

---

**Desenvolvimento de uma  
Metodologia para Predição de  
Sítios de Início de Tradução**

*Cristiane Neri Nobre*

---

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

# Desenvolvimento de uma Metodologia para Predição de Sítios de Início de Tradução

*Cristiane Neri Nobre*

**Orientador:** *Dr. Antônio Pádua Braga*

**Co-orientador:** *Dr. José Miguel Ortega*

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

**Belo Horizonte, Julho de 2007**

Aos meus pais  
*Wilson e Ana,*  
e aos meus irmãos,  
pelo amor, confiança e encorajamento.

# 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 torcida e amizade. Agradeço profundamente pelo apoio incondicional que eles sempre me proporcionaram.

Ao meu orientador Braga e co-orientador Miguel, pela orientação nestes quatro anos e meio, pela amizade, disponibilidade 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.

Ao Levi, pelas discussões sobre aprendizado transdutivo. Ao Thiago, 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 de 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 predição correta do início de tradução em seqüências de mRNA é um problema importante para a anotação genômica. No entanto, fazer uma predição 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, esta situação pode ser modelada como um problema de classificação entre as seqüências positivas (codificadoras de proteínas) e negativas (não codificadoras).

Para resolver este problema, os autores deste trabalho propõe a seguinte metodologia: (1) uma forma alternativa de extrair as seqüências negativas; (2) utilização de janelas menores; (3) alteração na forma de codificação dos nucleotídeos; (4) utilização do Smote - método para balanceamento das 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 transdutiva, além da inferência indutiva clássica; e, finalmente, 6) utilização do classificador Support Vector Machine - SVM - com funções simples de *kernel*.

Para testar esta 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 é uma acurácia, precisão, sensibilidade e especificidade acima de

95%, em média.

# 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 translations starts on the first ATG, but that is not a rule. Thus, this problem can be modeled as a classification solution between positive sequences (coding sequences) or negatives (non coding sequences).

To approach this problem authors of this work propose the following methodology: (1) an alternative extracton 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 high unbalanced (1:29 fold in average) for the bases used in this work; (5) use of a transductive approach besides the classical inductive inference; and finally, (6 ) use of the classificatory Support Vector machine - SVM - with simple kernel functions.

To test this methodology sequences collected by Petersen and Nielsen and Ref-Seq (Reference Sequences) sequences from NCBI (National Center for Biotechnology Information) from five organisms were used: Danio rerio, Drosophila melanogaster, Homo sapiens, Mus musculus and Rattus norvegicus, under six distinct inspection levels inspeção (reviewed, provisional, predicted, validated, model and inferred).

As result, accuracy, precision, sensibility and specificity over 95% were attained, in average.



# Sumário

---

---

Resumo . . . . .	i
Abstract . . . . .	iii
Lista de Figuras . . . . .	vi
Lista de Tabelas . . . . .	x
Lista de Abreviaturas . . . . .	xii
<b>1 Introdução</b>	<b>1</b>
1.1 Identificação do Sítio de Início da Tradução - SIT - uma breve introdução	1
1.1.1 Objetivos . . . . .	6
1.1.2 Contribuições . . . . .	7
1.1.3 Descrição da metodologia a ser utilizada . . . . .	9
1.1.4 Visão geral deste trabalho . . . . .	9
1.1.5 Conclusões do capítulo . . . . .	10
<b>2 Trabalhos relacionados ao tema</b>	<b>11</b>
2.1 Reconhecimento do Sítio de Início da Tradução - o estado da arte . . . . .	11
2.1.1 Descrição da base de dados de Pedersen e Nielsen . . . . .	12
2.1.2 Reconhecimento através de Redes Neurais Artificiais . . . . .	13
2.1.3 Reconhecimento através de SVM . . . . .	17
2.1.4 Reconhecimento por geração, seleção e integração de caracterís- ticas . . . . .	19

2.1.4.1	Geração de características . . . . .	20
2.1.4.2	Seleção de características . . . . .	23
2.1.4.3	Integração de características para tomada de decisão . . . . .	25
2.1.4.4	Melhorando o processo de reconhecimento do SIT por meio de geração, seleção e integração de características . . . . .	26
2.1.5	Reconhecimento por Função de Discriminante Linear . . . . .	34
2.1.6	Conclusões do Capítulo . . . . .	35
<b>3</b>	<b>Materiais e Métodos</b>	<b>36</b>
3.1	Bases de dados utilizadas . . . . .	37
3.2	Extração das seqüências . . . . .	38
3.3	Codificação utilizada . . . . .	43
3.4	Bases de dados desbalanceadas . . . . .	43
3.4.1	Algoritmo de balanceamento utilizado na predição do SIT . . . . .	45
3.5	Support Vector Machine - SVM . . . . .	47
3.6	Inferência indutiva <i>versus</i> transdutiva . . . . .	49
3.6.1	Inferência transdutiva e o problema de predição do SIT . . . . .	51
3.7	Medidas de desempenho utilizadas . . . . .	52
3.7.1	Validação cruzada com k-dobras . . . . .	53
3.7.2	Curva ROC . . . . .	55
3.8	Validação das seqüências através do modelo de escaneamento <i>versus</i> todas as seqüências da molécula . . . . .	57
3.9	Conclusões do capítulo . . . . .	57
<b>4</b>	<b>Resultados e Discussões</b>	<b>59</b>
4.1	Validação da metodologia proposta - uma comparação entre diferentes abordagens . . . . .	60
4.1.1	Seleção de seqüências não-SITs usadas no treinamento . . . . .	61
4.1.2	Tamanho da janela . . . . .	63
4.1.3	Desempenho em função da codificação utilizada . . . . .	66
4.1.4	Desempenho em função do método de balanceamento . . . . .	68

4.1.5	Desempenho do classificador em relação à abordagem indutiva e transdutiva . . . . .	69
4.2	Desempenho do classificador para as outras bases de dados . . . . .	74
4.3	Curva ROC para os organismos <i>Mus musculus</i> e <i>Rattus Norvegicus</i> . . .	81
4.4	Análise dos falsos positivos . . . . .	84
4.5	Parâmetros utilizados e recursos computacionais . . . . .	88
4.6	Conclusões do capítulo . . . . .	88
<b>5</b>	<b>Conclusões e propostas de continuidade</b>	<b>90</b>
5.1	Conclusões . . . . .	90
5.2	Sugestões de continuidade de trabalho . . . . .	91
<b>A</b>	<b>Apêndices</b>	<b>93</b>
A.1	Frequência de bases das seqüências . . . . .	94
A.2	Frequência de trinucleotídeos . . . . .	99
	<b>Referências</b>	<b>104</b>

# Lista de Figuras

---

---

- 1.1 Modelo de escaneamento e as posições dos nucleotídeos relativas ao códon AUG. Neste caso, o segundo AUG é o SIT. (*Imagem do autor.*) . . . 3
- 2.1 Um exemplo de seqüência da base de Pedersen e Nielsen. As 4 ocorrências de ATGs 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. O “.”, “i”, e “E” indicam se o nucleotídeo é *upstream* (.), SIT (i), ou *downstream* (E). . . . . 12
- 2.2 O módulo do consensus-RN de *DIANA-TIS*. Uma janela de 12 nucleotídeos é apresentada à rede neural. Um alto *score* na saída indica um possível SIT. (*Imagem de: Artemis Hatzigeorgiou.*) . . . . . 15
- 2.3 O módulo *coding-RN* de *DIANA-TIS*. Uma janela de 54 nucleotídeos é apresentada à rede neural. Um *score* alto na saída indica um nucleotídeo codificante. (*Imagem de: Artemis Hatzigeorgiou.*) . . . . . 16
- 2.4 As 9 características mais importantes selecionadas no trabalho de Li et al. pelo método de entropia em cada uma das 3 dobras. . . . . 28
- 2.5 Diagrama mostrando a transformação dos dados, com relação ao novo espaço de características. (*Imagem de: Huiqing et al.*) . . . . . 29

- 2.6 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., adaptada pelo autor) . . . 32
- 2.7 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.) . . . . . 32
- 3.1 Fragmento de um arquivo *RefSeq* no formato original. Este arquivo contém informações tais como organismo, número de acesso, nível de confiança, posição de início do CDS, dentre outras. (Imagem do autor.) 38
- 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.) . . . . 40
- 3.3 Diferença do método tradicional (indutivo-dedutivo) para a inferência transdutiva. No método tradicional, utilizando apenas o conjunto de treinamento, o hiperplano é construído para separar as duas classes (bolas verdes x quadrados pretos). Num segundo passo, este hiperplano é utilizado para deduzir (predizer) a classificação das amostras do conjunto de predição. Como ilustração, é mostrada a classe desconhecida a que pertencem as amostras do conjunto de predição, 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 os princípios da inferência transdutiva, o hiperplano é encontrado utilizando no treinamento os dois conjuntos de dados, treinamento e predição, sendo que os dados de predição ainda não estão classificados. Assim a classificação dos dados do conjunto de predição é feita em um único passo. Observa-se que agregando os dados do conjunto de predição ao treinamento do algoritmo, consegue-se um melhor desempenho. (Imagem de Semolini e Zuben.) . . . . . 50

- 3.4 Esquema mostrando o exemplo do método de validação cruzada com 4-dobras. . . . . 54
- 3.5 Ilustração da curva ROC e o efeito da técnica de *under-sampling*. Eliminando-se amostras da classe majoritária (negativa), o desempenho move-se do canto inferior esquerdo para o canto superior direito. (*Imagem de Nitesh et al., adaptada pelo autor*) . . . . . 56
- 4.1 Comparação entre a abordagem indutiva e transdutiva - Validação das seqüências até o SIT. . . . . 71
- 4.2 Comparação entre a abordagem indutiva e transdutiva - Validação de todos os ATGs da molécula. . . . . 73
- 4.3 Curva ROC para o *Mus musculus* com o conjunto balanceado de seqüências. Observe que o parâmetro J não influencia os resultados, visto que o conjunto de dados já está balanceado. . . . . 82
- 4.4 Curva ROC para o *Mus musculus* com o conjunto desbalanceado de seqüências. Observe que o parâmetro J influencia altamente os resultados, visto que o conjunto de dados considerado aqui é desbalanceado. 82
- 4.5 Curva ROC para o *Mus musculus* com o conjunto balanceado de seqüências. Observe que o parâmetro J não influencia os resultados, visto que o conjunto de dados já está balanceado. . . . . 83
- 4.6 Curva ROC para o *Mus musculus* com o conjunto desbalanceado de seqüências. Observe que o parâmetro J influencia altamente os resultados, visto que o conjunto de dados considerado aqui é desbalanceado. 84
- A.1 Frequência de bases das seqüências positivas e negativas, respectivamente, do *Homo sapiens reviewed*. . . . . 95
- A.2 Frequência de bases das seqüências positivas e negativas, respectivamente, do *Mus musculus reviewed*. . . . . 96
- A.3 Frequência de bases das seqüências positivas e negativas, respectivamente, do *Rattus norvegicus reviewed*. . . . . 97

- A.4 Freqüência de bases das seqüências positivas e negativas, respectivamente, da *Drosophila melanogaster reviewed*. . . . . 98
- A.5 Freqüência de trinucleotídeos das seqüências positivas e negativas *reviewed* do *Homo sapiens* e *Mus musculus*, respectivamente. A cor varia do mais claro, indicando uma alta freqüência, para o mais escuro, indicando baixa freqüência. . . . . 100
- A.6 Freqüência de trinucleotídeos das seqüências positivas e negativas *reviewed*, respectivamente, dos organismos estudados. A cor varia do mais claro, indicando uma alta freqüência, para o mais escuro, indicando baixa freqüência. . . . . 101

# Lista de Tabelas

---

---

3.1	Número de moléculas disponíveis no <i>Refseq</i> para os organismos investigados. <i>Download</i> em 11/02/07. . . . .	39
3.2	Quantidade de seqüências positivas e negativas para cada organismo, por grau de inspeção, utilizando-se janelas de 12 nucleotídeos <i>upstream</i> e <i>downstream</i> para cada ATG da molécula. . . . .	41
3.3	Quantidade de seqüências de acordo com a fase de leitura. UMF e UFF correspondem às seqüências que estão na região <i>upstream</i> do ATG na mesma fase de leitura ou fora de fase, respectivamente. DMF e DFF correspondem às seqüências que estão na região <i>downstream</i> do ATG na mesma fase de leitura ou fora de fase, respectivamente. . . . .	42
3.4	Descrição do algoritmo SMOTE . . . . .	46
4.1	Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função da seleção de seqüências não-SITs utilizadas no treinamento. . . . .	62
4.2	Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função do tamanho da janela utilizada. 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. . . . .	64



4.3	Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função do tamanho da janela utilizada. A validação foi realizada considerando-se todos os ATGs da molécula. . . . .	66
4.4	Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função da codificação utilizada. . . . .	67
4.5	Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade em função do método de balanceamento. . . . .	69
4.6	Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade para a base de dados de Petersen e Nielsen. . . . .	75
4.7	Desempenho do classificador para o <i>Danio rerio</i> utilizando-se a abordagem indutiva e transdutiva, validando as seqüências até o SIT. . . . .	76
4.8	Desempenho do classificador para o <i>Drosophila melanogaster</i> utilizando-se a abordagem indutiva e transdutiva, validando as seqüências até o SIT. . . . .	77
4.9	Desempenho do classificador para o <i>Mus musculus</i> utilizando-se a abordagem indutiva e transdutiva, validando as seqüências até o SIT. . . . .	78
4.10	Desempenho do classificador para o <i>Homo sapiens</i> utilizando-se a abordagem indutiva e transdutiva, validando as seqüências até o SIT. . . . .	79
4.11	Desempenho do classificador para o <i>Rattus norvegicus</i> utilizando-se a abordagem indutiva e transdutiva, validando as seqüências até o SIT. . . . .	80
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 <i>reviewed</i> . Onde, TamTotal é o número total de seqüências positivas <i>reviewed</i> e ocorrência é a quantidade de vezes que o trinucleotídeo aparece. As maiores porcentagens estão em negrito.	103

# Lista de Siglas e Abreviaturas

---

---

AUC	<i>Area under the ROC curve</i>
CDS	<i>CoDing Sequence - Seqüência Codificadora</i>
DNA	<i>Ácido Desoxiribonucleico</i>
cDNA	<i>Ácido Desoxiribonucleico complementar</i>
ESTs	<i>Expressed Sequence Tags - Seqüências de Expressão</i>
IA	<i>Inteligência Artificial</i>
KNN	<i>K Nearest Neighbor</i>
MCP	<i>McCulloch e Pitts</i>
NCBI	<i>National Center for Biotechnology Information</i>
NIH	<i>National Institutes of Health</i>
RefSeq	<i>Reference Sequences</i>
RBF	<i>Radial Basis Function</i>
RN	<i>Rede Neural Artificial</i>
RNA	<i>Ácido Ribonucléico</i>
ROC	<i>Receiver Operating Characteristic</i>
mRNA	<i>Ácido Ribonucléico Mensageiro</i>
SITs	<i>Sítios de Início de Tradução</i>
SVM	<i>Support Vector Machines</i>
Trepan	<i>TREes PARrotting Networks</i>
UTRs	<i>UnTranslated Regions - Região não codificadora</i>

---

# Introdução

---

---

## 1.1 *Identificação do Sítio de Início da Tradução - SIT - uma breve introdução*

Sistemas vivos são conhecidos pelas proteínas que produzem de acordo com sua informação genética. No entanto, somente parte das seqüências transcritas carregam esta 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).

Desta forma, dado um fragmento de DNA ou mRNA<sup>1</sup>, um problema central da biologia computacional é determinar se ele contém CDS; e, a partir daí, descobrir qual proteína ele codifica.

De acordo com Pedersen (Pedersen e Nielsen, 1997), o reconhecimento do Sítio de Início de Tradução (SIT) em eucariotos nem sempre começa na primeira metionina

---

<sup>1</sup>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 e Jambeck, 2001).

(AUG) em seqüências de mRNA ou cDNA<sup>2</sup>. Isto faz com que a predição de SIT não seja uma tarefa trivial, especialmente quando analisando ESTs<sup>3</sup> e dados genômicos, onde o mRNA completo não é conhecido ou, muitas vezes, quando é conhecido não é livre de erros.

Um dos trabalhos pioneiros em predição de início de tradução foi realizado por Stormo e colaboradores (Stormo et al., 1982). Mas, ao contrário da grande maioria dos autores que estudam o início de tradução em eucariotos, estes analisaram o SIT em *E. coli*. Utilizando-se de perceptron (Haykin, 1999; Braga et al., 2000; Zurada, 1992), com janelas de 51, 71 e 104 nucleotídeos e codificação binária, eles caracterizaram as seqüências de Shine e Dalgarno de 124 genes conhecidos.

Em 1978, Kozak e Shatki (Kozak e A.J., 1978) propuseram um modelo de escaneamento, o qual foi mais tarde atualizado por Kozak (Kozak, 1989). De acordo com o modelo, a tradução inicia-se no primeiro ATG que tem um contexto apropriado. Foi Kozak (Kozak, 1987) quem desenvolveu a primeira amostra de consenso para uma grande coleção de dados. Este consenso é GCC[A/G]CCAAUGG, onde a Guanina (G) aparece na posição +4 (posição que segue o códon ATG) e uma purina, preferencialmente a Adenina (A), aparece na posição -3. Ou seja, estas bases, nestas 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). Desta forma, começa-se a tradução dos códons para os aminoácidos. Esta é a teoria mais utilizada para identificação do SIT.

---

<sup>2</sup>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 e Jambeck, 2001).

<sup>3</sup>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 e Jambeck, 2001).

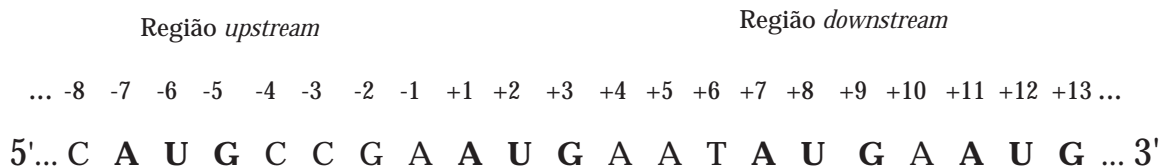


Figura 1.1: Modelo de escaneamento e as posições dos nucleotídeos relativas ao códon AUG. Neste caso, o segundo AUG é o SIT. (*Imagem do autor.*)

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

Pedersen e Nielsen (Pedersen e Nielsen, 1997) comunicaram um trabalho onde uma Rede Neural Artificial (RN) foi treinada com uma base de vertebrados e o desempenho da rede foi de 85%. Esta base, extraída do *Genbank* (Benson et al., 1997)<sup>4</sup>, foi processada para remover possíveis íntrons e juntar os exons. Destas seqüências, somente aquelas com o SIT anotado e com pelo menos 10 bases *upstream* e até 150 bases *downstream* foram selecionadas. As seqüências foram filtradas para remover aquelas pertencentes à mesma família gênica, genes homólogos de diferentes organismos e seqüências redundantes. Depois de processadas as moléculas, eles geraram seqüências com 100 nucleotídeos na região *upstream* e *downstream* cada cada ATG encontrado e usaram codificação binária (A=0001, C=0010, G=0100 e T=1000). O

<sup>4</sup>*GenBank* é o banco de dados de seqüências genéticas do NIH (National Institutes of Health), uma coleção anotada de todas as seqüências de DNA disponíveis publicamente. *GenBank* e PIR (*Protein Information Resource*) são bases de dados norte-americanas, EMBL (*European Molecular Biology Laboratory*) é uma base de dados européia, DDBJ (*DNA Data Bank of Japan*) é uma base japonesa e a *Swiss-Prot* é suíça.

resultado final foi a geração de uma base de dados contendo 13503 seqüências contendo ATGs, sendo que destas 3312 (24,5%) são SITs e 10190 (75,5%) são não-SITs. Estas seqüências têm sido utilizadas como referência em praticamente todos os trabalhos relativos a este tema.

Zien e colaboradores (Zien et al., 2000) usando Support Vector Machines (SVM), obtiveram resultados melhores para a mesma base de dados usada por Pedersen e Nielsen (Pedersen e Nielsen, 1997). Eles utilizaram o mesmo tamanho de janela (203 nucleotídeos) e a mesma codificação binária utilizada por eles. Eles relatam que SVM é capaz de retornar resultados comparáveis aos obtidos por Pedersen e Nielsen, usando-se funções padrões de *kernel*. Com modificações na função do *kernel*, eles conseguiram melhorar ainda mais os resultados, alcançando 88,1% de acurácia<sup>5</sup>. Mais tarde, usando-se uma função de *kernel* mais sofisticada, Salzberg kernel (Salzberg, 1997), eles alcançaram 88,6% de acurácia para o mesmo conjunto de dados.

Em seu trabalho, Hatzigeorgiou (Hatzigeorgiou, 2002) apresenta um programa, DIANA-TIS, usando duas RNs com seqüências humanas. A base de dados utilizada contém seqüências de cDNA completa e um desempenho de 94% é obtido. Este resultado foi encontrado por meio do conceito do escaneamento do ribossomo, utilizado por ela. Ou seja, o algoritmo inicia uma busca linear no início da região não codificadora e pára quando encontra um *score* positivo.

Zeng e colaboradores (Zeng et al., 2002), usando uma janela de 200 nucleotídeos, utilizam-se de um conceito de geração de características e partir da seleção das melhores (posição -3, ATG *upstream* em fase<sup>6</sup> e TAA, TAG, TGA, CTG, GAC, GAG, GCC *downstream* em fase) alcançam 90% de acurácia sobre a base do Pedersen e Nielsen. Além disso, incorporando o modelo de escaneamento do ribossomo, utilizado por Hatzigeorgiou (Hatzigeorgiou, 2002), eles obtêm uma acurácia de 94,4%.

Em trabalho posterior, Huiqing e colaboradores (Huiqing et al., 2004) trabal-

---

<sup>5</sup>Acurácia é proporção de todas as predições que são corretas, tanto de SIT quanto de não-SIT.

<sup>6</sup>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 ter ..., -9, -6, -3, +4, +7, +10,

haram com seqüências de aminoácidos para fazer a predição do SIT. Segundo eles, características candidatas são geradas usando seqüências de aminoácidos *k-gram*. Um *k-gram* é um padrão de *k* letras consecutivas, que podem ser símbolos de aminoácidos ou ácidos nucléicos. Usando este conceito, eles verificam a freqüência de ocorrência de certos padrões a partir das seqüências. O seção 2.1.4.1 descreve com detalhes esta técnica. Em um segundo momento, um pequeno número de características (considerados como principais) são selecionadas. No terceiro e último passo, um modelo de classificação é construído para reconhecer o verdadeiro SIT, aplicando SVM ou árvore de decisão a partir das características selecionadas. Os resultados obtidos por esta metodologia foram melhores do que aqueles obtidos pelo trabalho anterior (Zeng et al., 2002), utilizando-se a mesma base de dados.

Haifeng e Tao, em (Haifeng e Tao, 2004), introduziram uma classe de novos *kernels* baseada em similaridades das seqüências, chamada de *edit kernels*, para ser utilizada com SVM. Além disto, eles convertem a região *downstream* de uma seqüência de mRNA em uma seqüência de aminoácidos antes de aplicá-la ao SVM. Usando-se a mesma base de Petersen e Nielsen, eles encontraram uma acurácia, sensibilidade<sup>7</sup> e especificidade<sup>8</sup> de 99,9%, 99,92% e 99,82% a partir de uma janela de 30 bases *upstream* e 180 *downstream*.

Nos trabalhos de Tzanis e colaboradores, em (Tzanis et al., 2005), (Tzanis et al., 2006) e Tzanis e Vlahavas (Tzanis e Vlahavas, 2006), eles também propõem um novo conjunto de características importantes para o reconhecimento do SIT e conseguem uma acurácia de 96,25% extraindo estas características a partir de uma janela de 99 nucleotídeos nas regiões *upstream* e *downstream* do ATG.

O capítulo 2 apresenta uma explicação detalhada de todos estes métodos existentes na literatura.

Tendo em vista os trabalhos desenvolvidos na área de predição de SIT, este trabalho propõe um nova metodologia para este problema, a saber: (1) formas alternativas de se obter as seqüências negativas de uma dada molécula de mRNA; (2) a

---

<sup>7</sup>Sensibilidade refere-se à proporção de SIT corretamente classificada como SIT.

<sup>8</sup>Especificidade refere-se à proporção de não-SIT corretamente classificada como não-SIT.



utilização de janelas menores (12 nucleotídeos nas regiões *upstream* e *downstream* do ATG); (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, a partir desta metodologia, os classificadores padrão existentes na literatura conseguem alcançar um desempenho comparável aos obtidos até o momento.

A fim de validar esta metodologia, inicialmente serão apresentados os resultados obtidos a partir das seqüências de Pedersen e Nielsen (Pedersen e Nielsen, 1997), uma vez que a maioria dos trabalhos da literatura trabalha com esta base. Além disto, serão apresentados também os resultados a partir de seqüências RefSeq (Pruitt e Maglott, 2001) do NCBI<sup>9</sup> 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 novas metodologias que forneçam alta acurácia na predição do SIT;
- Analisar a qualidade das seqüências, segundo os seis níveis de inspeção existentes: *reviewed*, *provisional*, *predicted*, *inferred*, *model*, e *validated*;

---

<sup>9</sup><http://www.ncbi.nlm.nih.gov>.

- Verificar se é possível validar eficientemente o reconhecimento de seqüências de um dado organismo, a partir de seqüências de um outro organismo;

### 1.1.2 Contribuições

Como já foi mencionado anteriormente, apesar das referências apresentadas, ainda existem poucos trabalhos relacionados ao tema "Predição do SIT". E talvez, devido a isto, algumas necessidades, que são de suma importância neste contexto, não foram trabalhadas, ou o foram de forma parcial. Dentre as principais necessidades, destacam-se:

1. Formas alternativas de se obter as 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. Este fator pode, além de aumentar significativamente o número de seqüências (aumentando ainda mais a base de dados), afetar diretamente a qualidade do classificador.
2. Estudo do problema com janelas menores. As bases depositadas nos bancos públicos hoje 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 e, conseqüentemente, proibitivo. Neste trabalho, por exemplo, temos 373765 seqüências do *Homo sapiens reviewed* (sendo 10657 positivas e 363108 negativas) e o tempo de treinamento destas seqüências é de aproximadamente 25 dias em uma CPU comum.
3. Formas alternativas de codificação. A codificação, além de ser capaz de melhorar o desempenho de um classificador, também pode ser fundamental na redução do número de entradas do classificador. Com isto, podemos ter também uma redução no tempo de processamento.
4. Aplicação de um método de balanceamento. Este fator é decisivo para o desempenho do classificador utilizado, visto 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 prever a classe minoritária com precisão. Uma vez que este problema é altamente desbalanceado (da ordem de 1:29, em média, para as bases utilizadas neste trabalho), faz-se necessário a aplicação de um bom método de balanceamento.

5. Este problema de predição do 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 estas seqüências sem classificação podem ser utilizadas durante o treinamento para melhorar o desempenho do classificador.
6. Utilização do SVM com funções simples de *kernel*. Acredita-se que com todas estas etapas anteriores sendo trabalhadas, as funções simples de *kernel* já existentes possam oferecer um bom desempenho, sem a necessidade de se criar funções alternativas.
7. Analisar a qualidade das seqüências, segundo os seis níveis de inspeção existentes. Em 2003 havia apenas três níveis de inspeção: *reviewed*, *provisional* e *predicted*. Hoje temos seis níveis, com o acréscimo das seqüências *inferred*, *model* e *validated*. Analisar a qualidade destas seqüências pode ser uma grande contribuição para a área de predição de SIT.
8. Verificar se é possível validar eficientemente o reconhecimento de seqüências de um dado organismo, a partir de seqüências de um outro organismo. Muitas vezes têm-se poucas seqüências de um determinado organismo. Com base nisto, pretende-se analisar se o treinamento com um determinado organismo pode ser utilizado para validar seqüências de outro. Com isso, poderemos comparar organismos com base no estudo do SIT e, dependendo dos resultados, não ficarmos limitados ao tamanho das bases.

Desta forma, este trabalho vem contribuir de forma significativa para a área de bioinformática, em relação a predição do SIT.

### 1.1.3 Descrição da metodologia a ser utilizada

A metodologia aplicada no desenvolvimento deste trabalho pode ser dividida da seguinte maneira:

1. Levantar as principais técnicas existentes na literatura para predição do SIT;
2. Extrair seqüências dos organismos *Danio rerio*, *Drosophila melanogaster*, *Homo sapiens*, *Mus musculus* e *Rattus norvegicus* do RefSeq;
3. A partir das seqüências extraídas, propor novas metodologias que forneça uma acurácia superior, ou pelo menos comparável, aos métodos existentes;
4. Analisar a qualidade das seqüências, segundo os seis níveis de inspeção: *reviewed*, *provisional*, *predicted*, *inferred*, *model*, e *validated*;
5. Validar seqüências de um determinado organismo que possua poucas seqüências a partir de um outro organismo.

Todos estes passos serão descritos com detalhes no capítulo 3.

### 1.1.4 Visão geral deste trabalho

Este trabalho é organizado em capítulos, a saber:

O capítulo 2 apresenta uma descrição detalhada dos trabalhos relacionados ao problema de identificação do SIT.

O capítulo 3 apresenta 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.

### *1.1.5 Conclusões do capítulo*

Neste capítulo foi apresentada a motivação para a realização deste trabalho, além do estabelecimento de seus principais objetivos. Foi descrita também a metodologia utilizada durante o processo de desenvolvimento de novos métodos, capazes de atender a estes objetivos. E, finalmente, as contribuições que este trabalho proporcionará à área de Bioinformática foram relacionadas, uma vez alcançados os objetivos apresentados.

O capítulo 2 apresentará, com detalhes, o estado da arte sobre o tema de predição de SIT.

---

## Trabalhos relacionados ao tema

---

---

### *2.1 Reconhecimento do Sítio de Início da Tradução - o estado da arte*

Predição correta do SIT é uma tarefa importante em mRNA, cDNA ou outros tipos de seqüências genômicas. Alta acurácia na predição pode ajudar o entendimento da codificação de proteínas a partir das seqüências de nucleotídeos. No entanto, a predição correta do SIT não é uma tarefa trivial, uma vez que os mecanismos biológicos muitas vezes não são necessariamente entendidos e as seqüências não são completamente livres de erros, além de poderem estar incompletas.

Este capítulo destina-se à apresentação dos principais métodos de predição do SIT existentes na literatura, segundo as diferentes abordagens. No entanto, como a grande maioria dos trabalhos existentes sobre predição de SIT utilizam a base de dados criada por Pedersen e Nielsen (Pedersen e Nielsen, 1997), a próxima seção destina-se à uma pequena explicação sobre esta base.

### 2.1.1 Descrição da base de dados de Pedersen e Nielsen

A base de dados de Pedersen e Nielsen (Pedersen e Nielsen, 1997) composta pelos vertebrados *Bos taurus* (vaca), *Gallus gallus* (galinha), *Homo sapiens* (homem), *Mus musculus* (camundongo), *Oryctolagus cuniculus* (coelho), *Rattus norvegicus* (rato), *Sus scrofa* (porco) e *Xenopus laevis* (rã africana) é processada para remoção dos possíveis íntrons e união dos exons, obtendo assim as seqüências de mRNA correspondentes. Destas seqüências, somente aquelas com o SIT anotado e com pelo menos 10 nucleotídeos na região *upstream* e 150 nucleotídeos na região *downstream* foram selecionadas. Estas 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
CGTGTGTGCAGCAGCCTGCAGCTGCCCAAGCCATGGCTGAACACTGACTCCCAGCTGTG 80
CCCAGGGCTTCAAAGACTTCTCAGCTTCGAGCATGGCTTTTGGCTGTCAGGGCAGCTGTA 160
GGAGGCAGATGAGAAGAGGGAGATGGCCTTGGAGGAAGGGAAGGGCCTGGTGCCGAGGA
240 CCTCTCCTGGCCAGGAGCTTCTCCAGGACAAGACCTTCCACCCAACAAGGACTCCCCT
..... 80
.....iEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE 160
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE 240
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE

```

Figura 2.1: Um exemplo de seqüência da base de Pedersen e Nielsen. As 4 ocorrências de ATGs 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. O “.”, “i”, e “E” indicam se o nucleotídeo é *upstream* (.), SIT (i), ou *downstream* (E).

Neste exemplo, vê-se 4 ocorrências de ATGs, sendo que há 1 na região *upstream* do SIT e os outros 2 estão na região *downstream*. Existem um total de 13502 ATGs na base original de Pedersen e Nielsen, sendo que 3312 (24,5%) são SITs; enquanto 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 desta base, juntamente com o seu *download*, encontra-se em: <http://www.cbs.dtu.dk/databases/NetStart/>.

Quase todos os trabalhos existentes na literatura até o momento utilizam esta base de dados. Muitos deles extraem, 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 (Pedersen e Nielsen, 1997) também adotaram esta metodologia.

Vale ressaltar que esta base de dados foi construída em 1997; hoje é possível criar bases muito maiores e melhores a partir do *Genbank*<sup>1</sup> (Benson et al., 1997), *Refseq*<sup>2</sup> (Pruitt e Maglott, 2001) ou outras bases de dados existentes.

As seções seguintes apresentam os principais métodos de predição do SIT existentes na literatura. Para uma melhor visualização, eles foram divididos por categoria; assim as seções 2.1.2, 2.1.3, 2.1.4 e 2.1.5 apresentam os métodos que utilizam Redes Neurais Artificiais, SVM, geração de características e Reconhecimento por Função de Discriminante Linear, respectivamente (Li et al., 2004).

### 2.1.2 Reconhecimento através de Redes Neurais Artificiais

Um dos primeiros trabalhos de predição se deu em 1982 por Stormo e colaboradores (Stormo et al., 1982). No entanto, ao contrário da maioria dos trabalhos que fazem predição 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 - 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).

---

<sup>1</sup>Disponível em [www.ncbi.nlm.nih.gov/Genbank/](http://www.ncbi.nlm.nih.gov/Genbank/)

<sup>2</sup>Disponível em [www.ncbi.nlm.nih.gov/RefSeq/](http://www.ncbi.nlm.nih.gov/RefSeq/)



Stormo et al utilizaram codificação de 4 bits (A=1000, C=0100, G=0010 e T=0001) e janelas de 51, 71 e 101 nucleotídeos centradas no ATG. Destes tamanhos, eles avaliaram que 101, para o problema deles, era o ideal.

Em Pedersen e Nielsen (Pedersen e Nielsen, 1997), uma RN também foi treinada com uma janela de 203 nucleotídeos centrada no ATG. Eles utilizaram uma rede *feed-forward* (Braga et al., 2000; Haykin, 1999) com uma camada intermediária. 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 da RN. Além disto, testaram com janelas de tamanhos 13, 33, 53, 73, 93, 113, 133, 153, 173 e 203 nucleotídeos. Dentre estas configurações testadas, eles verificaram que o tamanho de 203 bases com 30 neurônios na camada oculta fornecia o melhor desempenho. Eles obtiveram uma sensibilidade, especificidade e acurácia de 78% e 87% e 85%, respectivamente, usando a base descrita na seção 2.1.1. Este sistema, chamado de NetStart, está disponível em <http://www.cbs.dtu.dk/services/NetStart>.

Pedersen e Nielsen realizaram também uma análise das seqüências para tentar descobrir 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 tentar descobrir o efeito desta 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 (Kozak, 1987). Eles avaliaram também seqüências não-SIT que foram classificadas erradamente como SIT. Como resultado, identificaram que a maioria destas seqüências estavam na mesma fase de leitura do SIT, independentemente da região *upstream* ou *downstream*.

Hatzigeorgiou (Hatzigeorgiou, 2002) apresenta um programa com alta acurácia na predição de SIT, *DIANA-TIS*, usando RNs em seqüências humanas. A base de dados é composta por seqüências de cDNA completas, filtradas para redução de erros. Uma acurácia 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 proposto. Esta idéia foi utilizada inicialmente para predição de início de *splice* por Brunak et al. (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 oculta e a topologia não totalmente conectada. Para cada molécula, foram extraídas uma seqüência positiva e uma negativa (a primeira depois da seqüência positiva). No total, foram extraídas 325 seqüências positivas e negativas.

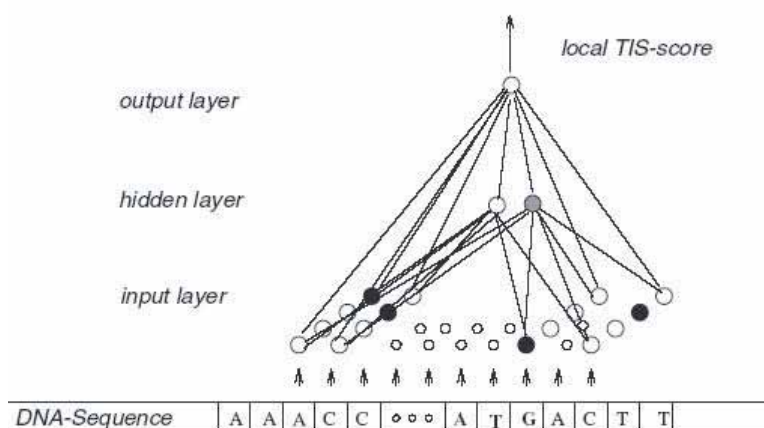


Figura 2.2: O módulo do *consensus-RN* de *DIANA-TIS*. Uma janela de 12 nucleotídeos é apresentada à rede neural. Um alto score na saída indica um possível SIT. (Imagem de: Artemis Hatzigeorgiou.)

O módulo *coding-RN* foi utilizado para reconhecer as regiões *upstream* e *downstream* do SIT candidato e trabalha com janelas de 54 nucleotídeos. Como cada três nucleotídeos forma um códon que traduz em aminoácido, existem 64 códon possíveis. Assim, para avaliar as seqüências dos 54 nucleotídeos, ela foi transformada em um vetor de 64 unidades antes de aplicar à rede neural. Cada unidade fornece a freqüência de determinado códon na seqüência. A estrutura de rede utilizada neste caso é apresentada na figura 2.3

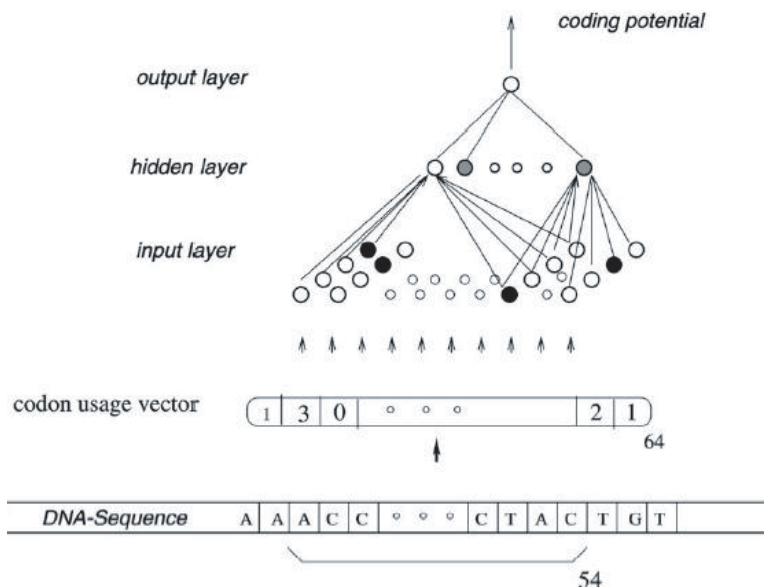


Figura 2.3: O módulo *coding-RN* de DIANA-TIS. Uma janela de 54 nucleotídeos é apresentada à rede neural. Um *score* alto na saída indica um nucleotídeo codificante. (Imagem de: Artemis Hatzigeorgiou.)

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 à 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  $S_1 \times (S_3 - S_2)$ . Este 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 da esquerda para a direita (da região 5' para a região 3') testando cada ATG até encontrar aquele que será classificado como SIT; todos os outros ATGs que estão à direita deste ATG são considerados como não-SITs. Ou seja, exatamente uma única predição é feita por molécula a partir desta 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 diferente da base de Pedersen e Nielsen.

### 2.1.3 Reconhecimento através de SVM

Zien et al (Zien et al., 1999; Zien et al., 2000) 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 (não conhecido)). Eles relataram que SVM, com *Kernel* polinomial, alcança desempenho comparável ao obtido por Pedersen e Nielsen usando RNAs.

*Kernel* polinomial de grau  $d$ ,

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

são comumente utilizados em SVM. A codificação utilizada, setando um *bit* em cada posição, indica se a base é A(00001), C(00010), G(00100), T(01000) ou N(10000). Conseqüentemente, o produto  $(X \cdot Y)$  é equivalente a contar o número de nucleotídeos que coincidem nas duas seqüências representadas pelos vetores  $X$  e  $Y$ . Similarmente,  $(X \cdot Y)^d$  é equivalente a correlação das freqüências dos nucleotídeos em todas as posições da seqüência  $d$ . Zien et al (Zien et al., 1999) relatam que SVM, como este tipo de *Kernel* padrão, consegue um desempenho no reconhecimento do SIT comparável à RN de Pedersen e de Nielsen.

No *Kernel* polinomial  $(X \cdot Y)^d$  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 (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 próximas 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. A função de kernel, *locality-improved kernel*, é definido, desta forma, como

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

onde,

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

Aqui,  $w_j$  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 o *locality-improved kernel*, Zien et al (Zien et al., 1999) obtêm uma sensibilidade de 69.9% e uma especificidade de 94.1%, dando uma acurácia total de 88.1% utilizando a base de dados de Pedersen e Nielsen descrita na seção 2.1.1. Eles mostram, assim, que através de funções simples de *Kernel* é possível conseguir uma acurácia melhor do que àquela obtida por Pedersen e Nielsen utilizando RN.

Mais tarde, Zien et al (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. Este *Kernel* fornece uma acurácia de 88.6% para a mesma base de dados.

Haifeng e Tao (Haifeng e Tao, 2004) utilizam duas novas propostas para identificação do SIT. Primeiro, eles introduzem uma classe de novos *kernels* baseados em *string edit distance*, chamado de *edit kernels* para ser utilizado com SVM. Segundo eles, os *edit kernels* são simples e possuem interpretações biológicas significativas e probabilísticas. Em um segundo momento, eles convertem a região *downstream* de um ATG em uma seqüência de aminoácidos antes de aplicar ao SVM. Eles mostram 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 àquelas que utilizavam SVM com *Kernel* polinomial ou *Salzberg*. Eles também testaram a metodologia proposta por eles 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/>.

#### 2.1.4 Reconhecimento por geração, seleção e integração de características

Wong (Zeng et al., 2002; Liu e Wong, 2003) mostram 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. seleção das características relevantes a partir das candidatas, e
3. integração das características selecionadas usando algum método de apren-

dizado de máquina para construir um sistema para reconhecer propriedades específicas das seqüências, neste caso o SIT.

Estes três passos serão apresentados nas subseções seguintes.

#### 2.1.4.1 Geração de características

Até o momento, os métodos apresentados extraem informações para predição do SIT basicamente através do conhecimento contido na própria seqüência. Ou seja, os próprios classificadores utilizados extraem estas informações das seqüências 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 símbolos de aminoácidos ou ácidos nucleicos. *K-gram* também pode ser restrita 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, esta técnica identifica em qual posição do fragmento da seqüência, o *k-gram* aparece.

Como mostrado na figura 1 do capítulo anterior, o modelo de escaneamento do ribossomo sugere a leitura dos nucleotídeos da região 5' para a região 3'. Além disto, foi mostrado também que a base "A" do ATG codificante é numerado 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 "A" é numerado com "-1", o segundo "-2" em direção à região 5'.

Desta forma, para uma melhor compreensão da técnica de *k-grams*, considere a figura 2.1 apresentada na seção 2.1.1. Como foi visto, o segundo ATG é o SIT e os 100 nucleotídeos *upstream* deste 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 frequência 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 disto, como já comentado no capítulo 1, as regiões *upstream* e *downstream* do SIT são respectivamente não-codificante e codificante. Assim, imagina-se que estas 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 frequência igual a 1 (uma vez que tem-se somente um ATG na região *upstream* do SIT); GCT, com frequência igual 5; e TTT, com frequência igual 0. Desta 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 frequência igual a 2 (uma vez que têm-se 2 ATGs na região *downstream* do SIT); GCT, com frequência igual a 3; e TTT, com frequência igual a 2. Neste caso, as características candidatas e os valores que correspondem a estes *k-grams* são “ATG *downstream*=2”, “GCT *downstream*=3”, 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 as posições que têm significado biológico, eles também são chamados de boas características candidadas. No total, existem  $2 \times 4^3$  possibilidades destes 3-grams. No exemplo em questão, alguns 3-grams *downstream* em fase são: GCT, TTT e ATG, com frequências iguais a 1 nestes 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 frequências 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 chamam de posições específicas de k-grams. Para este tipo de k-grams, eles simplesmente armazenam qual k-grams aparece em uma determinada posição da sequência a ser analisada. Neste caso, é suficiente considerar apenas 1-grams, 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 uma Guanina (G). As características candidatas e seus valores associados são: “posição-3=G” e “posição+4=G”.

Combinando todas as características discutidas acima, para  $k = 1, \dots, 5$ , cada sequência 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*=3”, “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=G”, ..., “posição+4=G”, ...}. A estas características, dá-se o nome de vetor de características.

Estas 4436 características, descritas acima, são geradas para cada uma das

13502 seqüências que contêm ATG na base de dados criada por Pedersen e Nielsen, descrita na seção 2.1.1. Além disto, muitas outras características podem ser extraídas a partir das seqüências (Li et al., 2004). No entanto, muitas vezes se torna inviável trabalhar com uma quantidade tão grande de características. Assim, a seção 4.1.1 mostra como pode ser feito o reconhecimento do SIT com base em um conjunto menor destas 4436 características candidatas.

#### 2.1.4.2 Seleção de características

Como descrito anteriormente, o número de características gerado é, claramente, muito grande. Muitas destas características podem ser ruído que muitas vezes atrapalham os algoritmos de aprendizagem de máquina típicos. Desta 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, 2000), teste  $\chi^2$  (Liu e Sectiono, 1995), medida de entropia (Fayyad e Irani, 1993) ou método de seleção de características baseado em correlação (CFS) (Hall, 1998).

Utilizando-se a técnica CFS, (Zeng et al., 2002) 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 estas 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[AG]CCAUGG, (apresentado na seção 1) para predição de início de tradução de vertebrados. Kozak mostra que a posição -3 é altamente conservativa para identificação do SIT e normalmente apresenta uma purina “A” ou “G” (preferencialmente a “A”). Pedersen e Nielsen também analisaram as seqüências e chegaram a mesma conclusão quanto à esta posição, como descrito na seção 2.1.2.

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, da esquerda para a direita) 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 *upstream* em fase com o SIT pode indicar que o SIT (previsto inicialmente) tem menos chances de ser o SIT. Isto também está de acordo com o trabalho de Rogozin et al. (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. Similarmente, uma correlação negativa entre a qualidade do contexto do início de tradução e o número de ATGs na região *upstream*. Tipicamente, cDNAs contendo longas regiões 5’ UTRs com vários ATGs nesta região têm um contexto pobre, e ao contrário, cDNAs contendo regiões 5’ UTR pequenas sem a presença de ATGs tem contextos fortes. Pedersen e Nielsen (Pedersen e Nielsen, 1997) também chegaram a estas conclusões quando fizeram uma avaliação detalhada das predições erradas feitas pela RN.

“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 destas três características sinaliza que existe um *stop-codon* em fase nos 100 nucleotídeos da região downstream do SIT (no exemplo considerado). Consequentemente, 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. Estas 3 características não foram relatadas 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”).

#### 2.1.4.3 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.

Em seu trabalho, eles apresentam os resultados obtidos por meio dos classificadores Bayesianos (Langley et al., 1992), SVM (Carvalho et al., 2002; Scholkopf et al., 1999), Redes neurais e árvore de decisão (com o C4.5) (Quinlan, 1993). De acordo com eles, utilizando-se o classificador Bayesiano, 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.

Neste mesmo trabalho, (Zeng et al., 2002) apresentam mais três maneiras para tentar aumentar a predição do SIT: 1) usando um *bagging* (Breiman, 1997; Chawla

et al., 2003)<sup>3</sup> dos 4 classificadores adotados: Bayesianos, SVM, Redes neurais e C4.5 (algumas combinações destes 4); 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 proposto por Hatzigeorgiou (Hatzigeorgiou, 2002), apresentado na seção 2.1.2.

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 ressalta a importância da técnica de geração de características. E, finalmente, usando o modelo de escaneamento proposto por Hatzigeorgiou (Hatzigeorgiou, 2002), eles conseguem melhorar as 4 taxas de desempenhos medidas (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%).

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

Como a cada 3 nucleotídeos temos a codificação de um aminoácido, seria interessante investigar se uma abordagem alternativa para geração de características baseada em aminoácidos pode produzir uma boa classificação do SIT. Além disto, na seção 4.1.1, (Zeng et al., 2002) usam a técnica CFS para geração de características. Assim, outras técnicas podem ser utilizadas para fazer esta seleção. (Li et al., 2003) e (Liu e Wong, 2003) analisam estas duas abordagens. Para a geração de características eles extraem segmentos de seqüências da mesma maneira que foi feito anteriormente: 100 nucleotídeos nas regiões *upstream* e *downstream* de cada ATG da molécula. Então, eles consideram *3-grams* para aqueles que estão em fase com o

---

<sup>3</sup>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 Bayesiano, 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.

ATG. Assim, aqueles 3-grams que codificam proteínas são convertidos nas suas letras de aminoácidos correspondentes, enquanto que os 3-grams que correspondem a *stop-codon*, são convertidos em letras especiais simbolizando um *stop-codon*. Para a convenção acima, (Li et al., 2003) geram os seguintes tipos de k-grams:

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

Além disto, (Li et al., 2003) também geram características booleanas a partir dos segmentos de seqüências extraídos da base do Pedersen e Nielsen: ATG-up, indicando a presença de um ATG na região *upstream*; pos-3AouG-up, indicando se existe uma “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 figura 2.4 mostra as 9 principais características selecionadas.

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

Figura 2.4: As 9 características mais importantes selecionadas no trabalho de Li et al. pelo método de entropia em cada uma das 3 dobras.

Interessantemente, a maioria destas características, exceto A-up e V-down, correspondem àquelas características selecionadas por CFS descritas na seção 4.1.1. 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”.

Eles também usaram os classificadores Bayesianos, 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 Bayesiano. Estes resultados são ligeiramente piores do que àqueles apresentados na seção 2.1.4.3 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, usando-se o mesmo classificador.

Li e colaboradores (Li et al., 2003) usam apenas as 10 características selecionadas por entropia e encontram resultados comparáveis a estes.

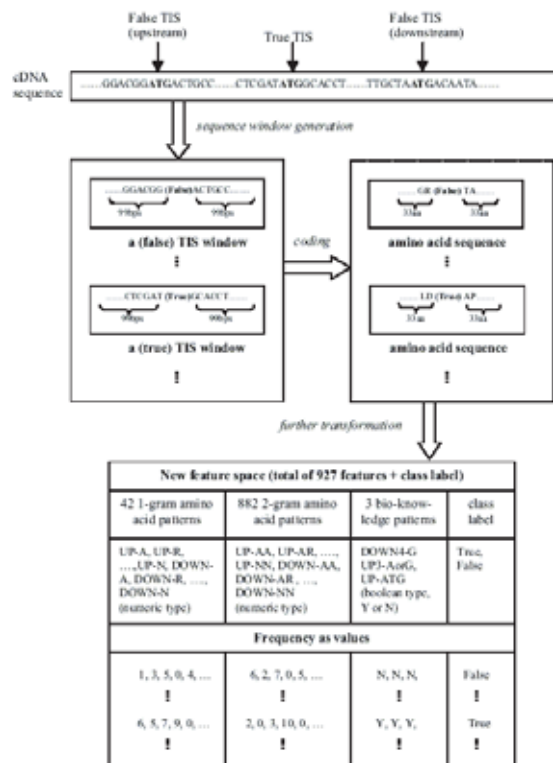


Figura 2.5: Diagrama mostrando a transformação dos dados, com relação ao novo espaço de características. (Imagem de: Huiqing et al.)

Mais tarde, Huiqing e colaboradores (Huiqing et al., 2004) utilizaram esta mesma metodologia para geração de características proposta por Li et al. (Li et al., 2003), mas utilizam-se apenas padrões de 1-gram e 2-gram, gerando assim um total de 927 características também (já com as características booleanas). A figura 2.5 apresenta um diagrama mostrando a transformação dos dados com relação ao novo espaço de características.

Eles testaram a metodologia proposta com três(3) bases de dados: 1) as bases de vertebrados criada por Pedersen e Nielsen (Pedersen e Nielsen, 1997), 2) a base de Hatzigeorgiou (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 Perderson e Nielsen.

Em 2005, Li et al. (Li et al., 2005) também desenvolveram um trabalho usando esta técnica de geração de características e propuseram um modelo Gaussiano para predição 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 estas características, locais e globais, eles encontraram uma sensibilidade 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/~lig1/software/TISglobal/TISglobal.htm/#ref1>.

Tzanis e colaboradores (Tzanis e Vlahavas, 2006) desenvolveram uma metodologia para construir uma sistema de vários classificadores para predição do SIT. Eles utilizaram as seguintes características:

1. “X<sub>up</sub>”: o número de aminoácidos X na região *upstream*.
2. “X<sub>down</sub>”: o número de aminoácidos X na região *downstream*.
3. “X<sub>{up-down}</sub>”: a diferença entre o número de aminoácidos X na região *upstream* e *downstream*, respectivamente.
4. “k-X<sub>{up-pos}</sub>”: 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]<sub>up</sub>”: característica booleana que analisa a presença dos nucleotídeos “A” ou “G” na posição -3.
6. “pos+4[G]<sub>down</sub>”: característica booleana que analisa a presença do nucleotídeo “G” na posição +4.
7. “ATG<sub>up</sub>”: característica booleana que analisa a presença de um códon ATG na região *upstream* e em fase com o SIT.
8. “stop<sub>down</sub>”: característica booleana que analisa a presença de algum dos 3 *stop-codon* (TAA, TAG e TGA) na região *downstream* 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 2.6

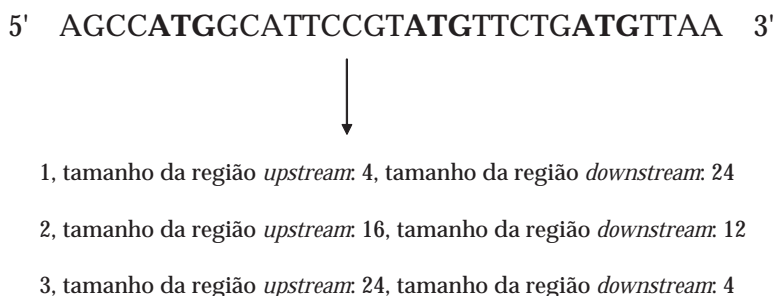


Figura 2.6: Todos os ATGs de uma determinada molécula são extraídos e os tamanhos das regiões *upstream* e *downstream* são calculados para separação entre os subgrupos. (Imagem de Tzanis et al., adaptada pelo autor.)

2. cada ATG candidato, encontrado no passo anterior, é agrupado 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 2.7.

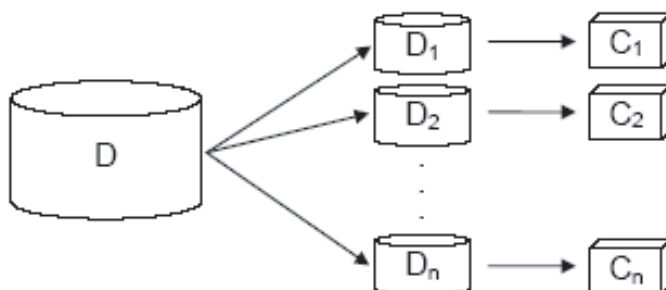


Figura 2.7: A base de dados inicial  $D$  é dividida em um número de base de dados menores  $D_i$  e, finalmente, um classificador é construído separadamente para cada  $D_i$ . (Imagem de Tzanis et al.)

3. para cada um dos ATGs candidatos, os valores das características consideradas (e 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 con-

struído para cada subconjunto de dados.

Segundo Tzanis (conversas por email), eles utilizaram janelas de 99 nucleotídeos na região *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  $m$  na região *upstream* e  $n$  na região *downstream*, eles utilizaram-se do seguinte artifício:  $m = \min(\text{tamanho\_upstream}, 99)$ ,  $n = \min(\text{tamanho\_downstream}, 99)$ . No total, foram 927 características extraídas.

Tzanis e colaboradores (Tzanis e Vlahavas, 2006) testaram 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 trabalho desenvolvido posteriormente, Tzanis e colaboradores (Tzanis e Vlahavas, 2006) adicionaram características baseadas em propriedades químicas dos aminoácidos. Estas características são as seguintes:

1. “X<sub>up</sub> e X<sub>down</sub>”: o número de aminoácidos X na região *upstream* e *downstream*.
2. “X<sub>up-down</sub>”: a diferença entre o número de aminoácidos X na região *upstream* e *downstream*, respectivamente.
3. “k-X<sub>up-pos</sub> e k-X<sub>down-pos</sub>”: o número de nucleotídeos X na k-ésima posição dos códons das regiões *upstream* e *downstream* que estão em fase ( $k \in \{1, 2, 3\}$ ).
4. “pos<sub>-3k</sub> e pos<sub>-3(k+1)</sub>”: indica a presença de aminoácidos nas posições que estão em fase nas regiões *upstream* e *downstream* ( $k \geq 1$ ), respectivamente.
5. “hidrofóbico<sub>up</sub> e hidrofóbico<sub>down</sub>”: o número de aminoácidos hidrofóbicos nas regiões *upstream* e *downstream*, respectivamente.
6. “hidrofilico<sub>up</sub> e hidrofilico<sub>down</sub>”: o número de aminoácidos hidrofílicos nas regiões *upstream* e *downstream*, respectivamente.
7. “acíclico<sub>up</sub> e acíclico<sub>down</sub>”: 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 ou não 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 *downstream* e que esteja em fase com o SIT.

Baseado nestas características, eles encontraram um desempenho melhor do que haviam encontrado em resultados anteriores.

### 2.1.5 Reconhecimento por Função de Discriminante Linear

O programa ATGpr desenvolvido por Salamov et al. (Salamov et al., 1998) usa uma função de discriminante linear que combina algumas características estatísticas derivadas da seqüência. Dentre as características observadas por eles, destacam-se: a presença da “C” na região *upstream*, a presença de um outro ATG na região

*upstream* e em fase com o ATG, a probabilidade da presença de um sinal peptídeo na região downstream do SIT, dentre outras características consideradas importantes na predição 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 predição de clones de cDNA. ATGpr está disponível em <http://www.hri.co.jp/atgpr/>.

### 2.1.6 Conclusões do Capítulo

Neste capítulo foram apresentados os principais métodos de predição de SIT existentes na literatura. Os métodos foram divididos segundo à abordagem utilizada: Redes Neurais, Support Vector Machine, Extração de características e Função de Discriminante Linear.

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

## Materiais e Métodos

---

---

Este capítulo destina-se à apresentação dos métodos propostos neste trabalho. As oito etapas, a seguir, foram utilizadas para que a predição 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 da inferência indutiva e transdutiva;
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 serão validadas, além da utilização do modelo de escaneamento utilizado por Hatzigeorgiou (Hatzigeorgiou, 2002), onde se valida apenas as amostras até o SIT.

As seções seguintes descreverão cada uma destas fases.

### 3.1 Bases de dados utilizadas

Uma vez que a maioria dos trabalhos relacionados à Predição de SIT trabalha com a base de dados de Petersen e Nielsen (Pedersen e Nielsen, 1997), apresentada na seção 2.1.1, 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 destes dois autores foi gerada em 1997 quando ainda não tínhamos um número muito grande de seqüências. Além disto, 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, buscou-se neste trabalho analisar seqüências de um outro banco de dados mais curado, o *RefSeq* do NCBI para os seguintes organismos: *Danio rerio*(peixe), *Drosophila melanogaster*(mosca), *Homo sapiens*(homem), *Mus musculus*(camundongo) e *Rattus norvegicus*(rato).

As seqüências de referência (*RefSeq*) fornecem uma coleção não redundante de seqüências de DNA, RNA e proteínas. No *RefSeq*, a redundância do GenBank é retirada.

As principais características do *RefSeq* são:

- Não redundância;
- Seqüências de nucleotídeos e proteínas explicitamente linkados;



- Validação dos dados e consistência no formato dos arquivos (normalmente dois caracteres, seguido pelo *underscore* e seis dígitos);
- Curado pelo *staff* do NCBI e colaboradores, com *status* de revisão, indicados em cada registro.

As seqüências de referência existem 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 destas 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. Este arquivo contém informações tais como organismo, número de acesso, nível de confiança, posição de início do CDS, dentre outras. (Imagem do autor.)

Através desta figura, pode-se verificar que o organismo em questão é o *Homo sapiens* (seqüência revisada), 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 extraído, a partir do *Refseq*, para cada organismo, por grau de inspeção.

Tabela 3.1: Número de moléculas disponíveis no *Refseq* para os organismos investigados. Download em 11/02/07.

	<i>Danio rerio</i>	<i>Drosophila melanogaster</i>	<i>Mus musculus</i>	<i>Homo sapiens</i>	<i>Rattus norvegicus</i>
<i>Reviewed</i>	3	20109	210	11226	1240
<i>Provisional</i>	5297	0	13464	5964	7620
<i>Predicted</i>	5334	0	2593	1484	915
<i>Validated</i>	9	0	3967	5702	368
<i>Model</i>	20792	0	26255	13574	29970
<i>Inferred</i>	11	0	35	39	3
<b>Total</b>	<b>31446</b>	<b>20109</b>	<b>46524</b>	<b>37989</b>	<b>40116</b>

A partir dos arquivos, no formato apresentado na figura 3.1, seqüências positivas e negativas foram extraídas através de um *parsing*, 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. Esta escolha foi motivada por um estudo realizado por Pedersen e Nielsen (descrito na seção 2.1.2) 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 esta suposição pode estar correta.

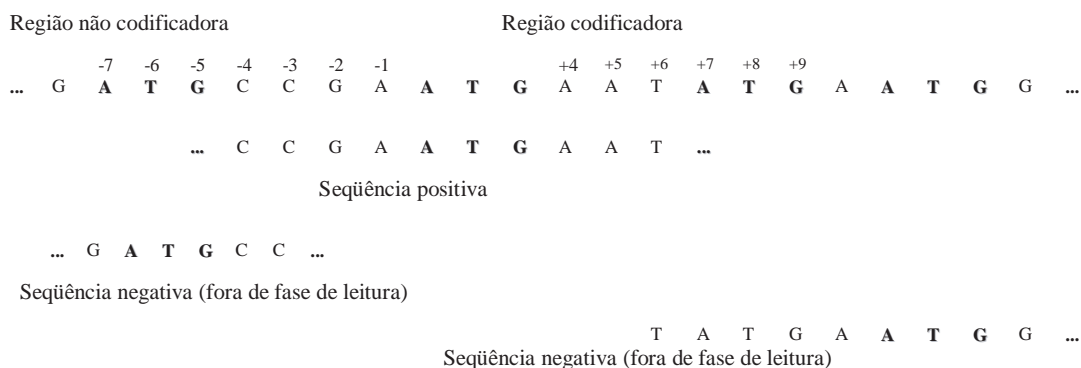


Figura 3.2: Construção das seqüências positivas e negativas usando-se uma janela de 24 nucleotídeos com códon ATG começando-se na décima terceira posição. As seqüências negativas foram obtidas fora da fase de leitura. A trinca ATG é desconsiderada da seqüência. (*Imagem do autor.*)

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.

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 predição de SIT, inclusive, adota este 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 Kokaz (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 apresentará 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). Desta forma, todas as seqüências contendo CDS abaixo deste número foram desconsideradas.

A tabela 3.2 apresenta o número total de seqüências negativas e positivas extraído 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.

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

Percebe-se claramente, por esta tabela, que este 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 fôssemos extrair todas as seqüências da molécula, o problema seria ainda mais desbalanceado.

A tabela 3.3 apresenta o número de seqüências negativas que estão nas regiões *upstream* e *downstream* de cada ATG; além disto que estão na mesma fase de leitura ou fora de fase, para cada organismo analisado, para cada grau de inspeção.

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

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

Até a data corrente (10 de junho de 2007), existem 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). Desta forma, inicialmente o SVM possuía 96 entradas ( $24 \text{ bases} \times 4 \text{ bits} = 96$ ) e uma saída (1 ou 0). Ou seja, a saída é 1 se a seqüência contém o códon ATG inicializador da tradução ou 0, no caso de a seqüência não conter este inicializador.

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. 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 esta nova codificação  $(24 \div 3) \times 6 \text{ bits} = 48$  entradas, contra as 96 entradas utilizando-se a codificação por base. Assim, temos uma redução de 50% do número das entradas. Os resultados apresentados no capítulo 4 mostram que esta 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).

Este 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 prever a classe minoritária com precisão na presença de dados com classes desbalanceadas. Muito freqüentemente o sistema irá fazer uma boa predição para a classe majoritária e uma predição 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 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* aumenta a probabilidade de ocorrer *overfitting* (sobreajuste ou a incapacidade de classificar itens semelhantes como pertencentes à mesma classe), uma vez que estes fazem cópias exatas dos exemplos das classes minoritárias. Desta forma, um classificador, por exemplo, pode construir regras que são aparentemente precisas, mas que cobre apenas os exemplos replicados. Por outro lado, o problema maior relacionado aos métodos *under-sampling* é que estes podem descartar amostras significativas durante o processo de indução.

Entretanto, existem métodos heurísticos 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 predição do SIT

O problema de predição 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 28, em média, conforme números apresentados na tabela 3.2.

Este fato pode gerar um classificador que prediz com alta precisão a classe dos não-SITs, enquanto falha na predição dos SITs. Para resolver este problema, neste trabalho, optou-se por utilizar um método de *over-sampling* utilizando o algoritmo Smote para replicação das amostras que codificam proteínas. Este algoritmo, descrito em Nitesh (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 deste método:



Tabela 3.4: Descrição do algoritmo SMOTE

---



---

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.      $N = (\text{int})(N/100)$  (\* 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.     newindex: 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**  $i \leftarrow 1$  to  $T$
15.         Compute  $k$  nearest neighbors for  $i$ , and save the indices in the nnarray
- Populate ( $N, i, \text{nnarray}$ )
16.     **endfor**
- Populate ( $N, I, \text{nnarray}$ ) (\* Function to generate the synthetic samples. \*)
17.     **while**  $N \neq 0$
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  $\leftarrow 1$  to numattrs
20.             Compute: dif = Sample[nnarray][nn][attr]-sample[i][attr]
21.             Compute: gap = random number 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.

---



---

Este algoritmo foi implementado em C++ e foi responsável por deixar as classes positivas e negativas com o mesmo número de amostras. No capítulo 4 será apresentado um quadro comparativo mostrando o efeito obtido pelo balanceamento das classes.

### 3.5 *Support Vector Machine - SVM*

SVM pode ser caracterizado como um algoritmo de aprendizado de máquina capaz de resolver problemas de classificação lineares e não lineares. A ideia principal da classificação por vetor de suporte é separar exemplos com uma superfície de decisão linear e maximizar a margem de separação entre as classes a serem classificadas.

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 apresentadas à máquina de forma que seus parâmetros sejam ajustados (Carvalho et al., 2002).

Seja o conjunto de treinamento  $\{x_i, y_i\}_i^N = 1$ , com cada vetor de entrada  $x_i \in \mathbb{R}_n$  e saída binária correspondente  $y_i \in \{-1, +1\}$ . Dado um vetor de entrada  $x$ , a saída da SVM é representada por  $f(x)$ . A SVM realiza um mapeamento não linear dos dados em um espaço de dimensão mais elevada.

Nesta 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  $f(x) = 0$  criada pela SVM é representada por:

$$\omega^t \varphi(x) + b = 0 \quad (3.1)$$

onde  $\omega \in \mathfrak{R}_n$  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 este espaço for ordem  $n$ , a superfície é um hiperplano  $(n-1)$ -dimensional. A superfície de decisão divide o espaço de características em dois sub-espacos, um para cada classe.

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

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

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

$$y_i [\omega^t \varphi(x_i) + b \geq 1] \quad (3.3)$$

para o padrão de entrada  $i$ .

O processo de treinamento de uma SVM consiste na obtenção de valores para os pesos  $\omega$  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<sup>light</sup>* implementada pelo T. Joachims (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<sup>light</sup>* 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 experimentos, visto que os dados deste trabalho pertencem a um espaço de dimensão 48 e o número de dados 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, junto à teoria do aprendizado estatístico (Vapnik, 1998) com a idéia 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 predição, em que as amostras não estão classificadas. O objetivo é classificar os dados pertencentes ao conjunto de predição. Treinando o classificador com estes dois conjuntos de dados, é possível classificar os dados do conjunto de predição 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).

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 predição, conforme ilustrado na Figura 3.3.

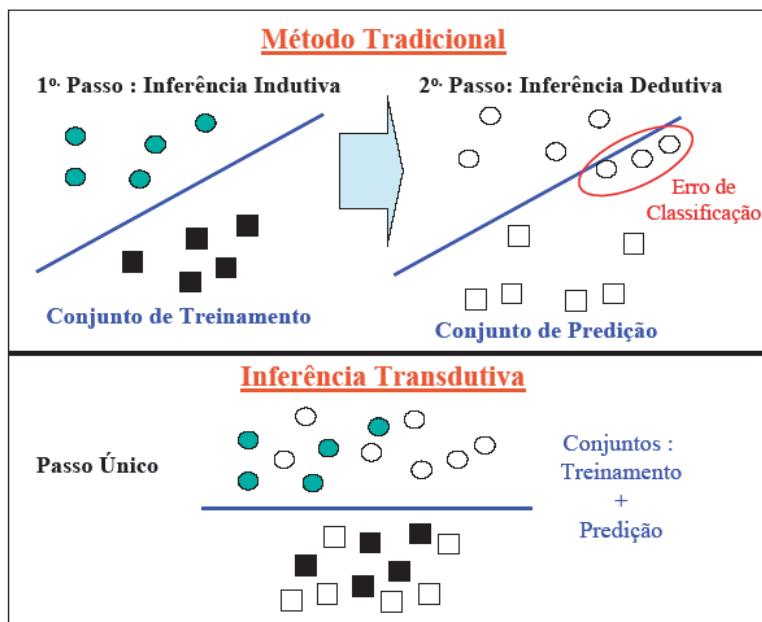


Figura 3.3: Diferença do método tradicional (indutivo-dedutivo) para a inferência transdutiva. No método tradicional, utilizando apenas o conjunto de treinamento, o hiperplano é construído para separar as duas classes (bolas verdes x quadrados pretos). Num segundo passo, este hiperplano é utilizado para deduzir (predizer) a classificação das amostras do conjunto de predição. Como ilustração, é mostrada a classe desconhecida a que pertencem as amostras do conjunto de predição, 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 os princípios da inferência transdutiva, o hiperplano é encontrado utilizando no treinamento os dois conjuntos de dados, treinamento e predição, sendo que os dados de predição ainda não estão classificados. Assim a classificação dos dados do conjunto de predição é feita em um único passo. Observa-se que agregando os dados do conjunto de predição ao treinamento do algoritmo, consegue-se um melhor desempenho. (Imagem de Semolini e Zuben.)

O passo indutivo consiste em descobrir a dependência funcional entre as variáveis de entrada-saída; enquanto o passo dedutivo utiliza esta dependência funcional para avaliar a saída dos pontos de interesse, ou seja, classificá-los.

Através da inferência indutiva, o aprendizado do classificador irá induzir a função de decisão de tal modo a minimizar a frequência do erro no conjunto de treinamento.

Porém, na maioria dos casos, o interesse não está diretamente ligado à escolha da função de decisão, e sim na classificação do conjunto de predição com o menor erro possível, o que constitui o princípio da inferência transdutiva.

Os conceitos da inferência transdutiva são ainda mais úteis quando o conjunto de treinamento consiste em apenas um pequeno número de amostras já classificadas e pertencentes a um espaço de alta dimensão. Para este caso, a solução do problema via um classificador que emprega o método indutivo vai estar demasiadamente suscetível ao sobreajuste (*overfitting*) dos dados do conjunto de treinamento. A idéia principal é explorar os dados ainda não classificados para gerar informação adicional sobre o problema, a qual será utilizada com o propósito de melhorar a generalização e aumentar o desempenho do classificador.

Há muitas aplicações práticas em que os dados não classificados são muitos e a quantidade de dados previamente classificados é pequena. Isto pode acontecer pelo motivo da classificação dos mesmos ser muito custosa, difícil ou demorada de ser obtida.

### 3.6.1 Inferência transdutiva e o problema de predição do SIT

Considerando o modelo de escaneamento do ribossomo apresentado na figura 1.1 da seção 1, temos que apenas os ATGs que estão na região *upstream* do SIT e o próprio SIT possuem classificação. Ou seja, o ribossomo não identificou o(s) primeiro(s) ATGs na seqüência como sendo o 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 ATGs que estão na região *downstream* do SIT.

Além disto, existe ainda a possibilidade de um ATG (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, este problema de predição de SIT possui características intrínsecas de inferência transdutiva e será, 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 predição, 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 classificados como negativos (não SIT) e, portanto, utilizados na inferência indutiva.

Em nossos experimentos, foi utilizada a versão TSVM (SVM transdutivo) implementada pelo T. Joachims (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 3.8, mede a proporção de predições, para SITs e não-SITs, que está correta.

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

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.

$$\text{Precisão} = 100 * \frac{VP}{VP + FP} \quad (3.5)$$

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 por:

$$\text{Sensibilidade} = 100 * \frac{VP}{VP + FN} \quad (3.6)$$

e

$$\text{Especificidade} = 100 * \frac{VN}{VN + FP} \quad (3.7)$$

A título de comparação, visto que alguns autores utilizam o conceito de acurácia ajustada, este trabalho também utilizou-se desta medida:

$$\text{Acurácia ajustada} = \frac{\text{Sensibilidade} + \text{Especificidade}}{2} \quad (3.8)$$

Todos os resultados apresentados no capítulo 4 são baseados nestas medidas, usando-se o conceito de validação cruzada. A subseção seguinte formaliza, portanto, este conceito.

### 3.7.1 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 3.4 mostra um esquema ilustrando o método de validação cruzada com  $k$ -dobras, mais precisamente uma validação cruzada com 4-dobras.



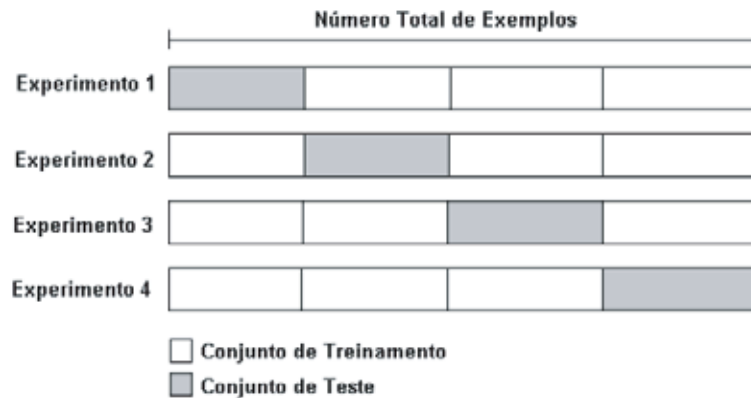


Figura 3.4: Esquema mostrando o exemplo do método de validação cruzada com 4-dobras.

A estimativa de Erro seria dada por:

$$\text{Erro} = \frac{1}{2} \sum_{i=1}^k E_i \quad (3.9)$$

Onde é  $0 \leq E_i \leq 1$  e representa a taxa de erro para a dobra e  $k$  é o número de dobras.

Todos os resultados deste trabalho foram obtidos a partir da validação cruzada com 5 dobras (*5-fold cross validation*). Ou seja, dividiu-se o conjunto em cinco partes; treinou-se com quatro(4) partes e testou-se com a outra. Isso foi repetido para as cinco partes e finalmente calculou-se a média e desvio padrão dos resultados obtidos.

Além das cinco medidas de desempenho, uma outra medida muito utilizada em problemas desbalanceados é a análise da curva ROC. A subseção seguinte descreve este conceito, visto que ele também será utilizado neste trabalho.

### 3.7.2 Curva ROC

Um gráfico ROC<sup>1</sup> 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, neste caso, é 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 3.5 é ilustrado um gráfico ROC.

---

<sup>1</sup>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.

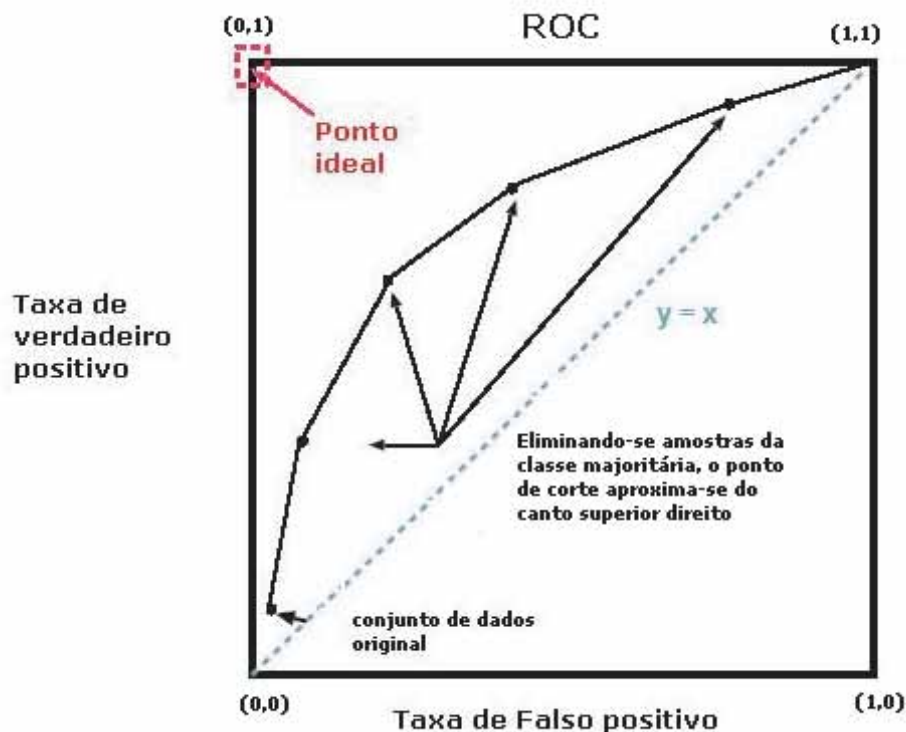


Figura 3.5: Ilustração da curva ROC e o efeito da técnica de *under-sampling*. Eliminando-se amostras da classe majoritária (negativa), o desempenho move-se do canto inferior esquerdo para o canto superior direito. (Imagem de Nitesh et al., adaptada pelo autor.)

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 o 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).

A partir de um gráfico ROC é possível calcular uma medida geral de qualidade,

a área sob a curva (AUC)<sup>2</sup>. 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).

### 3.8 *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 predição do SIT, seja a predição do verdadeiro SIT, dado uma molécula de mRNA, ao contrário da validação de todos os ATGs desta molécula (Zeng et al., 2002). Desta forma, apenas uma predição seria necessária. Isto corresponde ao modelo de escaneamento de Kozak (Kozak, 1989).

Este modelo de predição de SIT, introduzido por Agarwal e Bafna (Agarwal e Bafna, 1998) e também utilizado em Hatzigeorgiou (Hatzigeorgiou, 2002) e outros autores (Zeng et al., 2002) e Huiqing et al (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 este modelo. Em situações onde não se tem o mRNA completo, por exemplo, a validação de toda a molécula se faz necessária.

### 3.9 *Conclusões do capítulo*

Neste capítulo foram apresentados os métodos utilizados neste trabalho, mostrando as bases utilizadas; a forma de extração das seqüências positivas e negativas; a codificação empregada; a apresentação do método de balanceamento utilizado; a escolha do classificador, juntamente com a sua justificativa; a explicação da abordagem transdutiva aplicada ao problema de predição de SIT; as medidas de desempenhos utilizadas neste trabalho e, finalmente, a forma de validação das seqüências, validando todas as seqüências da molécula ou seguindo o modelo de escaneamento

---

<sup>2</sup>Area under the ROC curve

utilizado por Hatzigeorgiou (Hatzigeorgiou, 2002).

O capítulo 4 apresenta, então, os resultados obtidos a partir destes métodos.

---

## Resultados e Discussões

---

---

Neste capítulo são apresentados os resultados obtidos com os métodos descritos no capítulo 3, empregados nas bases de dados descritas nos capítulos 2.1.1 e 3. Estes resultados são analisados com a finalidade de verificar se os objetivos apresentados no capítulo 1 foram atingidos.

Está organizado da seguinte forma: a seção 4.1 apresenta uma comparação entre os diversos métodos propostos neste trabalho (descrito no capítulo 3). Uma vez que estes critérios influenciam diretamente no tempo de processamento, apenas as menores bases de dados serão utilizadas (*Mus musculus reviewed* com 182 seqüências positivas e 7130 seqüências negativas e *Rattus norvegicus reviewed* com 42 seqüências positivas e 927 seqüências negativas). 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 4.5 apresenta 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 abordagens*

Os resultados apresentados neste capítulo seguem a metodologia utilizada no capítulo 3. Desta forma, as subseções seguintes apresentam os resultados em função dos seguintes critérios:

1. forma de se obter as seqüências positivas (que codificam proteínas - da classe SIT) e as negativas (que não codificam proteínas - da classe não-SIT);
2. comparação do desempenho a partir de diferentes tamanhos de janelas;
3. escolha da codificação utilizada;
4. escolha do método de balanceamento que pudesse balancear as classes de SIT e não-SIT de forma eficiente;
5. a escolha do classificador que oferecesse um bom desempenho nesta classificação;
6. utilização da inferência indutiva e transdutiva;
7. a forma de validação das seqüências, onde todas as seqüências da molécula contendo ATG são validadas, além da utilização do modelo de escaneamento utilizado por Hatzigeorgiou (Hatzigeorgiou, 2002), onde se valida apenas as seqüências até o SIT;

As subseções 4.1.1, 4.2, 4.1.3 e 4.1.4 utilizam o conceito de escaneamento do ribossomo apresentado na seção 3.8, onde apenas uma predição é realizada por molécula, desconsiderando-se, assim, todos os ATGs que estão à direita do SIT (na região 3'), além da validação de todos os ATGs da molécula.

#### 4.1.1 Seleção de seqüências não-SITs usadas no treinamento

Um problema muito importante na predição do SIT é a seleção correta das seqüências consideradas negativas. Sabemos que, biologicamente, o ribossomo pode considerar um CDS mais para frente ou mais para trás. Ou seja, podemos ter um ATG codificante na região *upstream* ou *downstream* do ATG considerado verdadeiro.

Neste sentido, se considerarmos todos os ATGs que teoricamente são negativos como sendo realmente negativos podemos piorar o resultado da classificação, pois muitas destas 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. Na realidade, algumas proteínas podem ter dois SITs em fase. Esta proposta, além de melhorar o resultado da classificação, reduz enormemente o desbalanceamento entre as classes, fato que foi discutido na seção 3.4.1.

Os resultados apresentados na tabela 4.1 validam esta metodologia de eliminar do treinamento as seqüências negativas que estão na mesma fase de leitura do SIT.



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	
<i>Mus musculus Reviewed</i>				
	Média	DP	Média	DP
Acurácia	95,95	0,39	93,63	0,30
Acurácia ajustada	96,44	0,32	92,8	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
<i>Rattus Norvegicus Reviewed</i>				
	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

A eficiência do classificador é claramente reduzida usando-se seqüências negativas que estão na mesma fase de leitura do SIT, conforme pode ser visto principalmente pela taxa de sensibilidade e precisão. 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.

#### 4.1.2 Tamanho da janela

Uma das primeiras preocupações deste trabalho foi escolher o tamanho ideal de janela para se fazer a predição do SIT com uma alta acurácia. Como já citado anteriormente, os principais autores da literatura utilizam janelas de 200 nucleotídeos. No entanto, este número é proibitivo quando se trabalha com bases muito grandes, como é o caso da *Drosophila melanogaster reviewed*, por exemplo, que possui 15893 seqüências positivas e 379562 seqüências negativas (números apresentados na tabela 3.2 da 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. 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. 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.

	Janela 9+3		Janela 12+12		Janela 30+30		Janela 99+99	
<i>Mus musculus Reviewed</i>								
	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
<i>Rattus Norvegicus Reviewed</i>								
	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	0,71	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

Inicialmente, buscou-se testar a janela sugerida por Kozak, com apenas nove (9) bases na região *upstream* e três (3) na região *downstream* do ATG. Os estudos apresentam análises considerando-se apenas uma (1) base na região *dowstream*, mas devido à codificação proposta neste trabalho (codificação por trinca), três bases foram utilizadas. Vários outros testes foram realizados com diferentes tamanhos de janela, mas os mais significativos estão apresentados nesta tabela.

A título de exemplo, 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 considerarmos as bases de todos os organismos.

Percebe-se, por esta tabela, que seqüências de 9 nucleotídeos na região *upstream* e 3 na região *downstream* oferecem uma acurácia e sensibilidade comparáveis às obtidas por meio de janelas maiores, apesar de as medidas de precisão e especifici-

dade terem sido 6,83% e 4,65%, respectivamente, menores se considerarmos janelas de 30 bases na região *upstream* e *downstream* do ATG (para o organismo *Mus musculus*).

Por outro lado, vê-se que com 12 bases nas regiões *upstream* e *downstream* do ATG pode-se alcançar um desempenho comparável aos obtidos por 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 significativos.

Os resultados obtidos por janelas de 30 bases nas regiões *upstream* e *downstream* é bastante promissor e para o *Mus musculus* alcança resultados melhores do que aqueles obtidos por janelas de 99 bases nestas regiões.

É importante ressaltar que em seqüências onde as moléculas têm poucas bases na região 5' do CDS, como é o caso do *Rattus norvegicus* que teve 96,6% das seqüências desconsideradas por possuir o CDS menor que 13 (considerando-se janelas de 12 bases na região 5', conforme descrito na seção 3.2).

Além disto, 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 menor com janelas de tamanho 12+12 do que usando janelas de tamanho 99+99. Isto pode ser fundamental quando se trabalha com bases muito grandes, como é o caso das bases deste trabalho.

Ainda em relação ao efeito do tamanho da janela, um outro teste foi realizado para avaliar o efeito deste critério ao 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. A validação foi realizada considerando-se todos os ATGs da molécula.

	Janela 9+3		Janela 12+12		Janela 30+30		Janela 99+99	
<i>Mus musculus Reviewed</i>								
	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
<i>Rattus Norvegicus Reviewed</i>								
	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

Por esta tabela, percebe-se que a precisão é altamente afetada pelo tamanho da janela. Para o organismo *Rattus norvegicus*, a taxa de precisão é significativamente maior, considerando-se janelas de 99 bases nas regiões *upstream* e *downstream*. Para o *Mus musculus*, esta taxa continua muito baixa, mas não há uma diferença significativa 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. Desta 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 gasto.

Enquanto a codificação binária por base, utilizando-se 4 bits, é a mais comu-

mente utilizada na literatura, foi proposto neste trabalho uma codificação alternativa que codifica a seqüência por trinca (tri-nucleotídeos), ao invés da codificação por base. Esta 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 este 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 estas duas opções de codificação. Novamente, por limitação de tempo de processamento, serão apresentados apenas os resultados obtidos para o *Mus musculus* e *Rattus norvegicus reviewed*.

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

	Codificação com trinca		Codificação base a base	
<i>Mus musculus Reviewed</i>				
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
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
<i>Rattus Norvegicus Reviewed</i>				
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
Acurácia	99,80	0,00	98,56	0,36
Acurácia ajustada	97,58	0,00	96,67	0,98
Precisão	97,56	0,00	98,36	0,89
Sensibilidade	95,24	0,00	94,36	1,02
Especificidade	99,93	0,00	98,98	0,06

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 esta 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 predição de SIT em um computador *Desktop* simples.

Para se ter uma idéia, usando-se esta codificação por trinca, o tempo de processamento para a base do *Mus musculus* foi reduzido em 19.09 vezes. Este ganho, adicionado ao ganho obtido pelo tamanho da janela, pode fazer uma grande diferença em processamento de bases muito grandes.

#### 4.1.4 Desempenho em função do método de balanceamento

Como discutido na seção 3.4.1, o problema de predição 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 os classificadores tendem a classificar qualquer seqüência como pertencente à classe majoritária (neste caso, a classe 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, a tabela 4.5 apresenta os resultados obtidos com e sem balanceamento. Percebe-se claramente que o método Smote aumenta significativamente as taxas de sensibilidade e acurácia.

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

	Com balanceamento (Smote)		Sem balanceamento	
<i>Mus musculus Reviewed</i>				
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
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
<i>Rattus Norvegicus Reviewed</i>				
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
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

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*. Estas mesmas medidas vão 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 é tratar o problema de predição de SIT como um problema transdutivo. Ou seja, neste 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. E as seqüências que estão na mesma fase de leitura? O que foi feito delas? A abordagem transdutiva, descrita na seção 3.6, foi utilizada, então, neste contexto para tentar descobrir a classificação destas seqüências. Para serem utilizadas, todas as seqüências positivas são consideradas como sendo +1, as negativas são -1 e todas as restantes são consideradas “unlabelled ou não anotados”. Ou seja, durante o treinamento, estas seqüências também são consideradas; no entanto, elas são adicionadas ao conjunto de treinamento sem classificação.

A figura 4.1.5(a) apresenta os resultados obtidos considerando a abordagem indutiva e a figura 4.1.5(b) apresenta os mesmos resultados considerando-se a abordagem transdutiva. A validação neste caso é feita considerando-se o modelo de escaneamento, discutido anteriormente. Ou seja, desconsiderando-se os ATGs da região *downstream* do SIT.

	Janela 9+3		Janela 12+12		Janela 30+30		Janela 99+99	
<i>Mus musculus Reviewed</i>								
	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
<i>Rattus Norvegicus Reviewed</i>								
	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	0,71	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

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

	Janela 9+3		Janela 12+12		Janela 30+30		Janela 99+99	
<i>Mus musculus Reviewed</i>								
	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	0,12	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
<i>Rattus Norvegicus Reviewed</i>								
	Média	DP	Média	DP	Média	DP	Média	DP
Acurácia	97,65	0,44	99,81	0,46	99,16	0,23	99,65	0,85
Acurácia ajustada	97,61	0,49	99,95	0,01	98,13	0,18	99,69	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	99,38	1,51
Especificidade	98,00	0,00	99,89	0,01	98,62	0,87	100,00	0,00

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

Figura 4.1: Comparação entre a abordagem indutiva e transdutiva - Validação das seqüências até o SIT.

Por esta figura, percebe-se a abordagem transdutiva conseguiu melhorar os resultados em todos os casos.

A figura 4.1.5 apresenta os mesmos resultados validando-se todas as seqüências da molécula contendo ATGs.

	Janela 9+3		Janela 12+12		Janela 30+30		Janela 99+99	
<i>Mus musculus Reviewed</i>								
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
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
<i>Rattus Norvegicus Reviewed</i>								
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
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

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

	Janela 9+3		Janela 12+12		Janela 30+30		Janela 99+99	
<i>Mus musculus Reviewed</i>								
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
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
<i>Rattus Norvegicus Reviewed</i>								
	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>	<b>Média</b>	<b>DP</b>
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

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

Figura 4.2: Comparação entre a abordagem indutiva e transdutiva - Validação de todos os ATGs da molécula.

Observa-se, neste caso, que a abordagem transdutiva melhora significativamente a taxa de precisão em todos os casos. Isso comprova a relevância desta abordagem para o problema de predição 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% foi para 92,84%.

## 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 a partir da base de dados fornecida pelo Petersen e Nielsen e pelas outras bases *refseq* (*Homo sapiens*, *Drosophila melanogaster* e *Danio rerio*) propostas.

Como descrito na seção 2.1.1, a base de dados do 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 neste caso. Ou seja, foram extraídas seqüências de 24 bases (12+12), pegando-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 2.1.1, os autores completaram as seqüências com “N” (significando base “não conhecida”) quando não se tinha o tamanho desejado. Ou seja, ao extraírem as seqüências, se eles não possuem 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 4.6 apresenta os resultados obtidos para esta base de dados completando as seqüências com “N” e sem completá-las, ou seja, descartando as seqüências que não continham 12 bases nas regiões *upstream* ou *dowstream*. Neste caso, a codificação utilizada foi exatamente a codificação encontrada na literatura: A=00001, C=00010, G=00100, T=01000 e N=10000, uma vez que a codificação por

trinca adotada neste trabalho constrói trincas a partir das bases A, C, G e T, apenas.

Tabela 4.6: Acurácia, acurácia ajustada, precisão, sensibilidade e especificidade para a base de dados de Petersen e Nielsen.

	Janela 12+12 sem N	Janela 12+12 com N
Acurácia	92,79	99,63
Acurácia ajustada	84,99	94,38
Precisão	86,98	98,75
Sensibilidade	86,65	93,98
Especificade	83,32	94,78

Das 3212 seqüências positivas, 58 não continham 12 bases na região *upstream* ou *downstream* do ATG e das 10190 seqüências negativas, 987 não possuem o tamanho desejado. Assim, todas elas precisaram ser completadas com “N”. Esta é uma das razões pela qual acreditamos que o resultado tenha sido melhor completando-se as seqüências com N. Ou seja, o classificador possivelmente pode identificado este padrão.

No entanto, como já foi falado durante todo este trabalho, esta base foi utilizada apenas para servir como comparação, visto que praticamente todos os trabalhos relacionados ao tema utilizam-na. Mas sabemos que as seqüências do *Genback* não são as melhores existentes. Assim, o foco principal deste trabalho foi relativo a bases *refseq*.

Desta forma, as figuras 4.7, 4.8, 4.9, 4.10 e 4.11 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” indica que a base de dados é insuficiente (normalmente com menos de 11 seqüências). “BDNE” indica que não existe base de dados para aquele grau e inspeção e “ND” significa que os dados não foram determinados até o presente momento.

Tabela 4.7: Desempenho do classificador para o *Danio rerio* utilizando-se a abordagem indutiva e transdutiva, validando as seqüências até o SIT.

<i>Danio rerio - inferência indutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	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		
<i>Danio rerio - inferência transdutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	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		

Tabela 4.8: Desempenho do classificador para o *Drosophila melanogaster* utilizando-se a abordagem indutiva e transdutiva, validando as seqüências até o SIT.

<i>Drosophila melanogaster - inferência indutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	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										
<i>Drosophila melanogaster - inferência transdutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Acurácia	ND		BDNE		BDNE		BDNE		BDNE		BDNE	
Acurácia ajustada												
Precisão												
Sensibilidade												
Especificidade												



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

<i>Mus musculus - inferência indutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	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		
<i>Mus musculus - inferência indutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Acurácia	98,37	0,15	ND	ND	ND	ND	98,61	0,23	ND	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				

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

<i>Homo sapiens - inferência indutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	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	BDI	
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		
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		
<i>Homo sapiens - inferência transdutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP	Média	DP
Acurácia	ND		94,59	3,02	99,53	0,19	96,91	1,78	86,32	5,23	BDI	
Acurácia ajustada			96,05	3,30	99,60	1,35	98,09	1,45	79,86	2,91		
Precisão			96,40	3,94	96,34	0,27	94,65	2,40	78,98	6,45		
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		

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

<i>Rattus norvegicus - inferência indutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	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	BDI	
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		
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		
<i>Rattus norvegicus - inferência transdutiva</i>												
	<i>Reviewed</i>		<i>Provisional</i>		<i>Predicted</i>		<i>Validated</i>		<i>Model</i>		<i>Inferred</i>	
	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	BDI	
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		
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		

Como descrito na seção 3.6, na inferência indutiva padrão, 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).

Por estas figuras, percebe-se a abordagem transdutiva realmente é eficiente para este 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 rodando deste o dia 29 de maio e até hoje (dia 09 de julho) não terminou.

### 4.3 *Curva ROC para os organismos Mus musculus e Rattus Norvegicus*

As curvas apresentadas nas figuras 4.3 e 4.4 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.

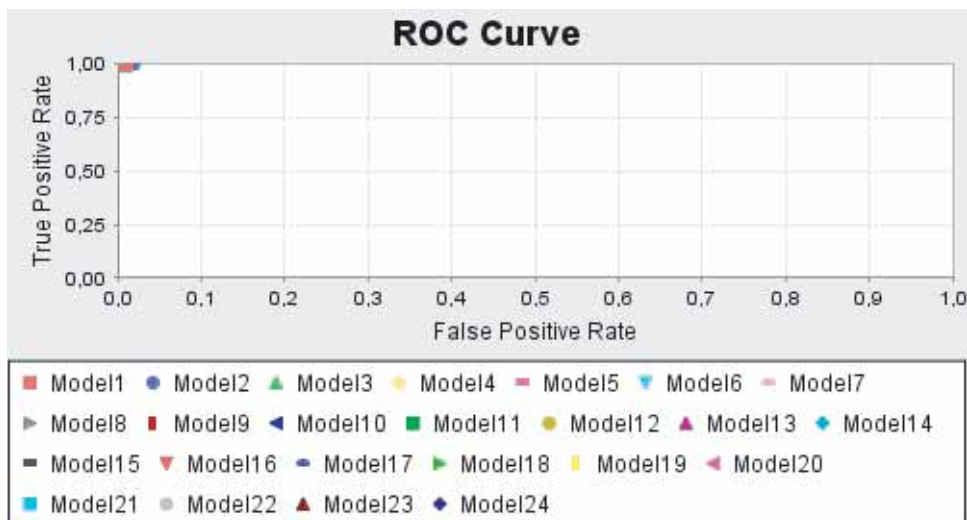


Figura 4.3: Curva ROC para o *Mus musculus* com o conjunto balanceado de seqüências. Observe que o parâmetro J não influencia os resultados, visto que o conjunto de dados já está balanceado.

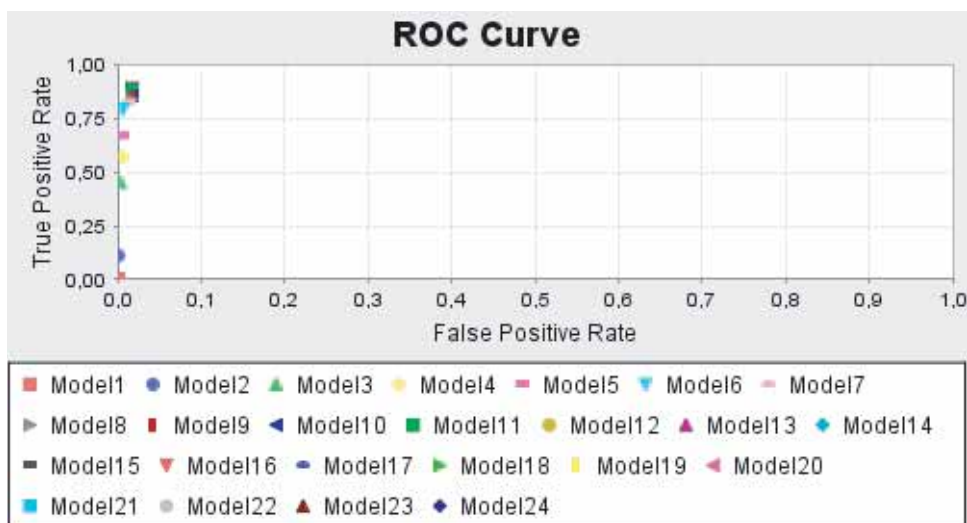


Figura 4.4: Curva ROC para o *Mus musculus* com o conjunto desbalanceado de seqüências. Observe que o parâmetro J influencia altamente os resultados, visto que o conjunto de dados considerado aqui é desbalanceado.

As figuras 4.5 e 4.6 apresentam os mesmos resultados para o *Rattus norvegicus*.

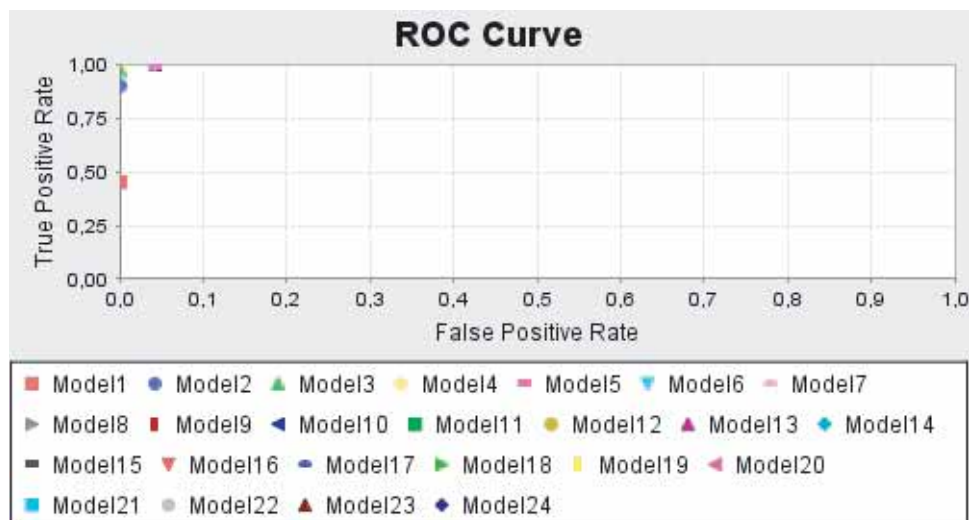


Figura 4.5: Curva ROC para o *Mus musculus* com o conjunto balanceado de seqüências. Observe que o parâmetro  $J$  não influencia os resultados, visto que o conjunto de dados já está balanceado.

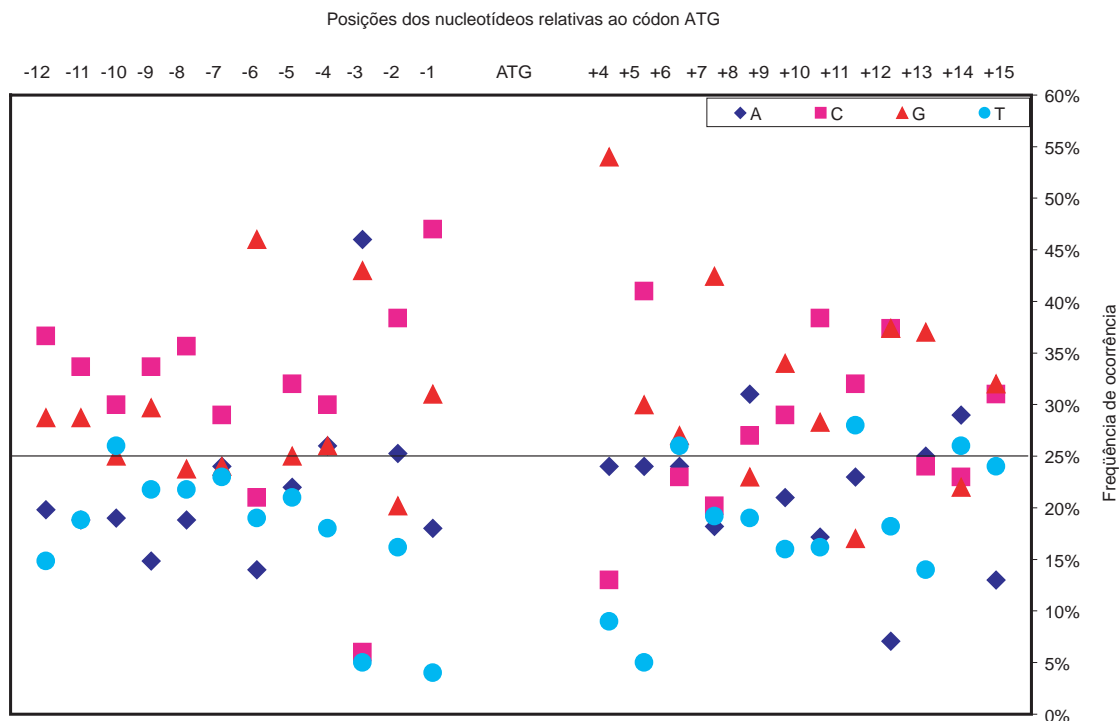
Figura 4.6: Curva ROC para o *Mus musculus* com o conjunto desbalanceado de seqüências. Observe que o parâmetro J influencia altamente os resultados, visto que o conjunto de dados considerado aqui é desbalanceado.

#### 4.4 *Análise dos falsos positivos*

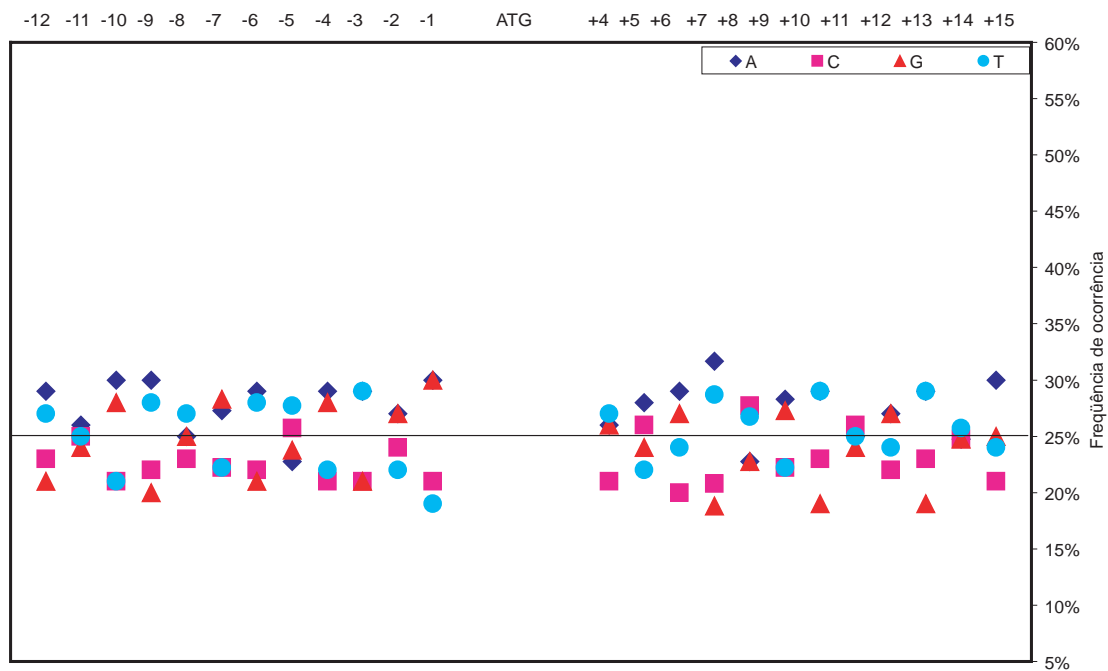
Uma análise da freqüência de cada base centrada sobre o TIS sugere que falsos positivos (figura 4.4 (c)), embora não reproduzindo completamente o padrão mostrado para as seqüências positivas (figura 4.4 (a)), não possuem uma distribuição aleatória que é peculiar às seqüências negativas (figura 4.4 (b)). A figura 4.4 (a) mostra o consenso de Kozak típico no qual obedece a regra para as posições -3 (purina) e +4 (guanina) e possui uma alta freqüência da G na posição -6. Claramente, como visto para os positivos verdadeiros, os falsos positivos não demonstram a dis-

tribuição aleatória observada nas seqüências negativas. Isso pode indicar que o classificador, na verdade, pode estar acertando a classificação e que estas seqüências na verdade podem ser verdadeiros positivos.

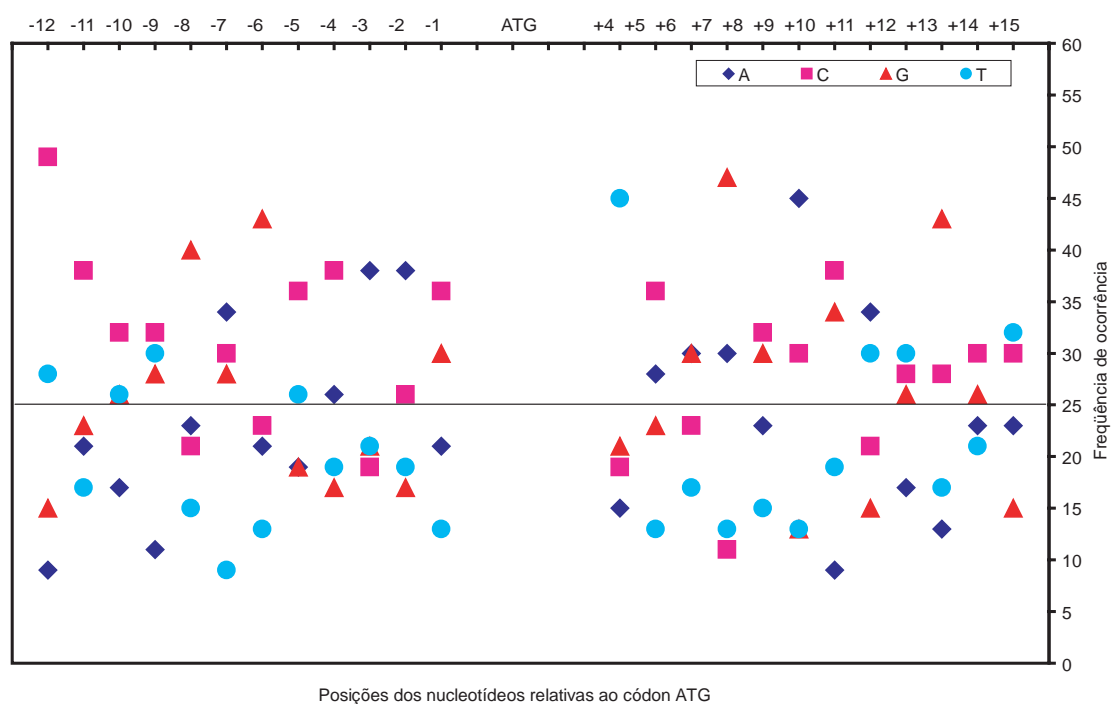




(a) Sequências positivas do *Mus musculus reviewed*



(b) Sequências negativas do *Mus musculus reviewed*



(c) Falsos positivos 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. 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 (10) 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 Pentiums 3GHz, 60GB de HD e 1 GB de RAM

Considerando-se a abordagem indutiva, o tempo de processamento variou de poucos minutos (8 min) para o *Mus musculus Reviewed* até 4,5 semanas para a *Drosophila melanogaster Reviewed*. No entanto, o tempo de processamento, considerando-se a abordagem transdutiva, é em média x% maior.

## 4.6 Conclusões do capítulo

Foram apresentados, neste capítulo, os resultados obtidos segundo a metodologia proposta, ou seja, foram apresentados os resultados obtidos variando-se o tamanho da janela, usando-se codificação base a base e por trinca, usando-se o método Smote para balanceamento entre as classes e sem a utilização deste método, os resultados

obtidos pela abordagem transdutiva, mostrando que esta técnica é bastante promissora para este problema de SIT, dentre outros métodos propostos.

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

# Conclusões e propostas de continuidade

---

---

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

## 5.1 *Conclusões*

A identificação do SIT é hoje um problema que necessita de muitas pesquisas. 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.

Alguns métodos já foram desenvolvidos até o momento, mas o desempenho para alguns organismos ainda precisa ser melhorado.

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

- Primeiramente, destaca-se a qualidade das bases analisadas. Todas elas foram extraídas do *RefSeq* que são sequências de referência;
- Escolha de toda uma metodologia que pudesse ser capaz de fazer uma boa predição. 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.

Quanto à análise das sequências, mostramos que as positivas estão de acordo com o consenso de Kozak (viés da Guanina (G) na posição +4 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

Alguns pontos neste trabalho ainda precisam ser aperfeiçoados; enquanto outros, precisam ser desenvolvidos. Os itens, a seguir, são sugestões para a continuidade deste trabalho:

- 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*, por exemplo, está rodando desde o dia 28 de maio e até hoje, dia 10 de julho, os resultados ainda não foram retornados.

- Expandir os testes para eucariotos mais primitivos, a partir daí, fazer uma comparação entre estes e metazoários superiores;
- 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. Com isso, estaremos dando uma grande contribuição para a área de Predição de SIT, que ainda é pouco estudada.
- ...

.....

---

# Apêndices

---

---

A codificação utilizada neste trabalho foi inspirada em uma análise das sequências positivas e negativas feita a partir das moléculas dos organismos extraídas. Inicialmente um dos propósitos deste trabalho foi exatamente analisar estas 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 (Kozak, 1984; Kozak, 1999; Kozak, 1989) 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. Neste sentido, dois tipos de análise foram realizadas:

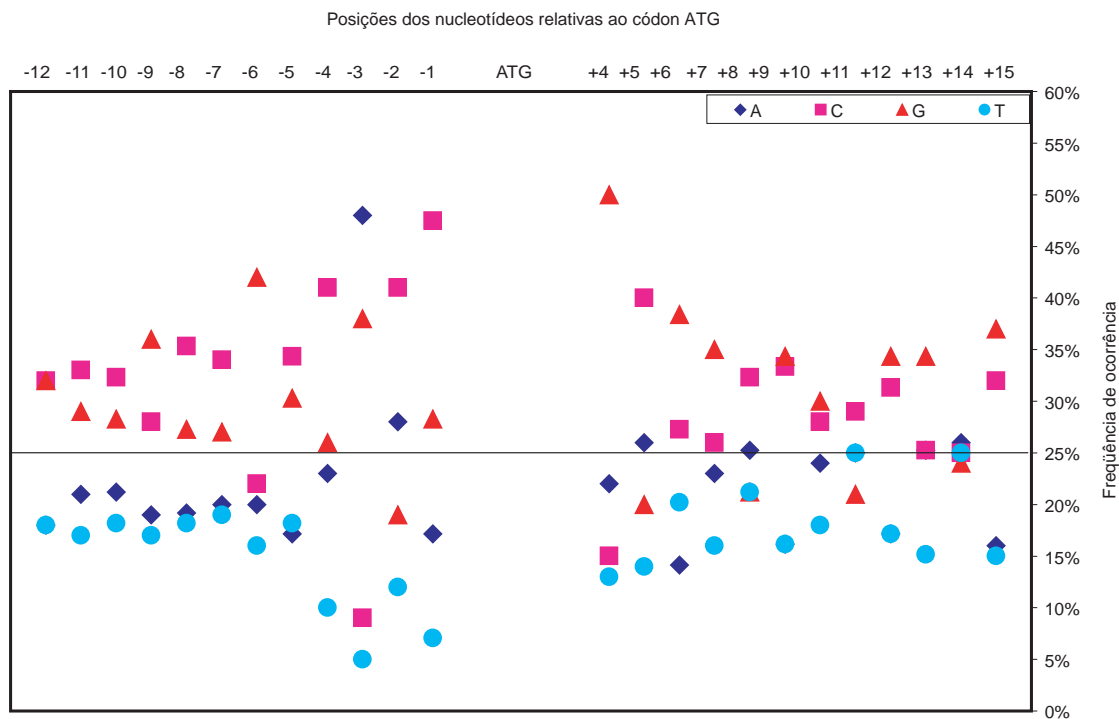
- Primeiramente, foi feita uma análise, por posição, da frequência de nucleotídeos de ambas as seqüências;
- Foi realizada, também, uma análise da frequência dos trinucleotídeos.



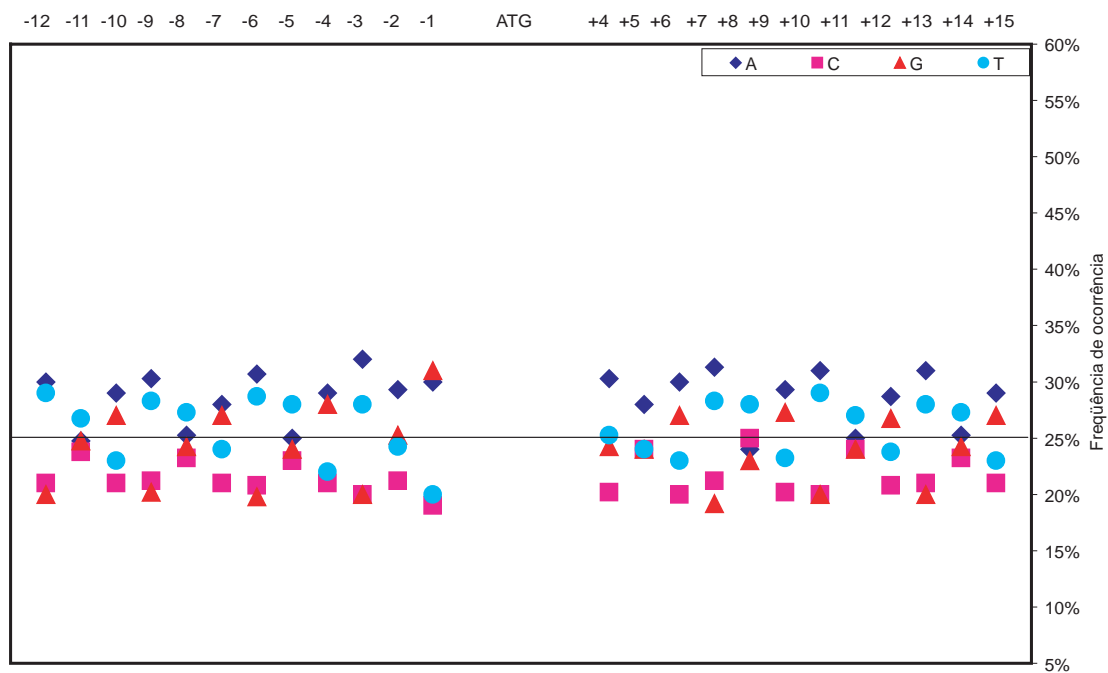
As seções seguintes apresentam os resultados destas análises.

### *A.1 Frequência de bases das seqüências*

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

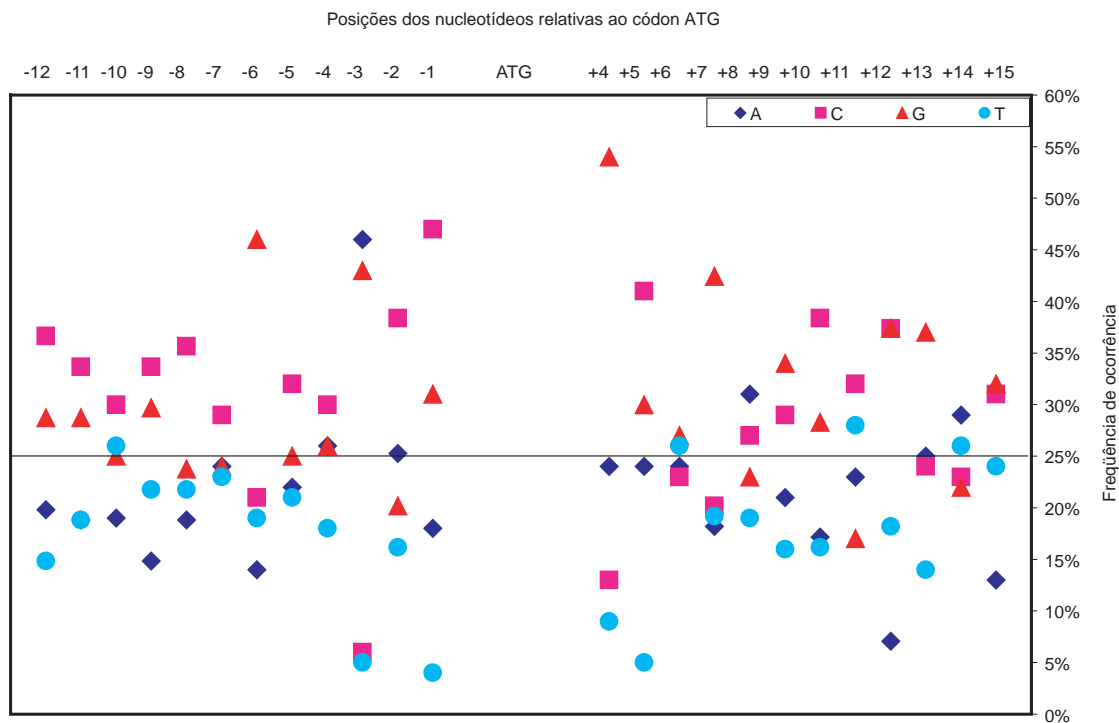


(d) Sequências positivas do *Homo sapiens reviewed*

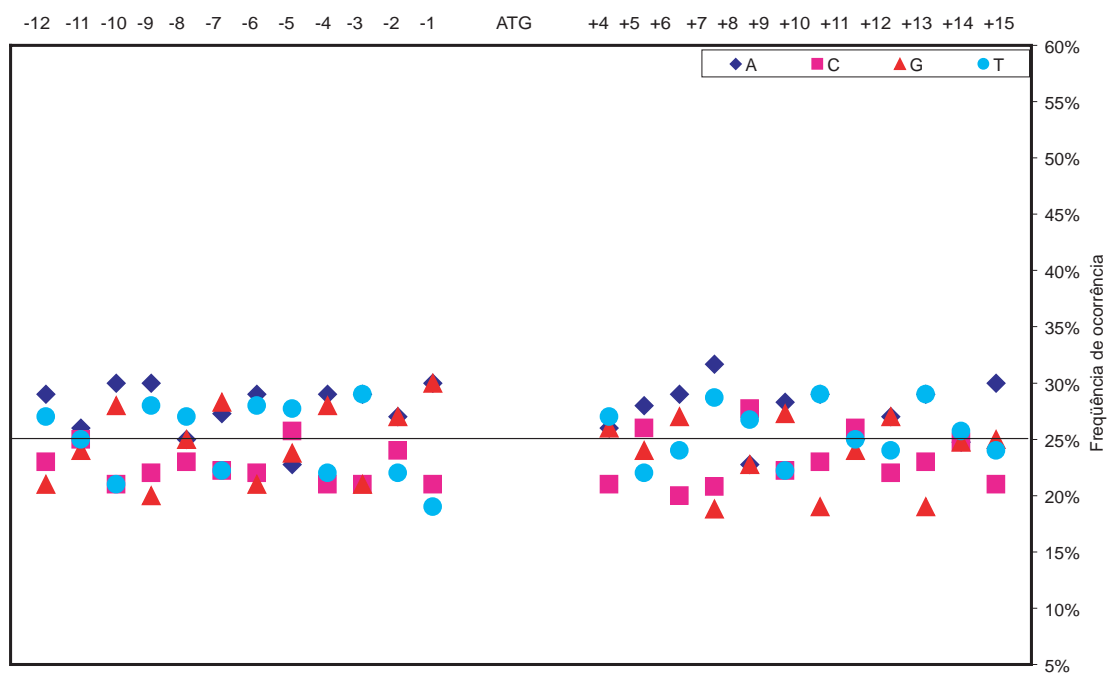


(e) Sequências negativas do *Homo sapiens reviewed*

Figura A.1: Frequência de bases das seqüências positivas e negativas, respectivamente, do *Homo sapiens reviewed*.

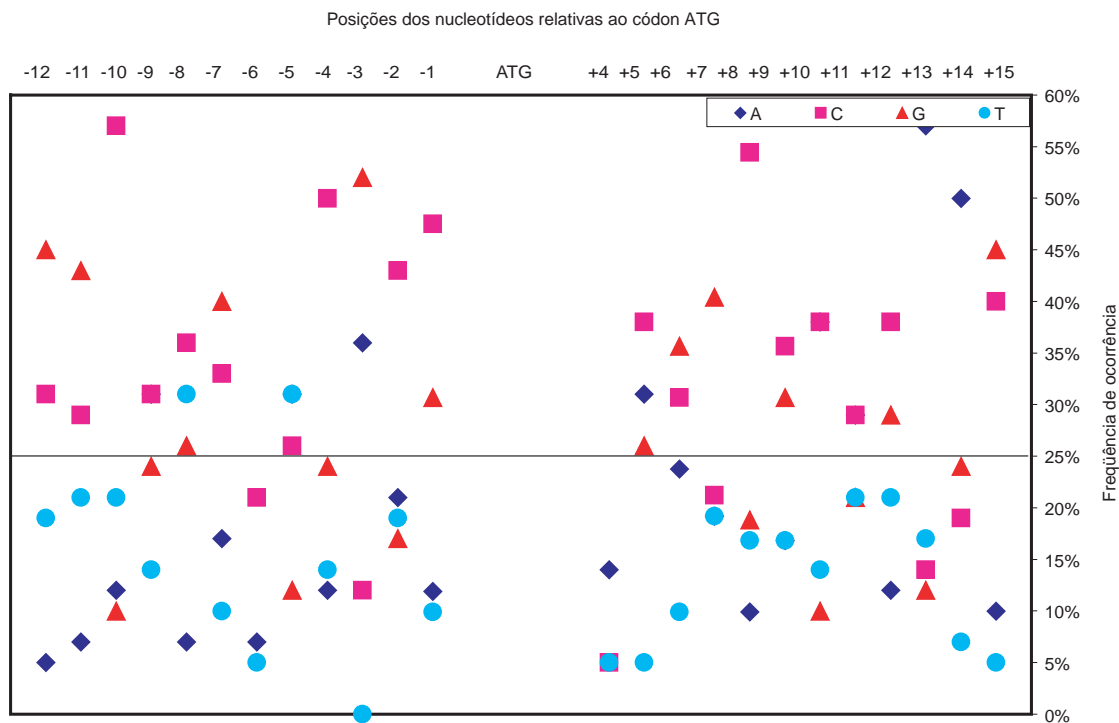


(a) Sequências positivas do *Mus musculus reviewed*

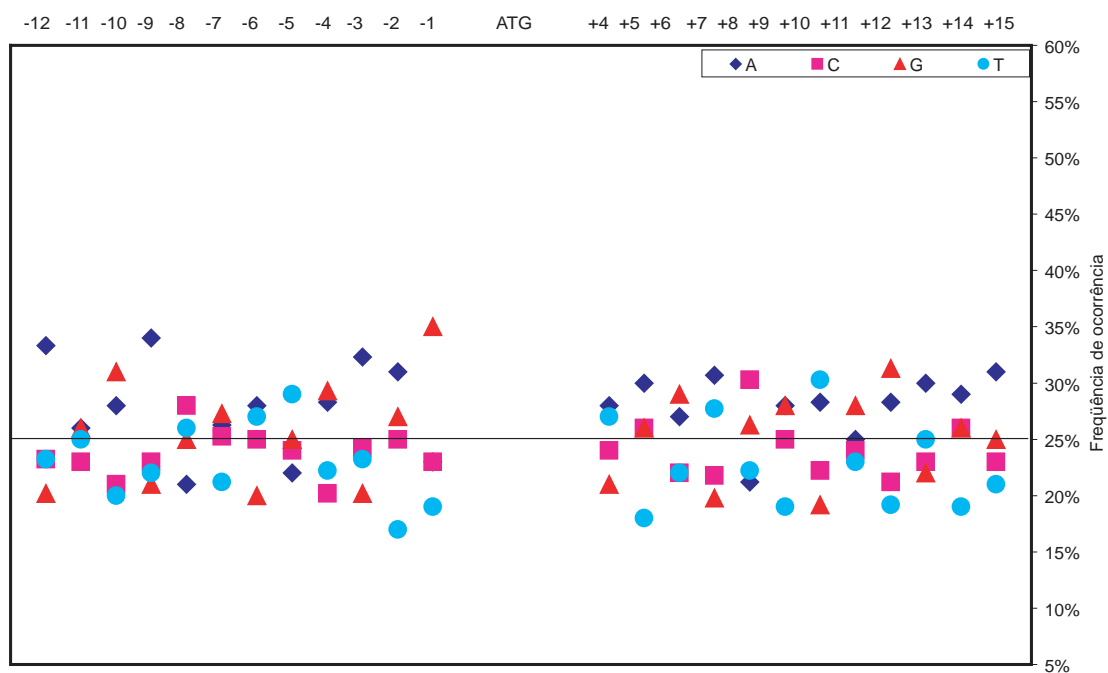


(b) Sequências negativas do *Mus musculus reviewed*

Figura A.2: Frequência de bases das seqüências positivas e negativas, respectivamente, do *Mus musculus reviewed*.

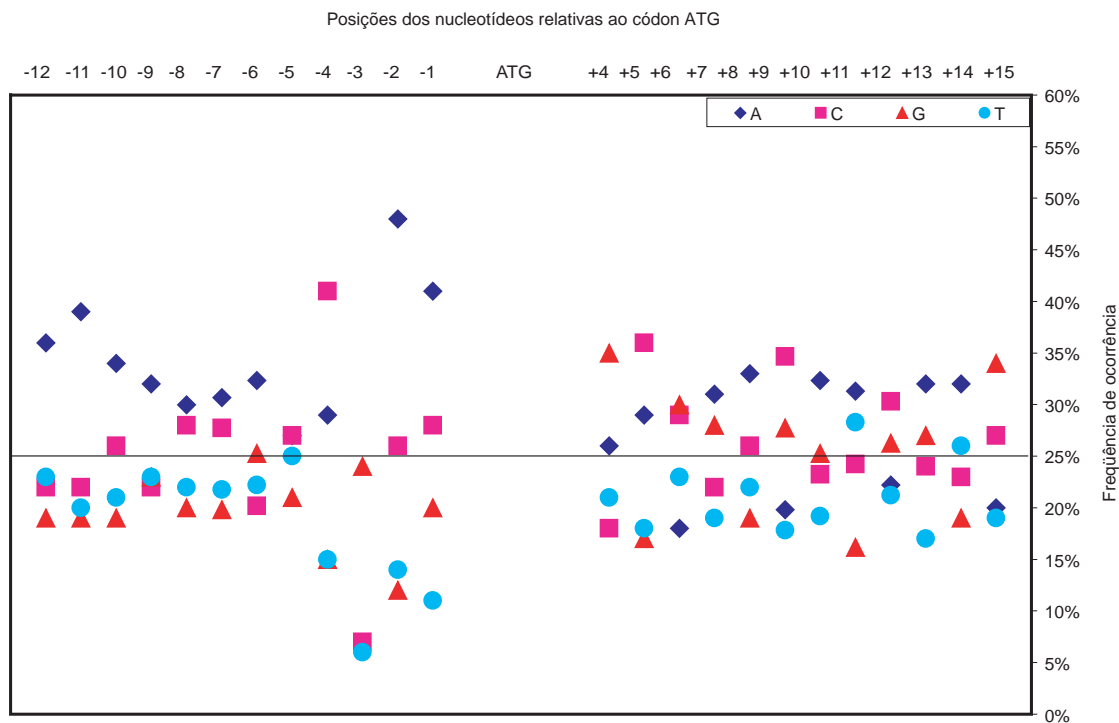


(a) Sequências positivas do *Rattus norvegicus reviewed*

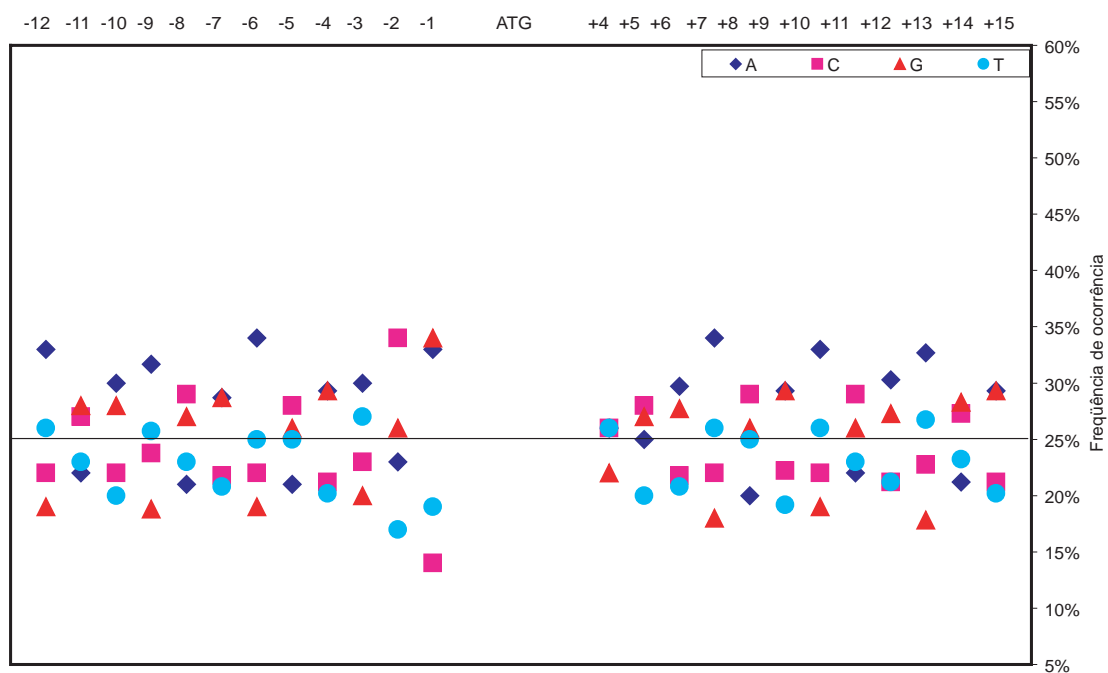


(b) Sequências negativas do *Rattus norvegicus reviewed*

Figura A.3: Frequência de bases das seqüências positivas e negativas, respectivamente, do *Rattus norvegicus reviewed*.



(a) Sequências positivas da *Drosophila melanogaster reviewed*



(b) Sequências negativas da *Drosophila melanogaster reviewed*

Figura A.4: Frequência de bases das seqüências positivas e negativas, respectivamente, da *Drosophila melanogaster reviewed*.

De acordo com estas 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 que 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. Para exemplificar, considere a seqüência: "**CGGACGCAT**"; neste caso, teríamos as seguintes composições: CGG=1, ACG=1 e CAT=1 e não teríamos o trinucleotídeo GGA, por exemplo. 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

e A.6 apresentam esta distribuição de freqüência para as seqüências *reviewed* do *Homo sapiens* e *Mus musculus*, *Rattus norvegicus* e *Drosophila melanogaster*. Não foi plotada a distribuição de freqüência para o *Danio rerio* porque este organismo só possui três(3) seqüências *reviewed*. Os resultados dos outros níveis de inspeção não estão apresentados porque seguem o mesmo padrão.

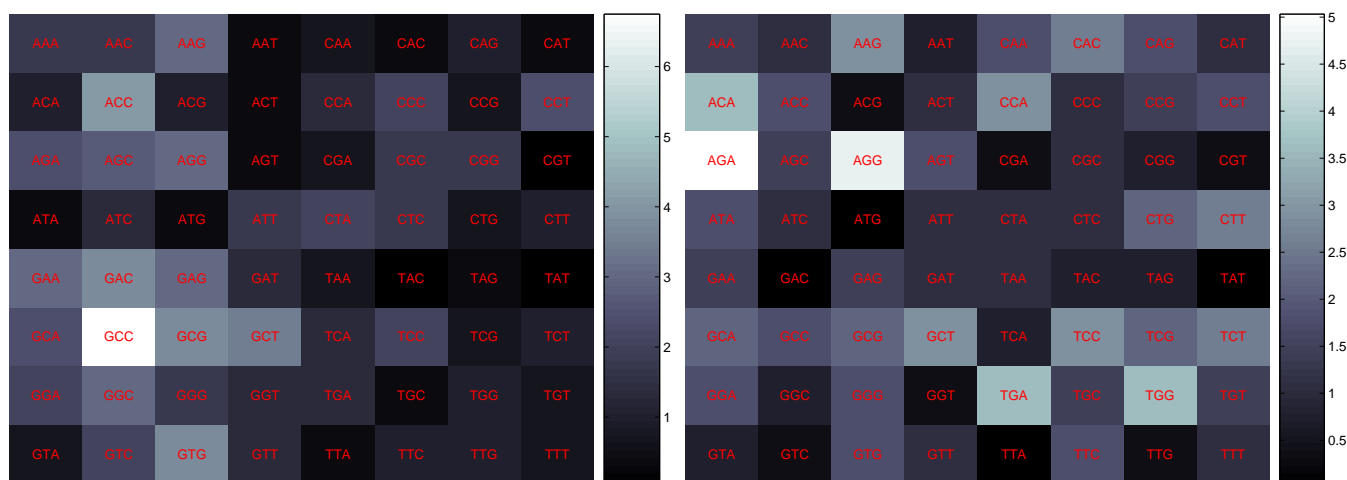
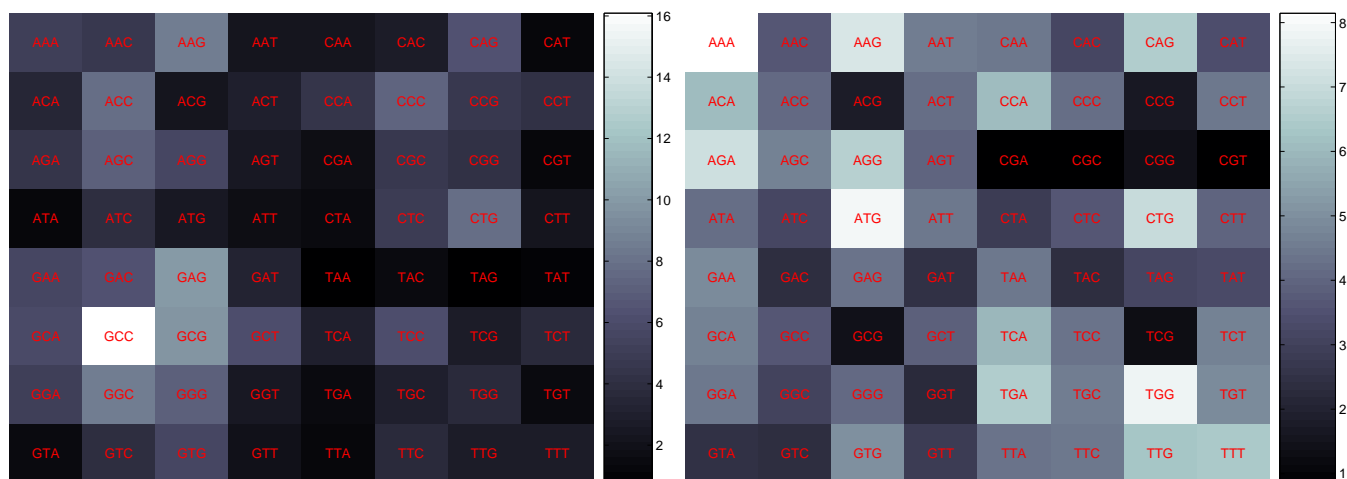
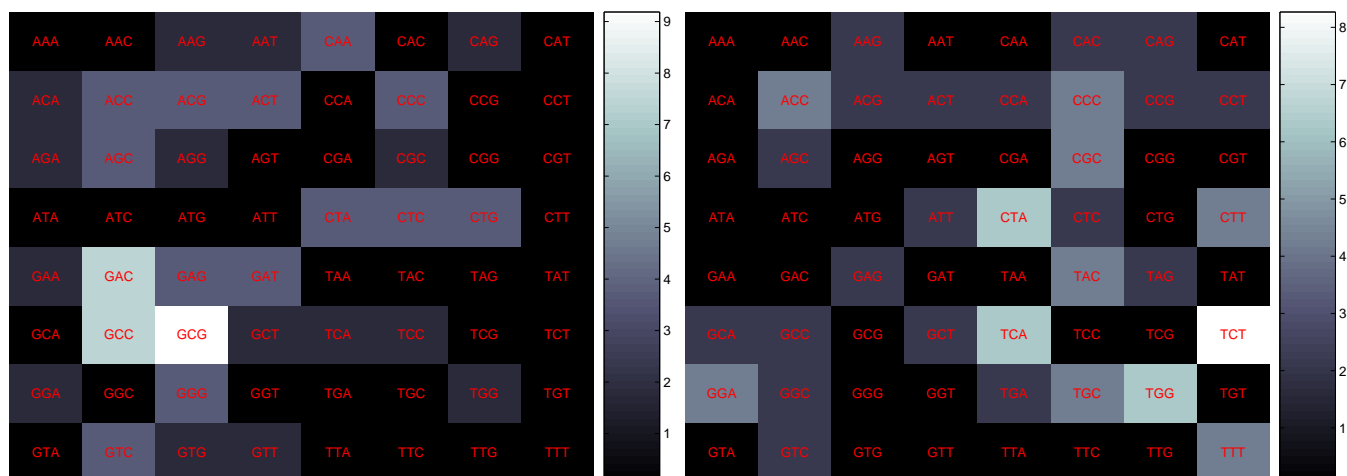
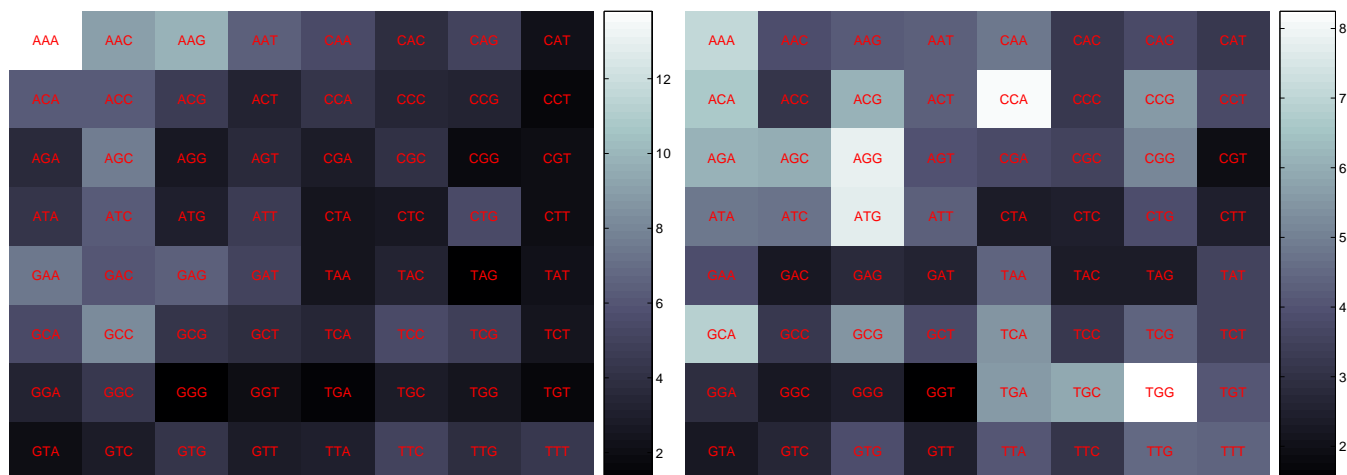


Figura A.5: Freqüência de trinucleotídeos das seqüências positivas e negativas *reviewed* do *Homo sapiens* e *Mus musculus*, respectivamente. A cor varia do mais claro, indicando uma alta freqüência, para o mais escuro, indicando baixa freqüência.



(a) Sequências positivas do *Rattus Norvegicus* re-  
viewed (b) Sequências negativas do *Rattus Norvegicus* re-  
viewed



(c) Sequências positivas do *Drosophila melanogaster* (d) Sequências negativas do *Drosophila*  
revisadas melanogaster revisadas

Figura A.6: Frequência de trinucleotídeos das seqüências positivas e negativas re-  
viewed, respectivamente, dos organismos estudados. 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 se-  
qüências 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 fre-  
qüência 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 frequência dos trinucleotídeos GCC e GCG (próximo de 9%) para os organismos *Mus musculus*, *Rattus norvegicus* e *Homo sapiens*. Para o *Danio rerio* e a *Drosophila melanogaster*, entretanto, o trinucleotídeo AAA aparece com maior frequência. Observando-se estas frequências percebe-se que estes dois organismos têm uma frequência bem diferente dos outros três. Nos negativos, entretanto, os trinucleotídeos TGG e TCT são os mais frequentes.

Ainda neste sentido, foram analisadas também as posições onde estes trinucleotídeos são mais frequentes. Ou seja, uma vez sabendo-se, por exemplo, que o GCC é o mais frequente no *Homo sapiens* queremos descobrir também qual é a posição onde se tem mais GCC. Será se este trinucleotídeo fica no início da sequência 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. Destas 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 frequência dos trinucleotídeos mais frequentes nestas três posições para todos os organismos analisados. Apenas as frequências referentes às seqüências positivas serão apresentadas, visto que a frequência obtida, neste caso, é bastante significativa. Além disso, os mesmos testes foram realizados com as seqüências negativas e observou-se uma frequência aleatória, em relação à posição, entre os trinucleotídeos mais frequentes (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*. Onde, TamTotal é o número total de seqüências positivas *reviewed* e ocorrência é a quantidade de vezes que o trinucleotídeo aparece. As maiores porcentagens estão em negrito.

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

Como pode ser observado, para todos os organismos, os trinucleotídeos são mais freqüentes na posição mais próxima do ATG. Esses dados, somados à análise independente de cada posição (figuras A.1 à A.4) ilustram a importância da utilização do classificador SVM que pondera as 24 posições simultaneamente.

# Referências

---

---

- Agarwal, P. e 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., M., B., D., L., e J., O. (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 Learning 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. (2000). 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. e Jambeck, P. (2001). *Desenvolvendo bioinformática: ferramentas de software para aplicações em biologia*. Campus.

- 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 Research 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](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 A.J., S. (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., 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 sim-

- ilarity 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 [cite-seer.ist.psu.edu/chawla02smote.html](http://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.
- 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*.
- Salamov, A. A., Nishikawa, T., e Swindells, M. A. (1998). Assessing protein coding region integrity in cdna sequencing projects. *Bioinformatics*, 14:384–390.

- Salzberg, S. (1997). A method for identifying splice sites and translational start sites in eukaryotic mrna. *Comput. Appl. Biosci.*, 13(4):365–376.
- 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, inferencia 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., Rättsch, 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.