Universidade Federal de Minas Gerais
Instituto de Ciências Biológicas
Programa de Pós-Graduação em Bioinformática

Desenvolvimento de uma Metodologia para Previsão de Sítios de Início de Tradução

Cristiane Neri Nobre

Belo Horizonte
2007

Cristiane Neri Nobre

Desenvolvimento de uma Metodologia para Previsão de Sítios de Início de Tradução

Tese submetida à Banca Examinadora designada pelo Programa de Pós-Graduação em Bioinformática da Universidade Federal de Minas Gerais.

Orientador: *Dr. Antônio Pádua Braga*

Co-orientador: *Dr. José Miguel Ortega*

Belo Horizonte, MG

2007

Aos meus pais

Wilson e Ana,

e aos meus irmãos;

carinhosamente dedico este trabalho.

Agradecimentos

A realização desta importante etapa da minha vida não seria possível sem a presença constante de Deus, Senhor da minha vida. A Ele, os meus eternos agradecimentos.

Aos meus queridos pais e irmãos pelo apoio, carinho e encorajamento.

Ao meu orientador Braga e co-orientador Miguel, pela orientação nestes quatro anos e meio, por incentivar idéias, pelo tempo despendido ao longo deste trabalho, pela amizade e confiança depositada em mim.

A todos os meus amigos que direta ou indiretamente contribuíram para a realização deste trabalho e que estiveram presentes durante todo este tempo. Não posso me esquecer daqueles que sempre estiveram orando por mim. Muitíssimo obrigada!

Ao Maurício e João por terem me ajudado na parte inicial deste trabalho, quando eu ainda não sabia extrair as seqüências do NCBI. Ao Gabriel e Adriano, por terem colaborado pela disponibilização do laboratório da Bioinfo para que os meus processos pudessem ter sido realizados. À Alessandra, pelas discussões sobre o tema deste trabalho. Ao Saulo pela revisão do inglês do artigo enviado ao BSB e a todos os outros membros do Biodados.

Ao Levi, pelas discussões sobre aprendizado transdutivo. Ao Mendonça, Francisco, Manoel, Cidney e a todos os membros do LITC que sempre me avisavam quando alguma coisa dava errado com os computadores (o computador reinicializava, faltava energia, dentre outros motivos).

Ao amigo Marcelo Azevedo, pelos comentários e conversas sobre os resultados deste trabalho.

À Pontifícia Universidade Católica de Minas Gerais, por ter me liberado 10 horas durante 3 anos e meio. Sem esta liberação, o desenvolvimento deste trabalho teria sido muito mais penoso.

A A.G. Pedersen e G. Tzanis pela generosidade e prontidão em fornecer os seus dados e responder as dúvidas relativas aos procedimentos utilizados em seus trabalhos.

Resumo

A previsão correta do início de tradução em seqüências de mRNA é uma tarefa importante para a anotação genômica. No entanto, fazer uma previsão correta nem sempre é uma tarefa trivial. Na maioria dos casos, a tradução começa no primeiro AUG da seqüência, mas isso nem sempre acontece. Desta forma, essa situação pode ser modelada como um problema de classificação entre as seqüências positivas (codificadoras de proteínas) e negativas (não codificadoras). Para resolvê-lo, os autores deste trabalho propõem a seguinte metodologia: (1) uma forma alternativa de extrair as seqüências negativas; (2) utilização de tamanho de janelas de nucleotídeos menores; (3) alteração na forma de codificação dos nucleotídeos; (4) utilização de um método de balanceamento de classes, visto que trata-se de um problema altamente desbalanceado (da ordem de 1:29, em média) para as bases utilizadas neste trabalho; (5) utilização de uma abordagem de inferência transdutiva, além da inferência indutiva tradicional; e, finalmente, 6) utilização do classificador Support Vector Machine - SVM - com funções simples de *kernel*. Para testar essa metodologia, foram utilizadas as seqüências de Petersen Nielsen e do *RefSeq* (Reference Sequences) do NCBI (National Center for Biotechnology Information) de cinco organismos: *Danio rerio*, *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus* e *Rattus norvegicus*, sob seis níveis de inspeção (*reviewed*, *provisional*, *predicted*, *validated*, *model* e *inferred*).

O resultado são uma acurácia[1], acurácia ajustada[2], precisão[3], sensibilidade[4] e especificidade[5] acima de 95%, em média, utilizando-se seqüências negativas fora de fase de leitura durante o treinamento, janelas de 24 bases, codificação por trinca, balanceamento das seqüências (com o Smote), o classificador SVM transdutivo e considerando-se o modelo de escaneamento, onde a validação é realizada até o SIT.

Palavras-chave: Sítio de Início de Tradução, Support Vector Machine, Smote, Conjunto de Dados Desbalanceado, Inferência Transdutiva.

[1]Acurácia é proporção de todas as predições que são corretas, tanto de SIT quanto de não-SIT.

[2]A acurácia ajustada é definida como sendo a média aritmética entre a sensibilidade e a especificidade.

[3]A precisão mede a proporção dos possíveis SITs que são certamente SITs.

[4]Sensibilidade refere-se à proporção de SIT corretamente classificada como SIT.

[5]Especificidade refere-se à proporção de não-SIT corretamente classificada como não-SIT.

Abstract

The correct prediction of the translation start site in mRNA sequences is an important task in genomic annotation. However, attaining a correct prediction is not trivial. Frequently the translation starts on the first AUG, but that is not a rule. Thus, this problem can be modeled as a classification problem between positive (coding sequences) and negative patterns (non coding sequences). To approach this problem the authors of this work propose the following methodology: (1) an alternative extration of negative patterns; (2) using of shorter sequence window; (3) modification of the codification for the nucleotides; (4) utilization of Smote - method for class balance, since the problem is highly unbalanced (1:29 fold in average) for the bases used in this work; (5) use of a transductive approach besides the traditional inductive inference; and finally, (6) use of the Support Vector Machine (SVM) classifier - with simple kernel functions. To test this methodology sequences collected by Petersen and Nielsen and *RefSeq* (*Reference Sequences*) sequences from NCBI (National Center for Biotechnology Information) from five organisms were used: *Danio rerio*, *Drosophila melanogaste*r, *Homo sapiens*, *Mus musculus* and *Rattus norvegicus*, under six distinct inspection levels (*reviewed*, *provisional*, *predicted*, *validated*, *model* and *inferred*). As a result, accuracy, adjusted accuracy, precision, sensitivity and specificity over 95% were attained, in average, by using negative patterns out of frame during training step, 24 nucleotide windows, codification by triples, pattern balancing with Smote, SVM classifier and by considering a scanning model, in which

validation is tested up to TIS.

Keywords: Translation Initiation Site, Support Vector Machine, Smote, Imbalanced Data Sets, Transductive Inference.

Lista de Figuras

Lista de Tabelas

Lista de Siglas e Abreviaturas

As principais siglas e abreviaturas utilizadas neste trabalho são apresentadas abaixo.

AUC	*Area under the ROC curve*
CDS	*CoDing Sequence* - Seqüência Codificadora
DNA	Ácido Desoxiribonucleico
cDNA	Ácido Desoxiribonucleico complementar
ESTs	*Expressed Sequence Tags* - Etiquetas de seqüências expressas
KNN	*K Nearest Neighbor*
MCP	McCulloch e Pitts
NCBI	*National Center for Biotechnology Information*
NIH	*National Institutes of Health*
RefSeq	*Reference Sequences* - Seqüências de Referência
RBF	*Radial Basis Function*
RN	Rede Neural Artificial
RNA	Ácido Ribonucléico
ROC	*Receiver Operating Characteristic*
mRNA	Ácido Ribonucléico Mensageiro
SITs	Sítios de Início de Tradução
SVM	*Support Vector Machines*
UTRs	*UnTranslated* Regions - Regiões não codificadoras

Sumário

CAPÍTULO 1

Introdução

Este capítulo destina-se à apresentação do problema de Sítio de Início de Tradução (SIT), juntamente com as principais dificuldades relativas ao tema. São apresentados os objetivos, as principais contribuições deste trabalho e uma descrição da metodologia utilizada para alcançar os objetivos propostos.

1.1 Identificação do SIT - uma breve introdução

Os sistemas vivos são conhecidos pelas proteínas que produzem de acordo com sua informação genética. No entanto, somente parte das seqüências transcritas carregam essa informação para codificar proteínas (CDS - CoDing Sequence - seqüência codificadora), enquanto outras partes não (UTR - UnTranslated Regions - regiões não traduzidas de um transcrito) (Zien et al., 2000). Dessa forma, dado um fragmento incompleto de DNA ou mRNA[1], um problema central da biologia computacional é determinar se ele contém CDS; e, a partir dai, descobrir qual proteína ele codifica.

[1] mRNA mensageiro é o produto da transcrição do DNA genômico. O mRNA pode ser editado pela célula para remover íntrons (em eucariotos) ou em outras formas que resultem em diferenças em relação ao DNA genômico transcrito (Gibas, 2001).

De acordo com Pedersen e Nielsen (1997), o reconhecimento do SIT em eucariotos nem sempre começa na primeira metionina (AUG) em seqüências de mRNA ou cDNA[2]. Isto faz com que a previsão de SIT não seja uma tarefa trivial, especialmente quando analisando ESTs[3] e dados genômicos, onde o mRNA completo não é conhecido ou, muitas vezes, quando é conhecido não é livre de erros.

Kozak e Shatki (1978) propuseram um modelo de escaneamento, o qual foi mais tarde atualizado por Kozak (1989). De acordo com esse modelo, a tradução inicia-se no primeiro AUG que tem um contexto apropriado. Foi Kozak (1987) quem desenvolveu a primeira amostra de consenso para uma grande coleção de dados. Esse consenso é GCC[A/G]CC**AUG**G, onde a Guanina (G) aparece na posição +4 (posição que segue o códon AUG) e uma purina, preferencialmente a Adenina (A), aparece na posição -3. Ou seja, essas bases, nessas posições, são grandes contribuintes para a identificação correta do SIT.

Em eucariotos, o modelo de escaneamento supõe que os ribossomos se ligam primeiro à região 5' do mRNA e percorre em direção à região 3' até encontrar o primeiro AUG da seqüência, de acordo com a Figura 1.1 (Kozak, 1984; Kozak, 1986; Kozak, 1999). Dessa forma, começa-se a tradução dos códons para os aminoácidos. Essa é a teoria mais utilizada para identificação do SIT.

[2]cDNA é uma seqüência de DNA gerada artificialmente por transcrição reversa do mRNA. O cDNA representa aproximadamente os componentes codificantes da região do DNA genômico que produziu o mRNA (Gibas, 2001).

[3]ESTs são seqüências curtas de cDNA preparadas a partir de mRNA extraído de uma célula em condições específicas ou em fases de desenvolvimento específicas. As ESTs são utilizadas para identificação rápida de genes, e não abrangem a seqüência codificante completa de um gene (Gibas, 2001).

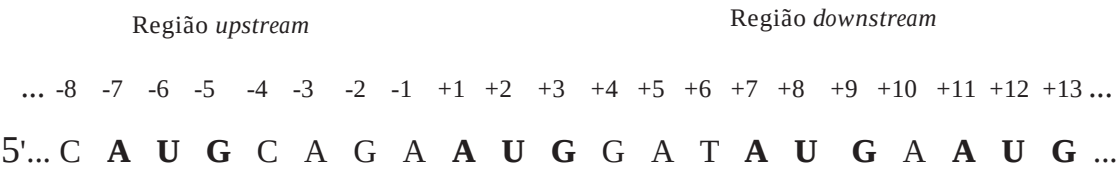


Figura 1.1: Modelo de escaneamento e as posições dos nucleotídeos relativas ao códon AUG. Nesse caso, o segundo AUG é o SIT.

No entanto, existem exceções: devido a um contexto pobre (com ruídos, por exemplo), esse primeiro AUG pode ser ignorado. A tradução pode ainda iniciar em um códon diferente do AUG (AUU e CTG, por exemplo), mas isso é raro em eucariotos (Pain, 1996; Kozak, 1999; Hatzigeorgiou, 2002).

Um dos trabalhos pioneiros em previsão de início de tradução foi realizado por Stormo e colaboradores (Stormo et al., 1982). Mas, ao contrário da grande maioria dos autores que estuda o sítio de início de tradução em eucariotos, esses analisaram o SIT em *Escherichia coli*.

Pedersen e Nielsen (1997) comunicaram um trabalho onde uma Rede Neural Artificial (RN) foi treinada com uma base de dados de vertebrados e o desempenho da rede foi de 85%.

Zien *et al* (2000) conseguiram uma acurácia de 88,6% para essa mesma base de dados, com Support Vector Machines (SVM) e modificando-se a função do *kernel*.

Hatzigeorgiou (2002) utilizou duas RNs e um conceito de escaneamento do ribossomo (validando-se as seqüências até o SIT e desconsiderando os AUGs a justante desse), alcançou uma acurácia de 94% para uma base de seqüências humanas.

Zeng *et al* (2002) alcançaram uma acurácia de 90% validando-se todos os AUGs

e 94,4% validando-se os AUGs até o SIT, com classificador baseado em SVM e utilizando-se um conceito de geração de características.

Em trabalho posterior, Huiqing *et al* (2004) utilizaram seqüências de aminoácidos para fazer a previsão do SIT e conseguiram uma acurácia de 95,15%.

Haifeng e Tao (2004) introduziram uma classe de novos *kernels* baseada em similaridades das seqüências e conseguiram uma acurácia de 99,9%.

Tzanis *et al* (2006) também propuseram um novo conjunto de características importantes para o reconhecimento do SIT e conseguiram uma acurácia de 96,25%.

O Capítulo 2 desta tese apresenta uma revisão detalhada de todos esses métodos de reconhecimento do SIT, classificando-os segundo as técnicas de RN, SVM, Extração de características e Função Discriminante Linear. No entanto, a maioria desses trabalhos precisam criar funções de *kernel* bastante sofisticadas para se obter os desempenhos supracitados, além de utilizarem janelas muito grandes.

Este trabalho propõe, portanto, um nova metodologia para o problema de previsão de SIT, a saber: (1) formas alternativas de se obter as seqüências negativas de uma dada molécula de mRNA; (2) a utilização de janelas menores (12 nucleotídeos nas regiões *upstream* e *downstream* do AUG); (3) uma nova forma de codificação das seqüências; (4) a utilização de um método eficiente de balanceamento, visto que um problema muito freqüentemente observado neste tipo de trabalho de reconhecimento do SIT é o grande desbalanceamento entre as classes, uma vez que, dada uma seqüência de mRNA, temos, a priori, apenas um SIT; o restante corresponde a não-SITs; (5) uma nova abordagem de treinamento utilizando-se a inferência transdutiva que utiliza seqüências não classificadas durante a fase de treinamento; e (6) a utilização de SVM com funções simples de *kernel*. Assim, com essa metodologia, os classificadores padrão existentes na literatura, com função simples de *kernel*, alcançam um desempenho comparável aos obtidos até o momento.

Com o objetivo de validar esses métodos, inicialmente serão apresentados os resultados obtidos a partir das seqüências de Pedersen e Nielsen (1997), uma vez que a maioria dos trabalhos da literatura considera essa base. Além disso, serão apresen-

tados também os resultados a partir de seqüências *RefSeq*[4] (Pruitt e Maglott, 2001) do NCBI[5] de cinco organismos: *Danio rerio*, *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus* e *Rattus norvegicus*, sob seis diferentes graus de inspeção *reviewed*, *provisional*, *predicted*, *validated*, *model* e *inferred*.

1.1.1 Objetivos

Este trabalho tem por objetivos:

- Extrair seqüências dos organismos *Danio rerio*, *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus* e *Rattus norvegicus* do banco de dados público *RefSeq*;
- Estudar e desenvolver novas metodologias que forneçam alta acurácia na previsão do SIT;
- Analisar a qualidade das seqüências, segundo os seis níveis de inspeção existentes: *reviewed*, *provisional*, *predicted*, *inferred*, *model*, e *validated*.

1.1.2 Contribuições

Em virtude do que foi mencionado, este trabalho traz as seguintes contribuições: Inicialmente, é proposta uma *forma alternativa de se obter seqüências negativas para o treinamento*. Um fator que afeta diretamente o número de seqüências negativas e sua qualidade é a forma de extraí-las de uma determinada molécula. Esse fator pode, além de aumentar significativamente o número de seqüências (aumentando ainda mais a base de dados), afetar diretamente a qualidade do classificador.

Ao contrário do grande tamanho de janela atualmente utilizada na previsão do SIT, sugere-se *o uso de janelas menores* com desempenho compáravel aos obtidos. As bases depositadas nos bancos públicos possuem um número muito grande de seqüências e trabalham com janelas muito grandes (200 nucleotídeos, por exemplo) o que acaba gerando um tempo de processamento muito grande. Nesse trabalho,

[4]Disponível em `www.ncbi.nlm.nih.gov/RefSeq/`
[5]`http://www.ncbi.nlm.nih.gov`.

por exemplo, temos 373765 seqüências do *Homo sapiens reviewed* (sendo 10657 positivas e 363108 negativas) e o tempo de treinamento dessas seqüências foi de aproximadamente 28 dias em uma CPU comum (considerando-se a inferência indutiva). Trabalhar com janelas menores pode ajudar nesse sentido.

Traz-se, também, uma *nova forma de codificação das seqüências,* ao contrário da codificação base a base normalmente utilizada. Esse fator pode reduzir significativamente o número de entradas e, além disso, melhorar o desempenho do classificador. Com isso, tem-se também uma redução no tempo de processamento.

A identificação da tarefa de previsão de SIT como sendo um problema transdutivo. O problema de previsão de SIT é inerentemente transdutivo (Goutte et al., 2002), visto que, a priori, temos um número muito grande de seqüências não classificadas (todas as seqüências contendo ATG que estão na região *downstream* do SIT, por exemplo). Além disto, sabemos que muitos ATGs na região *upstream* do SIT podem ser na verdade um SIT alternativo ou ancestral. Assim, todas essas seqüências sem classificação podem ser utilizadas durante o treinamento para melhorar o desempenho do classificador.

Propõe-se a *utilização de SVM com funções simples de kernel.* Acredita-se que, com todas essas etapas anteriores sendo trabalhadas, as funções simples de *kernel* já existentes possam oferecer um bom desempenho, sem a necessidade de se criar funções especiais. Com isso, esse método poderá ser aplicável a várias técnicas de aprendizado de máquina existentes, por exemplo.

Realizar uma análise da qualidade das seqüências, segundo os seis níveis de inspeção existentes. Em 2003 haviam apenas três níveis de inspeção: *reviewed*, *provisional* e *predicted*. Hoje têm-se seis níveis, com o acréscimo das seqüências *inferred*, *model* e *validated*. A análise da qualidade dessas seqüências tem grande contribuição para a área de previsão de SIT.

Assim, este trabalho vem contribuir para a área de bioinformática, em relação à previsão do SIT.

1.1.3 Visão geral deste trabalho

Este trabalho é organizado em capítulos, a saber:

O Capítulo 2 traz uma descrição detalhada dos trabalhos relacionados ao problema de identificação do SIT.

O Capítulo 3 descreve os materiais e métodos propostos, descrevendo as bases de dados, a codificação, o classificador utilizado, dentre outros critérios sugeridos neste trabalho.

O Capítulo 4 apresenta os resultados obtidos e as principais discussões a respeito do SIT.

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

O Capítulo 6 apresenta o apêndice, fonte de inspiração para a codificação utilizada neste trabalho.

E, finalmente, o Capítulo 7 apresenta as principais referências utilizadas.

1.1.4 Conclusões do capítulo

Nesse capítulo foi introduzido o conceito de previsão de SIT, juntamente com os seus desafios. Estabeleceram-se os principais objetivos e as metodologias utilizadas capazes de atender a esses objetivos. E, finalmente, foram relacionadas as contribuições que esse trabalho proporcionará à área de Bioinformática.

CAPÍTULO 2

Previsão do SIT - o estado da arte

A previsão correta do SIT é uma tarefa importante e uma alta acurácia nessa previsão pode ajudar no entendimento do padrão utilizado para iniciar a codificação de proteínas a partir das seqüências de nucleotídeos. No entanto, essa tarefa não é trivial, uma vez que os mecanismos biológicos envolvidos neste reconhecimento não são conhecidos e algumas vezes as seqüências não são completamente livres de erros, além de poderem estar incompletas.

Este capítulo destina-se à apresentação detalhada dos principais métodos de previsão do SIT existentes na literatura, segundo diferentes abordagens. No entanto, como esses métodos utilizam a base de dados gerada por Pedersen e Nielsen (1997), a próxima seção destina-se à uma pequena explanação dessa base. Essa descrição será importante também para o entendimento dos métodos apresentados na Seção 2.4.

2.1 Descrição da base de dados de Pedersen e Nielsen

A base de dados de Pedersen e Nielsen (1997) composta de DNA genômico dos vertebrados *Bos taurus* (boi), *Gallus gallus* (galinha), *Homo sapiens* (homem), *Mus musculus* (camundongo), *Oryctolagus cuniculus* (coelho), *Rattus norcevigus* (rato), *Sus scrofa* (porco) e *Xenopus laevis* (rã) é processada para remoção dos possíveis íntrons e união dos exons, obtendo assim as seqüências de mRNA correspondentes. Dessas seqüências, somente aquelas com o SIT anotado e com pelo menos 10 nucleotídeos na região *upstream* e 150 nucleotídeos na região *downstream* foram selecionadas. Essas seqüências foram então filtradas para remover aquelas pertencentes a uma mesma família gênica, genes homólogos de diferentes organismos e seqüências repetidas.

