JOACHIMS, T. **Making Large-Scale SVM Learning Practical**. MIT Press, Cambridge, MA, USA, p. 169–184, 1999. Disponível em: <http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims_99a.pdf>.

KOHAVI, R. **A study of cross-validation and bootstrap for accuracy estimation and model selection**. In: *IJCAI'95: Proceedings oh 14h International Joint Conference on Artificial Intelligence*. [S.l.]: Morgan Kaufmann Publishers Inc., 1995. p. 1137–1143.

KOZAK, M. **Compilation and analysis of sequences upstream from the translational start site in eukaryotic mRNAs**. *Nucleic Acids Research*, v. 12, p. 857–872, 1984.

KOZAK, M. **Initiation of translation in prokaryotes and eukaryotes**. *Gene*, v. 234, p. 187–208, 1999.

LI, G.-L.; LEONG, T.-Y. **Feature Selection for the Prediction of Translation Initiation Sites**. *Geno. Prot. Bioinfo.*, v. 3, n. 2, p. 73–83, 2005.

LI, H.; JIANG, T. **A class of edit kernels for SVMs to predict translation initiation sites in eukaryotic mRNAs**. In: *RECOMB '04: Proceedings of the eighth annual international conference on Resaerch in computational molecular biology*. [S.l.]: ACM, 2004. p. 262–271.

LI, J.; SEE-KIONG, N.; WONG, L. **Bioinformatics Adventures in Database Research**. In: CALVANESE, D.; LENZERINI, M.; MOTWANI, R. (Ed.). *Database Theory ICDT 2003*. [S.l.]: Springer Berlin / Heidelberg, 2002. (Lecture Notes in Computer Science, v. 2572), p. 31–46.

LIU, H. et al. **Using amino acid patterns to accurately predict translation initiatin sites**. *In Silico Biol*, v. 4, n. 3, p. 255–269, 2004.

MACHADO, E. L. ***Um estudo de limpeza em base de dados desbalanceada e com sobreposiao de classes***. Tese (Dissertaao de Mestrado - Programa de Mestrado em Informatica) — Universidade de Brasilia, Brasilia, 2007.

MACHADO, E. L.; LADEIRA, M. **Um Estudo de Limpeza em Base de Dados Desbalanceada e com Sobreposição de Classes**. In: *XXVII Congresso da Sociedade Brasileira de Computação*. [S.l.]: SBC, 2007. p. 330–340.

MACQUEEN, J. **Some methods for classification and analysis of multivariate observations**. In: *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*. [S.l.: s.n.], 1967. v. 1, p. 281–297.

NAKAGAWA, S. et al. **Diversity of preferred nucleotide sequences around the translation initiation codon in eukayote genomes**. *Nucleic Acids Research*, v. 36, n. 3, p. 861–871, 2008.

NOBRE, C. N.; ORTEGA, J. M.; BRAGA, A. de P. **High efficiency on prediction of Translation Initiation Site (TIS) of RefSeq sequences**. *Lecture Notes in Computer Science*, v. 4643/2, p. 138–148, 2007.

PEDERSEN, A. G.; NIELSEN, H. **Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis**. In: *Proc. 5th International Conference on Intelligence Systems for Molecular Biology*. [S.l.: s.n.], 1997. p. 226–233.

PRUITT, K.; MAGLOTT, D. **Refseq and locuslink: NCBI gene-centered resources**. *Nucleic Acids Research*, v. 29, p. 137–140, 2001.

ROGOZIN, I. et al. **Presence of atg triplets in 5' untranslated regions of eukaryotic cdnas correlates with a 'weak' context of the start codon**. *Bioinformatics*, v. 10, n. 17, p. 890–900, 2001.

SEMOLINI, R. **Support Vector Machines, Inferência Transdutiva e o Problema de Classificação**. *Dissertação de Mestrado, Universidade Estadual de Campinas*, 2002.

STORMO, G. D.; SCHNEIDER, T. D.; GOLD, L. M. **Characterization of translational initiation sites in e. coli**. *Nucleic Acid Res*, v. 10, n. 9, p. 2971–2996, 1982.

TIKOLE, S.; SANKARARAMAKRISHNAN, R. **Prediction of translation initiation sites in human mRNA sequences with AUG start codon in weak Kozak context: A neural network approach**. *Biochemical and Biophysical Research Communications*, v. 369, n. 4, p. 1166 – 1168, 2008. ISSN 0006-291X.

TZANIS, G.; BERBERIDIS, C.; VLAHAVAS, I. **A novel data mining approach for the accurate prediction of translation initiation sites**. In: *7th International Symposium on Biological and Medical Data Analysis*. [S.l.: s.n.], 2006. p. 92–103.

TZANIS, G.; BERBERIDIS, C.; VLAHAVAS, I. **MANTIS: A Data Mining Methodology for Effective Translation Initiation Site Prediction**. In: *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.: s.n.], 2007. p. 6343–6347.