A Figura 2.1 apresenta um exemplo desta base.

```
299 HSU27655.1 CAT U27655 Homo sapiens
CGTGTGTGCAGCAGCCTGCAGCTGCCCCAAGCCATGGCTGAACACTGACTCCCAGCTGTG 80
CCCAGGGCTTCAAAGACTTCTCAGCTTCGAGCATGGCTTTTGGCTGTCAGGGCAGCTGTA 160
GGAGGCAGATGAGAAGAGGGAGATGGCCTTGGAGGAAGGGAAGGGGCCTGGTGCCGAGGA
240 CCTCTCCTGGCCAGGAGCTTCCTCCAGGACAAGACCTTCCACCCAACAAGGACTCCCCT
.............................................................  80
.............................iEEEEEEEEEEEEEEEEEEEEEEEEE 160
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE 240
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

Figura 2.1: Um exemplo de seqüência da base de Pedersen e Nielsen. As 4 ocorrências de ATG estão sublinhadas e em negrito. O segundo ATG é o SIT, enquanto que os outros 3 são não-SIT. Os 100 nucleotídeos na região *upstream* do SIT são marcados com uma linha simples e os 100 nucleotídeos da região *downstream* são marcados por uma linha dupla. Símbolos ".", "i", e "E" indicam se o nucleotídeo é *upstream* (.), SIT (i), ou *downstream* (E). (*Imagem de: Li et al. (2004), adaptada pelo autor.*)

Nesse exemplo, existem 4 ocorrências de ATGs, sendo que há 1 na região *upstream* do SIT e os outros 2 estão na região *downstream*. Existem 13502 ATGs na base original de Pedersen e Nielsen, sendo que 3312 (24,5%) são SITs; en-

quanto que os outros 10190 (75,4%) são não-SITs. Para cada ATG, seqüências de 100 nucleotídeos das regiões *upstream* e *downstream* foram extraídas. Uma descrição detalhada dessa base, juntamente com o seu *download*, encontra-se em: `http://www.cbs.dtu.dk/databases/NetStart/`.

A maioria dos trabalhos que utiliza essa base de dados extrai, para cada ATG, 100 nucleotídeos a partir das regiões *upstream* e *downstream*. Caso a molécula não contenha este número, os autores completam a seqüência com a letra "N", significando "não conhecido". Pedersen e Nielsen (1997) também adotaram esta metodologia.

Vale ressaltar que essa base de dados foi construída em 1997. Hoje é possível criar bases muito maiores e melhores a partir do *Genbank*[1] (Benson et al., 1997), *RefSeq* (Pruitt e Maglott, 2001) ou outras bases de dados existentes.

As seções seguintes apresentam os principais métodos de previsão do SIT existentes. Para uma melhor visualização, eles foram divididos por categoria; assim as Seções 2.2, 2.3, 2.4 e 2.5 apresentam os métodos que utilizam Redes Neurais Artificiais, SVM, geração de características e Reconhecimento por Função Discriminante Linear, respectivamente (Li et al., 2004).

2.2 Reconhecimento através de Redes Neurais Artificiais

Um dos primeiros trabalhos de previsão se deu em 1982 por Stormo e colaboradores (1982). No entanto, ao contrário da maioria dos trabalhos que fazem previsão em eucariotos, eles reconheceram SIT em *Escherichia coli*. Eles utilizaram *perceptron* (Braga et al., 2000; Haykin, 1999) como classificador. O *perceptron*, modelo proposto por Rosenblatt, é composto por uma estrutura de rede, tendo como unidades básicas nodos MCP - proposto por McCulloch e Pitts (1943) apud (Haykin, 1999) - e por uma regra de aprendizado (Rosenblatt, 1958). Rosenblat demonstrou o teorema de convergência do *perceptron*, que mostra que um nodo MCP treinado com o algoritmo de aprendizado do *perceptron* sempre converge, caso o problema seja linearmente separável (Rosenblatt, 1962). Eles utilizaram codificação de 4 bits

[1] Disponínel em `www.ncbi.nlm.nih.gov/Genbank/`

(A=1000, C=0100, G=0010 e T=0001) e janelas de 51, 71 e 101 nucleotídeos centradas no ATG. Dessas janelas, eles avaliaram que o tamanho 101 oferecia os melhores resultados.

Pedersen e Nielsen (1997) utilizaram uma Rede Neural Artificial (RN), perceptron de múltiplas camadas, treinada com uma janela de 203 nucleotídeos centrada no ATG. Eles utilizaram uma rede *feed-forward* com uma camada intermediária (Braga et al., 2000; Haykin, 1999; Zurada, 1992). A camada de saída tem dois neurônios: o primeiro prediz se a entrada é um SIT; enquanto que o segundo prediz se é não-SIT. Aquele que fornecer maior *score* é o vencedor. Eles utilizaram 0, 1, 2, 5, 10, 20, 30 e 50 neurônios na camada intermediária. Além disso, testaram com janelas de tamanhos 13, 33, 53, 73, 93, 113, 133, 153, 173 e 203 nucleotídeos. Dentre essas configurações testadas, eles verificaram que o tamanho de 203 bases com 30 neurônios na camada intermediária fornecia o melhor desempenho. Eles obtiveram uma sensibilidade, especificidade e acurácia de 78% e 87% e 85%, respectivamente, usando a base descrita na Seção 2.1. Este sistema, chamado de NetStart, está disponível em `http://www.cbs.dtu.dk/services/NetStart`.

Pedersen e Nielsen realizaram ainda uma análise das seqüências 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, fato que já havia sido identificado por Kozak (1987). Avaliaram também as seqüências não-SIT que foram classificadas erradamente como SIT. Como resultado, identificaram que a maioria dessas seqüências estava na mesma fase de leitura do SIT[2], independentemente da região *upstream* ou *downstream*.

Hatzigeorgiou (2002) apresenta um programa com alta acurácia na previsão de SIT, *DIANA-SIT*, usando RNs em seqüências humanas. A base de dados é composta por seqüências de cDNA completas, filtradas para redução de erros. Uma acurácia

[2]Dizer que uma seqüência está em fase com o SIT, significa que ela está alinhada com o SIT. Por exemplo, segundo o modelo de escaneamento do ribossomo, poderíamos dizer que as ...,-9, -6, -3, +4, +7, +10, ..., estão alinhadas com o SIT.

de 94% é obtida utilizando-se um método integrado por dois módulos o qual combina um *consensus-RN* e um *coding-RN*, além do modelo de escaneamento utilizado, onde apenas as seqüências até o SIT são validadas. Esta idéia foi utilizada inicialmente para previsão de início de *splice* por Brunak *et al.* (1991).

O módulo *consensus-RN* avalia o SIT candidato e sua vizinhança mais imediata por meio de uma janela de 12 nucleotídeos. As seqüências foram extraídas a partir das posições -7 a +5 e a codificação de 4 bits (adotada em trabalhos anteriores) foi utilizada. A Figura 2.2 apresenta a estrutura de rede utilizada pela autora, mostrando os dois neurônios na camada intermediária e a topologia não totalmente conectada (onde cada neurônio não está conectado a todos os outros). Para cada molécula, foram extraídas uma seqüência positiva e uma negativa (a primeira depois da seqüência positiva). No total, foram geradas 325 seqüências positivas e negativas.

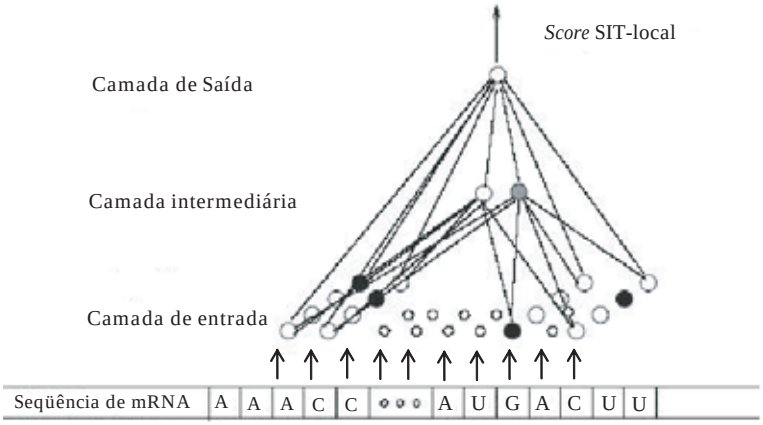


Figura 2.2: O módulo consensus-RN de *DIANA-SIT*. Uma janela de 12 nucleotídeos é apresentada à rede neural. Um *score* alto na saída indica um possível SIT. (*Imagem de: Hatzigeorgiou (2002), adaptada pelo autor.*)

O módulo *coding-RN* foi utilizado para avaliar 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 seqüências dos 54 nucleotídeos, estas são transformadas em um vetor de 64 unidades correspondendo à freqüência de determinado códon na seqüência. A estrutura de rede utilizada nesse caso é apresentada na Figura 2.3.

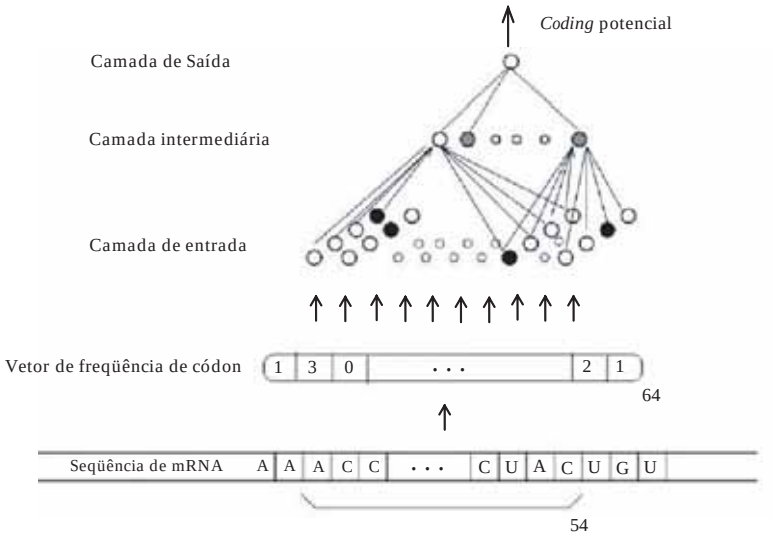


Figura 2.3: O módulo *coding-RN* de DIANA-SIT. Uma janela de 54 nucleotídeos é apresentada à rede neural. Um *score* alto na saída indica um nucleotídeo codificante. (*Imagem de: Hatzigeorgiou (2002), adaptada pelo autor.*)

Os dois módulos propostos são integrados da seguinte forma: dado um ATG candidato, o *consensus-RN* é aplicado a uma janela de 12 nucleotídeos para calcular um consenso S_1. O *coding-RN* é aplicado então às 60 posições que estão na região *upstream* e em fase com o ATG, através de uma janela de 54 nucleotídeos e somando as saídas do *coding-RN* em cada posição para obter um *score* S_2 da região *upstream*. Isso se repete também para a região *downstream* e o *score* S_3 é calculado. O *score* final para o ATG é então obtido por S_1x(S_3-S_2). 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 seqüência.

Note que no modelo de escaneamento do ribossomo, uma seqüência de mRNA é escaneada do início para o final (da região 5' para a 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 (de 94%) obtida por este modelo não pode ser comparada àquela obtida por outros modelos que avaliam todos os ATGs da molécula. Além disto, Hatzigeorgiou usa uma base de dados que é distinta da base de Perdersen e Nielsen.

2.3 Reconhecimento através de SVM

Zien *et al* (1999) trabalharam com a mesma base de dados de Pedersen e Nielsen utilizando SVM. Eles também utilizaram o mesmo tamanho de janela (203 nucleotídeos) e a mesma codificação de cinco *bits* (A, C, G, T ou N). Eles relataram que SVM, com *kernel* polinomial, alcança desempenho comparável ao obtido por Perdersen e Nielsen usando RNs.

Kernels polinomiais de grau d,

$$k(X,Y) = (X \cdot Y)^d = \sum_{i_1} \ldots \sum_{i_d} X[i_1] \times \ldots \times X[i_d] \times Y[i_1] \times \ldots \times Y[i_d]$$

são freqüentemente utilizados em SVM. A codificação utilizada, ajustando um *bit* em cada posição, indica se a base é A (00001), C (00010), G (00100), T (01000) ou N (10000). Conseqüentemente, o produto $(X.Y)$ equivale a contar o número de nucleotídeos que coincidem nas duas seqüências representadas pelos vetores X e Y. Similarmente, $(X.Y)^d$ é equivalente à correlação das freqüências dos nucleotídeos em todas as posições da seqüência d. Esses autores relatam que SVM, com esse tipo de *kernel* padrão, consegue um desempenho no reconhecimento do SIT comparável à RN de Pedersen e de Nielsen.

No *kernel* polinomial $(X.Y)^d$ acima, a correlação de freqüências de nucleotídeos em cada posição da seqüência é utilizada. Entretanto, há um número de razões biológicas que sugerem que devemos considerar somente as posições da seqüência que não estão muitos distantes entre si. Em particular, cada aminoácido é codificado por um tripla de nucleotídeos adjacentes e a região *upstream* do SIT é não codificadora, mas a região *downstream* é codificadora. Desta forma, um *kernel* que considera tais correlações pode ser apropriado para o reconhecimento de SIT.

Inspirado neste raciocínio, Zien *et al.* (1999) mostraram como obter melhorias a partir de uma função de *kernel*, usando uma nova função de *kernel*, chamada de *locality-improved kernel*, com uma pequena janela em cada posição. O *locality-improved kernel* enfatiza correlações entre as posições da seqüência que são próxi-

mas entre si, e um tamanho de 3 nucleotídeos *upstream* e *downstream* é 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. Essa função é definida, dessa forma, como

$$k(X,Y) = \sum_{p=1}^{l} win_p(X,Y)$$

onde,

$$\begin{aligned} win_p(X,Y) &= \left(\sum_{j=-3}^{3} w_j \times (X =_{p+j} Y) \right)^4 \\ &= \sum_{j_1=-3}^{3} \ldots \sum_{j_4=-3}^{3} w_{j_1} \times (X =_{p+j_1} Y) \times \ldots \times w_{j_4} \times (X =_{p+j_4} Y) \end{aligned}$$

Sendo que w_j são pesos que vão aumentando das extremidades para o centro da janela, e

$$(X =_{p+j} Y) = \begin{cases} 1, & \text{se os nucleotídeos na posição } p+j \text{ de} \\ & X \text{ e } Y \text{ são os mesmos} \\ 0, & \text{caso contrário} \end{cases}$$

Com esta função de *kernel*, Zien *et al.* (1999) obtiveram uma sensibilidade de 69,9% e uma especificidade de 94,1%, chegando a uma acurácia total de 88,1%, utilizando a base de dados de Pedersen e Nielsen descrita na Seção 2.1. Eles mostraram, assim, que através de funções simples de *kernel* é possível conseguir uma acurácia melhor do que aquela obtida por Pedersen e Nielsen utilizando RN.

Mais tarde, Zien *et al.* (2000) melhoraram estes resultados através de uma função de *kernel* mais sofisticada, também chamada de *kernel* de Salzberg. O *kernel* de Salzberg é essencialmente um modelo probabilístico condicional das posições de di-

nucleotídeos. Esse *kernel* fornece uma acurácia de 88,6% para a mesma base de dados.

Haifeng e Tao (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 seqüência 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 do Pedersen e Nielsen) do que as abordagens levantadas até o momento, inclusive sobre aquelas que utilizavam SVM com *kernel* polinomial ou *Salzberg*. Eles também testaram essa metodologia com a base de dados de *Homo sapiens* obtidas a partir do *RefSeq* e encontraram uma acurácia de 96,7%.

Este programa, chamado de TISHunter, encontra-se disponível em `http://bioinfo.ucr.edu/~hli/`.

2.4 Reconhecimento por geração, seleção e integração de características

Zeng *et al.* (2002) e Li *et al.* (2004) mostraram que um bom desempenho, comparável aos melhores resultados já descritos, pode ser obtido por uma metodologia baseada em três passos:

1. geração de características candidatas a partir das seqüências;
2. selecão das características relevantes a partir das candidatas, e
3. integração das características selecionadas usando algum método de aprendizado de máquina para construir um sistema de reconhecimento das propriedades específicas das seqüências, neste caso o SIT.

Esses três passos serão apresentados nas subseções seguintes.

2.4.1 Geração de características

Até o momento, os métodos apresentados extraem informações para previsão do SIT basicamente através do conhecimento contido na própria seqüência. Ou seja, os próprios classificadores utilizados extraem estas informações das seqüencias fornecidas. No entanto, uma geração de novos métodos de classificação surge, a partir de 2002, utilizando características extraídas *a priori* como entradas para os classificadores.

Zeng *et al.* (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. Um *K-gram* também pode ser restrito àqueles que estão em fase com o ATG codificante. Cada *k-gram* e sua freqüência no fragmento da seqüência transformam-se em uma característica candidata. Uma outra técnica para produzir características candidatas é a idéia da posição específica do *k-gram*. Ou seja, essa técnica identifica em qual posição do fragmento da seqüência o *k-gram* aparece.

Como mostrado na Figura 1.1 do Capítulo 1, o modelo de escaneamento do ribossomo sugere a leitura dos nucleotídeos da região 5' para a região 3'. Além disso, foi mostrado também que a base "A" do ATG codificante é numerada com +1, o "T" com +2 e assim sucessivamente. Ou seja, a primeira base depois do ATG é numerada com "+4". Da mesma forma, o nucleotídeo que está imediatamente à esquerda do ATG é numerado com "-1", o segundo "-2" em direção à região 5'.

Dessa forma, para uma melhor compreensão da técnica de *k-grams*, considere a Figura 2.1 apresentada na Seção 2.1. Como foi visto, o segundo ATG é o SIT e os 100 nucleotídeos *upstream* desse estão marcados com uma linha simples, enquanto que os 100 nucleotídeos *downstream* estão marcados com linha dupla.

Para o *k-grams* básico, k é o tamanho da sequência de nucleotídeo a ser gerado. Alguns valores típicos para k são 1, 2, 3, 4, e 5. Uma vez que existem 4 possibilidades de letras para cada posição (A, C, G e T) , existem 4^k possíveis *k-grams* para cada

valor de k. Por exemplo, para k=3, um dos *k-grams* é ATG e a freqüência deste *k-gram* é 4 para o exemplo apresentado na Figura 2.1. Assim, a característica candidata é ATG e o seu valor associado é 4 ("ATG=4").

Além disso, como já comentado no Capítulo 1, as regiões *upstream* e *downstream* do SIT são respectivamente não-codificante e codificante. Assim, imagina-se que essas regiões possuam características específicas que as tornam diferentes. Desta forma, é interessante introduzir classes adicionais de *k-grams* para tentar capturar estas diferenças. Estes são os *k-grams upstream* e *downstream*.

Para os *k-grams upstream*, Zeng *et al.* (2002) contam somente ocorrências dos padrões que estão na região *upstream* do SIT. Novamente, para cada valor de k, existem 4^k *k-grams upstream*. Ainda para o caso apresentado na Figura 2.1, para k=3, existem algumas possibilidades de *k-grams*: ATG, com freqüência igual a 1 (uma vez que se tem somente um ATG na região *usptream* do SIT); GCT, com freqüência igual 5; e TTT, com freqüência igual 0. Dessa forma, as características candidatas e os valores que correspondem a estes *k-grams* são "ATG *upstream*=1", "GCT *upstream*=5", e "TTT *upstream*=0".

Da mesma forma, para os *k-grams downstream*, Zeng *et al.* (2002) contam somente ocorrências dos padrões que estão na região *downstream* do SIT. Assim, para cada valor de k, existem 4^k *k-grams downstream*. Para esta região da Figura 2.1, para k=3, algumas possibilidades de *k-grams* são as seguintes: ATG, com freqüência igual a 2 (uma vez que têm-se 2 ATGs na região *downstream* do SIT); GCT, com freqüência igual a 4; e TTT, com freqüência igual a 2. Neste caso, as características candidatas e os valores que correspondem a esses *k-grams* são "ATG *downstream*=2", "GCT *downstream*=4", e "TTT *downstream*=2".

Partindo-se do fato de que o processo biológico de traduzir nucleotídeos em aminoácidos a partir de 3 nucleotídeos (também chamado de códon) inicia-se no SIT, *3-grams* nas posições ..., –9, –6, –3, +4, +7, +10,... são alinhados ao SIT. Zeng *et al.* (2002) chamam os 3-*grams* nas posições ..., –9, –6, e –3, de 3-*grams upstream* em fase, e os 3-*grams* nas posições +4, +7, +10, ..., de 3-*grams downstream* em fase. Como estes 3-*grams* são posições consideradas de grande significado biológico, eles

também são chamados de boas características candidadas. No total, existem 2×4^3 possibilidades destes 3-*grams*. No exemplo em questão, alguns 3-grams *downstream* em fase são: GCT, TTT e ATG, com freqüências iguais a 1 nesses três casos. Assim, as características candidatas com os seus respectivos valores são: "GCT *downstream* em fase=1", "TTT *downstream* em fase=1", e "ATG *downstream* em fase=1". Da mesma forma, alguns 3-grams *upstream* em fase são: GCT, TTT e ATG, com freqüências iguais a 2, 0 e 0, respectivamente. Assim, as características candidatas, agora na região *upstream*, com os seus respectivos valores são: "GCT *upstream* em fase=2", "TTT *upstream* em fase=0", e "ATG *upstream* em fase=0".

Um outro tipo de característica utilizada por Zeng *et al.* (2002) é o que eles denominam de posições específicas de k-*grams*. Para este tipo de k-*grams*, eles armazenam o k-*grams* que aparece em uma determinada posição da seqüência a ser analisada. Neste caso, é suficiente considerar apenas 1-*gram*, isto é, k-*grams* para k=1. Uma vez que o exemplo apresentado na Figura 2.1 apresenta 100 nucleotídeos na região *upstream* e *downstream*, existem 200 posições a serem consideradas. Ainda no exemplo considerado, nas posições –3 e +4 têm-se, respectivamente, uma Adenina (A) e uma Guanina (G). As características candidatas e seus valores associados são: "posição -3 = A" e "posição +4 = G".

Combinando-se todas as características discutidas acima, para $k = 1, ..., 5$, cada seqüência é codificada tendo $(\sum_{k=1}^{5} 4^k + 4^k + 4^k) + 2 \times 4^3 + 200 = 4436$ características. Assim, para o exemplo considerado teríamos: {..., "ATG=4", ..., "ATG *upstream*=1", "GCT *upstream*=5", "TTT *upstream*=0", ...,"ATG *downstream*=2", "GCT *downstream*=4", "TTT *downstream*=2", ..., "GCT *downstream* em fase=1", "TTT *downstream* em fase=1", "ATG *downstream* em fase=1", ..., "GCT *upstream* em fase=2", "TTT *upstream* em fase=0", "ATG *upstream* em fase=0", ..., "posição -3 = A", ...,"posição +4 = G", ...}. A essas características, dá-se o nome de vetor de características.

Essas 4436 características, descritas acima, são geradas para cada uma das 13502 seqüências que contêm ATG na base de dados criada por Pedersen e Nielsen. No entanto, muitas vezes se torna inviável trabalhar com uma quantidade tão grande de características. Assim, a Seção 2.4.2 mostra como pode ser feito o re-

conhecimento do SIT com base em um conjunto menor dessas 4436 características candidatas.

2.4.2 Seleção de características

Como descrito anteriormente, o número de características gerado é, claramente, muito grande. Muitas dessas características podem ser ruído que muitas vezes atrapalham os algoritmos de aprendizagem de máquina típicos. Dessa forma, uma outra etapa importante na metodologia proposta por Zeng *et al.* (2002) é a seleção das características mais significativas para distinguir SIT de não-SIT. Para isto, várias técnicas podem ser utilizadas, incluindo teste-t (Caria, 2001), teste χ^2 (Liu e Setiono, 1995), medida de entropia (Fayyad e Irani, 1993) ou método de seleção de características baseado em correlação (Correlation Feature Selection - CFS) (Hall, 1998).

Utilizando-se a técnica CFS, estes autores selecionaram 9 características, descritas abaixo, a partir das 4436 candidatas extraídas.

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”, e
9. “GCC *downstream* em fase”.

Eles mostraram que essas 9 características são fundamentais para a classificação de SIT e não-SIT. Além disto, eles apresentaram razões biológicas para a maioria delas.

A "posição -3" pode ser explicada pelo já conhecido consenso de Kozak (Kozak, 1984), GCC[A/G]CC**AUG**G, (apresentado na Seção 1.1) para previsão de início de tradução em vertebrados. Kozak mostra que a posição -3 é altamente conservada para identificação do SIT e normalmente apresenta uma purina "A" ou "G" (preferencialmente "A"). Pedersen e Nielsen também analisaram as seqüências e chegaram à mesma conclusão quanto a essa posição, como descrito na Seção 2.2.

O "ATG *upstream* em fase" também pode ser explicado pelo modelo de escaneamento do ribossomo (descrito na Figura 1.1 da Seção 1). Vimos que o ribossomo escaneia o mRNA da região 5' para a região 3' (ou seja, do início para o final) 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. Conseqüentemente, a presença de um ATG na região *usptream* em fase com o SIT pode indicar que o SIT (previsto inicialmente) tem menos chances de ser o SIT verdadeiro. Isto também está de acordo com o trabalho de Rogozin *et al.* (2001) quando mostraram que existe uma correlação negativa entre o conteúdo de informação do sinal de início de tradução e o tamanho da região 5' UTR. Tipicamente, cDNAs contendo longas regiões 5' UTRs com vários ATGs têm um contexto pobre, e ao contrário, cDNAs contendo regiões 5' UTR pequenas sem a presença de ATGs têm contextos fortes. Pedersen e Nielsen (1997) também chegaram a essas conclusões quando fizeram uma avaliação detalhada das previsões erradas feitas pela RN.

Os "TAA *downstream* em fase", "TAG *downstream* em fase" e "TGA *downstream* em fase" também podem ser explicados porque eles correspondem aos *stop-codon* que estão em fase (na região *downstream*) com o SIT. Estas 3 triplas, chamadas de *stop-codon*, não codificam aminoácidos. O processo biológico de traduzir os códons que estão em fase para aminoácidos pára quando um *stop-codon* em fase é encontrado. Desta forma, a presença de qualquer uma dessas três características sinaliza que existe um *stop-codon* em fase nos 100 nucleotídeos da região *downstream* do SIT

(no exemplo considerado). Conseqüentemente, a proteína produzida não deverá ter mais de 33 aminoácidos. Isto é menor do que a maioria das proteínas existentes, indicando que o ATG inicialmente considerado como SIT pode não ser realmente o SIT verdadeiro. Essas 3 características (a presença de um dos três códons de parada) não foram avaliadas por nenhum dos autores mencionados anteriormente.

Zeng *et al.* (2002) não apresentaram explicações biológicas para as 4 características restantes ("CTG *downstream* em fase", "GAC *downstream* em fase", "GAG *downstream* em fase" e "GCC *downstream* em fase").

2.4.3 Integração de características para tomada de decisão

Zeng *et al.* (2002) afirmam que praticamente todos os métodos de aprendizado de máquina podem ser treinados com estas 9 características resultando em uma classificação de SIT com desempenho comparável a outras técnicas desenvolvidas até então.

No trabalho, eles apresentam os resultados obtidos por meio dos classificadores Baysianos (Langley et al., 1992), SVM (Carvalho et al., 2002; Scholkopf et al., 1999), redes neurais e árvore de decisão (como C4.5) (Quinlan, 1993). De acordo com estes autores, utilizando-se o classificador Baysiano, é possível obter uma sensibilidade de 84,3%, especificidade de 86,1%, precisão de 66,3% e acurácia de 85,7%. Eles obtiveram sensibilidade = 73,9%, especificidade = 93,2%, precisão = 77,9% e acurácia = 88,5% por meio de SVM; encontraram sensibilidade = 77,6%, especificidade = 93,2%, precisão =78,8% e acurácia = 89,4% usando redes neurais e sensibilidade = 74,0%, especificidade = 94,4%, precisão = 81,1%, e acurácia = 89,4% usando-se o C4.5.

Nesse mesmo trabalho, eles apresentam mais três maneiras para tentar aumentar a previsão do SIT: 1) usando um *bagging* (Breiman, 1997; Chawla et al., 2003)[3]

[3]A idéia principal do *Bagging* é selecionar (com reposição) n pontos do conjunto de treinamento e usar m classificadores (que, neste caso, foi uma combinação entre os classificadores Baysiano, SVM, Redes Neurais e C4.5) para classificar os padrões de teste. O classificador ótimo será definido pelo voto majoritário. Ou seja, apresenta-se um padrão teste para os m classificadores. A partir das respostas, escolhe-se qual a classe mais votada para este padrão. Isto é repetido para todos os padrões do conjunto de teste.

dos 4 classificadores adotados: Baysianos, SVM, Redes neurais e C4.5; 2) adicionando mais uma característica que seria a distância, contada pelo número de bases, do início da molécula até o SIT; 3) usando o conceito de modelo de escaneamento também utilizado por Hatzigeorgiou (2002), apresentado na Seção 2.2.

Quanto à primeira maneira (usando *Bagging*), eles não encontraram melhorias significativas. Usando a característica de distância eles encontraram uma pequena melhoria nas medidas de desempenho analisadas. Eles sugerem que esta característica poderia ser utilizada em outros trabalhos da área e ressaltam a importância da técnica de geração de características. E, finalmente, usando o modelo de escaneamento, eles conseguem melhorar as quatro medidas de desempenho (sensibilidade = 88,5%, especificidade = 96,3%, precisão = 88,6% e acurácia = 94,4% com o classificador SVM) e compara os resultados obtidos com aqueles obtidos por Hatzigeorgiou (acurácia = 94,4%).

2.4.4 Melhorando o processo de reconhecimento do SIT por meio de geração, seleção e integração de características

Li *et al.* (2003) e Liu e Wong (2003) investigaram uma abordagem alternativa de extração de características baseada em aminoácidos. Para isso, eles extraíram segmentos de seqüências da mesma maneira que foi feito em trabalhos anteriores (100 nucleotídeos nas regiões *upstream* e *downstream* de cada ATG) e consideraram 3-*grams* para aqueles que estão em fase com o ATG. Assim, todos os 3-*grams* que codificam proteínas são convertidos para suas letras de aminoácidos correspondentes, enquanto que os 3-*grams* que correspondem aos *stop-codons*, são convertidos em letras especiais simbolizando um *stop-codon*. Desta forma, os seguintes *k-grams* foram gerados (Li et al., 2003):

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 dois aminoácidos XY aparecem como uma *subtring*

na região *upstream*.

4. XY-down, número de vezes que dois aminoácidos XY aparecem como uma *substring* na região *downstream*.

Além disso, Li *et al.* (2003) também geraram características *booleanas* a partir dos segmentos de seqüências extraídos da base de Pedersen e Nielsen: ATG-up, indicando a presença de um ATG na região *upstream*; pos-3AouG-up, indicando a presença de um "A" ou "G" na posição -3; Pos+4G-down, indicando a presença da "G" na posição +4. Estas duas últimas características também foram inspiradas no consenso de Kozak (Kozak, 1984). Assim, um total de $2 \times 21 + 2 \times 21^2 + 3 = 927$ características foram geradas.

Para selecionar as características mais relevantes dentre as 927, eles utilizaram medida de entropia usando validação cruzada com 3 dobras (Kohavi, 1995) e selecionaram 100 características. A Tabela 2.1 apresenta as nove principais características selecionadas.

Tabela 2.1: As nove características mais importantes selecionadas no trabalho de Li *et al.* (2003) pelo método de entropia em cada uma das 3 dobras.

Dobra	ATG up	STOP down	pos–3AouG up	A down	V down	A up	L down	D down	E down
1	1	2	4	3	6	5	8	9	7
2	1	2	3	4	5	6	7	8	9
3	1	2	3	4	5	6	8	9	7

Interessantemente, a maioria dessas características, exceto A-up e V-down, correspondem àquelas características selecionadas por CFS descritas na Seção 2.4.2. Assim, o "ATG-up" corresponde a "ATG *upstream* em fase"; o *stop-down* corresponde a "TAA *downstream* em fase", "TAG *upstream* em fase" 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"; e "A-down" corresponde à "GCC *downstream* em fase".

Liu e Wong (2003) também usaram os classificadores Baysianos, SVM e C4.5 para medir a acurácia utilizando-se as 100 características selecionadas e obtiveram uma sensibilidade = 70,53%, especificidade = 87,76%, precisão = 65,47%, e acurácia = 83,49% para o classificador Baysiano. Estes resultados são ligeiramente piores do que aqueles apresentados na Seção 2.4.3 utilizando-se a técnica de CFS.

Utilizando-se SVM, eles obtiveram sensibilidade = 80,19%, especificidade = 96,48%, precisão = 88,24%, e acurácia = 92,45%. Estes resultados são melhores do que àqueles obtidos por CFS e melhores também do que os resultados obtidos por Pedersen e Nielsen com RN usando-se apenas a própria seqüência.

Para o classificador C4.5 eles encontraram uma sensibilidade de 74,88%, especificidade de 93,65%, precisão de 79,51% e acurácia de 89,00%. Estes resultados são comparáveis àqueles obtidos por CFS, utilizando-se o mesmo classificador.

Liu e Wong (2003) usam apenas as 9 características selecionadas por entropia e encontram resultados comparáveis a esses.

Mais tarde, Huiqing *et al* (2004) utilizaram essa mesma metodologia para geração de características proposta por Li *et al.* (2003), mas utilizaram apenas padrões de 1-*gram* e 2-*gram*, gerando assim um total de 927 características também (já com as características *booleanas*). A Figura 2.4 apresenta um diagrama com a transformação dos dados em relação ao novo espaço de características adotado por eles.

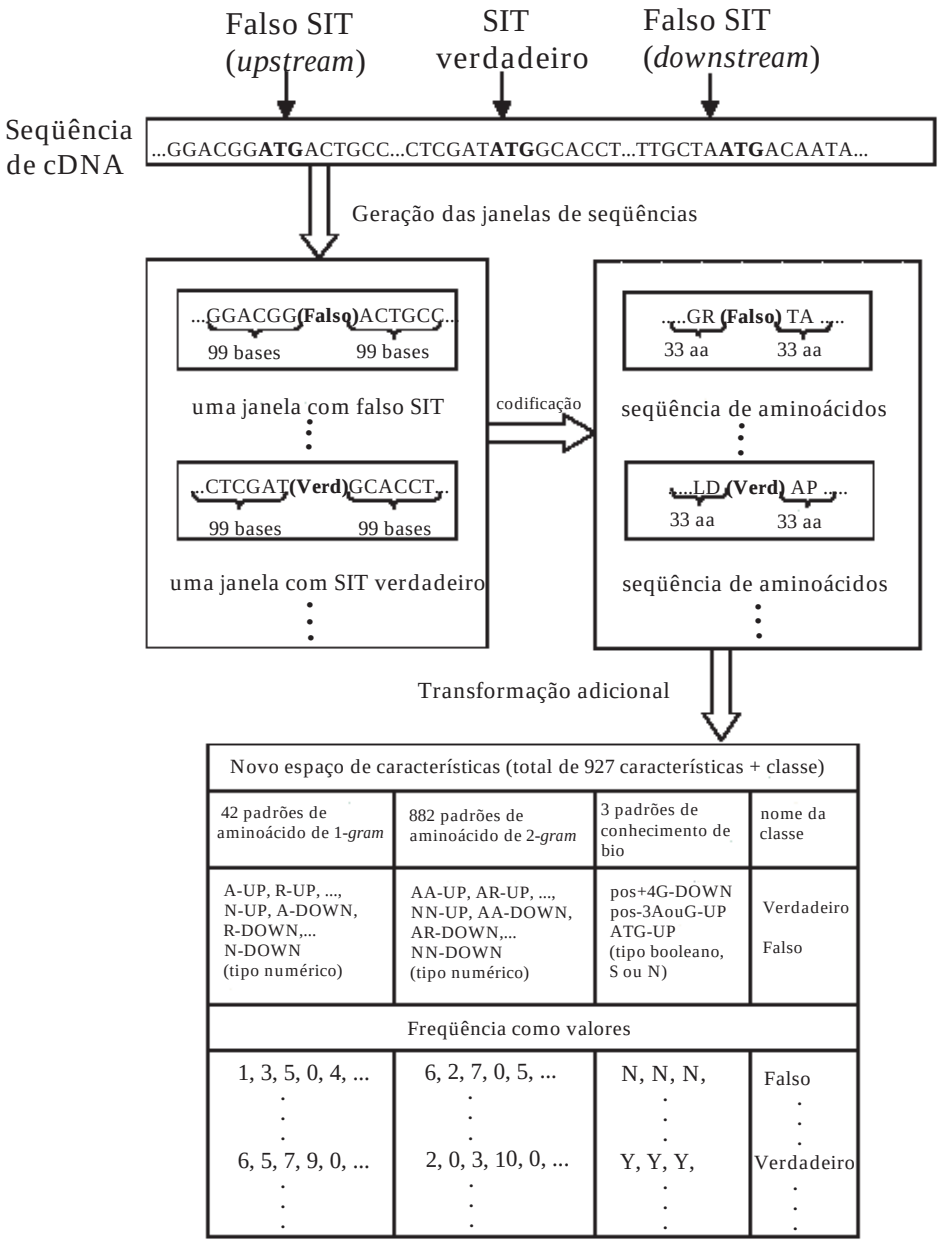


Figura 2.4: Diagrama com a transformação dos dados em relação ao novo espaço de características. (*Imagem de: Huiqing et al. (2004), adaptada pelo autor.*)

Eles testaram a metodologia proposta com três bases de dados: (1) as bases de vertebrados criada por Pedersen e Nielsen (1997), (2) a base de Hatzigeorgiou (2002) que contém 480 seqüências humanas de cDNA, e (3) e uma base formada pelos próprios autores de genes humanos dos cromossomos X e 21. Eles encontraram uma sensibilidade de 86,05%, especificidade de 98,14%, precisão de 93,84%, e acurácia de 95,15%, usando-se SVM com *kernel* quadrático para a base do Perdersen e Nielsen.

Em 2005, Li *et al.* (2005) também desenvolveram um trabalho usando esta técnica de geração de características e propuseram um modelo *Gaussiano* para previsão do SIT. Eles identificaram 16 características, a saber:

1. "tamanho da região *upstream* do ATG",
2. "tamanho da região *downstream* do ATG",
3. "o valor de $log(2)/(1)$",
4. "número de ATGs na região *upstream* do ATG",
5. "número de ATGs na região *downstream* do ATG",
6. "o valor de $log(5)/(4)$",
7. "o número de ATGs na região *upstream* que estão em fase com o ATG",
8. "o número de ATGs na região *downstream* que estão em fase com o ATG",
9. "o valor de $log(8)/(7)$",
10. "o número de *stop-codon* na região *upstream* do ATG"
11. "o número de *stop-codon* na região *downstream* do ATG"
12. "o valor de $log(11)/(10)$",
13. "o número de *stop-codon* na região *upstream* que estão em fase com o ATG"
14. "o número de *stop-codon* na região *downstream* que estão em fase com o ATG"

15. “o valor de $log(14)/(13)$”, e finalmente,

16. “o tamanho da região codificadora do ATG”.

Usando-se essas características, locais e globais, eles encontraram uma especificidade de 98% e uma sensibilidade de 93,6%. O programa para extração de características que eles desenvolveram encontra-se disponível em `http://www.comp.nus.edu.sg/~ligl/software/TISglobal/TISglobal.htm/#ref1`.

Tzanis *et al* (2006) desenvolveram uma metodologia para construir um sistema de vários classificadores para previsão do SIT. Eles utilizaram as seguintes características:

1. “X_up”: o número de aminoácidos X na região *upstream*.

2. “X_down”: o número de aminoácidos X na região *downstream*.

3. “X_{up-down}”: a diferença entre o número de aminoácidos X na região *upstream* e *downstream*, respectivamente.

4. “k-X_{up-pos}”: o número de nucleotídeos x na k-ésima posição dos códons *upstream* que estão em fase ($k \in \{1, 2, 3\}$).

5. “pos-3[AG]_up”: característica *booleana* que analisa a presença dos nucleotídeos “A” ou “G” na posição -3.

6. “pos+4[G]_down”: característica *booleana* que analisa a presença do nucleotídeo “G” na posição +4.

7. “ATG_up”: característica *booleana* que analisa a presença de um códon ATG na região *upstream* e em fase com o SIT.

8. “stop_down”: característica *booleana* que analisa a presença de algum dos 3 *stop-codon* (TAA, TAG e TGA) na região *dowstream* e que esteja em fase com o SIT.

E consideraram a seguinte metodologia:

1. todas as seqüências são escaneadas e todo ATG candidato é identificado, conforme apresentado na Figura 2.5

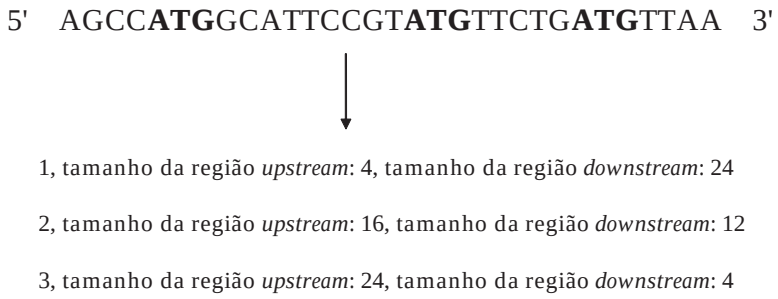


Figura 2.5: Todos os ATGs de uma determinada molécula são extraídos e os tamanhos das regiões *upstream* e *downstream* são calculados para separação entre os subgrupos. (*Imagem de Tzanis et al. (2006), adaptada pelo autor.*)

2. Os ATGs candidatos, encontrados no passo anterior, são agrupados de acordo com o tamanho das regiões *upstream* e *downstream* das seqüências. Desta forma, o conjunto contendo todas as seqüências é dividido em subconjuntos menores e cada classificador trabalha com subconjuntos diferentes. No trabalho, eles dividiram o conjunto maior em 4 subconjuntos. Esta fase é representada pela Figura 2.6.

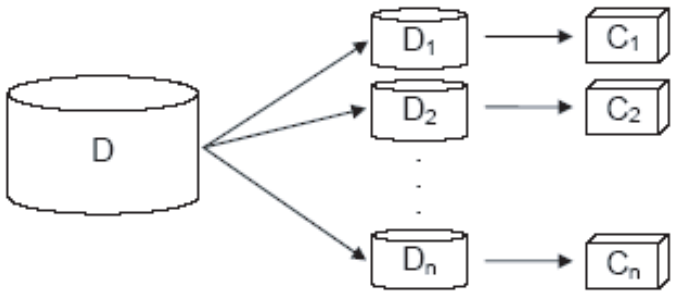


Figura 2.6: A base de dados inicial D é dividida em um número de base de dados menores D_i e, finalmente, um classificador é construído separadamente para cada D_i. (*Imagem de Tzanis et al. (2006)*)

3. para cada um dos ATGs candidatos, os valores das características consideradas

(descritas acima) foram calculados.

4. é realizada uma avaliação das características para cada subgrupo de amostras.

5. as características mais relevantes são selecionadas e um classificador é construído para cada subconjunto de dados.

Segundo Tzanis (comunicação pessoal), eles utilizaram janelas de 99 nucleotídeos nas regiões *upstream* e *downstream*, tanto para as seqüências consideradas SIT como àquelas consideradas não-SIT. Ainda segundo ele, como algumas seqüências continham menos de 99 nucleotídeos nessas regiões, eles consideraram *m* nucleotídeos na região *upstream* e *n* na região *downstream*; sendo, *m* = min(tamanho_*upstream*, 99) e *n* = min(tamanho_*downstream*, 99). No total, foram extraídas 927 características a partir das seqüências.

Tzanis e Vlahavas (2006) compararam a metodologia que eles propuseram com as características sugeridas por Zeng *et al.* (2002) e concluíram que a abordagem deles é de 3,51% a 3,67% melhor em relação à acurácia.

Em trabalho desenvolvido posteriormente, Tzanis *et al.* (2006) adicionaram características baseadas em propriedades químicas dos aminoácidos. Essas características são as seguintes:

1. "X_up e X_down": o número de aminoácidos X na região *upstream* e *downstream*.

2. "X_up-down": a diferença entre o número de aminoácidos X na região *upstream* e *downstream*, respectivamente.

3. "k-X_{up-pos} e k-X_{down-pos}": o número de nucleotídeos X na k-ésima posição dos códons das regiões *upstream* e *dowstream* que estão em fase ($k \in \{1, 2, 3\}$).

4. "pos_–3k e pos_–3(k+1)": indica a presença de aminoácidos nas posições que estão em fase nas regiões *upstream* e *downstream* ($k \geq 1$), respectivamente.

5. “hidrofóbico_up e hidrofóbico_down”: o número de aminoácidos hidrofóbicos nas regiões *upstream* e *downstream*, respectivamente.

6. “hidrofílico_up e hidrofílico_down”: o número de aminoácidos hidrofílicos nas regiões *upstream* e *downstream*, respectivamente.

7. “acíclico_up e acíclico_down”: o número de aminoácidos acíclicos nas regiões *upstream* e *downstream*, respectivamente.

8. “básico_up e básico_down”: o número de aminoácidos básicos nas regiões *upstream* e *downstream*, respectivamente.

9. “aromático_up e aromático_down”: o número de aminoácidos aromáticos nas regiões *upstream* e *downstream*, respectivamente.

10. “alifático_up e alifático_down”: o número de aminoácidos alifáticos nas regiões *upstream* e *downstream*, respectivamente.

11. “não-aromático/não-alifático_up e não-aromático/não-alifático_down”: o número de aminoácidos que não são aromáticos nem alifáticos nas regiões *upstream* e *downstream*, respectivamente.

12. “pos-3[AG]_up”: característica *booleana* que analisa a presença dos nucleotídeos “A” ou “G” na posição -3.

13. “pos+4[G]_down”: característica *booleana* que analisa a presença do nucleotídeo “G” na posição +4.

14. “ATG_up”: característica *booleana* que analisa a presença de um códon ATG na região *upstream* e em fase com o SIT.

15. “stop_down”: característica *booleana* que analisa a presença de algum dos 3 *stop-codon* (TAA, TAG e TGA) na região *dowstream* e que esteja em fase com o SIT.

Baseados nessas características, eles encontraram um desempenho melhor do que haviam encontrado em resultados anteriores.

Eles ainda discutiram sobre a efetividade da característica "distância" considerada em trabalhos anteriores (Liu e Wong, 2003; Zeng et al., 2002; Tzanis et al., 2005), afirmando que essa característica é altamente afetada por características intrínsecas das seqüências. No caso das seqüências de Perdersen e Nielsen (1997), por exemplo, essa característica não faria sentido, visto que somente aquelas que continham pelo menos 10 nucleotídeos na região *upstream* e 150 nucleotídeos na região *downstream* foram selecionadas. Além disso, muitas vezes, não se sabe se a seqüência está completa.

2.5 Reconhecimento por Função de Discriminante Linear

O programa ATGpr desenvolvido por Salamov *et al.* (1998) usa uma função discriminante linear que combina algumas características estatísticas derivadas da seqüencia. Dentre as características observadas por eles, destacam-se: a presença de "C" na região *upstream*, a presença de um outro ATG na região *upstream* e em fase com o ATG candidato, a probabilidade da presença de um sinal peptídeo na região downstream do SIT, dentre outras características consideradas importantes na previsão do SIT.

Em trabalho mais recente (Nishikaw et al., 2000), foi apresentada uma versão melhorada do ATGpr chamada de ATGpr_sim usando informações estatísticas e similaridades com outras proteínas conhecidas para obter uma alta acurácia na previsão de clones de cDNA. ATGpr está disponível em `http://www.hri.co.jp/atgpr/`.

2.6 Conclusões do Capítulo

Nesse capítulo foram apresentados os principais métodos de previsão de SIT existentes na literatura. Os métodos foram divididos segundo à abordagem utilizada: Redes Neurais, Support Vector Machine, Extração de características e Função Dis-

criminante Linear.

Com base nesse métodos, o Capítulo 3 destina-se à apresentação dos materiais e métodos utilizados neste trabalho.

CAPÍTULO
3

Materiais e Métodos

Este capítulo destina-se à apresentação dos métodos propostos neste trabalho. A metodologia seguinte foi utilizada para que a previsão do SIT fosse efetivamente realizada: (1) a escolha das bases a serem analisadas; (2) a forma de se obter as seqüências positivas (que codificam proteínas - da classe SIT) e as negativas (que não codificam proteínas - da classe não-SIT); (3) a escolha da codificação utilizada; (4) a escolha do método de balanceamento que pudesse balancear as classes de SIT e não-SIT; (5) a escolha do classificador que oferecesse um bom desempenho nesta classificação; (6) utilização de inferência transdutiva, além da indutiva tradicional; (7) a escolha das medidas de desempenho a serem utilizadas; e, finalmente, (8) a forma de validação das seqüências, onde todas as seqüências da molécula foram validadas, além da utilização do modelo de escaneamento utilizado por Hatzigeorgiou (2002), onde se valida apenas as amostras até o SIT.

As seções seguintes descrevem cada uma destas fases.

3.1 Bases de dados utilizadas

Uma vez que a maioria dos trabalhos relacionados à previsão de SIT trabalha com a base de dados de Petersen e Nielsen (1997), apresentada na Seção 2.1, este trabalho também buscou analisá-la. Desta forma, inicialmente, foram analisadas as 13502 seqüências de vertebrados obtidas do *Genbank*, sendo 3312 (24,5%) SITs e 10190 (75,4%) não-SITs.

No entanto, como foi citado no Capítulo 2, a base de dados desses dois autores foi gerada em 1997 quando ainda não havia um número muito grande de seqüências disponíveis. Além disso, a forma de extração das seqüências adotada por eles também descartou muitas bases das seqüências, visto que eles cortaram o mRNA em uma região de -150 a +150 bases a partir do SIT.

Assim, além desta base, seqüências de um outro banco de dados mais curado, o *RefSeq* do NCBI, foram analisadas para os seguintes organismos: *Danio rerio* (peixe), *Drosophila melanogaster* (mosca), *Homo sapiens* (homem), *Mus musculus* (camundongo) e *Rattus norvegicus* (rato).

As seqüências *RefSeq* fornecem uma coleção não redundante de seqüências de DNA, RNA e proteínas. As principais características dessas seqüências são:

- Não redundância;
- 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, model,* correspondendo à diminuição do grau de inspeção. As seqüências *reviewed,* por serem revisadas por membros do *staff* do NCBI e seus colaboradores, são, de maneira geral, as melhores seqüências disponíveis de um determinado organismo.

Assim, a Seção 3.2 apresenta o método adotado por este trabalho para a extração das seqüências positivas e negativas a partir dessas seqüências *RefSeq*.

3.2 Extração das seqüências

A Figura 3.1 apresenta um fragmento de um arquivo *RefSeq* original extraído do NCBI.

LOCUS IL2 1047 bp mRNA linear PRI 31-JAN-2003
DEFINITION **Homo sapiens** interleukin 2 (IL2), mRNA.
ACCESSION NM_000586
VERSION NM_000586.2 GI:28178860KEYWORDS
SOURCE Homo sapiens (human)
ORGANISM Homo sapiens Eukaryota; Metazoa; Chordata; Craniata; Vertebrata;

...

COMMENT **REVIEWED REFSEQ**: This record has been curated by NCBI staff.

...

CDS 295..756

```
1   cgaattcccc tatcacctaa gtgtgggcta atgtaacaaa gagggatttc acctacatcc
61  attcagtcag tctttggggg tttaaagaaa ttccaaagag tcatcagaag aggaaaaatg
121 aaggtaatgt tttttcagac aggtaaagtc tttgaaaata tgtgtaatat gtaaaacatt
181 ttgacacccc cataatattt ttccagaatt aacagtataa attgcatctc ttgttcaaga
241 gttccctatc actctcttta atcactactc acagtaacct caactagcga ccgaatgaat
301 atgaatggac tcctgtcttg cattgcacta agtcttgcac ttgtcacaaa cagtgcacct
...
```

Figura 3.1: Fragmento de um arquivo *RefSeq* no formato original. Esse arquivo contém informações tais como organismo, número de acesso, nível de confiança, posição de início do CDS, dentre outras.

Através dessa figura, pode-se verificar que o organismo em questão é o *Homo sapiens* (seqüência *reviewed*), com número de acesso igual N_000586 e cujo CDS é iniciado no nucleotídeo 295; ou seja, o início da tradução desta seqüência começa na posição 295 do mRNA.

A Tabela 3.1 apresenta o número total de moléculas extraídas, a partir do *RefSeq*, para cada organismo, por grau de inspeção.

Tabela 3.1: Número de moléculas disponível no RefSeq para os organismos investigados*.

Grau de inspeção do *Refseq*	**Organismos analisados**				
	D. rerio	*D. melanogaster*	*M. musculus*	*H. sapiens*	*R. norvegicus*
Reviewed	3	20109	210	11226	1240
Provisional	5297	0	13464	5964	7620
Predicted	5334	0	2593	1484	915
Valitaded	9	0	3967	5702	368
Model	20792	0	26255	13574	29970
Inferred	11	0	35	39	3
Total	**31446**	**20109**	**46524**	**37989**	**40116**

* *Download* em 11/02/2007.

A partir dos arquivos, no formato apresentado na Figura 3.1, seqüências positivas e negativas foram extraídas através de um *parsing* (desenvolvido em C++), que selecionava seqüências positivas de 24 bases próximo ao códon ATG (12 nucleotídeos nas regiões *upstream* e *downstream* do ATG) e todas as seqüências negativas usando *ATGs fora da fase de leitura*, conforme ilustrado na Figura 3.2. Essa escolha foi motivada por um estudo realizado por Pedersen e Nielsen (descrito na Seção 2.2) onde eles verificaram que a maioria das seqüências classificadas incorretamente como SIT estavam na mesma fase de leitura do SIT. Isso nos ajudou a imaginar que estas seqüências, na verdade, poderiam ser realmente SITs, e, portanto, foram eliminadas do processo de treinamento. A Seção 4.4 apresenta um estudo dos falsos positivos e traz uma indicação de que essa suposição pode estar correta.

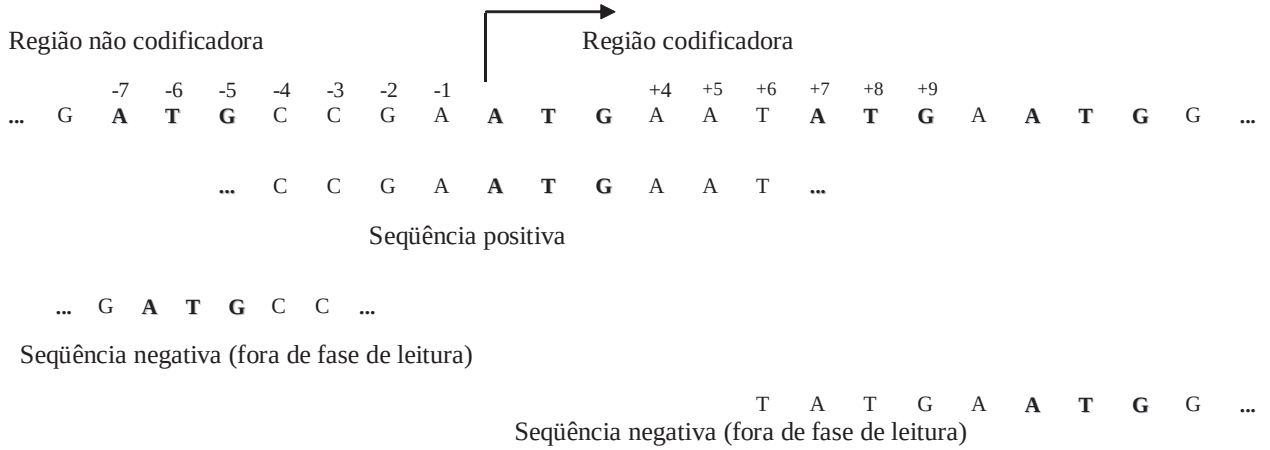


Figura 3.2: Construção das seqüências positivas e negativas usando-se uma janela de 24 nucleotídeos com códon ATG começando-se na décima terceira posição. As seqüências negativas foram obtidas *fora da fase de leitura*. A trinca ATG é desconsiderada da seqüência. (*Imagem do autor.*)

Os testes foram realizados pegando-se todas as seqüências *fora e na mesma fase de leitura* do SIT. No entanto, verificou-se que o melhor desempenho foi obtido pegando-se os negativos *fora de fase de leitura*. Isso pode indicar que provavelmente muitos ATGs que estão na mesma fase de leitura do SIT podem ser, na verdade, prováveis SITs ou SITs alternativos.

Em Pedersen e Nielsen (1997) existe uma indicação de que o tamanho da janela varia entre 13 a 203 nucleotídeos e que o tamanho ideal (que oferece os melhores resultados) é de 203 (100 bases nas regiões *upstream* e *downstream* do SIT). A maioria dos métodos existentes na área de previsão de SIT, inclusive, adota esse tamanho de janela como referência.

Todavia, neste trabalho, optou-se por usar uma janela com apenas 24 bases, favorecendo o estudo do padrão apresentado originalmente por Kozak (Kozak, 1984; Kozak, 1986), que revela que existe um consenso na região -9 a +4 do ATG. No entanto, foram testadas neste trabalho janelas com tamanhos menores e maiores do que o sugerido por Kozak e a janela de -12 a +15 ofereceu resultados bastante satisfatórios. O Capítulo 4 apresenta uma comparação entre os desempenhos obtidos

através de diferentes tamanhos de janela.

Assim, estas seqüências foram extraídas apenas de arquivos contendo a posição de início do CDS maior ou igual a 13 (para se obter as 12 bases antes do ATG). Dessa forma, todas as seqüências contendo CDS abaixo desse número foram desconsideradas.

A Tabela 3.2 apresenta o número total de seqüências positivas e negativas extraídas para cada organismo com base no método utilizado neste trabalho; ou seja, pegando-se todos os negativos *fora de fase de leitura*.

Tabela 3.2: Quantidade de seqüências positivas e negativas para cada organismo, por grau de inspeção, utilizando-se janelas de 12 nucleotídeos *upstream* e *downstream* para cada ATG da molécula.

Grau de inspeção do *Refseq*	**Organismos analisados**									
	D. rerio		*D. melanogaster*		*M. musculus*		*H. sapiens*		*R. norvegicus*	
	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg	Pos	Neg
Reviewed	1	16	15893	379562	182	7130	10657	363108	42	927
Provisional	4597	121052	0	0	11220	322966	5053	153059	6166	151127
Predicted	5015	114046	0	0	2451	70719	1374	33893	774	15818
Valitaded	9	205	0	0	3754	152433	5478	204277	338	8187
Model	7588	167042	0	0	11894	363932	3348	71781	12966	362501
Inferred	5	301	0	0	16	288	11	147	3	113
Total	**17215**	**402662**	**15893**	**379562**	**29517**	**917468**	**25921**	**826265**	**20289**	**538673**

Percebe-se claramente, por essa tabela, que esse problema é altamente desbalanceado, justificando inclusive, investimentos em métodos de balanceamento, objeto que também será estudado neste trabalho. Vale ressaltar aqui que as seqüências negativas foram obtidas *fora de fase de leitura*. Se fossem extraídas todas as seqüências negativas da molécula, o problema seria ainda mais desbalanceado. Um outro ponto muito importante de ser comentado nesse contexto é que nenhum outro trabalho de previsão de SIT tratou desse problema de desbalanceamento.

A Tabela 3.3 apresenta o número de seqüências negativas que estão nas regiões *upstream* e *downstream* de cada ATG; além disso, mostra informações sobre a fase de leitura, para cada organismo analisado, por grau de inspeção.

Tabela 3.3: Quantidade de seqüências negativas de acordo com a fase de leitura*.

Organismo, por grau de inspeção	**Fase de leitura**			
	UMF	**UFF**	**DMF**	**DFF**
Reviewed				
D. rerio	0	0	18	16
D. melanogaster	10706	24902	241335	357875
M. musculus	96	221	3990	6943
H. sapiens	4628	11381	191929	353105
R. norvegicus	15	35	485	900
Provisional				
D. rerio	2337	4814	72130	116848
M. musculus	4233	9869	145447	314893
H. sapiens	2491	5789	78702	147861
R. norvegicus	2020	4602	85144	147251
Predicted				
D. rerio	3304	6318	63573	108384
M. musculus	1963	4516	35477	66642
H. sapiens	1216	2877	17042	31231
R. norvegicus	331	709	8586	15154
Validated				
D. rerio	2	10	97	198
M. musculus	1489	3787	80127	149271
H. sapiens	2572	6455	105488	198425
R. norvegicus	78	215	4723	8005
Model				
D. rerio	5748	12165	91159	157273
M. musculus	13535	30841	176582	335843
H. sapiens	4638	9853	32545	62768
R. norvegicus	6155	13396	195580	351720
Inferred				
D. rerio	6	8	182	295
M. musculus	2	8	145	286
H. sapiens	2	16	65	134
Rattus norvegicus	0	2	57	112

* UMF e UFF correspondem às seqüências que estão na região *upstream* do ATG na mesma fase de leitura ou fora de fase, respectivamente.

DMF e DFF correspondem às seqüências que estão na região *downstream* do ATG na mesma fase de leitura ou fora de fase, respectivamente.

Até a data de 15 de julho de 2007, existiam apenas seqüências *reviewed* para a *Drosophila melanogaster*.

3.3 Codificação utilizada

Seguindo os principais autores da literatura, inicialmente as entradas foram apresentadas ao classificador SVM codificando-se as seqüências (positivas e negativas) em uma *string* binária, usando um esquema de codificação onde cada nucleotídeo é representado por 4 dígitos binários: A=0001, C=0010, G=0100 e T=1000 (codificação com espaço). Dessa forma, inicialmente o SVM possuía 96 entradas ($24\ bases \times 4\ bits = 96$) e uma saída (1 ou 0). Ou seja, a saída é 1 se a seqüência contêm o códon ATG inicializador da tradução ou 0, caso contrário.

No entanto, inspirados por um estudo realizado a partir da freqüência de bases e trincas das seqüências positivas e negativas (apresentado no apêndice A), os autores deste trabalho sugerem uma codificação por trinca. Assim, ao invés de codificar base a base, a codificação foi feita por trinca, com janela deslizante de 3. Ou seja, o deslizamento na janela é feito de três em três posições. Para exemplificar, considere a seqüência: "**CGGACGCAT**"; nesse caso, teríamos as seguintes composições: CGG=1, ACG=1 e CAT=1 e não teríamos o trinucleotídeo GGA, por exemplo. Neste trabalho, chamou-se AAA de 000000, AAC de 000001, AAG de 000010, e assim por diante. Para o mesmo tamanho de janela (24 bases), temos para essa nova codificação $(24 \div 3) \times 6$ bits $= 48$ entradas, contra as 96 entradas utilizando-se a codificação base a base. Assim, temos uma redução de 50% do número de entradas. Os resultados apresentados no Capítulo 4 mostram que essa codificação é bastante promissora.

3.4 Bases de dados desbalanceadas

O problema de classes desbalanceadas ocorre quando existem mais exemplos de uma classe do que da outra. Em certas situações a proporção entre uma classe e outra pode chegar a 1 para 100, 1 para 1000 ou até mais (Chawla et al., 1997).

Esse tipo de problema é de grande importância uma vez que conjuntos de dados com essa característica podem ser encontrados em diversos domínios. Por exemplo, em detecção de fraudes em chamadas telefônicas (Fawcett e Provost, 1997), o número de transações legítimas é muito maior que o número de transações fraudulentas. Em análise de risco para seguradoras (Pednault et al., 2000), somente uma pequena porcentagem dos clientes acionam o seguro em um dado período de tempo. Muitos outros problemas com grande desbalanceamento entre as classes podem ser encontrados na literatura (Batista et al., 2004).

O problema é que 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. Muito freqüentemente o sistema irá fazer uma boa previsão para a classe majoritária e uma previsão ruim para a minoritária.

Existem basicamente duas classes de métodos para o balanceamento da distribuição das classes (Batista et al., 2004):

- *over-sampling* é um método não-heurístico[1] que replica exemplos da classe minoritária com o objetivo de obter uma distribuição mais balanceada,
- *under-sampling* também é um método não-heurístico que objetiva balancear o conjunto de dados pela eliminação de exemplos da classe majoritária.

Vários autores discutem que os métodos *over-sampling* aumentam a probabilidade de ocorrer *overfitting* (sobreajuste ou a incapacidade de classificar itens semelhantes como pertencentes à mesma classe), uma vez que esses fazem cópias exatas dos exemplos das classes minoritárias. Desta forma, um classificador, por exemplo, pode construir regras que são aparentemente precisas, mas que cobrem apenas os exemplos replicados. No caso deste trabalho, inicialmente foram realizados testes apenas replicando os exemplos da classe minoritária, mas os resultados foram bastante negativos (sensibilidade abaixo de 70%, na média). Por outro lado, o problema

[1]Métodos não-heurísticos, também chamados de métodos cegos ou sem informação, não utilizam nenhuma informação adicional a respeito do problema (Russel, 2004). Nesse caso em particular, a replicação e eliminação das seqüências são realizadas sem nenhuma informação sobre elas.

maior relacionado aos métodos *under-sampling* é que esses podem descartar amostras significativas durante o processo de indução (Pednault et al., 2000), (Batista, 2003), (Nitesh et al., 2002), (Batista et al., 2004).

Entretanto, existem métodos heurísticos[2] que tentam superar as limitações intrínsecas dos métodos não-heurísticos, tais como: *Tomek links*, *Condensed Nearest Neighbor* (CNN) *Rule*, *One-sided selection*, *CNN+Tomek Links*, *Synthetic Minority Over-sampling Technique* (Smote), dentre outros (Batista et al., 2004).

3.4.1 Algoritmo de balanceamento utilizado na previsão do SIT

O problema de previsão do SIT é inerentemente desbalanceado, visto que dada uma molécula de mRNA temos, inicialmente, apenas um ATG que codifica proteína; enquanto que todos os outros são não-SITs. Nas bases utilizadas neste trabalho, exceto para a base do Pedersen e Nielsen, temos uma desproporção de 1 para 29, em média, conforme números apresentados na Tabela 3.2.

Esse fato pode gerar um classificador que prediz com alta precisão a classe dos não-SITs, enquanto falha na previsão dos SITs. Para resolver esse problema, neste trabalho, optou-se por utilizar um método de *over-sampling* utilizando o algoritmo Smote para replicação das amostras que codificam proteínas. Esse algoritmo, descrito em Nitesh *et al.* (2002), tem como objetivo principal gerar novos exemplos da classe minoritária através da interpolação entre vários exemplos da amostra. Desta forma, o problema de *overfitting* é evitado.

A Tabela 3.4 apresenta o pseudo código desse método:

[2]Métodos heurísticos, também chamados de métodos com informação, são algoritmos que utilizam o conhecimento específico do problema e podem encontrar solução de forma mais eficiente que uma estratégia sem informação.

Tabela 3.4: Descrição do algoritmo SMOTE, de acordo com Nitesh *et al.* (2002)

Algoritmo SMOTE (T, N, k)

Input: Number of minority class samples T; Amount of SMOTE N%; number of nearest neighbors k
Output: (N/100) * T synthetic minority class samples

```
1.    (* if N is less than 100% randomize the minority class samples as only a random
      percent of them will be SMOTEd. *)
2.    if N < 100
3.       then Randomize the T minority class samples
4.         T = (N/100)*T
5.         N = 100
6.    endif
7.    N = (int)(N/100) (* The amount of SMOTE is assumed to be in integral multiples
      of 100. *)
8.    k = Number of nearest neighbors
9.    numattrs = Number of attributes
10.   Sample[][]: array for original minority class samples
11.   newindes: keeps a count of number of synthetic samples generated, initialized to 0
12.   Synthetic[][]: array for synthetic samples
13.   (*Compute k nearest neighbors for each minority class sample only. *)
14.   for i ⟵ 1toT
15.        Compute k nearest neighbors for i, and save the indices in the nnarray
           Populate (N, i, nnarray)
16.   endfor
      Populate (N, I, nnarray) (* Function to generate the synthetic samples.*)
17.   while N ≠ 0
18.        Choose a random number between 1 and k, call it nn. This step chooses one of the
           k nearest neighbors of i.
19.        for attr ⟵ 1 to numattrs
20.             Compute: dif = Sample[nnarray][nn][attr]-sample[i][attr]
21.             Compute: gap = random nmber between 0 and 1
22.             Synthetic[newindex][attr] = Sample[i][attr] + gap*dif
23.        Endfor
24.        newindex++
25.        N=N-1
26.   endwhile
27.   Return (*End of Populate.*)
      End of Pseudo-Code.
```

Neste trabalho, esse algoritmo foi implementando em C++ e foi responsável por deixar as classes positivas e negativas com o mesmo número de amostras. Com isso, a fase de treinamento se deu de forma balanceada. No Capítulo 4 será apresentado um quadro comparativo mostrando o efeito obtido pelo balanceamento das classes.

3.5 Support Vector Machine - SVM

A SVM pode ser caracterizada como um algoritmo de aprendizado de máquina capaz de resolver problemas de classificação lineares e não lineares. A idéia 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 (Carvalho et al., 2002).

Originalmente denominada "classificador de margem ótima", ela foi introduzida em Boser *et al* (1992) para aplicação em problemas de classificação binária. Em (Cortes e Vapnik, 1995), sendo chamada de "rede 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. O nome "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 e Buxton, 2001).

Ela se baseia na Teoria da Aprendizagem Estatística, por meio da utilização do princípio indutivo de Minimização do Risco Estrutural (Vapnik, 1998). Seu processo de aprendizagem é do tipo supervisionado, em que os dados de treinamento, juntamente com suas saídas correspondentes, são apresentados à máquina de forma que seus parâmetros sejam ajustados (Carvalho et al., 2002).

Seja o conjunto de treinamento $\{\mathbf{x}_i, y_i\}_i^N = 1$, com cada vetor de entrada $\mathbf{x}_i \in \Re^n$ e saída binária correspondente $y_i \in \{-1, +1\}$. Dado um vetor de entrada $\mathbf{x}$, a saída da SVM é representada por $f(\mathbf{x})$. 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 (Scholkopf et al., 1999). 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 $f(\mathbf{x}) = 0$ criada pela SVM é representada por:

$$\omega^t \varphi(\mathbf{x}) + b = 0 \tag{3.1}$$

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

$$\begin{cases} \omega^t \varphi(\mathbf{x}_i) + b \geq +1 & \text{se} \quad y_i = +1 \\ \omega^t \varphi(\mathbf{x}_i) + b \geq -1 & \text{se} \quad y_i = -1 \end{cases} \tag{3.2}$$

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

$$y_i \left[\omega^t \varphi(\mathbf{x}_i) + b \right] \geq 1 \tag{3.3}$$

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 $J_P(\omega, b)$.

Em nossos experimentos, foi utilizada a versão SVM^{light} implementada pelo T.

Joachims (1999) e disponível em `http://svmlight.joachims.org/`. Os resultados que serão apresentados no Capítulo 4 foram realizados com a função polinomial de ordem 4.

O motivo da escolha do algoritmo SVM^{light} para implementar o treinamento da SVM foi baseado nos seguintes fatores:

- É projetado para operar com grande número de dados de treinamento não tendo problema com quantidade de informação armazenada 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;

Estas características são todas necessárias para viabilizar os nossos experimentos, visto que os dados deste trabalho pertencem a um espaço de dimensão 48 e o número de seqüências de treinamento é grande para a maioria das bases utilizadas, conforme apresentado na Tabela 3.2.

3.6 Inferência indutiva versus transdutiva

A inferência transdutiva foi introduzida por Vapnik (1998), junto à teoria do aprendizado estatístico com o objetivo principal de construir um classificador utilizando dois conjuntos de dados: o tradicional de treinamento, em que as amostras já estão previamente classificadas, e o conjunto de previsão, em que as amostras não estão classificadas. Deseja-se com isso, classificar os dados não anotados.

Treinando-se o classificador com esses dois conjuntos de dados, é possível classificar os dados do conjunto de previsão diretamente em um único passo, e como principal vantagem teremos o aumento de informação disponível para o treinamento do algoritmo, e conseqüentemente uma melhora da generalização e desempenho do classificador (Semolini e Zuben, 2002).

Desta forma, a inferência transdutiva representa um modo alternativo para o método tradicional, a inferência indutiva, a qual necessita de dois passos para classificar as amostras do conjunto de previsão, conforme ilustrado na Figura 3.3.

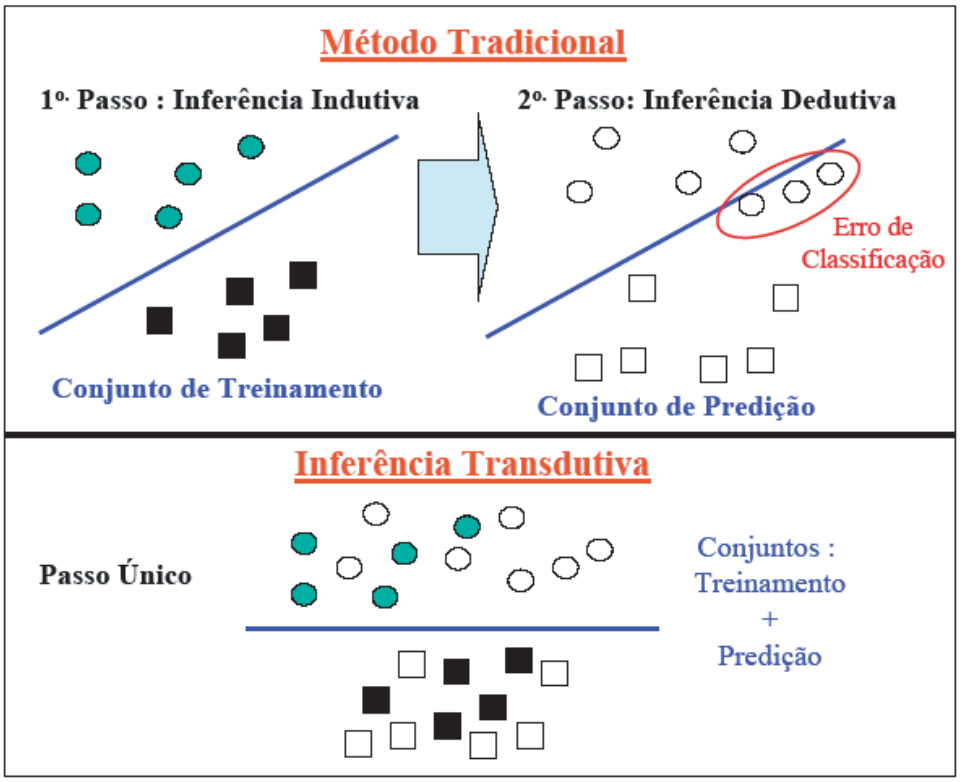


Figura 3.3: Diferença do método tradicional (indutivo-dedutivo) para a inferência transdutiva. (*Imagem de Semolini e Zuben. (2002)*)

No método tradicional, o hiperplano é construído para separar as duas classes (por exemplo, bolas verdes x quadrados pretos na Figura 3.3). Num segundo passo, esse hiperplano é utilizado para deduzir a classificação das amostras do conjunto de previsão. Como ilustração, é mostrada a classe desconhecida a que pertencem as amostras do conjunto de previsão, obedecendo à seguinte notação: bolas brancas pertencem à classe bola verde, enquanto que quadrados brancos pertencem à classe quadrado preto. Verifica-se que houve erro de classificação. Empregando-se os princípios da inferência transdutiva, o hiperplano é encontrado utilizando no treinamento os dois conjuntos de dados, treinamento e previsão, sendo que os dados de previsão ainda não estão classificados. Assim a classificação dos dados do

conjunto de previsão é feita em um único passo. Observa-se que agregando os dados do conjunto de previsão ao treinamento do algoritmo, consegue-se um melhor desempenho.

Assim, na inferência indutiva tradicional, os dados anotados são utilizados para inferir um modelo, o qual é então aplicado aos dados não anotados. A inferência consiste na relação entre o tamanho da margem (associado com a generalização) e o erro do treinamento. Por outro lado, a inferência transdutiva visa maximizar a margem entre os positivos e negativos, enquanto minimiza não somente o número atual de predições incorretas nos exemplos anotados, mas também o número esperado de predições incorretas no conjunto dos exemplos não classificados (Goutte et al., 2002).

3.6.1 Inferência transdutiva e o problema de previsão do SIT

Considerando o modelo de escaneamento do ribossomo apresentado na Figura 1.1, temos que apenas os AUGs que estão na região *upstream* do SIT e o próprio SIT possuem classificação. Ou seja, o ribossomo não identificou o(s) primeiro(s) ATGs da seqüência como sendo SIT, ele seguiu até o segundo ou o terceiro ou mais ATGs, classificando-os como não-SITs até encontrar o ATG que ele classifica como sítio de início de tradução. Neste sentido, não se tem uma classificação para nenhum dos AUGs que estão na região *downstream* do SIT.

Além disto, existe ainda a possibilidade de um AUG (classificado como não-SIT) na região *upstream* do SIT (preferencialmente na mesma fase de leitura) ser um SIT e não poder ter sido selecionado devido a um contexto pobre, por exemplo.

Assim, o problema de previsão de SIT possui características intrínsecas de inferência transdutiva e foi, portanto, trabalhado desta forma, além da abordagem indutiva tradicional.

Para isto, todos os ATGs da molécula que estavam nas regiões *upstream* e *downstream* do SIT e na mesma fase de leitura foram colocados no conjunto de previsão, onde as amostras não são classificadas. Todos os ATGs (tanto da região *upstream* quanto *downstream*) que estavam fora de fase de leitura com o SIT foram classifica-

dos como negativos (não SIT) e, portanto, utilizados na inferência indutiva.

Em nossos experimentos, foi utilizada a versão TSVM (SVM transdutivo) implementada pelo T. Joachims (1999) e disponível no endereço `http://svmlight.joachims.org/`.

3.7 Medidas de desempenho utilizadas

Para se calcular o desempenho do classificador SVM, cinco medidas foram avaliadas: a acurácia, a precisão, a sensibilidade, a especificidade e a acurácia ajustada.

A acurácia, definida pela equação 3.4, mede a proporção de predições, para SITs e não-SITs, que está correta.

$$\text{Acurácia} = 100 * \frac{VP + VN}{VP + VN + FN + FP} \tag{3.4}$$

onde, VP, VN, FP e FN denotam o número de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, respectivamente.

A precisão, definida pela equação 3.5, mede a proporção dos possíveis SITs que são certamente SITs.

$$\text{Precisão} = 100 * \frac{VP}{VP + FP} \tag{3.5}$$

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 SITs que foi corretamente classificada como SITs. A taxa de especificidade, também conhecida como taxa de verdadeiro-negativo, refere-se à porcentagem do número de acertos dentro da classe negativa, ou seja, é a proporção de não-SITs que foi corretamente reconhecida como não-SITs. Estas duas medidas são definidas pelas equações 3.6 e 3.7:

$$\text{Sensibilidade} = 100 * \frac{VP}{VP + FN} \tag{3.6}$$

e

$$\text{Especificidade} = 100 * \frac{VN}{VN + FP} \tag{3.7}$$

Existe ainda o conceito de acurácia ajustada, utilizada por alguns autores (Zeng et al., 2002; Tzanis et al., 2005):

$$\text{Acurácia ajustada} = \frac{Sensibilidade + Especificidade}{2} \tag{3.8}$$

Todos os resultados apresentados no Capítulo 4 são baseados nestas medidas, usando-se o conceito de validação cruzada. A Seção 3.8 formaliza esse conceito.

Além dessas cinco medidas de desempenho, um outro método alternativo para avaliação do desempenho desses classificadores é a análise de curvas ROC[3]. A subseção seguinte descreve esse conceito, visto que ele também foi utilizado neste trabalho.

3.7.1 Curva ROC

Um gráfico ROC pode ser utilizado para analisar a relação entre falsos negativos (FN) e falsos positivos (FP), ou verdadeiros negativos (VN) e verdadeiros positivos (VP), para um determinado classificador (Batista, 2003).

Considerando-se que a classe minoritária, cujo desempenho é o principal foco da análise (nesse caso, a classe positiva), em um gráfico ROC, $VP = 1 - FN$ é associado ao eixo Y e FP é associado ao eixo X. Alguns classificadores possuem parâmetros para os quais diferentes ajustes podem produzir pontos em um gráfico ROC. O desenho de todos os pontos que podem ser produzidos por meio da variação dos parâmetros do classificador produz uma curva ROC. Na prática, essa curva é um conjunto discreto de pontos, incluindo os pontos (0,0) e (1,1), os quais são conectados por segmentos de reta. Na Figura 3.4 é ilustrado um gráfico ROC.

[3]ROC é uma sigla para *Receiver Operating Characteristic*, um termo utilizado em detecção de sinais para caracterizar a relação de perda e ganho entre a taxa de acerto e a taxa de falso alarme em um canal com ruído.

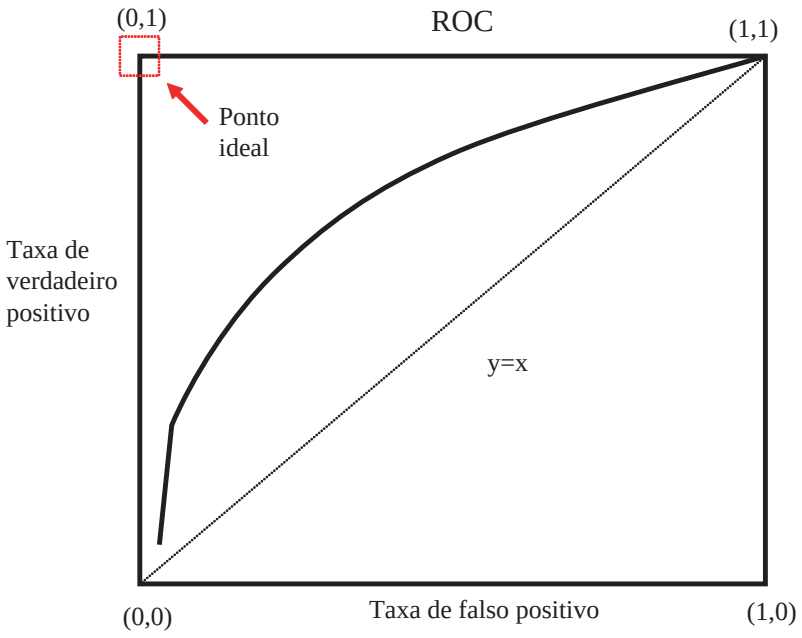


Figura 3.4: Ilustração da curva ROC.

Alguns pontos em um gráfico ROC são importantes de serem observados. O ponto no canto inferior esquerdo (0,0) representa a estratégia de classificar todos os exemplos como pertencentes à classe negativa. O ponto no canto superior direito (1,1) representa a estratégia de classificar todos os exemplos como pertencentes à classe positiva. O ponto no canto superior esquerdo (0,1) representa o classificador perfeito; ou seja, todos os exemplos positivos são classificados corretamente (isto é, não existe nenhum falso negativo) e nenhum exemplo negativo é classificado como positivo (isto é, não existe nenhum falso positivo). A linha $x = y$ representa a estratégia de tentar adivinhar a classe aleatoriamente (Batista, 2003; Bradley, 1997; Nitesh et al., 2002; Prati et al., 2003).

A partir de um gráfico ROC é possível calcular uma medida geral de qualidade, a área sob a curva (AUC)[4]. A AUC é a fração da área total que se situa sob a curva ROC. Essa medida é equivalente a diversas medidas estatísticas para avaliação de modelos de classificação (Bradley, 1997).

[4] *Area under the ROC curve*

3.8 Validação cruzada com k-dobras

No método de validação cruzada com k-dobras (*k-fold cross-validation*) (Kohavi, 1995) o conjunto D, de tamanho N, é dividido em k subconjuntos (dobras) mutuamente excludentes de tamanhos aproximadamente iguais. Onde $1 < k \leq N$. 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 3.5 mostra um esquema ilustrando o método de validação cruzada com k-dobras, mais precisamente uma validação cruzada com 5-dobras.

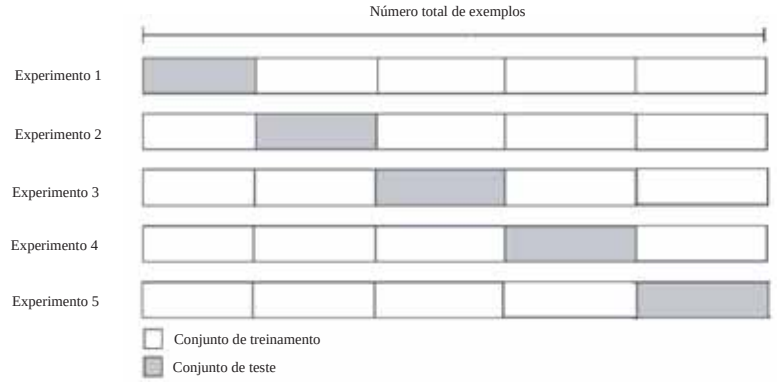


Figura 3.5: Esquema mostrando o exemplo do método de validação cruzada com 5-dobras.

Todos os resultados deste trabalho foram obtidos a partir da média e desvio padrão usando-se validação cruzada com 5 dobras (5-*fold cross validation*).

3.9 Validação das seqüências através do modelo de escaneamento versus todas as seqüências da molécula

Alguns autores acreditam que para os biólogos, talvez um dos pontos mais importantes em relação à previsão do SIT, seja a previsão do verdadeiro SIT, dado uma molécula de mRNA, ao contrário da validação de todos os ATGs desta molécula (Zeng

et al., 2002). Dessa forma, apenas uma previsão seria necessária. Isso corresponde ao modelo de escaneamento de Kozak (Kozak, 1989).

Esse modelo de previsão de SIT, introduzido por Agarwal e Bafna (1998), utilizado também por Hatzigeorgiou (2002) e outros autores (Zeng et al., 2002; Huiqing et al., 2004) desconsidera todos os ATGs que estão depois do verdadeiro ATG.

Neste trabalho, além da validação de todos os ATGs da molécula, foi calculado também o desempenho segundo esse modelo. Em situações onde não se tem o mRNA completo, por exemplo, a validação de toda a molécula se faz necessária.

3.10 Conclusões do capítulo

Nesse capítulo foram descritos os passos da metodologia utilizados neste trabalho, apresentando-se: (1) as bases utilizadas; (2) a forma de extração das seqüências positivas e negativas; (3) a codificação empregada; (4) a apresentação do método de balanceamento utilizado; (5) a escolha do classificador, juntamente com a sua justificativa; (6) a explicação da abordagem transdutiva aplicada ao problema de previsão de SIT; (7) as medidas de desempenho utilizadas e, finalmente, (8) a forma de validação das seqüências, validando todas as seqüências da molécula ou seguindo o modelo de escaneamento.

O Capítulo 4 apresenta os resultados obtidos a partir desses métodos.

CAPÍTULO 4

Resultados e Discussões

Neste capítulo são apresentados os resultados obtidos com os métodos descritos no Capítulo 3, e está organizado da seguinte forma: a Seção 4.1 traz uma comparação entre os diversos métodos propostos neste trabalho. A Seção 4.2, já com a metodologia proposta e validada, apresenta os resultados obtidos para a base de Pedersen e Nielsen e as outras bases *RefSeq* propostas neste trabalho. A Seção 4.3 apresenta algumas curvas ROC obtidas, a Seção 4.4 apresenta uma análise das seqüências falso-positivas e, finalmente, a Seção 4.5 descreve os recursos computacionais gastos no processamento das bases de dados deste trabalho, além dos parâmetros utilizados na classificação.

4.1 Validação da metodologia proposta - uma comparação entre diferentes critérios

As subseções seguintes apresentam os resultados em função dos seguintes critérios:

1. forma de se obter as seqüências positivas (classe SIT) e as negativas (classe

não-SIT);

2. comparação do desempenho com diferentes tamanhos de janelas;
3. escolha da codificação utilizada;
4. escolha de método de balanceamento que pudesse balancear as classes de SIT e não-SIT de forma eficiente;
5. escolha do classificador que oferecesse um bom desempenho nesta classificação;
6. utilização de inferência indutiva e transdutiva;
7. análise de outras formas de validação das seqüências, onde todos os padrões contendo ATG são validados, e utilização do modelo de escaneamento utilizado por Hatzigeorgiou (2002), onde se valida apenas os padrões até o SIT verdadeiro;

Uma vez que os critérios analisados influenciam diretamente no tempo de processamento, apenas as menores bases de dados foram utilizadas (*Mus musculus reviewed* com 182 seqüências positivas e 7130 seqüências negativas e *Rattus norvegicus reviewed* com 42 seqüências positivas e 927 seqüências negativas).

4.1.1 Seleção de seqüências não-SITs usadas no treinamento

Um problema muito importante na previsão do SIT é a seleção correta das seqüências consideradas negativas. Sabemos que, biologicamente, o ribossomo pode utilizar um SIT alternativo mais a montante ou a jusante. Ou seja, podemos ter um ATG codificante na região *upstream* ou *downstream* do ATG realmente verdadeiro.

Nesse sentido, se considerarmos todos os ATGs que teoricamente são negativos como sendo realmente negativos podemos piorar o resultado da classificação, pois muitas dessas seqüências negativas, na verdade, podem ser positivas.

Assim, uma solução encontrada neste trabalho foi considerar, durante a fase de treinamento, apenas os negativos que estão fora de fase com o SIT, desconsiderando, assim, todos os outros que estariam na mesma fase de leitura. Esta proposta, além

de melhorar o resultado da classificação, reduz sensivelmente o desbalanceamento entre as classes.

Os resultados apresentados na Tabela 4.1 validam esta metodologia na eliminação do treinamento das seqüências negativas que estão na mesma fase de leitura do SIT. Neste experimento, utilizou-se uma janela de 12 bases nas regiões *upstream* e *downstream* do SIT e uma codificação por trinca.

Tabela 4.1: Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função da seleção de seqüências não-SITs utilizadas no treinamento*.

	Fora de fase de leitura		**Mesma fase de leitura**	
Mus musculus Reviewed				
	Média	**DP**	**Média**	**DP**
Acurácia	95,95	0,39	93,63	0,30
Acurácia ajustada	96,44	0,32	92,78	0,29
Precisão	91,50	0,87	85,95	0,53
Sensibilidade	98,44	0,22	93,98	0,28
Especificidade	94,43	0,62	91,59	0,39
Rattus Norvegicus Reviewed				
	Média	**DP**	**Média**	**DP**
Acurácia	97,75	0,71	94,39	0,45
Acurácia ajustada	97,64	0,71	90,23	0,49
Precisão	99,59	1,00	91,32	1,23
Sensibilidade	95,64	0,97	90,46	0,97
Especificidade	99,65	0,87	90,00	0,00

* Estes resultados foram obtidos usando-se codificação por trinca, janelas de 12 bases centradas no ATG, classes balanceadas e a validação foi realizada segundo o modelo do ribossomo, ou seja, validando-se todos os ATGs até o CDS e desconsiderando-se todos os outros.

Note que a eficiência do classificador é claramente reduzida usando-se seqüências negativas que estão na mesma fase de leitura do SIT durante o treinamento, conforme pode ser visto principalmente pelas taxas de precisão, sensibilidade e especificidade. Isso mostra que as seqüências negativas fora de fase são realmente

mais negativas, fazendo com que o classificador consiga traçar uma melhor superfície de separação entre as classes.

Vale ressaltar que foram realizados testes pegando-se todos os ATGs (da região 5' e fora de fase), apenas os ATGs da região 5' do CDS (mesma fase de leitura ou não) e os ATGs da região 3' do CDS (mesma fase de leitura ou não). Os melhores resultados são obtidos pegando-se todos os negativos fora de fase de leitura (dados não mostrados).

4.1.2 Tamanho da janela

Uma das primeiras preocupações deste trabalho foi escolher o tamanho ideal de janela para se fazer a previsão do SIT com uma alta acurácia e minimizando o custo computacional. Como já citado anteriormente, os principais trabalhos da literatura utilizam janelas de 200 nucleotídeos. No entanto, este número é inviável quando se trabalha com bases muitos grandes, como é o caso da *Droshophila melanogaster reviewed*, por exemplo, que possui 15893 seqüências positivas e 379.562 seqüências negativas (Tabela 3.2, Seção 3.2).

Neste trabalho, vários testes empíricos foram realizados para se chegar a um tamanho ideal. A Tabela 4.2 apresenta os principais resultados obtidos com quatro diferentes tamanhos de janela. Estes resultados são baseados na validação de todos os ATGs que estão na região 5' do CDS (a montante). Ou seja, os ATGs que estão na região 3' foram desconsiderados.

Tabela 4.2: Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função do tamanho da janela utilizada*.

	Janela 9+3		**Janela 12+12**		**Janela 30+30**		**Janela 99+99**	
Mus musculus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	93,86	0,54	95,95	0,39	97,56	0,22	96,55	0,42
Acurácia ajustada	94,54	0,57	96,44	0,32	97,90	0,19	97,11	0,38
Precisão	87,76	0,79	91,50	0,87	94,59	0,46	92,15	0,82
Sensibilidade	97,28	0,84	98,44	0,22	99,34	0,23	99,76	0,37
Especificidade	91,80	0,60	94,43	0,62	96,45	0,32	94,47	0,62
Rattus Norvegicus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	96,20	0,91	97,75	0,71	96,95	0,67	99,65	0,85
Acurácia ajustada	96,31	0,84	97,64	0,73	96,98	0,62	99,69	0,76
Precisão	94,28	1,78	99,59	1,00	96,12	1,37	100,00	0,00
Sensibilidade	97,62	0,00	95,64	0,97	97,37	0,00	99,38	1,51
Especificidade	95,00	1,67	99,65	0,87	96,59	1,25	100,00	0,00

* Estes resultados foram obtidos usando-se codificação por trinca, classes balanceadas e a validação foi realizada segundo o modelo do ribossomo, ou seja, validando-se todos os ATGs até o CDS e desconsiderando-se todos os outros.

Inicialmente, buscou-se testar a janela sugerida por Kozak (1987), com apenas nove bases na região *upstream* e três na região *downstream* do ATG. Tais estudos apresentam análises considerando-se apenas *uma* base na região *dowstream*, mas, devido à codificação proposta neste trabalho (codificação por trinca), *três* bases foram utilizadas a jusante do SIT. Vários outros testes foram realizados com diferentes tamanhos de janela, mas os mais significativos estão apresentados nessa tabela.

Os testes foram realizados considerando-se apenas as seqüências do *Mus musculus* e *Rattus norvegicus reviewed*, uma vez que o tempo de processamento seria

muito grande se considerássemos as bases de todos os organismos.

Percebe-se, por essa tabela, que seqüências de nove nucleotídeos na região *upstream* e três na região *downstream* oferecem uma acurácia e sensibilidade comparáveis às obtidas com uso de janelas maiores, apesar de as medidas de precisão e especificidade terem sido 6,83% e 4,65% menores, respectivamente, se considerarmos janelas de 30 bases na região *upstream* e *downstream* do ATG (para o organismo *Mus musculus*).

Por outro lado, percebe-se que com 12 bases nas regiões *upstream* e *downstream* do ATG pode-se alcançar um desempenho comparável aos obtidos com janelas maiores, apesar de as medidas de desempenho, para o *Mus musculus*, terem sido ligeiramente menores. No entanto, para o organismo *Rattus norvegicus* este tamanho de janela alcançou resultados bastante satisfatórios.

Os resultados obtidos com janelas de 30 bases nas regiões *upstream* e *downstream* são bastante promissores e para o *Mus musculus* são melhores do que aqueles obtidos por janelas de 99 bases nessas regiões.

É importante ressaltar que em moléculas onde se têm poucas bases na região 5' do CDS, trabalhar com janelas maiores pode ser um problema, como é o caso do *Rattus norvegicus* que teve 96,6% das seqüências desconsideradas por possuir o CDS iniciado em posição menor que 13 (ou seja, não há informação no arquivo de seqüência suficiente para se analisar janelas de 12 bases na região 5').

Além disso, como o número de entradas para o classificador SVM aumenta linearmente com o número de bases usadas, o tempo de processamento é otimizado utilizando-se janelas menores. Por exemplo, para a base do *Mus musculus*, o tempo de processamento é 7,3 vezes menor com janelas de tamanho 12+12 do que usando janelas de tamanho 99+99. Isto pode ser um fator decisivo quando se trabalha com bases muito grandes.

Ainda em relação ao efeito do tamanho da janela, um outro teste foi realizado para avaliar o efeito desse critério ao se validar todos os ATGs da molécula. Assim, a Tabela 4.3 apresenta os mesmos resultados da Tabela 4.2, validando-se, no entanto, todos os ATGs da molécula.

Tabela 4.3: Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função do tamanho da janela utilizada, validando-se todos os ATGs da molécula*.

	Janela 9+3		**Janela 12+12**		**Janela 30+30**		**Janela 99+99**	
Mus musculus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	94,41	0,03	96,63	0,05	97,64	0,04	93,89	0,08
Acurácia ajustada	95,82	0,42	97,53	0,12	98,38	0,21	96,90	0,04
Precisão	21,86	0,18	31,99	0,31	40,03	0,42	19,54	0,21
Sensibilidade	97,28	0,84	98,44	0,22	99,15	0,40	100,00	0,00
Especificidade	94,37	0,02	96,60	0,05	97,61	0,03	95,29	2,21
Rattus Norvegicus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	92,67	0,62	98,67	0,12	90,63	1,01	99,78	0,05
Acurácia ajustada	95,07	0,32	97,20	0,52	93,90	0,52	99,61	0,77
Precisão	27,76	1,68	69,49	1,92	23,21	1,93	92,54	1,25
Sensibilidade	97,62	0,00	95,64	0,97	97,37	0,00	99,38	1,51
Especificidade	92,53	0,64	98,76	0,11	90,43	1,04	99,78	0,04

* Estes resultados foram obtidos usando-se codificação por trinca, classes balanceadas e validando-se todos os ATGs da molécula.

Por essa tabela, percebe-se que, para o organismo *Rattus norvegicus*, a taxa de precisão é significativamente maior, considerando-se janelas de 99 bases nas regiões *upstream* e *downstream*. Todavia, para o Mus musculus, essa taxa continua muito baixa, e não há uma diferença relevante entre os tamanhos analisados.

4.1.3 Desempenho em função da codificação utilizada

Além do tamanho da janela, um outro fator que afeta diretamente o tempo de processamento e a qualidade do classificador é a codificação utilizada. Dessa forma, foram realizados vários testes para se encontrar a codificação que oferecesse a melhor relação custo-benefício entre as medidas de desempenho calculadas e o tempo

de processamento.

Enquanto a codificação binária por base, utilizando-se 4 bits, é a mais comumente utilizada na literatura, foi proposta neste trabalho uma codificação alternativa que codifica a seqüência por trinca (tri-nucleotídeos), ao invés da codificação por base. Essa codificação foi inspirada em um estudo, apresentado no Apêndice A, realizado a partir das seqüências positivas e negativas extraídas. Este estudo mostrou que existe uma distribuição de freqüência de bases e trincas muito característica.

Vale a pena ressaltar ainda que esse procedimento não usa o conceito de janela deslizante da forma tradicional (de 1 em 1). Aqui a janela é deslizada de 3 em 3, o que diminui ainda mais a quantidade de entradas. A Tabela 4.4 apresenta uma comparação entre essas duas opções de codificação. Novamente, por limitação de tempo de processamento, as análises foram concentradas em *Mus musculus* e *Rattus norvegicus reviewed*.

Tabela 4.4: Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função da codificação utilizada*.

	Codification by tri-nucleot		**Codification by base**	
Mus musculus Reviewed				
	Média	**DP**	**Média**	**DP**
Acurácia	95,95	0,39	93,56	0,23
Acurácia ajustada	96,44	0,32	95,79	0,32
Precisão	91,50	0,87	90,59	0,65
Sensibilidade	98,44	0,22	98,32	0,32
Especificidade	94,43	0,62	93,26	0,36
Rattus Norvegicus Reviewed				
	Média	**DP**	**Média**	**DP**
Acurácia	97,75	0,71	98,56	0,36
Acurácia ajustada	97,64	0,71	96,67	0,98
Precisão	99,59	1,00	98,36	0,89
Sensibilidade	95,64	0,97	94,36	1,02
Especificidade	99,65	0,87	98,98	0,06

* Estes resultados foram obtidos usando-se janelas de 12 bases centradas no ATG, classes balanceadas e a validação foi realizada segundo o modelo do ribossomo, ou seja, validando-se todos os ATGs até o CDS e desconsiderando-se todos os outros.

Estes resultados mostram que a codificação proposta neste trabalho alcança resultados comparáveis aos obtidos utilizando-se a codificação de bases individuais. É relevante ressaltar que essa nova codificação reduz o número de entradas pela metade (conforme apresentado na Seção 3.3), resultando em uma redução no tamanho do arquivo de entrada, favorecendo inclusive o trabalho de previsão de SIT em um computador pessoal.

Para se ter uma idéia, usando-se essa codificação por trinca, o tempo de processamento para a base de *Mus musculus* foi reduzido em 19 vezes. Este ganho, adicionado ao ganho obtido pelo tamanho da janela, pode ser bastante significativo em processamento de bases muito grandes, sem perda relevante de desempenho.

4.1.4 Desempenho em função do método de balanceamento

Como discutido na Seção 3.4.1, o problema de previsão de SIT é intrinsecamente desbalanceado, visto que para cada molécula de mRNA temos apenas uma seqüência positiva e várias negativas. Isso pode afetar diretamente o desempenho do classificador diminuindo consideravelmente a taxa de sensibilidade, já que esse sempre tende a classificar qualquer seqüência como pertencente à classe majoritária (neste caso, a negativa).

Neste sentido, várias alternativas de balancear os conjuntos de dados foram testadas, mas o algoritmo Smote, apresentado na Seção 3.4.1, foi o que ofereceu os melhores resultados.

Para se comprovar a relevância de se balancear as classes e a eficiência do Smote, a Tabela 4.5 apresenta os resultados obtidos *com* e *sem* balanceamento.

Tabela 4.5: Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função do método de balanceamento*.

	Com balanceamento (Smote)		**Sem balanceamento**	
Mus musculus Reviewed				
	Média	**DP**	**Média**	**DP**
Acurácia	95,95	0,39	75,99	0,15
Acurácia ajustada	96,44	0,32	68,32	0,27
Precisão	91,50	0,87	99,275	1,02
Sensibilidade	98,44	0,22	36,81	0,78
Especificidade	94,43	0,62	99,84	0,23
Rattus Norvegicus Reviewed				
	Média	**DP**	**Média**	**DP**
Acurácia	97,75	0,71	79,78	4,77
Acurácia ajustada	97,64	0,71	78,57	5,05
Precisão	99,59	1,00	100,00	0,00
Sensibilidade	95,64	0,97	57,15	10,10
Especificidade	99,65	0,87	100,00	0,00

* Estes resultados foram obtidos usando-se codificação por trinca, janelas de 12 bases centradas no ATG e a validação foi realizada segundo o modelo do ribossomo, ou seja, validando-se todos os ATGs até o CDS e desconsiderando-se todos os outros.

Percebe-se claramente que o método Smote aumenta significativamente as taxas de sensibilidade e acurácia. Vemos que a taxa de sensibilidade sem o balanceamento é de apenas 36,81% para o *Mus musculus* e de 57,15% para o *Rattus norvegicus*. Essas mesmas medidas se elevam para 98,44% e 95,64%, respectivamente, usando-se o Smote. Isso corrobora o que foi supracitado: o classificador acabou classificando as amostras como sendo pertencentes à classe majoritária, diminuindo, conseqüentemente, os acertos entre a classe minoritária.

4.1.5 Desempenho do classificador em relação à abordagem indutiva e transdutiva

Uma das propostas deste trabalho foi tratar o problema de previsão de SIT como um problema transdutivo. Ou seja, nesse problema, o que se conhece com certeza é que os ATGs da região *upstream* do CDS são não-SITs. No entanto, não conhecemos a classificação dos ATGs que estão na região *downstream* do SIT.

Como foi discutido na Seção 4.1.1, os ATGs da região *upstream* e *downstream* que estão fora de fase de leitura foram considerados negativos durante o processo de treinamento. E quanto aos ATGs que estão na mesma fase de leitura? O que foi feito deles? A abordagem transdutiva, descrita na Seção 3.6, foi utilizada, então, nesse contexto para tentar descobrir a classificação dessas seqüências. Para serem utilizadas, todas as seqüências positivas são consideradas como sendo +1, as negativas, como sendo -1 e todas as restantes são consideradas “unlabelled” ou “não anotadas”. Ou seja, durante o treinamento, essas seqüências também são consideradas; no entanto, elas são adicionadas ao conjunto de treinamento sem classificação.

A Tabela 4.6(a) apresenta os resultados obtidos considerando-se a abordagem indutiva e a Tabela 4.6(b) apresenta os mesmos resultados considerando-se a abordagem transdutiva. A validação nesse caso foi feita considerando-se o modelo de escaneamento, discutido anteriormente. Ou seja, desconsiderando-se os ATGs da região *downstream* do SIT. Além disso, foi utilizada a codificação por trinca e classes balanceadas.

Tabela 4.6: Comparação entre a abordagem indutiva e transdutiva - Validação das seqüências até o SIT.

(a) Abordagem Indutiva - Validação das seqüências até o SIT

	Janela 9+3		**Janela 12+12**		**Janela 30+30**		**Janela 99+99**	
Mus musculus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	93,86	0,54	95,95	0,39	97,56	0,22	96,55	0,42
Acurácia ajustada	94,54	0,57	96,44	0,32	97,90	0,19	97,11	0,38
Precisão	87,76	0,79	91,50	0,87	94,59	0,46	92,15	0,82
Sensibilidade	97,28	0,84	98,44	0,22	99,34	0,23	99,76	0,37
Especificidade	91,80	0,60	94,43	0,62	96,45	0,32	94,47	0,62
Rattus Norvegicus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	96,20	0,91	97,75	0,71	96,95	0,67	99,65	0,85
Acurácia ajustada	96,31	0,84	97,64	0,73	96,98	0,62	99,69	0,76
Precisão	94,28	1,78	99,59	1,00	96,12	1,37	100,00	0,00
Sensibilidade	97,62	0,00	95,64	0,97	97,37	0,00	99,38	1,51
Especificidade	95,00	1,67	99,65	0,87	96,59	1,25	100,00	0,00

(b) Abordagem Transdutiva - Validação das seqüências até o SIT

	Janela 9+3		**Janela 12+12**		**Janela 30+30**		**Janela 99+99**	
Mus musculus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	96,25	0,35	98,37	0,15	98,36	0,12	99,02	0,99
Acurácia ajustada	96,12	1,58	99,76	0,16	98,54	0,13	99,57	0,29
Precisão	90,00	2,83	96,20	0,21	96,51	0,28	98,81	0,69
Sensibilidade	99,16	0,23	99,63	0,28	99,34	0,23	99,83	0,25
Especificidade	94,76	2,48	99,89	0,03	97,75	0,18	99,61	0,11
Rattus Norvegicus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	97,65	0,44	99,81	0,46	99,16	0,23	99,76	0,85
Acurácia ajustada	97,61	0,49	99,95	0,01	98,13	0,18	99,82	0,76
Precisão	97,61	0,02	99,61	0,95	97,28	1,01	100,00	0,00
Sensibilidade	97,22	0,97	100,00	0,00	98,34	0,03	99,78	0,03
Especificidade	98,00	0,00	99,89	0,01	98,62	0,87	100,00	0,00

Percebe-se que a abordagem transdutiva conseguiu melhorar os resultados em praticamente todos os casos. Ou seja, as seqüências que inicialmente não foram classificadas corretamente com a abordagem indutiva tradicional foram corretamente classificadas usando-se a abordagem transdutiva.

A Tabela 4.7 apresenta os mesmos resultados validando-se todos os ATGs da molécula.

Tabela 4.7: Comparação entre a abordagem indutiva e transdutiva - Validação de todos os ATGs da molécula.

(a) Abordagem Indutiva - Validando-se todos os ATGs da molécula

	Janela 9+3		**Janela 12+12**		**Janela 30+30**		**Janela 99+99**	
Mus musculus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	94,41	0,03	96,63	0,05	97,64	0,04	93,89	0,08
Acurácia ajustada	95,82	0,42	97,53	0,12	98,38	0,21	96,90	0,04
Precisão	21,86	0,18	31,99	0,31	40,03	0,42	19,54	0,21
Sensibilidade	97,28	0,84	98,44	0,22	99,15	0,40	100,00	0,00
Especificidade	94,37	0,02	96,60	0,05	97,61	0,03	95,29	2,21
Rattus Norvegicus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	92,67	0,62	98,67	0,12	90,63	1,01	99,78	0,05
Acurácia ajustada	95,07	0,32	97,20	0,52	93,90	0,52	99,61	0,77
Precisão	27,76	1,68	69,49	1,92	23,21	1,93	92,54	1,25
Sensibilidade	97,62	0,00	95,64	0,97	97,37	0,00	99,38	1,51
Especificidade	92,53	0,64	98,76	0,11	90,43	1,04	99,78	0,04

(b) Abordagem Transdutiva - Validando-se todos os ATGs da molécula

	Janela 9+3		**Janela 12+12**		**Janela 30+30**		**Janela 99+99**	
Mus musculus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	97,57	0,06	98,89	0,04	99,51	0,03	99,93	0,07
Acurácia ajustada	98,05	0,22	98,62	0,02	99,38	0,17	98,97	1,37
Precisão	38,62	0,68	59,06	0,96	76,62	1,08	92,84	0,23
Sensibilidade	98,40	0,38	98,35	0,00	99,24	0,33	99,50	0,71
Especificidade	97,70	0,06	98,90	0,04	99,51	0,03	99,93	0,07
Rattus Norvegicus Reviewed								
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	96,53	0,31	99,80	0,00	96,25	0,35	99,98	0,04
Acurácia ajustada	96,87	0,44	97,58	0,00	95,75	1,06	99,69	0,76
Precisão	45,05	2,07	97,56	0,00	64,00	1,41	100,00	0,00
Sensibilidade	97,22	0,97	95,24	0,00	96,50	2,12	99,38	1,51
Especificidade	96,51	0,32	99,93	0,00	94,00	0,00	100,00	0,00

Observa-se, nesse caso, que a abordagem transdutiva melhora significativamente a taxa de precisão em todos os casos. Isso comprova a relevância dessa abordagem para o problema de predicação de SIT, ou seja, as amostras "unlabelled" ou "não anotadas" ajudam a classificar corretamente as amostras positivas e negativas. Para o organismo *Mus musculus*, por exemplo, com janelas de 99+99, a precisão de 19,54% aumentou para 92,84% e para o *Rattus norvegicus*, usando-se janelas de 12+12, a precisão aumentou de 69,49% para 97,56%.

Isso comprova que o problema de previsão de SIT pode ser visto como um problema transdutivo, e que melhora significativamente as taxas de desempenho analisadas.

4.2 Desempenho do classificador para as outras bases de dados

Uma vez validada a metodologia proposta neste trabalho, foi avaliado o desempenho do classificador sobre a base de dados fornecida por Petersen e Nielsen e posteriormente para as outras bases *RefSeq* (*Homo sapiens*, *Drosophila melanogaster* e *Danio rerio*).

Como descrito na Seção 2.1, a base de dados de Pedersen e Nielsen consiste de seqüências de vertebrados obtidas do *Genbank*, sendo 3312 SITs e 10190 não-SITs. Para avaliarmos os resultados, a metodologia supracitada também foi utilizada nesse caso. Ou seja, foram extraídas seqüências de 24 bases (12+12), utilizando-se apenas os negativos fora de fase de leitura, aplicando-se o método Smote para o balanceamento das classes e usando-se a codificação por trinca.

Como apresentado na Seção 2.1, a maioria dos autores completava as seqüências com "N" (significando base "não conhecida") quando não se tinha o tamanho de janela desejado. Ou seja, ao extraírem as seqüências, se elas não possuíssem o tamanho desejado na região *upstream* ou *dowstream* de cada ATG (no caso 100 bases), eles completavam a seqüência com a letra "N".

Assim, a Tabela 4.8 apresenta os resultados obtidos para essa base de dados

completando-se as seqüências com “N” e sem completá-las, nesse último caso descartando-se as seqüências que não continham 12 bases nas regiões *upstream* ou *downstream*. Além disso, a base foi testada também com janelas de 100 bases nas regiões *upstream* e *downstream* do SIT. A codificação utilizada, nesses dois últimos casos, foi exatamente a codificação encontrada na literatura: A=00001, C=00010, G=00100, T=01000 e N=10000.

Tabela 4.8: Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade para a base de dados de Petersen e Nielsen*.

	Janela de 12+12 sem N	**Janela de 12+12 com N**	**Janela de 100+100 com N**
Acurácia	92,79	95,63	99,99
Acurácia ajustada	84,99	93,38	99,78
Precisão	86,98	91,75	99,96
Sensibilidade	86,65	92,98	99,89
Especificidade	83,32	93,78	99,67

* “N” corresponde a uma base não conhecida. Ou seja, quando não se tinha o tamanho de janela desejado, as seqüências eram completadas com essa letra.

Das 3212 seqüências positivas, 58 não continham 12 bases na região *upstream* ou *downstream* do ATG e das 10190 seqüências negativas, 987 não possuiam o tamanho desejado. Assim, todas elas precisaram ser completadas com “N”. Essa é uma das razões pela qual acreditamos que o resultado tenha sido melhor, completando-se as seqüências com N. Isso pode ser aplicado também para a janela de 100+100, onde muitas seqüências precisaram ser completadas com “N” e com um número muito grande de “N”, inclusive. Ou seja, o classificador possivelmente pode ter identificado esse padrão.

Vale lembrar que Pedersen e Nielsen (1997) encontraram 85% de acurácia, 78% de sensibilidade e 87% de especificidade para essa mesma base, usando-se RN, janelas de 100+100 bases e completando-se o tamanho da janela com a letra “N”. Zien *et al* (2000), utilizando SVM com alterações na funções do *kernel* e usando-se o mesmo esquema de codificação anterior, encontraram 88,6% de acurácia. Zeng *et al* (2002), usando uma janela de 200 nucleotídeos e o conceito de geração de

características, alcançaram 90% de acurácia sobre a mesma base de dados. Além disso, incorporando o modelo de escaneamento do ribossomo, eles obtiveram uma acurácia de 94,4%. Huiqing *et al* (2004), trabalhando com SVM, janelas de 99+99 e seqüências de aminoácidos, obtiveram uma sensibilidade de 86,05%, especificidade de 98,14%, precisão de 93,84% e acurácia de 95,15%. Haifeng e Tao (2004) introduziram uma classe de novos *kernels* baseada em similaridades das seqüências e encontraram uma acurácia, sensibilidade e especificidade de 99,9%, 99,92% e 99,82% a partir de janelas de 30 bases *upstream* e 180 *downstream*. Tzanis *et al* (2005) propõem um novo conjunto de características e conseguem uma acurácia de 96,25% extraindo essas características a partir de uma janela de 99 nucleotídeos nas regiões *upstream* e *downstream* do ATG.

Dessa forma, os resultados apresentados na Tabela 4.8 estão de acordo com os encontrados na literatura para essa mesma base de dados e possuem algumas vantagens: trabalha-se com janelas menores, usando funções simples de *kernel* e com um tempo de processamento bastante reduzido.

No entanto, como já foi exposto essa base foi utilizada apenas para servir como comparação, visto que praticamente todos os trabalhos relacionados ao tema a utilizam. Mas sabemos que as seqüências do *Genback* não são as melhores existentes. Assim, o foco principal deste trabalho foi relativo a bases *RefSeq*. Aparentemente, vários trabalhos que utilizam a base de Petersen e Nielsen podem ter sido afetados pelo uso de “N”.

Dessa forma, as Tabelas 4.9, 4.10, 4.11, 4.12 e 4.13 apresentam os resultados das bases *RefSeq* do *Danio rerio, Drosophila melanogaster, Mus musculus, Homo sapiens* e *Rattus norvegicus* considerando-se a abordagem indutiva e transdutiva. “BDI” (Base de Dados Insuficiente) indica que a base de dados é insuficiente (normalmente com menos de 11 seqüências). “BDNE” (Base de Dados Não Existente) indica que não existe base de dados para aquele grau de inspeção e “ND” (Não Determinado) significa que os dados não foram determinados até o presente momento.

Os resultados foram obtidos usando-se codificação por trinca, janela de 12+12 bases centrada no ATG, classes balanceadas e a validação foi realizada segundo o

modelo do ribossomo, ou seja, validando-se todos os ATGs até o CDS e desconsiderando-se todos os ATGs da região 3' do CDS.

Tabela 4.9: Desempenho do classificador para o *Danio rerio* utilizando-se a abordagem indutiva e transdutiva, validando-se as seqüências até o SIT.

Danio rerio - inferência indutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	BDI		95,19	4,08	95,33	1,03	BDI		62,54	15,39	BDI	
Acurácia ajustada			91,26	6,36	97,12	2,74			19,16	26,32		
Precisão			98,33	1,80	94,29	5,24			66,99	21,31		
Sensibilidade			91,63	9,46	96,79	3,92			17,42	39,97		
Especificidade			90,89	3,26	97,45	1,56			20,89	12,68		
Danio rerio - inferência transdutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	BDI		97,45	0,02	98,00	1,41	BDI		86,47	15,38	BDI	
Acurácia ajustada			96,54	0,16	98,73	0,73			84,17	22,57		
Precisão			99,14	0,04	99,01	0,01			39,23	19,67		
Sensibilidade			95,73	0,08	97,89	1,46			82,65	40,00		
Especificidade			97,35	0,23	99,56	0,01			85,68	5,14		

BDI = Base de dados insuficiente. As bases do *Danio rerio reviewed, validated e inferred* possuem apenas 1, 9 e 5 seqüências positivas, respectivamente.

Tabela 4.10: Desempenho do classificador para o *Drosophila melanogaster* utilizando-se a abordagem indutiva e transdutiva, validando-se as seqüências até o SIT.

Drosophila melanogaster - inferência indutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	89,26	2,36	BDNE		BDNE		BDNE		BDNE		BDNE	
Acurácia ajustada	76,46	5,89										
Precisão	87,56	3,24										
Sensibilidade	73,37	6,45										
Especificidade	79,56	5,32										
Drosophila melanogaster - inferência transdutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	ND		BDNE		BDNE		BDNE		BDNE		BDNE	
Acurácia ajustada												
Precisão												
Sensibilidade												
Especificidade												

BDNE = Base de dados não existente. Até o presente momento, existem somente seqüências *reviewed* para a *Drosophila melanogaster*.
ND = Valores não determinados. Devido ao grande tempo de processamento, não foi possível, até o momento, obter estes resultados.

Tabela 4.11: Desempenho do classificador para o *Mus musculus* utilizando-se a abordagem indutiva e transdutiva, validando-se as seqüências até o SIT.

Mus musculus - inferência indutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	95,95	0,39	74,54	6,48	90,96	0,16	84,23	84,23	75,36	11,54	BDI	
Acurácia ajustada	96,44	0,32	71,89	4,05	76,27	0,60	79,32	79,32	84,56	4,35		
Precisão	91,50	0,87	92,70	2,35	90,55	0,24	83,53	83,53	89,58	14,82		
Sensibilidade	98,44	0,22	68,32	3,48	75,73	0,56	78,77	78,77	86,75	3,25		
Especificidade	94,43	0,62	75,46	4,62	76,82	0,65	79,87	79,87	82,36	5,45		
Mus musculus - inferência transdutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	98,37	0,15	ND		ND		98,61	0,23	ND		BDI	
Acurácia ajustada	99,76	0,16					98,26	0,70				
Precisão	96,20	0,21					99,51	0,03				
Sensibilidade	99,63	0,28					97,73	0,95				
Especificidade	99,89	0,03					98,78	0,45				

BDI = Base de dados insuficiente. A base do *Mus musculus inferred* possui apenas 16 seqüências positivas.
ND = Valores não determinados. Devido ao grande tempo de processamento, não foi possível, até o momento, obter estes resultados.

Tabela 4.12: Desempenho do classificador para o *Homo sapiens* utilizando-se a abordagem indutiva e transdutiva, validando-se as seqüências até o SIT.

Homo sapiens - inferência indutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	91,45	2,56	79,79	10,38	98,35	0,13	74,51	10,35	71,48	16,66		
Acurácia ajustada	92,18	2,75	73,66	15,97	98,90	0,45	52,49	20,75	59,49	24,14		
Precisão	82,56	3,09	90,91	12,35	94,61	0,29	90,07	10,45	58,72	10,07	BDI	
Sensibilidade	91,21	2,47	71,43	31,07	99,24	0,36	43,72	37,94	60,28	8,02		
Especificidade	93,15	3,02	75,89	0,87	98,56	0,54	61,26	3,56	58,69	0,26		
Homo sapiens - inferência transdutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia			94,59	3,02	99,53	0,19	96,91	1,78	86,32	5,23		
Acurácia ajustada			96,05	3,30	99,60	1,35	98,09	1,45	79,86	2,91		
Precisão	ND		96,40	3,94	96,34	0,27	94,65	2,40	78,98	6,45	BDI	
Sensibilidade			96,44	4,04	99,73	0,11	97,59	0,59	80,24	3,25		
Especificidade			95,65	2,56	99,47	2,58	98,59	2,30	79,48	2,56		

BDI = Base de dados insuficiente. A base do *Homo sapiens inferred* possui apenas 11 seqüências positivas.
ND = Valores não determinados. Devido ao grande tempo de processamento, não foi possível, até o momento, obter estes resultados.

Tabela 4.13: Desempenho do classificador para o *Rattus norvegicus* utilizando-se a abordagem indutiva e transdutiva, validando-se as seqüências até o SIT.

Rattus norvegicus - inferência indutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	97,75	0,71	78,94	11,36	98,29	0,49	96,64	0,40	88,74	0,2		
Acurácia ajustada	97,64	0,71	61,54	14,33	98,31	0,53	99,11	0,11	79,99	2,24		
Precisão	99,59	1,00	97,11	2,14	97,26	1,09	94,285	0,64	91,17	0,05	BDI	
Sensibilidade	95,64	0,97	59,84	25,21	98,97	0,70	100,00	0,00	80,67	2,12		
Especificidade	99,65	0,87	63,23	3,45	97,65	0,36	98,23	0,23	79,32	2,36		
Rattus norvegicus - inferência transdutiva												
	Reviewed		***Provisional***		***Predicted***		***Validated***		***Model***		***Inferred***	
	Média	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**	**Média**	**DP**
Acurácia	99,81	0,46	89,16	10,28	99,95	4,50	98,09	0,34	98,50	0,58		
Acurácia ajustada	99,95	0,01	76,19	8,99	99,58	0,32	99,48	0,15	98,44	0,58		
Precisão	99,61	0,95	95,33	4,29	98,79	8,54	96,94	0,55	99,60	0,97	BDI	
Sensibilidade	100,00	0,00	75,49	12,31	99,27	0,60	100,00	0,00	97,22	0,97		
Especificidade	99,89	0,01	76,89	5,68	99,89	0,04	99,25	0,03	99,65	0,87		

BDI = Base de dados insuficiente. A base do *Rattus norvegicus inferred* possui apenas 3 seqüências positivas.

Por esses dados, percebe-se a abordagem transdutiva realmente é eficiente para tratar esse problema. Vale ressaltar ainda que nem todos os dados foram obtidos devido ao tempo de processamento ser muito grande para as bases grandes. Para se ter uma idéia, a base do *Homo sapiens Reviewed* transdutiva está sendo processada desde o final de junho de 2007 em servidor de alto desempenho e até o dia da defesa desta tese (06 de agosto de 2007) ainda não havia terminado.

De maneira geral, as bases *Reviewed* fornecem bons resultados, sob inferência transdutiva. Recentemente foram acrescentadas as categorias *Validated*, *Model* e *Inferred*. Nota-se que a categoria *Model* apresenta resultados insatisfatórios, com exceção de *Rattus norvegicus*. Isto sugere que a produção desses registros pode ser melhorada utilizando-se o procedimento relatado neste trabalho. Assim, juntamente com as propostas que foram feitas para o ajuste do classificador, ressalta-se a importância da utilização dos melhores dados possíveis.

A utilização de Smote como uma técnica eficiente de balanceamento sugere que seja possível a utilização de um número não muito alto de seqüências de referência para o treinamento. Isso é muito importante quando se aplica a abordagem em um projeto genoma novo.

4.3 Curva ROC para os organismos Mus musculus e Rattus norvegicus

As curvas apresentadas nas Figuras 4.1 e 4.2 foram obtidas variando-se os parâmetros C e J do classificador SVM, onde C é o *trade-off* entre o erro de treinamento e a margem, e J é um fator de custo, utilizado para balancear os dados. Os resultados foram obtidos validando-se as seqüências até o SIT.

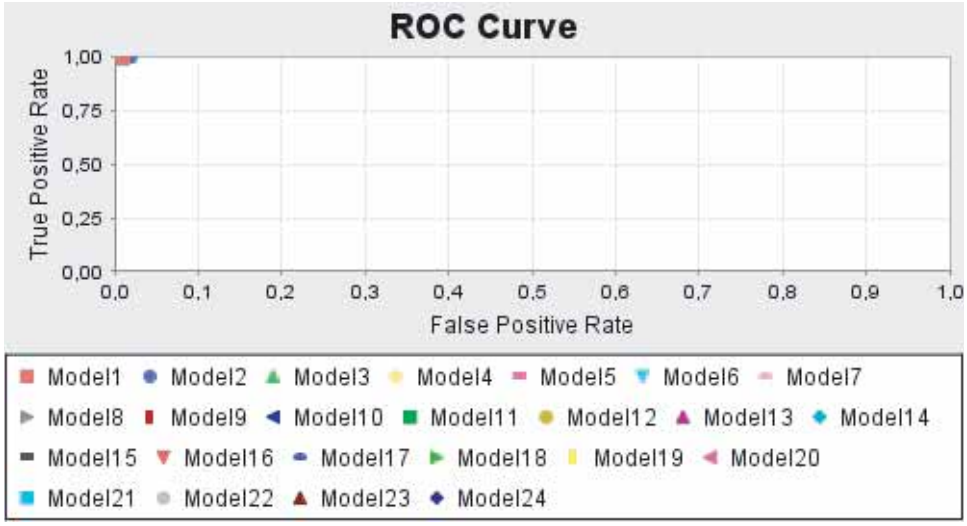


Figura 4.1: Curva ROC para *Mus musculus* com o conjunto balanceado de seqüências, variando-se o parâmetro C de 0 a 0,00001 com intervalo de 0,1.

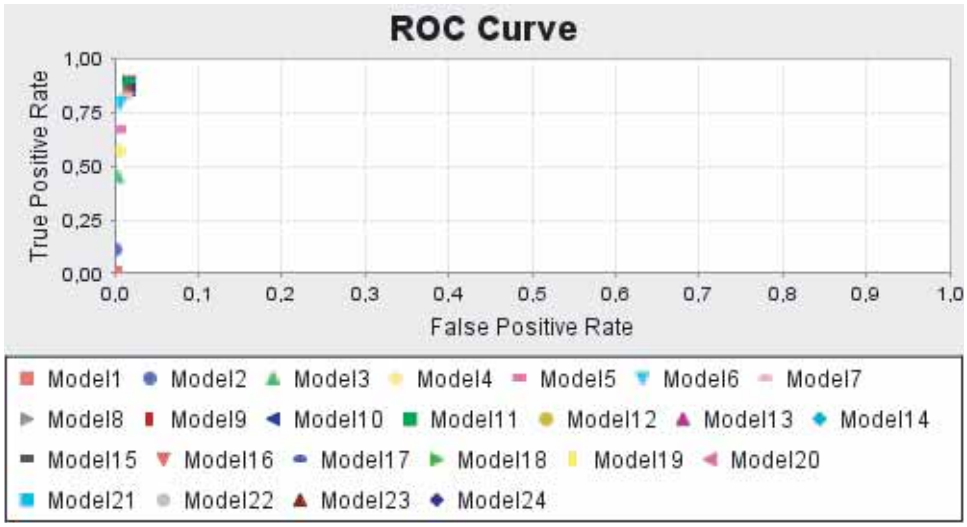


Figura 4.2: Curva ROC para *Mus musculus* com o conjunto desbalanceado de seqüências, variando-se o parâmetro J de 0 a 1 com intervalo de 0,1.

As Figuras 4.3 e 4.4 apresentam os mesmos resultados para o *Rattus norvegicus*.

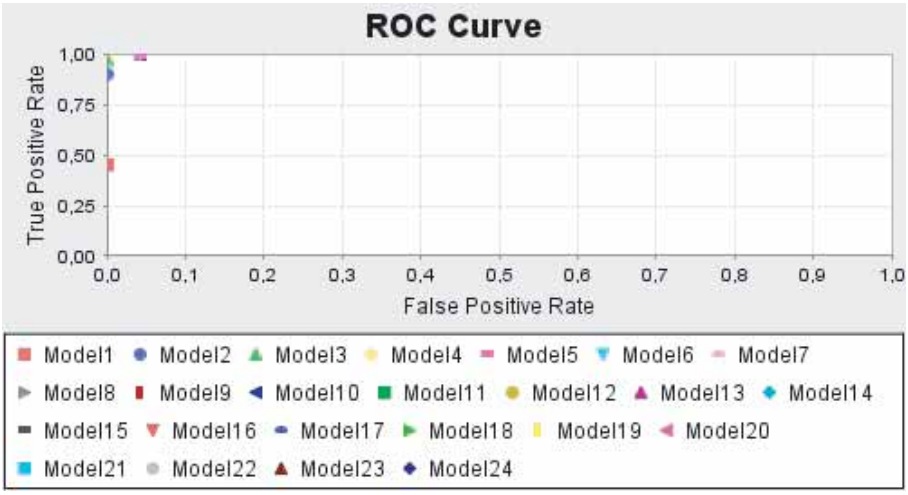


Figura 4.3: Curva ROC para *Rattus norvegicus* com o conjunto balanceado de seqüências, variando-se o parâmetro C de 0 a 0,00001 com intervalo de 0,1.

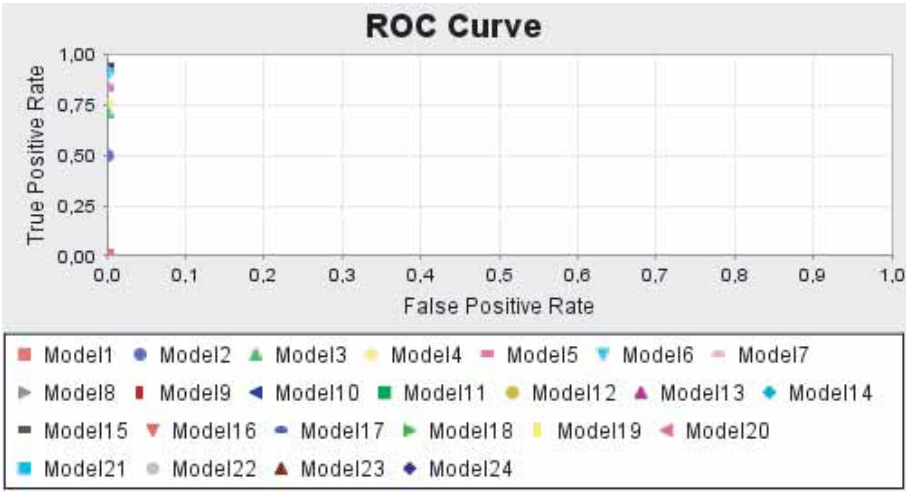
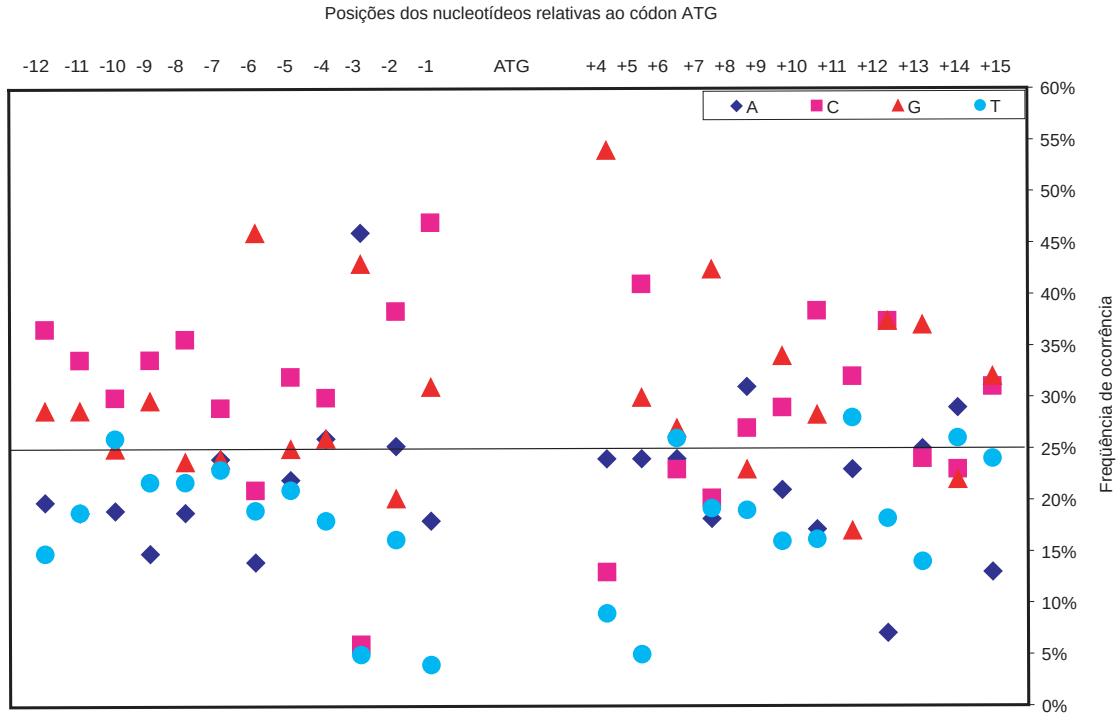


Figura 4.4: Curva ROC para *Rattus norvegicus* com o conjunto desbalanceado de seqüências, variando-se o parâmetro J de 0 a 1 com intervalo de 0,1.

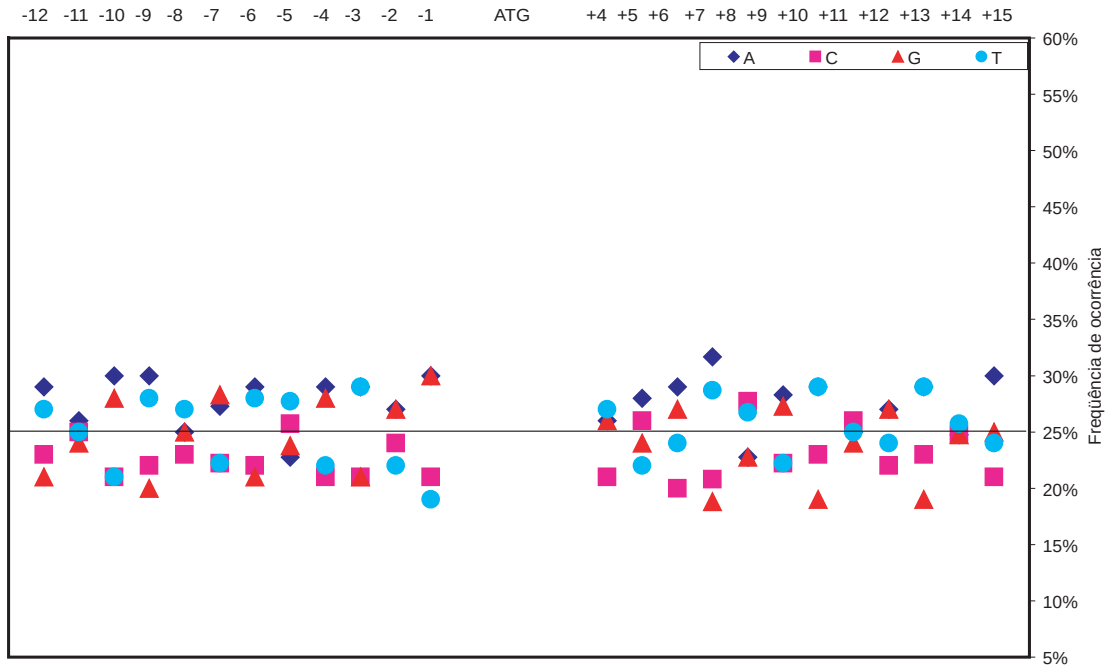
Esses resultados mostram que o balanceamento de dados realizado foi satisfatório e que a curva se aproxima do ponto ótimo (1,0), retornando pouquíssimos falsos positivos.

4.4 Análise dos falsos positivos

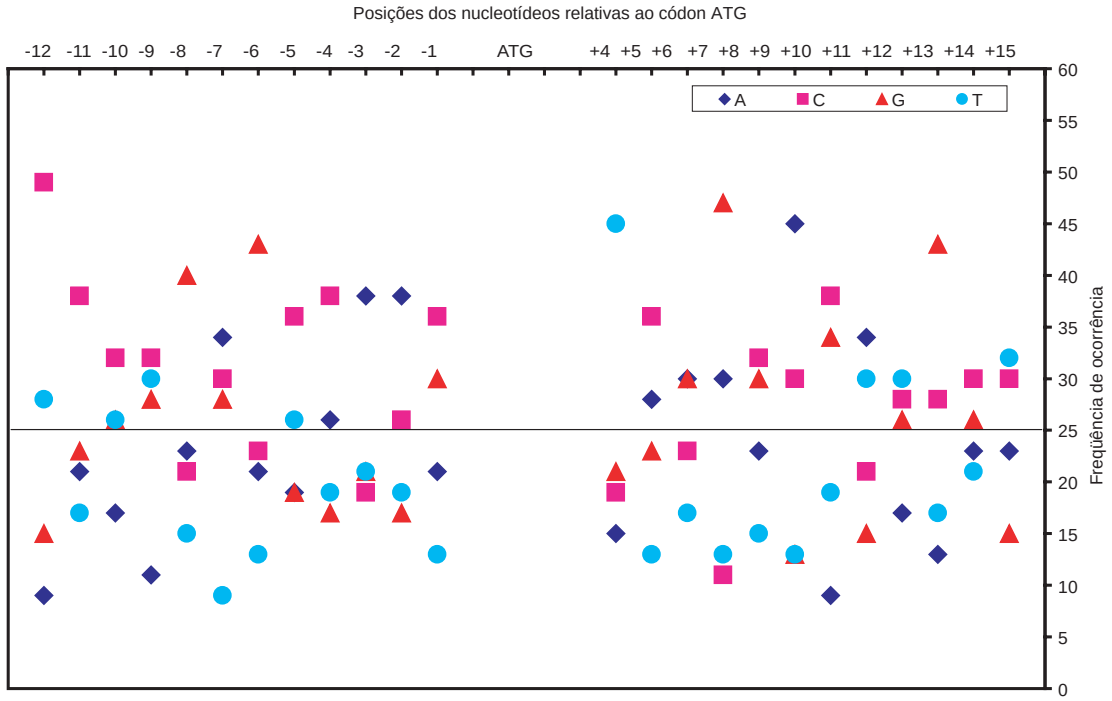
Uma análise da freqüência de cada base flanqueando o SIT sugere que falsos positivos (Figura 4.5 (c)), embora não reproduzindo completamente o padrão mostrado para as seqüências positivas (Figura 4.5 (a)), não possuem uma distribuição aleatória, que é peculiar às seqüências negativas (Figura 4.5 (b)). A Figura 4.5 (a) mostra um consenso de Kozak típico, no qual são obedecidas as regras para as posições -3 (purina) e +4 (guanina) e apresenta uma alta freqüência de G na posição -6. Claramente, como visto para os positivos verdadeiros, os falsos positivos não demonstram a distribuição aleatória observada nas seqüências negativas. Isso pode indicar que o classificador, na verdade, pode estar acertando a classificação e que essas seqüências se aproximam de verdadeiros positivos.



(a) Seqüências positivas do *Mus musculus reviewed*



(b) Seqüências negativas do *Mus musculus reviewed*



(c) Falsos positivos do *Mus musculus reviewed*

Figura 4.5: Freqüência de bases das seqüências positivas (a), negativas (b) e falsos positivos (c), respectivamente, do *Mus musculus reviewed*.

4.5 Parâmetros utilizados e recursos computacionais

Para a classificação usando o SVM, o tipo de função de *kernel* que produziu os melhores resultados para as bases *RefSeq* foi o polinomial de grau 4, apesar de terem sido testados vários outros tipos de função (não mostrado). Para a base de Petersen e Nielsen a função RBF (*Radial Basis Function*) com $\sigma = 0.002$ retornou os melhores resultados.

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

1. Athlon, 2.08 GHZ com 2,5GB de RAM
2. Dell Inspiron 6400, 1.83GHz Core Duo, 1GB DDR
3. Pentium IV 3.0 GHz, 1 GB de RAM
4. Pentium IV 3.2GHz, 1 GB de RAM
5. Pentium IV 3.0 GHz, 2 GB de RAM
6. Pentium IV Core 2 Duo 6600 2.4GHz, 3 GB de RAM
7. Quatro Pentiuns 3GHz, 60GB de HD e 1 GB de RAM

Considerando-se a abordagem indutiva, o tempo de processamento variou de poucos minutos (37 min) para o *Mus musculus Reviewed* até 5,5 semanas para a *Droshopila melanogaster Reviewed*. No entanto, o tempo de processamento, considerando-se a abordagem transdutiva, é, em média, 8 vezes maior.

4.6 Conclusões do capítulo

Foram apresentados, nesse capítulo, os resultados obtidos segundo a metodologia proposta, além de uma comparação desses resultados com os obtidos na literatura.

O próximo capítulo destina-se à apresentação das conclusões e propostas de trabalhos futuros.

CAPÍTULO 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, apresentamos 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

Em vista dos argumentos apresentados em todo o desenvolvimento deste trabalho, é possível concluir que o tema de identificação do SIT exige um grande investimento em pesquisa. Trabalhar com diferentes organismos (metazoários superiores e eucariotos mais primitivos), fazer uma análise detalhada das seqüências, melhorar o desempenho dos classificadores obtidos até então, fazer uma comparação entre diversos organismos, com base no SIT e representar o conhecimento adquirido por

estes classificadores, de forma que seja fácil de ser interpretado, são algumas das necessidades para quem trabalha com a biologia molecular computacional e a bioinformática.

Foi apresentada nesta tese uma metodologia que vem ao encontro de algumas dessas necessidades. Vimos que todos os critérios considerados no Capítulo 3 foram bastante significativos para se obter um bom desempenho.

Neste trabalho, foi utilizado o classificador SVM que mostrou ser bastante promissor para o problema em questão. Esses resultados foram obtidos utilizando-se a seguinte metodologia:

- Primeiramente, destaca-se a qualidade das bases analisadas. Todas elas foram extraídas do *RefSeq* que são seqüências de referência; nota-se, no entanto, que as seqüências com maior curadoria manual (*Reviewed* e *Validated*), tendem a oferecer os melhores resultados. Sob nossas condições de análise, ficou evidente que a utilização de "N" em experimentos com a base de Petersen e Nielsen causa melhora artefatual dos resultados, o que pode ter comprometido alguns trabalhos anteriores.

- Escolha de uma metodologia que pudesse ser capaz de fazer uma boa previsão com bom desempenho e custo computacional reduzido. Os métodos apresentados neste trabalho foram de fundamental importância para o desempenho obtido. Destaca-se para a abordagem transdutiva que conseguiu melhorar os resultados de forma bastante significativa. Ainda neste sentido, vale a pena ressaltar que a aplicação da metodologia proposta favorece o uso do classificador SVM com funções simples de *kernel*, ou seja, sem a necessidade adicional de criar funções especiais, como é o caso dos trabalhos de Zien *et al.* (2000) e Haifeng e Tao (2004), ou de Hatzigeorgiou (2002) que utilizou um método integrado combinando duas RNs. Desta forma, acredita-se que os resultados apresentados são de aplicabilidade geral para várias técnicas de aprendizado de máquina e são vantajosos uma vez que os classificadores existentes, e disponíveis, podem ser facilmente utilizados.

Quanto à análise das seqüências, mostramos que as classificadas como positivas estão de acordo o consenso de Kozak (viés da Guanina (G) na posição +4 e uma purina, preferencialmente a Adenina (A), na posição -3). Quanto às negativas, mostramos que elas possuem um comportamento aleatório. Além disto, vimos que existe uma indicação de que os falsos positivos podem ser, na verdade, um SIT alternativo ou ancestral.

5.2 Sugestões de continuidade de trabalho

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

- Investir em técnicas de processamento distribuído para que todos os testes possam ser realizados em tempo hábil, principalmente para o aprendizado transdutivo. A base do *Homo sapiens* (com o aprendizado transdutivo), por exemplo, está rodando em um servidor deste o dia 28 de maio e até hoje, dia 15 de julho, e os resultados ainda não foram retornados;

- Extrair regras do classificador SVM para que o conhecimento adquirido seja de fácil entendimento. Uma grande contribuição deste trabalho é 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 seqüências 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;

- Implementar as funções aqui utilizadas em um aplicativo independente e, futuramente, em um servidor web;

- Analisar o desempenho de treinamento e validação utilizando organismos distintos em cada fase. Isso poderá nos trazer informações sobre a conservação evolutiva da sinalização do SIT;

- Expandir os testes para eucariotos mais primitivos, a partir daí, fazer uma comparação entre esses e metazoários superiores.

Com isso, estaremos dando uma grande contribuição para a área de previsão de SIT, que ainda é pouco estudada.

APÊNDICE A

Apêndices

A codificação utilizada nesse trabalho foi inspirada em uma análise das seqüências positivas e negativas feita a partir das moléculas extraídas dos organismos. Inicialmente um dos propósitos desse trabalho foi exatamente analisar essas seqüências.

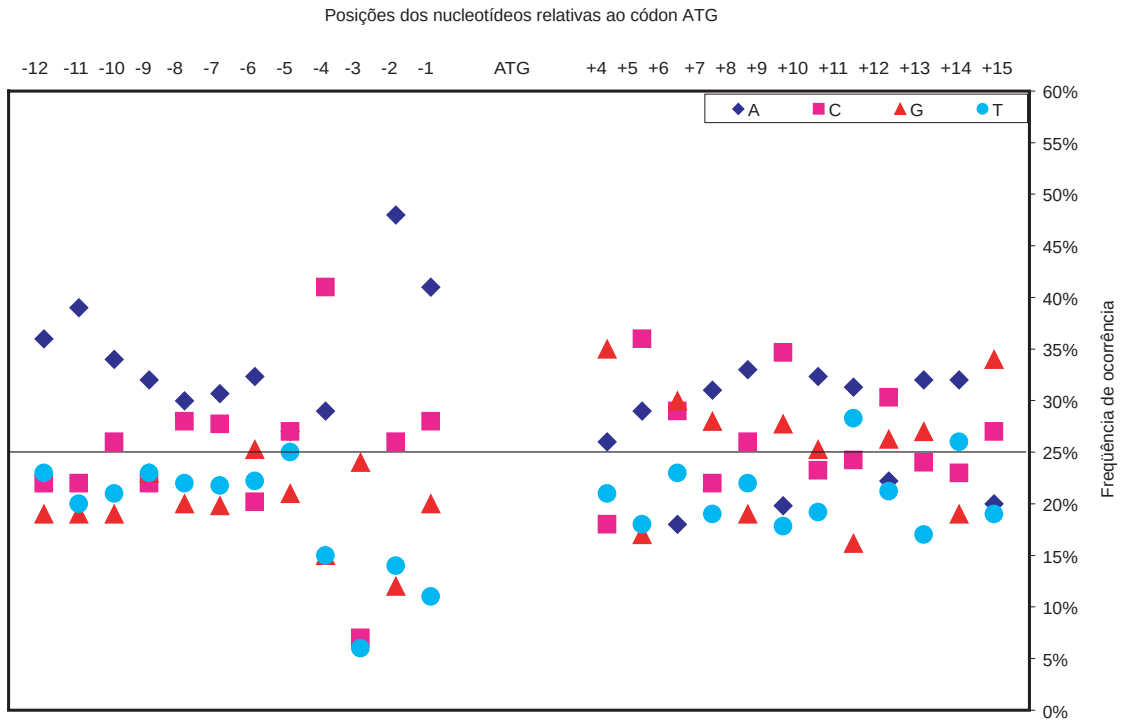
É plausível supor que o sinal utilizado para reconhecimento do SIT possa depender de (i) um padrão reconhecido para início da tradução e um padrão randômico em torno do negativo ou (ii) um padrão positivo somado a algum sinal específico em torno do ATG negativo. O consenso sugerido por Kozak (1987) desde o início das investigações sobre esse tema sugere fortemente que o mecanismo de reconhecimento não se deve a sinalização exclusiva em torno do padrão negativo e, portanto, essa possibilidade é descartada. Nesse sentido, dois tipos de análise foram realizadas:

- Primeiramente, foi feita uma análise, por posição, da frequência de nucleotídeos de ambas as seqüências;
- Foi realizada, também, uma análise da freqüência dos trinucleotídeos.

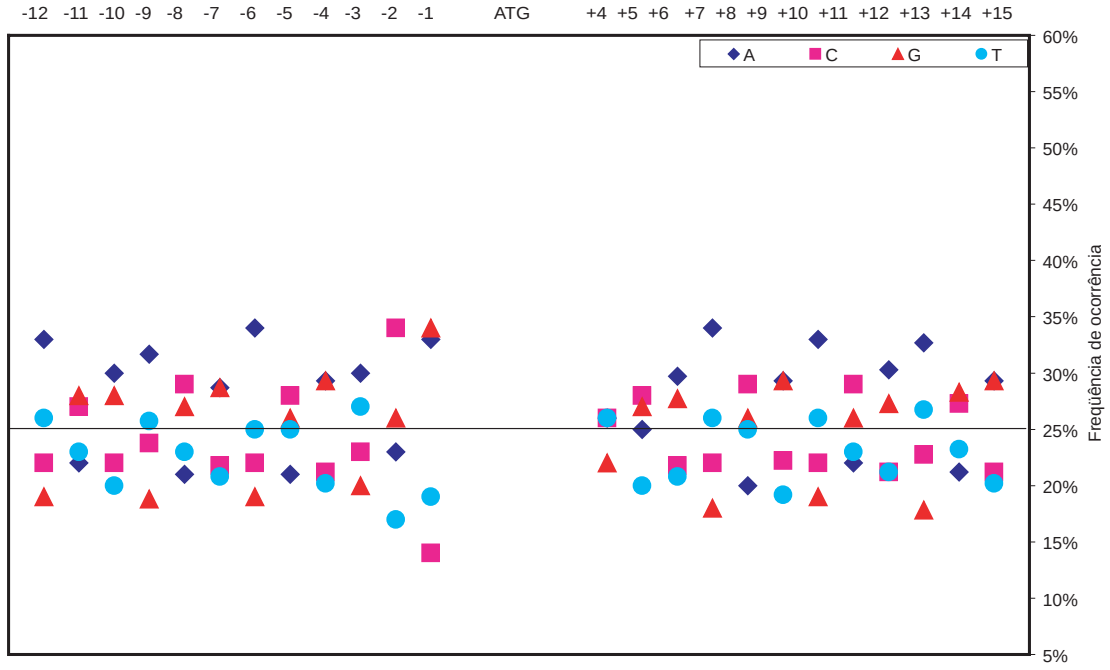
As seções seguintes apresentam os resultados dessas análises.

A.1 Frequência de bases das seqüências

Com o objetivo de verificar a freqüência de bases existentes nas seqüências positivas e negativas, foram geradas as Figuras A.1, A.2, A.3 e A.4.

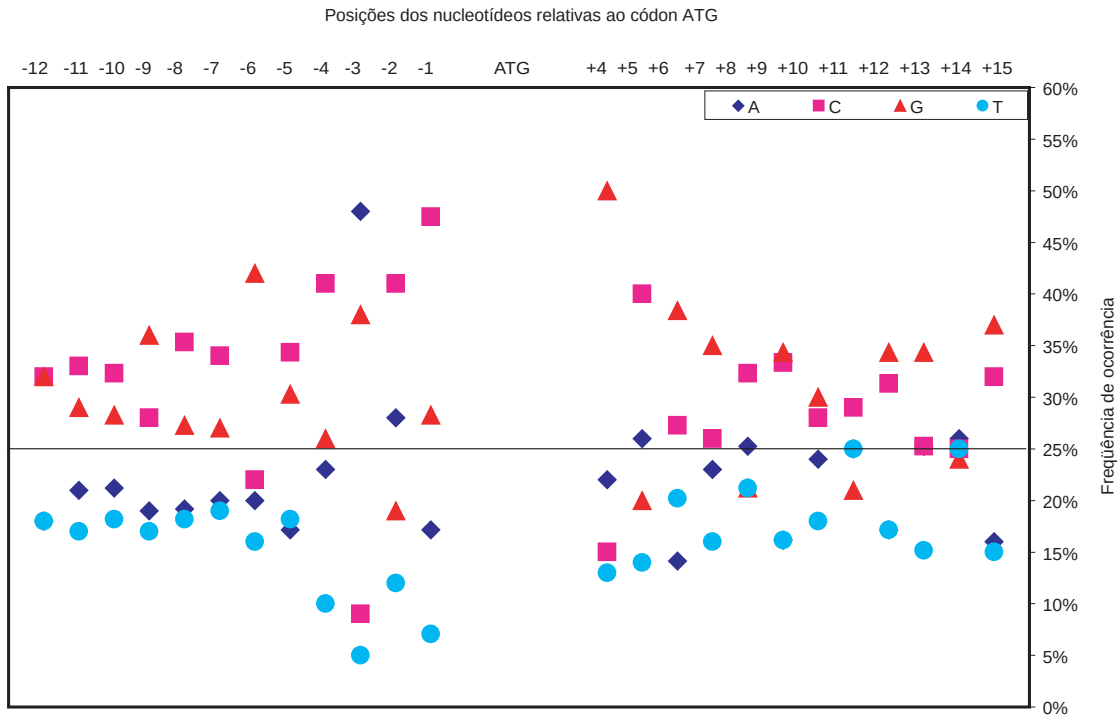


(a) Seqüências positivas da *Drosophila melanogaster reviewed*

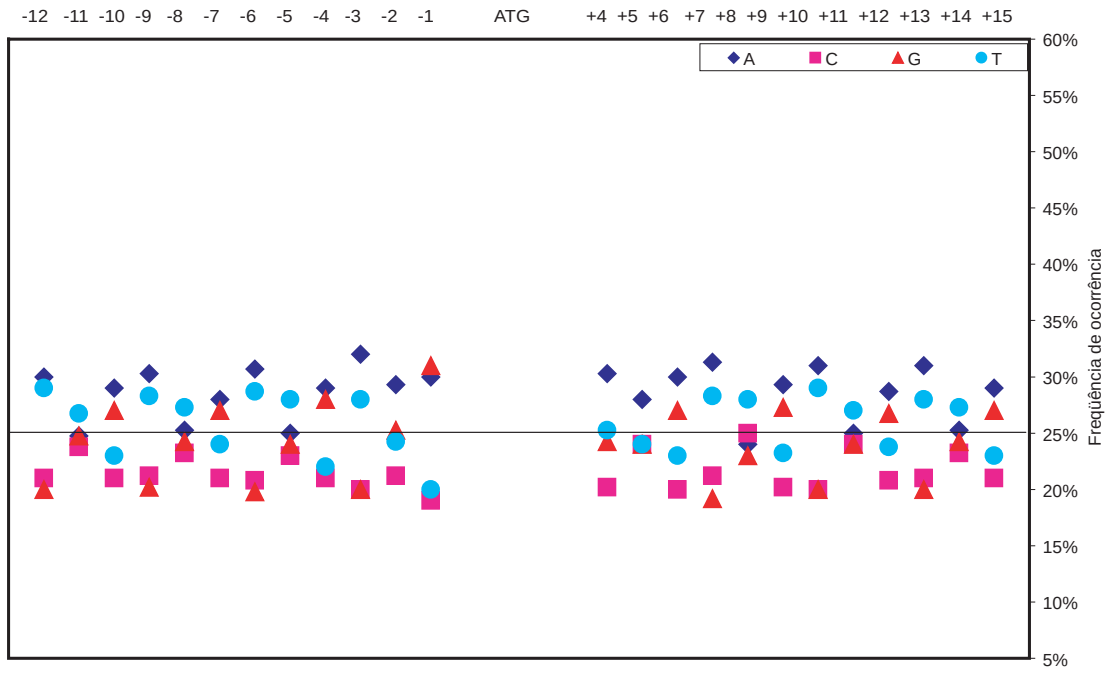


(b) Seqüências negativas da *Drosophila melanogaster reviewed*

Figura A.1: Freqüência de bases das seqüências positivas e negativas, respectivamente, da *Drosophila melanogaster reviewed*.

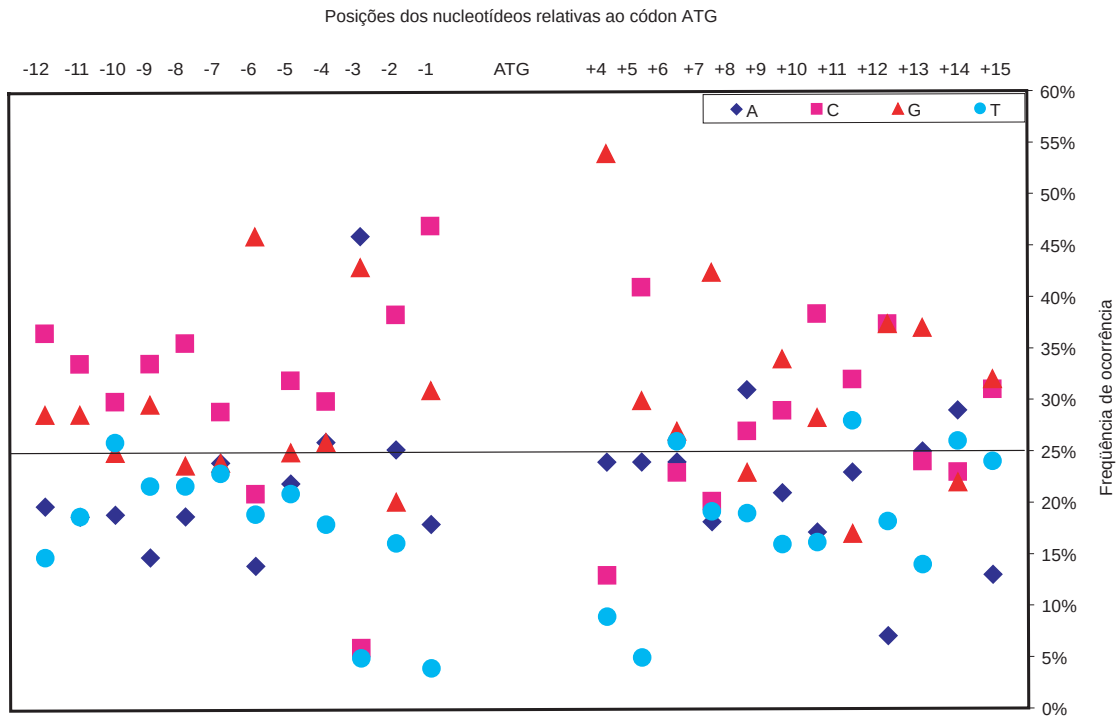


(a) Seqüências positivas do *Homo sapiens reviewed*

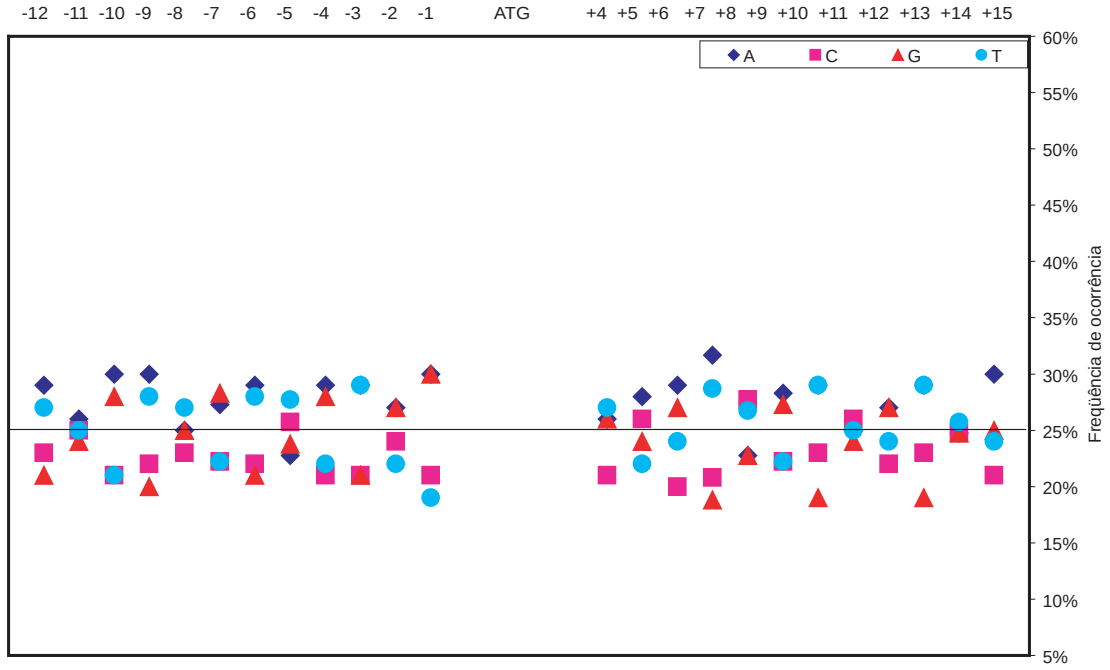


(b) Seqüências negativas do *Homo sapiens reviewed*

Figura A.2: Freqüência de bases das seqüências positivas e negativas, respectivamente, do *Homo sapiens reviewed*.

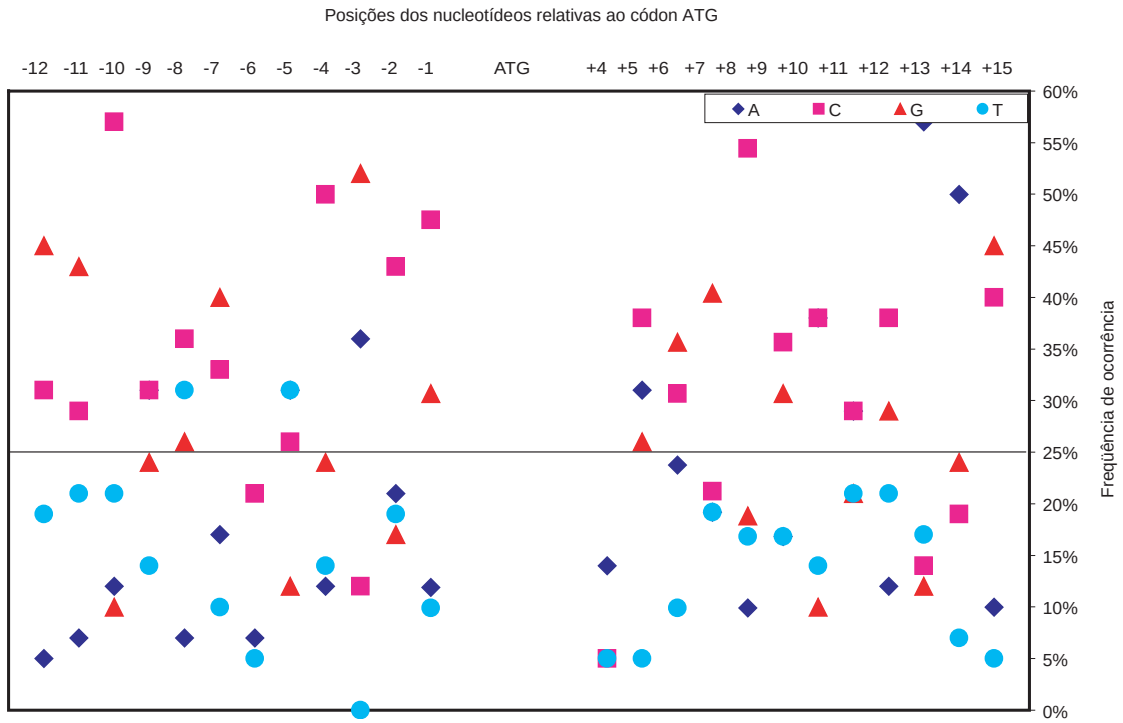


(a) Seqüências positivas do *Mus musculus reviewed*

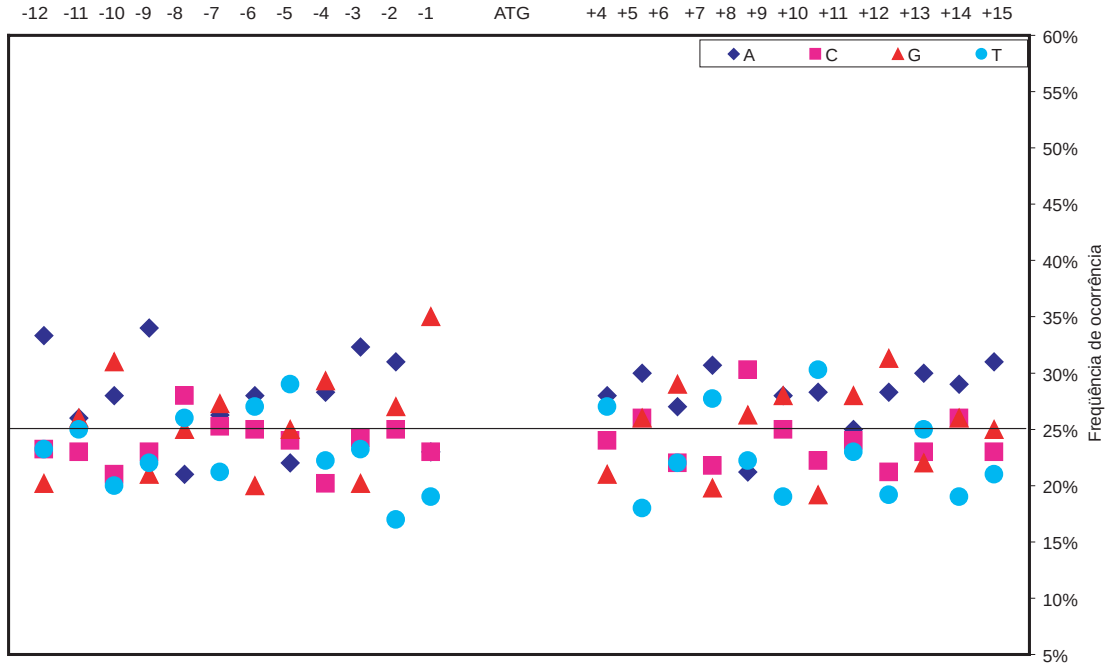


(b) Seqüências negativas do *Mus musculus reviewed*

Figura A.3: Freqüência de bases das seqüências positivas e negativas, respectivamente, do *Mus musculus reviewed*.



(a) Seqüências positivas do *Rattus norvegicus reviewed*



(b) Seqüências negativas do *Rattus norvegicus reviewed*

Figura A.4: Freqüência de bases das seqüências positivas e negativas, respectivamente, do *Rattus norvegicus reviewed*.

De acordo com essas figuras, pode-se perceber que as seqüências positivas estão de acordo com o padrão do consenso de Kozak, mostrando que a presença da purina (Adenina ou Guanina) na posição -3 é muito importante para a identificação correta do SIT. Uma outra posição importante é a +4, onde normalmente aparece uma Guanina.

Quanto às seqüências negativas, percebe-se pelas figuras que não existe um padrão, à exceção de uma leve tendência à presença da base G imediatamente anterior ao ATG (posição -1) ser um pouco superior à freqüência de G no mRNA, em quase todos os casos. Ou seja, em geral a freqüência de cada base fica perto da ocorrência dela no mRNA, que é sempre em torno de 25% (como pode ser observado nas Figuras A.1(b), A.2(b), A.3(b) e A.4(b). Outra observação interessante é que, tanto nas seqüências negativas como positivas, a presença da Timina é aparentemente menor que no mRNA como um todo, sugerindo que sua ausência nos padrões positivos não é determinante. No entanto, se compararmos a quantidade de Timina nestas seqüências (positivas e negativas), podemos perceber que as negativas possuem uma quantidade ligeiramente maior desta base.

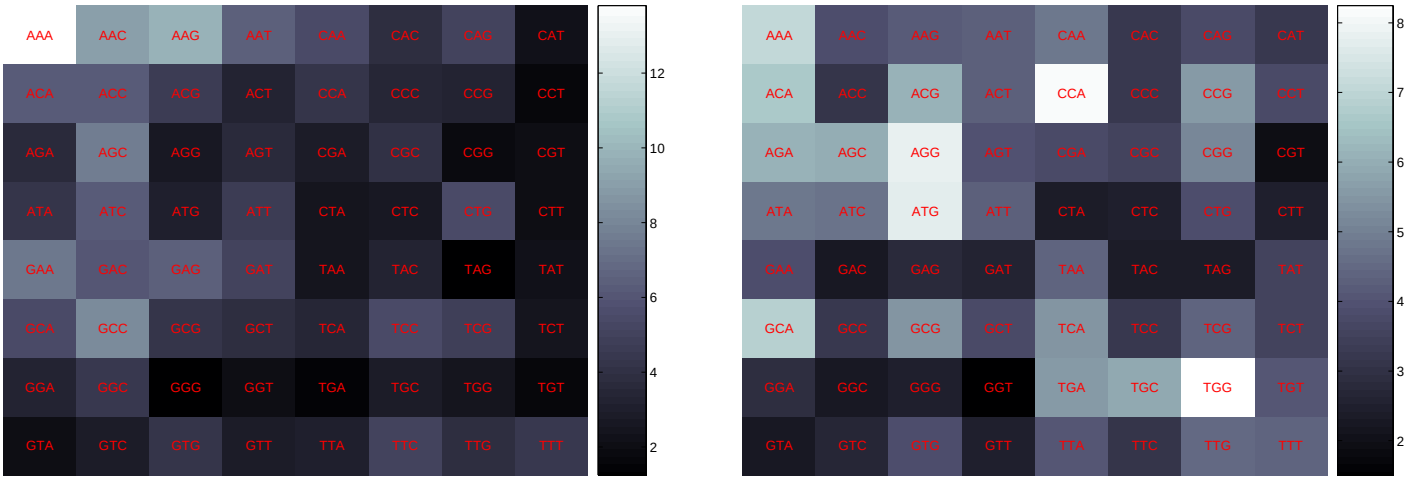
Um outro fato interessante é que a Adenina (A) não aparece na mesma proporção que a Timina (T); o mesmo acontece com a Citosina e Guanina. Isso se deve ao fato de estarmos trabalhando com fitas simples de cDNA.

A.2 Freqüência de trinucleotídeos

Ainda com o objetivo de analisar as seqüências positivas e negativas, foi realizada, também, uma análise da freqüência de trinucleotídeos a partir das 24 posições.

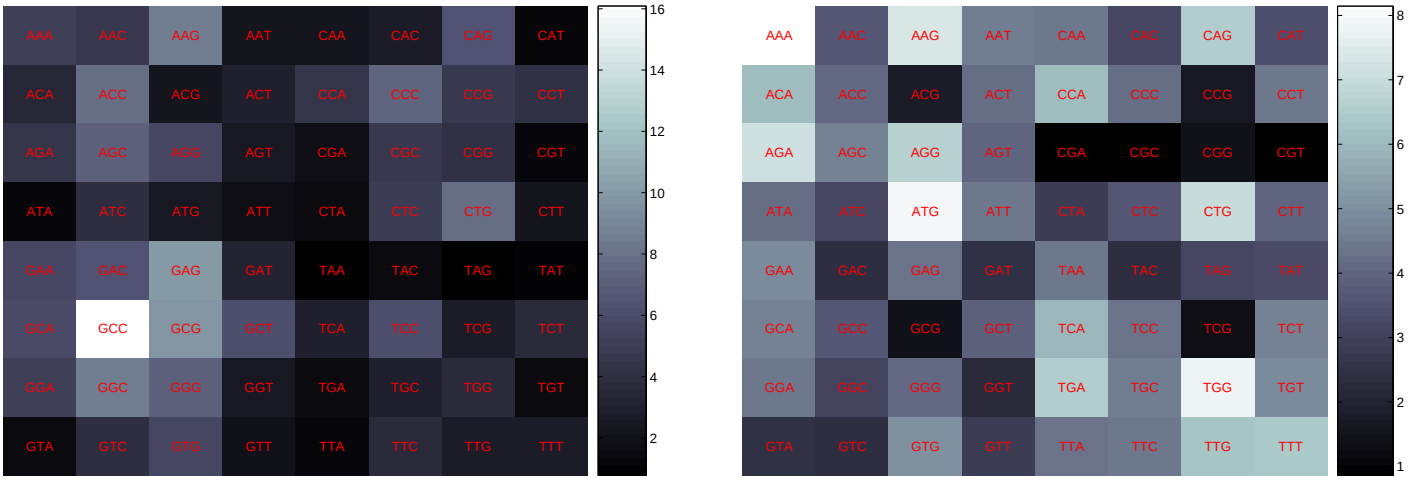
O programa percorre as seqüências de três em três bases; ou seja, o deslizamento na janela é feito de três em três posições. Desta forma, composições de todas as seqüências (*reviewed*, *validated*, *inferred*, *provisional*, *predicted*, *model*) foram calculadas e as freqüências obtidas para cada organismo. As Figuras A.5, A.6, A.7 e A.8 apresentam essa distribuição de freqüência para as seqüências *reviewed* de *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus* e *Rattus norvegicus*, respec-

tivamente. Não foi plotada a distribuição de freqüência para o *Danio rerio* porque este organismo só possui três seqüências *reviewed*. Os resultados dos outros níveis de inspeção não estão apresentados porque seguem o mesmo padrão.



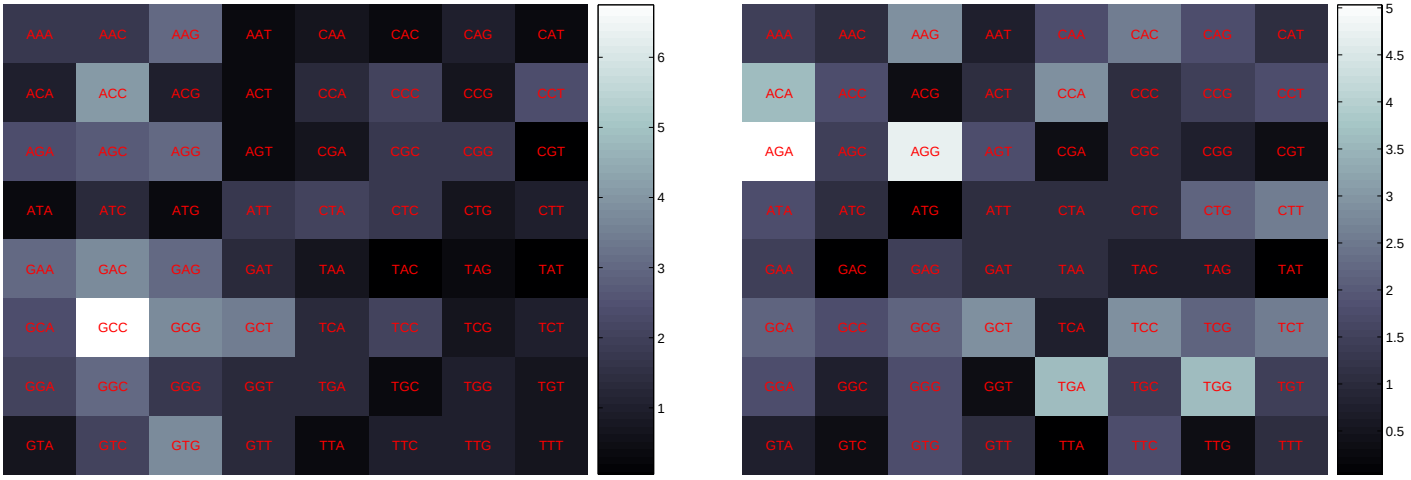
(a) Seqüências positivas *reviewed* - Dm (b) Seqüências negativas *reviewed* - Dm

Figura A.5: Freqüência de trinucleotídeos das seqüências positivas e negativas *reviewed* de *Drosophila melanogaster*, respectivamente. A cor varia do mais claro, indicando uma alta freqüência, para o mais escuro, indicando baixa freqüência.



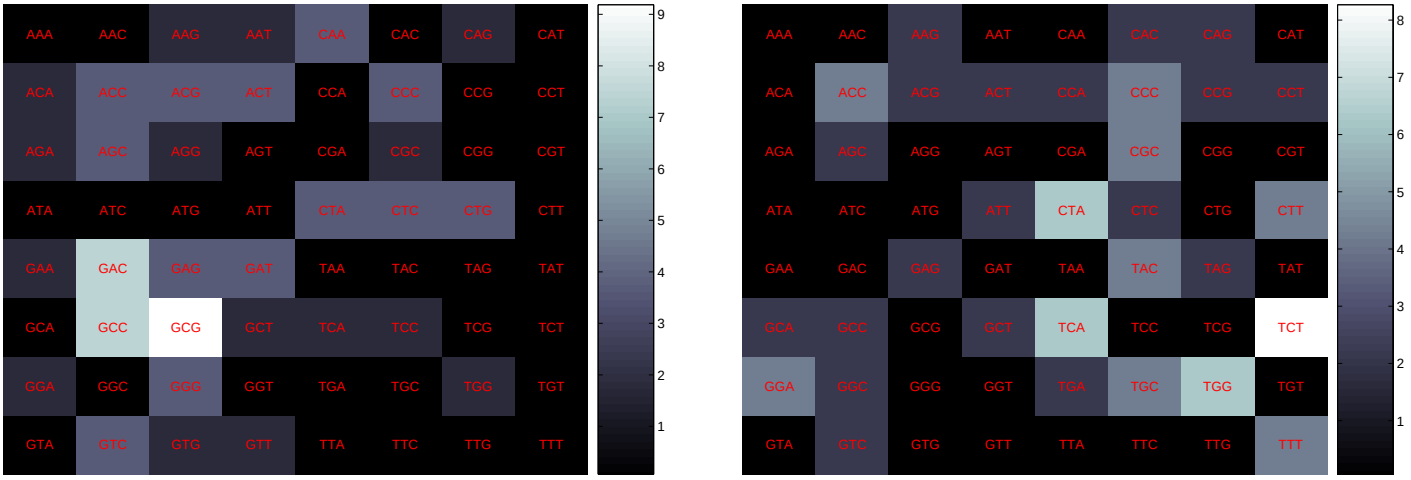
(a) Seqüências positivas *reviewed* - Hs (b) Seqüências negativas *reviewed* - Hs

Figura A.6: Freqüência de trinucleotídeos das seqüências positivas e negativas *reviewed* de *Homo sapiens*, respectivamente. A cor varia do mais claro, indicando uma alta freqüência, para o mais escuro, indicando baixa freqüência.



(a) Seqüências positivas *reviewed* - Mm (b) Seqüências negativas *reviewed* - Mm

Figura A.7: Freqüência de trinucleotídeos das seqüências positivas e negativas *reviewed* de *Mus musculus*, respectivamente. A cor varia do mais claro, indicando uma alta freqüência, para o mais escuro, indicando baixa freqüência.



(a) Seqüências positivas *reviewed* - Rn (b) Seqüências negativas *reviewed* - Rn

Figura A.8: Freqüência de trinucleotídeos das seqüências positivas e negativas *reviewed* de *Rattus Norvegicus*, respectivamente. A cor varia do mais claro, indicando uma alta freqüência, para o mais escuro, indicando baixa freqüência.

Pelas figuras, é possível perceber que a freqüência de trinucleotídeos das seqüências positivas normalmente é diferente das seqüências negativas para todos os organismos estudados. De maneira geral, as maiores freqüências nos positivos se concentram no lado esquerdo e para os negativos, no lado direito (onde tem-se mais Timina). Estes resultados estão de acordo com os encontrados no estudo sobre a

freqüência de bases das seqüências, onde se percebia uma presença maior da Timina nas seqüências negativas, em relação às seqüências positivas.

Outro ponto importante é que os testes apontam, nos positivos, uma maior freqüência dos trinucleotídeos GCC e GCG (próximo de 9%) para os organismos *Mus musculus*, *Rattus norvegicus* e *Homo sapiens*. Para a *Drosophila melanogaster*, entretanto, o trinucleotídeo AAA aparece com maior freqüência. Observando-se estas freqüências percebe-se que estes organismo têm uma freqüência bem diferente dos outros três. Nos negativos, entretanto, os trinucleotídeos TGG e TCT são os mais freqüentes.

Ainda neste sentido, foram analisadas também as posições onde estes trinucleotídeos são mais freqüentes. Ou seja, uma vez sabendo-se, por exemplo, que o GCC é o mais freqüente no *Homo sapiens* queremos descobrir também qual é a posição onde se tem mais GCC. Será se esse trinucleotídeo fica no início da seqüência de 24 ou mais próximo do ATG?

Para o caso do *Homo sapiens*, considerando-se as 10657 seqüências revisadas positivas, encontramos 5056 ocorrências do trinucleotídeo GCC. Dessas 5056 ocorrências, 7,97% estão localizadas nas posições de -12 a -10, 12,06% nas posições de -9 a -7, 15,63% nas posições de -6 a -4, 28,16% nas posições de -3 a -1 (região *upstream* do ATG), 13,67% nas posições de +4 a +6, 8,45% nas posições de +7 a +9, 6,92% nas posições de +10 a +12 e 7,14% nas posições de +13 a +15 (região *downstream* do ATG). A Tabela A.1 apresenta a freqüência dos trinucleotídeos mais freqüentes nestas três posições para todos os organismos analisados. Apenas as freqüências referentes às seqüências positivas serão apresentadas, visto que a freqüência obtida, neste caso, é bastante significativa. Além disso, os mesmos testes foram realizados com as seqüências negativas e observou-se uma freqüência aleatória, em relação à posição, entre os trinucleotídeos mais freqüentes (dados não mostrados).

Tabela A.1: Freqüência, nas posições de -12 a -10, -9 a -7, -6 a -4, -3 a -1, +4 a +6, +7 a +9, +10 a +12, +13 a +15 de trinucleotídeos mais freqüentes para os quatro organismos *reviewed*.

Organismos	TamTotal	Ocorrência	Tri-nucleotídeo	-12 a -10	-9 a -7	-6 a -4	-3 a -1	+4 a +6	+7 a +9	+10 a +12	+13 a +15
		3834	GCC	5,92%	9,34%	15,44%	17,21%	**20,50%**	13,25%	9,91%	8,43%
D.	15893	6624	AAA	15,79%	8,80%	8,04%	**39,43%**	5,62%	7,21%	8,73%	6,37%
melanogaster		1980	GCG	7,68%	8,79%	9,79%	13,28%	**24,69%**	13,89%	8,94%	12,93%
		5056	GCC	7,97%	12,06%	15,63%	**28,16%**	13,67%	8,45%	6,92%	7,14%
H. sapiens	10657	1723	AAA	7,89%	9,75%	8,88%	**27,33%**	7,60%	11,96%	13,29%	13,29%
		2949	GCG	8,48%	11,59%	9,46%	11,19%	**26,86%**	12,44%	9,87%	10,10%
		68	GCC	8,82%	17,65%	16,18%	**25,00%**	11,75%	8,82%	11,76%	0,00%
M. musculus	182	15	AAA	0,00%	0,00%	6,67%	**26,67%**	20,00%	13,33%	6,67%	**26,67%**
		29	GCG	0,00%	10,34%	13,79%	**24,14%**	17,24%	13,79%	13,79%	6,89%
		25	GCC	20,00%	8,00%	20,00%	**28,00%**	4,00%	20,00%	0,00%	0,00%
R. norvegicus	42	1	AAA	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	**100,00%**
		11	GCG	0,00%	9,09%	27,27%	9,09%	**45,45%**	9,09%	0,00%	0,00%

TamTotal é o número total de seqüências positivas *reviewed*.

Ocorrência é a quantidade de vezes que o tri-nucleotídeo aparece. As maiores porcentagens estão em negrito.

Como pode ser observado, para todos os organismos, os trinucleotídeos GCC, AAA e CCG são mais freqüentes nas posições mais próxima do ATG. Esses dados, somados a análise independente de cada posição (Figuras A.1 a A.4) ilustram a importância da utilização do classificador SVM que pondera as 24 posições simultaneamente, além da codificação por trinca e um tamanho menor de janela utilizados.

Referências

Agarwal, P.; Bafna, V. (1998). The ribosome scanning model for translation initiation for gene prediction and full-length cdna detection. *Proc. 5th International Conference on Intelligent Systems for Molecular Biology*, pages 2–7.

Almeida, M. B. (2002). Svms training using re-sampling based on error and a priori strategies for sample selection. *Dissertação de Mestrado, Universidade Federal de Minas Gerais*.

Batista, G. E. A. P. A. (2003). Pré-processamento de dados em aprendizado de máquina supervisionado. *Department of Computer Science and Statistics. University of São Paulo*. PhD. Thesis.

Batista, G. E. A. P. A., Prati, R. C., e Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6:20–29.

Benson, D., Boguski, M., Lipman, D., e Ostell, J. (1997). Genbank. *Nucleic Acids Research*, 25:1–6.

Boser, B., Guyon, I., e Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *In Computational Learing Theory*, pages 144–152.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 6(30):1145–1159.

Braga, A., Carvalho, A., e Ludermir, T. (2000). *Redes neurais artificiais: teoria e aplicações*. Livros Técnicos e Científicos.

Breiman, L. (1997). Bagging predictions. *Machine Learning.*, 24:123–140.

Brunak, S., Engelbrecht, J., e Knudsen, S. (1991). Prediction of human mrna donor and acceptor sites from the dna sequence. *J. Mol. Biol*, (220):49–65.

Burbidge, R. e Buxton, B. (2001). An introduction to support vector machines for data mining. *Operational Research Society: Operational Research Society*. In M. Sheppee (Ed.), Keynote Papers, Young OR12, University of Nottingham.

Caria, M. (2001). Measurement analysis: An introduction to the statistical analysis of laboratory data in physics, chemistry, and the life sciences. Imperial College Press.

Carvalho, B., Almeida, M., e Braga, A. (2002). Support vector machines - um estudo sobre técnicas de treinamento. *Technical Report Monografia interna no.3 Universidade Federal de Minas Gerais, Belo Horizonte, MG*.

Chawla, N., Moore, T. E., Hall, L. O., Bowyer, K., Kegelmeyer, W., e Springer, C. (2003). Distributed learning with bagging-like performance. *Pattern Recognition Letters*, 24:455–471.

Chawla, N. V., Japkowicz, N., e Kotcz, A. (1997). Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, pages 1–6.

Cortes, C. e Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 3(20):273–297.

Fawcett, T. e Provost, F. J. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316.

Fayyad, U. e Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *In Proceedings of 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029.

Gibas, C.; Jambeck, P. (2001). *Desenvolvendo bioinformática: ferramentas de software para aplicações em biologia*. Campus, Rio de Janeiro.

Goutte, C., Déjean, H., Gaussier, E., Cancedda, N., e Renders, J.-M. (2002). Combining labelled and unlabelled data: a case study on fisher kernels and transductive inference for biological entity recognition. *In: Dan Roth and Antal van den Bosch (eds.), Proceedings of CoNLL-2002*, pages 1–7. Taipei, Taiwan.

Haifeng, L. e Tao, J. (2004). A class of edit kernels for svms to predict translation initiation sites in eukaryotic mrnas. *ACM Press*, pages 262–271. In Proceedings of the Eighth Annual international Conference on Resaerch in Computational Molecular Biology (San Diego, California, USA, March 27-31, 2004). RECOMB '04. ACM Press, New York, NY, 262-271. DOI= `http://doi.acm.org/10.1145/974614.974649`.

Hall, M. A. (1998). Correlation-based feature selection machine learning. *PhD thesis, Department of Computer Science*. University of Waikato, New Zealand.

Hatzigeorgiou, A. (2002). Translation initiation start prediction in human cdnas with high accuracy. *Bioinformatics*, 18(2):343–350.

Haykin, S. (1999). *Neural networks: a comprehensive foundation*. Prentice Hall, 2 edition.

Huiqing, L., Hao, H., Jinyan, L., e Limsoon, W. (2004). Using amino acid patterns to accurately predict translation initiation sites. *In Silico Biology*, 4. 0022.

Joachims, T. (1999). Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press*. `http://www-ai.cs.uni-dortmund.de/DOKUMENTE/joachims\_99a.pdf`.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *In Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI)*.

Kozak, M. (1984). Compilation and analysis of sequences upstream from the translational start site in eukaryotic mrnas. *Nucl. Acids Res*, 12:857–872.

Kozak, M. (1986). Point mutations define a sequence flanking the aug initiator

codon that modulates translation by eukaryotic ribosomes. *Cell*, 44:283–292.

Kozak, M. (1987). An analysis of 5'-noncoding sequences from 699 vertebrate messenger rnas. *Nucleic. Acids Research*, 15:8125–8148.

Kozak, M. (1989). The scanning model for translation: an update. *J. Cell. Biol.*, 108:229–241.

Kozak, M. (1999). Initiation of translation in prokaryotes and eukaryotes. *Gene*, 234:187–208.

Kozak, M. e Shatkin, A. J. (1978). Migration of 40 s ribosomal subunits on messenger rna in the presence of edeine. *Journal of Biological Chemistry*, 18(253):6568–6577.

Langley, P., Iba, W., e Thompson, K. (1992). An analysis of bayesian classifier. *In Proceedings of 10th National Conference on Artificial Intelligence*, pages 223–228. AAAI Press.

Li, G., Leong, T., e Zhang, L. (2005). Translation initiation sites prediction with mixture gaussian models in human cdna sequences. *Knowledge and Data Engineering, IEEE Transactions*, 17(8):1152–1160.

Li, J., Liu, H., Wong, L., e Yap, R. (2004). *Techniques for Recognition of Translation Initiation Sites*. The Practical Bioinformatician, edited by Limsoon Wong, Chapter 4.

Li, J., Ng, S. K., e Wong, L. (2003). Bioinformatics adventures in database research. *In LNCS 2572: Proceedings of 9th International Conference on Database Theory*, pages 31–46.

Liu, H. e Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. *In Proceedings of IEEE 7th International Conference on Tools with Artificial Intelligence*, pages 338–391.

Liu, H. e Wong, L. (2003). Data mining tools for biological sequences. *Journal of Bioinformatics and Computational Biology*, (1):139–167.

Nishikaw, T., Ota, T., e Isogai, T. (2000). Prediction whether a human cdna sequence contains initiation codon by combining statistical information and similarity with protein sequences. *Bioinformatics*, (16):960–967.

Nitesh, V. C., Bowyer, K. W., Hall, L. O., e Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence and Research*, 16:321–357. Disponível em `citeseer.ist.psu.edu/chawla02smote.html`.

Pain, V. (1996). Initiation of proteins synthesis in eukaryotes cells. *Eur. J. Biochem*, 236:747–771.

Pedersen, A. e Nielsen, H. (1997). Neural network prediction of translation initiation sites in eukaryotes: perspectives for est and genome analysis. *Proc. 5th International Conference on Intelligent Systems for Molecular Biology*, pages 226–233.

Pednault, E. P., Rosen, B. K., e Apte, C. (2000). Handling imbalanced data sets in insurance risk modeling. *Technical Report RC-21731, IBM Research Report*. March.

Prati, R., Batista, G., e Monard, M. C. (2003). Uma experiência no balanceamento artificial de conjuntos de dados para aprendizado com classes desbalanceadas utilizando análise roc. *ATAI*. IV Workshop de Inteligência Artificial.

Pruitt, K. e Maglott, D. (2001). Refseq and locuslink: Ncbi gene-centered resources. *Nucleic Acids Research*, 29:137–140.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Rogozin, I., Kochetov, A., Kondrashov, F., Koonin, E., e Milanesi, L. (2001). Presence of atg triplets in 5' untranslated regions of eukaryotic cdnas correlates with a 'weak' context of the start codon. *Bioinformatics*, 10(17):890–900.

Rosenblatt, R. (1958). The perceptron. a probabilistic model for information storage and organization in the brain. *Psychological Review*, (65):386–408.

Rosenblatt, R. (1962). Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. *Spartan Books, New York*.

Russel, Stuart J.; Russel, P. N. (2004). *Inteligência Artificial*. Campus.

Salamov, A. A., Nishikawa, T., e Swindells, M. A. (1998). Assessing protein coding region integrity in cdna sequencing projects. *Bioinformatics*, 14:384–390.

Scholkopf, B., Mika, S., Burges, C., Knirsch, P., Muller, K., Ratsch, G., e Smola, A. (1999). Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10:1000–1017.

Semolini, R. e Zuben, F. J. V. (2002). Support vector machines, inferência transdutiva e o problema de classificação. *Dissertação de Mestrado*. Campinas, SP.

Stormo, G. D., Schneider, T. D., e Gold, L. M. (1982). Characterization of translational initiation sites in e. coli. *Nucleic Acid Res*, 10(9):2971–2996.

Tzanis, G., Berberidis, C., Alexandridou, A., e Vlahavas, I. (2005). Improving the accuracy of classifiers for the prediction of translation initiation sites in genomic sequences. *10th Panhellenic Conference on Informatics (PCI'2005)*, pages 426 – 436. P. Bozanis and E.N. Houstis (Eds.), Springer-Verlag, LNCS, Volos, Greece, 11-13 November.

Tzanis, G., Berberidis, C., e Vlahavas, I. (2006). A novel data mining approach for the accurate prediction of translation initiation sites. *7th International Symposium on Biological and Medical Data Analysis, Nicos Maglaveras et al. (Ed.), Springer-Verlag, Thessaloniki, Greece*, pages 92–103.

Tzanis, G. e Vlahavas, I. (2006). Prediction of translation initiation sites using classifier selection. *Proc. 4th Hellenic Conference on Artificial Intelligence (SETN-06)*, pages 367–377. G. Antoniou, G. Potamias, D. Plexousakis, C. Spyropoulos (Ed.), Springer-Verlag, LNAI 3955, Heraklion, Crete, May.

Vapnik, V. (1998). *Statistical learning theory*. John Wiley & Sons. New York.

Zeng, F., Yap, R., e Wong, L. (2002). Using feature generation and feature selection for accurate prediction of translation initiation sites. *Proceedings of 13th*

International Conference on Genome Informatics,, 13:192–200.

Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lemmem, C., Smola, A., Lengauer, T., e Müller, K. (1999). Engineering support vector machine kernels that recognize translation initiation sites. *In Proceedings of German Conference on Bioinformatics*, pages 37–43.

Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., e Müller, K. (2000). Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807.

Zurada, J. M. (1992). *Introduction to Artificial Neural Systems*. Prentice Hall.