YEN, S.-J.; LEE, Y.-S. **Cluster-based under-sampling approaches for imbalanced data distributions**. *Expert Systems with Applications*, Pergamon Press, Inc., v. 36, n. 3, p. 5718–5727, 2009.

ZENG, F.; YAP, R.; WONG, L. **Using feature generation and feature selection for accurate prediction of translation initiation sites**. In: *Proceedings of 13th International Conference on Genome Informatics*. [S.l.: s.n.], 2002. p. 192–200.

ZIEN, A. et al. **Engineering support vector machine kernels that recognize translation initiation sites**. *Bioinformatics*, v. 16, n. 9, p. 799–807, 2000.

APÊNDICE A – FERRAMENTA PREDICTIS

O objetivo geral da ferramenta PredicTIS é a extração de sequências positivas (que possuem SIT) e negativas (que não possuem SIT) a partir de características determinadas pelo usuário da mesma. Sendo desenvolvida em Borland C++ Builder 6 pela facilidade com que a autora tem essa linguagem, além de oferecer características facilitadoras para implementação de um ambiente com interface de fácil entendimento por parte dos usuários.

Para extração das sequências pode-se selecionar o tamanho da janela a ser utilizado, ou seja, tamanho das regiões *upstream* e *downstream*. Essa característica é importante para validação do tamanho da janela a ser utilizado.

Para a seleção dos negativos há várias possibilidades: inclusão somente de negativos na mesma fase de leitura; inclusão somente de negativos de fora de fase de leitura; ou inclusão de todos os negativos (ambos). Outra característica a ser selecionada é a inclusão de sequências negativas fora da região codificadora ou a exclusão das mesmas. Essas propriedades podem ser selecionadas no formulário de Configuração de Propriedades, podendo ser visualizado na figura 20.

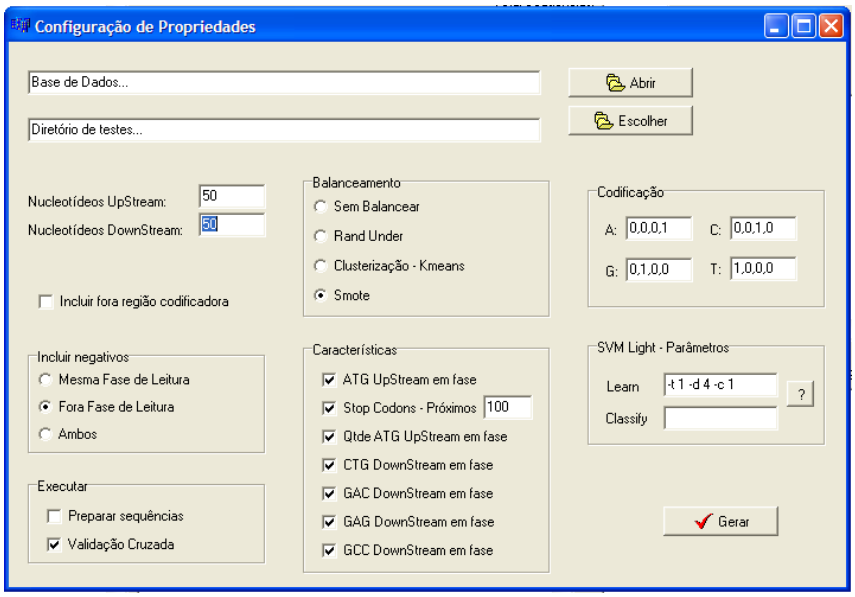

Figura 20: Formulário de configurações de propriedades da ferramenta PredicTIS

A ferramenta também contabiliza a quantidade das seqüências e gera gráficos sobre a distribuição das sequências.

A ferramenta PredicTIS permite:

1.Seleção da base de dados e do diretório onde se irá salvar todas as informações sobre a simulação;

2.Extração das sequências a partir da base selecionada. Para isso, alguns parâmetros de configuração são:

- •Seleção do tamanho da janela. Item importante para esse trabalho, já que vários testes para descobrir o melhor tamanho foram realizados;
- •Forma de extração dos negativos, que podem ser: negativos fora de fase de leitura, negativos mesma fase de leitura e ambos;
- •Inclusão ou não dos negativos fora da região codificadora.

3.Codificação das sequências extraídas, sendo informado pelo usuário da ferramenta o modelo da codificação;

4.Particionamento das sequências positivas e negativas em dez dobras, a fim de se utilizar o algoritmo SVM^{light};

5.Configuração dos parâmetros do algoritmo SVM^{light}, bem como interface e chamada do algoritmo;

6.Cálculo dos resultados através das taxas avaliadas, a saber: acurácia, precisão, sensibilidade, especificidade e acurácia ajustada, e também cálculo das médias das dez execuções (validação cruzada) e do desvio padrão.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo