HMM's para Identificação de Regiões Promotoras de Genes de Reparação

Gustavo Bastos (gbs@cin.ufpe.br)
Orientadora: Katia Silva Guimarães
Universidade Federal de Pernambuco - Centro de Informática - CIn

Março 2003

Resumo

Bio-informática é uma área recente da computação, que lida com problemas relacionados com a manipulação e análise de seqüências de nucleotídeos ou aminoácidos. Muitos dos problemas nesta área são NP-completos, de forma que abordagens alternativas são necessárias. Modelos Escondidos de Markov (do inglês *Hidden Markov Models*), que são modelos probabilísticos, são uma abordagem cada dia mais utilizada. Neste trabalho, pretendemos estudar a aplicação de HMM's na predição de regiões promotoras e desenvolver um HMM para reconhecer regiões promotoras de genes de reparação de fungos da espécie dos sacaramídeos.

Agradecimentos

Agradeço a minha mãe pelo apoio que me deu durante toda a minha vida, à professora Katia Guimarães pela orientação e oportunidade oferecida durante o meu período de bolsa de iniciação científica e trabalho de graduação e a Taciana Pontual Falcão, uma pessoa muito especial, que sempre me dá forças para continuar.

Assinaturas

Katia Silva Guimarães

Gustavo Bastos

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

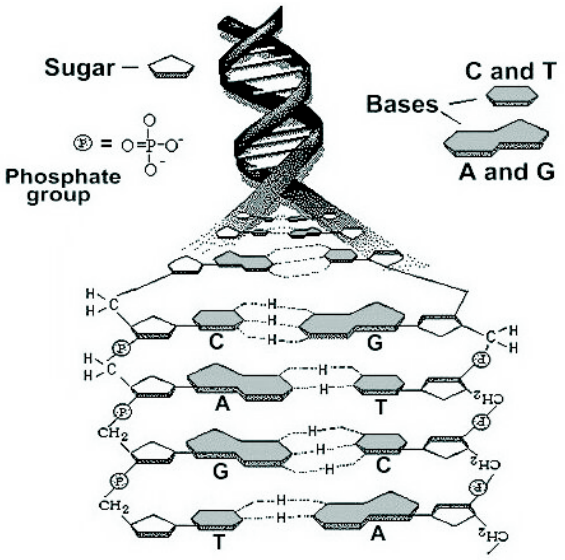

Figura 1: Estrutura do DNA

O ácido desoxirribonucléico (DNA) é o material genético primário de todas as células. Os organismos eucariontes e procariontes normalmente o possuem como uma fita dupla. O DNA é uma macromolécula composta por quatro bases nitrogenadas: adenina, citosina, guanina e timina. Estas bases ligam-se duas a duas (adenina com timina e citosina com guanina) e formam uma dupla hélice, que é a forma normal da molécula. As bases nitrogenadas são chamadas nucleotídeos e podem ser representadas respectivamente pelas letras **A**, **C**, **G** e **T**. Uma representação da estrutura do DNA pode ser vista na **Figura 1**.

Todo o DNA de um organismo é armazenado e 'empacotado' nos seus cromossomos e dentro deles existem genes, que são segmentos de DNA que codificam proteínas. O processo de transformar DNA em proteína é chamado de expressão do gene e é composto de duas partes: transcrição e tradução.

Existem vários tipos de genes e, geralmente, eles possuem uma função específica [8], como por exemplo: no processamento e armazenamento de informação (tradução, transcrição, reparação, recombinação e replicação de DNA), nos processos celulares (divisão celular), no metabolismo (conversão e produção de energia, metabolismo e transporte de aminoácidos), e em outras funções não muito bem caracterizadas. Uma maneira de se descobrir a funcionalidade de um gene é expressá-lo e verificar qual o resultado gerado e onde ele age, porém esta estratégia é lenta e trabalhosa.

No decorrer dos anos, descobriu-se que o processo de expressão do gene é muito mais complexo do que se imaginava. Foi descoberto por exemplo, que existem regiões dentro das seqüências de DNA que controlam este processo: as regiões promotoras. Elas apresentam uma alta conservação de bases, representando um padrão, e geralmente aparecem antes do início do gene (veja **Figura 2**). Foi descoberto também, que um gene pode ter várias

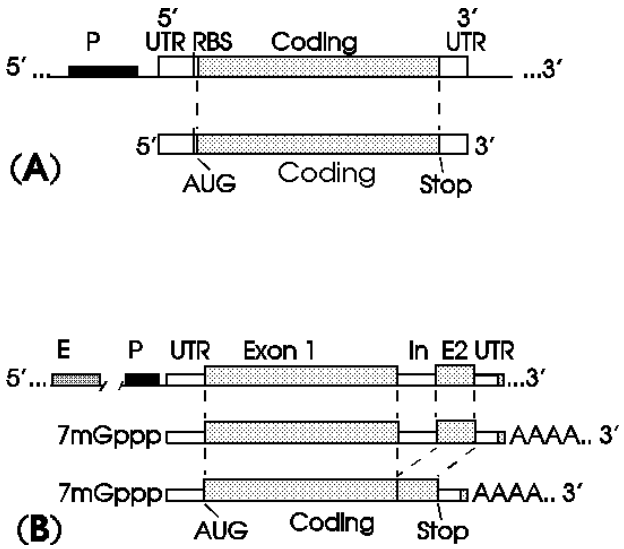

Figura 2: Região Promotora em uma Seqüência de DNA

regiões promotoras, cada uma delas fazendo-o traduzir uma proteína diferente. Então, a idéia de que um gene produziria um único produto foi alterada e verificou-se que o trabalho de descobrir a função de um gene é ainda mais complexa e difícil do que pensado anteriormente. Mas como dito acima, as regiões promotoras são altamente conservadas e determinam aos genes que elas controlam que executem certa funcionalidade. Logo, se for possível descobrir a existência de uma determinada região promotora que regula uma funcionalidade específica, pode-se supor que um gene controlado por esta região possuirá aquela função. No processo de determinar a existência de uma região promotora, a computação é de grande ajuda.

Como já citado, os nucleotídeos pode ser representados por letras, logo as seqüências de DNA serão longos *strings* de A, C, G e T. As regiões promotoras são subseqüências que apresentam alta similaridade entre si, representado padrões do ponto de vista computacional. Então, o problema recai em procura de palavras dentro de um texto.

Existem muitos programas eficientes que resolvem este problema e poderiam ser usados para solucionar a questão de procurar padrões nas seqüências de DNA, mas eles freqüentemente falham. Isto ocorre porque as seqüências biológicas são mais flexíveis do que a linguagem natural. Regiões promotoras, dentro do mesmo organismo e regulando a mesma função, podem variar muito, seja em tamanho, seja na composição das bases da seqüência. Apesar destas diferenças ainda há similaridades entre as seqüências, então o problema é como encontrar estas semelhanças.

A variação nas seqüências promotoras pode ser descrita estatisticamente e isto é a base para a maioria dos métodos usados para análise de seqüências biológicas [6]. Dois métodos muito utilizados são: as matrizes de substituição, como matrizes PAM, que são usadas no alinhamento de seqüências e as matrizes de peso que são usadas para encontrar padrões no DNA. Existem muitos outros métodos que podem ser usados nos problemas com seqüências

biológicas, mas neste trabalho vamos dar atenção ao *Hidden Markov Model*.

Um *Hidden Markov Model* (HMM) é um modelo estatístico muito adequado para tarefas na bio-informática. O mais popular uso do HMM na biologia molecular é como 'perfil probabilístico' (do inglês *probabilistic profile*) de uma família de proteína (ou DNA), que é chamado 'perfil HMM'. O 'perfil HMM' pode ser treinado para uma família de proteína (ou DNA) e depois pode procurar em bancos de dados por outros membros desta família. Provavelmente, a maior vantagem dos 'perfis HMM' é que eles lidam com *gaps* - espaço no alinhamento entre seqüências, seja por deleção ou inserção - de uma maneira sistemática.

Este trabalho irá mostrar uma utilização de um 'perfil HMM' para localizar regiões promotoras de sacaramídeos, usando a abordagem de alinhamento local. Na próxima seção, vamos falar um pouco sobre a história do HMM e de como ele está sendo usado hoje na bio-informática. Na Seção 3, uma revisão sobre a teoria de HMM e o modo como um 'perfil HMM' é gerado serão mostrados. Na Seção 4, será dada uma explicação sobre outras abordagens possíveis para a resolução do problema e por que não foram escolhidas. Esta seção também compreenderá a abordagem escolhida e a implementação da mesma. A Seção 5 mostra os testes e na 6 temos a conclusão e futuros trabalhos. Na Seção 7 existe um glossário cujo objetivo é explicar ao leitor os significados de alguns termos da biologia utilizados no decorrer do texto. As explicações encontrados no glossário são feitas de maneira simples e acessível. Para informações mais detalhadas e maiores explicações sobre os termos da biologia, olhar as referências no fim do trabalho.

2 História e contexto do HMM

Um HMM é um modelo probabilístico que é a extensão dos processos discretos de Markov. Existem três problemas fundamentais que um HMM deve resolver: a estimativa da probabilidade de uma seqüência de observações dado um específico HMM; a escolha da melhor seqüência de estados do modelo, e o ajuste dos parâmetros do modelo para se basear nos sinais observados.

Nem a teoria do HMM nem suas aplicações são novas. A teoria básica foi publicada por Baum e seus colegas no fim dos anos 60 e início dos anos 70 [4]. A primeira aplicação do HMM foi no 'reconhecimento de fala' (*speech recognition*), mas só com o passar dos anos é que um vasto entendimento e uma grande quantidade de aplicações em 'processamento da fala', e ultimamente bio-informática, têm ocorrido. Existem várias razões para isto ter acontecido. Primeiro, a teoria base de HMM foi publicada em jornais matemáticos que, geralmente, não eram lidos por engenheiros que trabalhavam no processamento da fala. A segunda razão foi que as aplicações originais da teoria para processamento da fala não forneciam material suficiente de tutorial para a maioria dos leitores entender a teoria e ser capaz de aplicá-la em sua própria pesquisa. Então, vários artigos com tutoriais foram escritos, fazendo assim com que muitos laboratórios pudessem começar suas pesquisas [4].

Antes de mostrarmos todo formalismo teórico, vamos fazer uma pequena comparação entre a forma que o HMM é aplicado em processamento da fala e como pode ser utilizado na análise de seqüências biológicas [5].

Um sinal de fala é dividido em pedaços (chamados *quadros*) de 10 a 20 milisegundos. Depois de alguns pré-processamentos, cada quadro é rotulado como uma de uma grande quantidade de categorias por um processo chamado quantização de vetor (*vector quantisation*). Geralmente, existem 256 categorias. O sinal de fala é, então, representado como uma longa seqüência de rótulos de categorias e, a partir disto, o reconhecedor de fala tem que descobrir qual seqüência de fonemas (ou palavras) foi falada. O problema é que há variações do som real e também do tempo gasto para dizer cada parte da palavra.

Muitos problemas de análise de seqüências biológicas têm a mesma estrutura: baseado em uma seqüência de símbolos de algum alfabeto, descobrir o que a seqüência representa. Para DNA, as seqüências consistem de símbolos de um alfabeto de 4 nucleotídeos e nós podemos querer encontrar uma região dentro desta seqüência. Aqui, a seqüência primária de nucleotídeos é análoga ao sinal de fala e o processo de encontrar certa região é análogo a saber se uma determinada palavra foi falada. A variação de tempo no sinal de fala corresponde a ter inserções e deleções nas seqüências de DNA.

Como pode ser percebido, depois de seu uso inicial, HMM pode ser utilizado em diversos problemas, alguns deles são [12]:

- Inferir a gramática de linguagens simples;
- Modelar a relação entre pares de seqüências de DNA;
- Modelar a relação entre várias seqüências dada uma árvore filogenética;
- Descobrir padrões em seqüências de DNA;

Um trabalho que apresenta bons resultados na classificação de genes corregulados baseada na região promotora [14] utiliza o pacote de software MEME [19]. Este pacote possui um software que descobre e procura *motifs* e o resultado pode ser utilizado por outro software do pacote para criar um HMM para ser usado na procura em bancos de dados. Este pacote pode ser utilizado pela internet.

Existem muitos softwares e aplicações [12] que utilizam HMM na resolução de problemas específicos, existem pacotes [11] que usam HMM em problemas de diferentes áreas e existem até bancos de dados de alinhamentos e de 'perfis HMM' [13]. Porém, o uso de HMM, geralmente, é feito baseado em uma classe similar de seqüências, famílias de proteína ou de DNA. Assim, a utilização de HMMs gerais podem produzir resultados não tão bons quanto os de um HMM específico.

Existem outros softwares que utilizam HMM na bio-informática; muitos deles não disponibilizam códigos fontes. Há poucos trabalhos dirigido a regiões promotoras de sacaramídeos, logo uma possibilidade existente seria estudar e utilizar um HMM para encontrar regiões promotoras em seqüências de DNA dos sacaramídeos com uma abordagem diferente. Baseado na literatura existente e sabendo que outros trabalhos na área apresentaram bons resultados, este seria um trabalho inovador e relevante.

3 Base Teórica do Trabalho

3.1 Processos Discretos de Markov

Considere um sistema que pode ser descrito em qualquer tempo como estando em um estado de um conjunto de N estados distintos S_1, S_2, ..., S_n. Em tempos regulares, o sistema sofre uma mudança de estado (podendo voltar para o mesmo estado) seguindo um conjunto de probabilidades associado com o estado. Os instantes de tempo associados com as mudanças de estados são denominados $t = 1, 2, \ldots$, e o estado atual no tempo t é chamado q_t. Para determinar o estado atual no tempo t, depende-se, apenas, dos n estados anteriores. Esta é a suposição dos processos de Markov. Existe o caso do processo discreto de Markov de primeira ordem, no qual, para se determinar o estado atual só é necessário o estado imediatamente anterior. Formalmente:

$$\begin{aligned} P[q_t &= S_j | q_{t-1} = S_i, q_{t-2} = S_k, \ldots] \\ &= P[q_t = S_j | q_{t-1} = S_i] \end{aligned} \tag{1}$$

Posteriormente, nós apenas consideramos aqueles processos nos quais o lado direito de (1) é independente de tempo, com isto criando o conjunto de probabilidades de transição de estados a_{ij} da forma:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad 1 \leq i, j \leq N \tag{2}$$

com os coeficientes da transição de estados tendo as propriedades:

$$a_{ij} \geq 0 \tag{3}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{4}$$

visto que eles obedecem às restrições estocásticas padrão.

Qualquer processo estocástico que possa ser descrito assim é chamado *um modelo observável de Markov*, visto que a saída do processo é o conjunto de estados em cada instante de tempo, onde cada estado corresponde a um evento físico (observável) (para maiores explicações e alguns exemplos, veja o tutorial de Rabiner [4]).

3.2 Modelos Escondidos de Markov - HMM

Na subseção anterior, nós consideramos modelos de Markov nos quais cada estado correspondia a um evento observável (físico). Este modelo é muito restritivo para ser aplicado ao problema de nosso interesse. Nesta subseção, nós estenderemos os conceitos dos modelos de Markov para incluir o caso no qual a observação é uma função probabilística do estado, isto é, o modelo resultante, chamado Modelo Escondido de Markov (do inglês *Hidden Markov Model*), é um processo estocástico duplo com um processo estocástico fundamental

que não é observável (ele fica escondido), mas pode ser observado através de um outro conjunto de processos estocásticos que produz a seqüência de observações.

Para tentar explicar a idéia de um HMM, vamos considerar o sistema de urnas e bolas.[1] Assumamos que existem N urnas em uma sala e dentro de cada uma existe um grande número de bolas. Assumamos, também, que existem M cores distintas de bolas. O processo de obter observações seria o seguinte: de uma forma randômica, uma urna inicial é escolhida. A partir dela, uma bola é escolhida randomicamente e sua cor é registrada como observação. A bola é recolocada na urna da qual foi retirada. Uma nova urna é selecionada de acordo com um processo de seleção randômico associado com a urna atual e o processo de escolha da bola é repetido. O processo inteiro gera uma seqüência finita de observação de cores que nós gostaríamos de modelar como uma saída observável de um HMM.

O mais simples HMM que corresponde ao processo urna e bola é aquele no qual cada estado corresponde a uma urna específica e no qual uma probabilidade de cor é definida para cada estado. A escolha de urnas é feita com base na matriz de transição de estados do HMM [4, 15].

3.3 Elementos de um HMM

Na subseção anterior, nós introduzimos a idéia de HMM e agora vamos mostrar o formalismo para criar um HMM.

Um HMM é definido como uma tripla $\mu = (\Sigma, \mathrm{S}, \Theta)$, onde:

1. Σ é o conjunto que contém o alfabeto de símbolos. Os símbolos de observação correspondem à saída física do sistema. Existe um número finito de símbolos M e os símbolos individuais podem ser representados como $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_M\}$.

2. S é o conjunto finito de estados no modelo. Existem N estados no modelo. Geralmente, os estados possuem interconexões de tal modo que qualquer estado pode ser alcançado a partir de qualquer outro estado. Os estados podem emitir símbolos de Σ. Os estados individuais são denotados $S = \{s_1, s_2, \ldots, s_N\}$ e o estado no tempo t é denotado q_t.

3. Θ é o conjunto de probabilidades, formado de três conjuntos:

   (a) $\mathrm{A} = \{a_{ij}\}$ é a distribuição de probabilidade da transição de estados, onde :

   $$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N \tag{5}$$

   (b) $\mathrm{B} = \{b_j(k)\}$ é a distribuição de probabilidade de emissão (observação) em um estado j, onde :

   $$b_j(k) = P[\sigma_k \; em \; t | q_t = S_j], \quad 1 \leq j \leq N \;\; e \;\; 1 \leq k \leq M \tag{6}$$

[1] O modelo 'urna e bola', do inglês *the urn and ball model*, foi apresentado por Jack Fergunson e outros, em palestras sobre a teoria do HMM.

(c) $\pi = \{\pi_j\}$ é a distribuição dos estados iniciais onde:

$$\pi_j = P[q_1 = S_j], \quad 1 \leq j \leq N \tag{7}$$

Maiores informações e explicações podem ser encontradas nas referências no final do trabalho, principalmente Rabiner [4].

3.4 Os três problemas do HMM

Dada a forma do HMM da seção anterior, há três problemas básicos que devem ser resolvidos para o HMM ser útil em aplicações do mundo real. Eles são:

- Problema 1: Dada uma seqüência de observação $O = O_1 O_2 \ldots O_T$, e o modelo $\mu = (\Sigma, S, \Theta)$, como computar eficientemente $P(O|\mu)$, a probabilidade da seqüência de observação dado o modelo?
- Problema 2: Dada uma seqüência de observação $O = O_1 O_2 \ldots O_T$, e o modelo μ, como escolher uma seqüência de estados correspondente $Q = q_1 q_2 \ldots q_T$ que é ótima de algum modo significativo?
- Problema 3: Como nós ajustamos os parâmetros do modelo $\mu = (\Sigma, S, \Theta)$ para maximizar $P(O|\mu)$?

O problema 1 é um problema de avaliação, ou seja, dado um modelo e uma seqüência de observações, como nós podemos computar a probabilidade de que a seqüência observada tenha sido produzida pelo modelo. O algoritmo *forward* é utilizado para resolver este problema.

O problema 2 é aquele no qual tentamos descobrir a parte escondida do modelo, isto é, achar a seqüência de estados 'correta'. O algoritmo *Viterbi* é usado para resolver este problema.

No problema 3, nós tentamos otimizar os parâmetros do modelo para descrever melhor como uma seqüência observada acontece. O algoritmo *forward-backward* é utilizado na resolução deste problema.

Como o objetivo do trabalho é descobrir regiões promotoras dentro de seqüências de DNA, vamos nos concentrar na resolução do problema 2, encontrar a melhor seqüência de estados para uma seqüência observada. Na próxima subseção, será mostrado o algoritmo que resolve este tipo de problema. O algoritmo *forward-backwaqrd* também possui uma grande importância quando queremos treinar um HMM, porém este trabalho não possui a intenção de apresentar uma ferramenta completa, mas sim um modelo que pode apresentar bons resultados no reconhecimento de regiões promotoras em sacaramídeos. Para um maior conhecimento e entendimento sobre os problemas com HMM e os algoritmos utilizados para resolvê-los, veja as referências no fim do trabalho [1, 2, 3, 4, 5].

3.5 Encontrando um caminho ótimo

O problema 2 tenta encontrar o caminho 'ótimo' associado com a seqüência de observação dada. A dificuldade é justamente definir a seqüência de estados ótima, ou seja há vários critérios possíveis. Vamos mostrar agora um algoritmo baseado em programação dinâmica, o *algoritmo Viterbi* [4].

Para encontrar a melhor seqüência de estados, $Q = \{q_1 q_2 \dots q_T\}$, para um seqüência de observação dada $O = \{O_1 O_2 \dots O_T\}$, precisamos definir a quantidade

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_t = i, O_1 O_2 \dots O_T | \mu] \tag{8}$$

ou seja, $\delta_t(i)$ é o melhor escore (mais alta probabilidade) ao longo de um caminho, no tempo t, que leva em conta as primeiras t observações e termina em S_i. Por indução temos:

$$\delta_{t+1}(i) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(O_{t+1}) \tag{9}$$

Para podermos recuperar a seqüência de estados, nós precisamos guardar a trilha do argumento que maximizou (9), para cada t e j. Isto pode ser feito através do *array* $\psi_t(j)$. O procedimento completo é mostrado a seguir.

1. Início:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N \tag{10}$$
$$\psi_1(i) = 0 \tag{11}$$

2. Recursão :

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \le t \le T \ \ e \ \ 1 \le j \le N \tag{12}$$
$$\psi_t(j) = \operatorname{argmax}_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}], \quad 2 \le t \le T \ \ e \ \ 1 \le j \le N \tag{13}$$

3. Término:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \tag{14}$$
$$q_T^* = \operatorname{argmax}_{1 \le i \le N} [\delta_T] \tag{15}$$

4. Rastreamento do caminho (dos estados):

$$q_t^* = \psi_t(q_{t+1}^*), \quad t = T-1, T-2, \cdots, 1. \tag{16}$$

A complexidade deste algoritmo pode ser calculada como se segue. Nós calculamos os valores de $O(|S| \cdot T)$ células da matriz V, gastando $O(|S|)$ operações por célula. A complexidade do tempo total é $O(|S|^2 \cdot T)$ e a complexidade do espaço é $O(|S| \cdot T)$.

Visto que estamos lidando com probabilidades, a repetida utilização de operações de multiplicação que é realizada pode levar a um *underflow*. Isto pode ser evitado se usarmos escores logarítmicos. Então, $\delta_t(j)$ pode ser definido como o escore logarítmico do caminho mais provável. Os novos valores são apresentados abaixo (veja [1, 2, 3, 5] para maiores explicações e provas).

1. Início:

$$\delta_1(i) = \log(\pi_i) + \log(b_i(O_1)), \quad 0 \leq i \leq N \tag{17}$$

2. Recursão :

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) + \log(a_{ij})] + \log(b_j(O_t)), \quad 2 \leq t \leq T \ \ e \ \ 1 \leq j \leq N \tag{18}$$

3.6 Perfil HMM

Um *perfil HMM* Υ de tamanho L é definido como um conjunto de probabilidades denotadas $e_i(b)$ (a probabilidade de observar o símbolo b na *i-ésima* posição, onde b $\in \Sigma$). Portanto, a probabilidade de um *string* $O = (O_1, \ldots, O_L)$ dado o perfil Υ será:

$$P(X \mid \Upsilon) = \prod_{i=1}^{L} e_i(x_i) \tag{19}$$

Nós podemos calcular o escore de *likelihood* para um alinhamento sem *gaps* de X contra o perfil Υ:

$$Escore(X|\Upsilon) = \sum_{i=1}^{L} log \frac{e_i(x_i)}{p(x_i)} \tag{20}$$

onde $p(b)$ á freqüência *background* de ocorrências do símbolo b.

Esta idéia de perfil leva à definição do HMM formado por estados *match* seqüencialmente ligados $M_1, M_2, \ldots, M_L$, que corresponderia a 'casar' com o mais provável símbolo em cada posição do perfil.

Estados *insertion*, $I_0, I_1, \ldots, I_L$, são adicionados para permitir inserções no alinhamento e cada estado I_j tem um elo de entrada a partir do estado M_j e um elo de saída para M_{j+1}. Os estados *insertion* também possuem um *self-loop*, uma ligação para o próprio estado, para permitir inserções múltiplas na mesma posição.

O último tipo de estado deste HMM são os estados *deletion*, $D_1, D_2, \ldots, D_L$, que são obviamente introduzidos para permitir deleções. Eles diferem dos dois tipos anteriores por serem 'silenciosos', ou seja, eles não emitem símbolos, logo não há $e_i(b)$ associado aos estados *deletion*. Os estados de deleção são seqüencialmente ligados, bem como conectados com os estados *match* do mesmo modo que os estados *insertion*. Estados *deletion* também são ligados aos estados *insertion* por dois elos: de D_j para I_j e de I_j para D_{j+1}.

O perfil final Υ tem o comprimento de L camadas. Cada camada do HMM é formada por três estados : M_j, I_j, D_j. Uma camada extra, compreendendo os estados I_0 e BEGIN, é adicionada no início e outra camada, compreendendo o estado END. Os estados BEGIN e END também são silenciosos. O modelo é mostrado na **Figura 3**.

3.7 Algoritmo Viterbi para um Perfil HMM

Dado um perfil HMM Υ de tamanho L e uma seqüência $X = (x_1, x_2, \ldots, x_T)$, os escores logarítmicos de Viterbi para cada estado são:

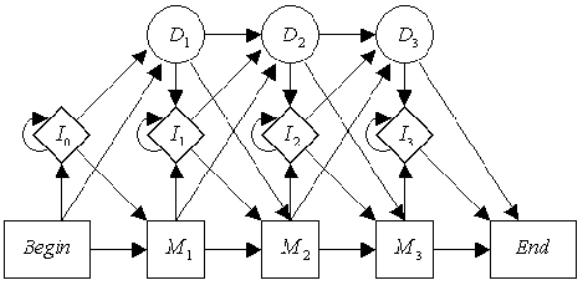

Figura 3: Perfil HMM

- $v_j^M(t)$ - escore do melhor caminho para casar a seqüência $X = (x_1, x_2, \ldots, x_t)$ com o perfil HMM Υ, terminando com x_t emitido pelo estado M_j (onde $1 \leq$ j $\leq$ L).
- $v_j^I(t)$ - escore do melhor caminho para casar a seqüência $X = (x_1, x_2, \ldots, x_t)$ com o perfil HMM Υ, terminando com x_t emitido pelo estado I_j (onde $1 \leq$ j $\leq$ L).
- $v_j^D(t)$ - escore do melhor caminho para casar a seqüência $X = (x_1, x_2, \ldots, x_t)$ com o perfil HMM Υ, terminando no estado D_j, sem emitir nenhum símbolo (onde $1 \leq$ j $\leq$ L).

Porém, será necessário fazer pequenas modificações para os valores iniciais e para o cálculo dos escores logarítmicos. Isto acontece porque os estados BEGIN e END são usados no cálculo e porque os estados não possuem ligações com todos os estados (veja **Figura 3**). Os estados BEGIN e END são chamados de M_0 e M_{l+1}, respectivamente, para facilitar os cálculos. O procedimento para calcular os escores Viterbi para cada um dos tipos de estado é:

1. Início:

$$v_0^M(0) = 0 \tag{21}$$

2. Recursão :

$$v_j^M(t) = \log\left(\frac{e_{M_j}(x_t)}{p(x_t)}\right) + \max\left\{\begin{array}{l} v_{j-1}^M(t-1) + \log\left(a_{M_{j-1},M_j}\right) \\ v_{j-1}^I(t-1) + \log\left(a_{I_{j-1},M_j}\right) \\ v_{j-1}^D(t-1) + \log\left(a_{D_{j-1},M_j}\right) \end{array}\right. \tag{22}$$

$$v_j^I(t) = \log\left(\frac{e_{I_j}(x_t)}{p(x_t)}\right) + \max\left\{\begin{array}{l} v_j^M(t-1) + \log\left(a_{M_j,I_j}\right) \\ v_j^I(t-1) + \log\left(a_{I_j,I_j}\right) \\ v_j^D(t-1) + \log\left(a_{D_j,I_j}\right) \end{array}\right. \tag{23}$$

$$v_j^D(t) = \max \begin{cases} v_{j-1}^M(t) + \log\left(a_{M_{j-1},D_j}\right) \\ v_{j-1}^I(t) + \log\left(a_{I_{j-1},D_j}\right) \\ v_{j-1}^D(t) + \log\left(a_{D_{j-1},D_j}\right) \end{cases} \quad (24)$$

A complexidade deste algoritmo pode ser calculada da seguinte forma. Nós calculamos $O(L \cdot T)$ valores, gastando $O(1)$ (visto que nós precisamos considerar os escores de no máximo três predecessores) para calcular cada valor. A complexidade do tempo total é $O(L \cdot T)$ e a complexidade do espaço é $O(L \cdot T)$ (para maiores informações sobre a criação do perfil HMM, veja as referências no fim do trabalho [1, 2, 3, 5]).

4 Desenvolvimento do Trabalho

4.1 Outras abordagens possíveis

Ao se aplicar o algoritmo Viterbi em um perfil HMM, consegue-se um alinhamento global [5], mas nosso interesse é em alinhamento local. Existem diversas abordagens para fazermos um alinhamento local [5], duas delas estão nas figuras apresentadas nesta subseção.

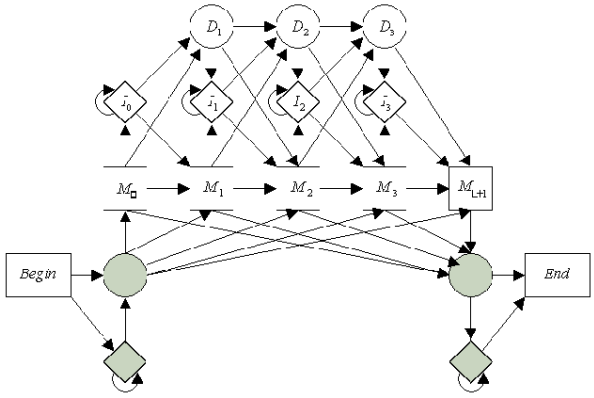

Figura 4: Perfil HMM para alinhamento local

Basicamente, o funcionamento delas é o seguinte. Incluem-se dois estados novos para 'consumir' os símbolos da seqüência antes e depois da região que se deseja alinhar. Nas figuras, estes estados são representados pelos losangos sombreados. Os círculos sombreados servem de estados de 'escalonamento'. Estes estados estão ligados ao perfil HMM que representa a região da seqüência que nós desejamos alinhar.

Ao analisarmos a **Figura 4**, percebemos que este modelo não permite que exista mais de uma região promotora na seqüência de entrada. Percebe-se que o primeiro losango irá consumir os símbolos que estão antes da região desejada, e a partir do primeiro círculo nós alcançamos o perfil HMM. Mas, uma vez que nós alcançarmos o segundo círculo não

poderemos entrar novamente no perfil, podendo, apenas, consumir os símbolos finais da seqüência. Por este motivo, este modelo foi descartado.

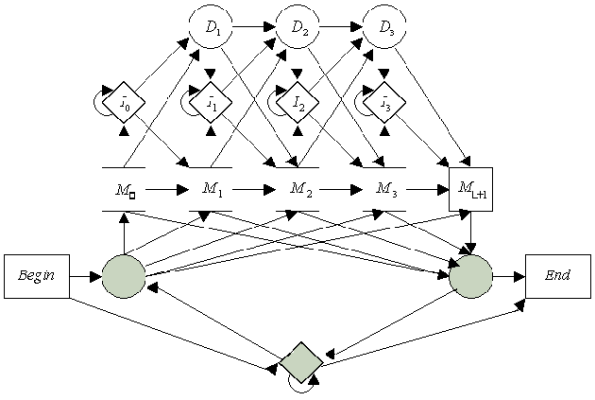

Figura 5: Outro perfil HMM para alinhamento local

Ao analisarmos a **Figura 5**, percebemos que diferentemente do modelo anterior este modelo permite que exista mais de uma região promotora na seqüência de entrada, logo poderíamos utilizá-lo. Porém, ele não diferencia as regiões promotoras, ou seja, como o modelo usa apenas um estado de consumo (losango) a probabilidade de ir para uma segunda região promotora é a mesma de ir para a primeira região. Achamos que isto não é interessante, se já foi encontrada um região, a probabilidade de encontrar outra deveria ser menor. Este modelo apresenta também a característica de fazer a ligação direta do estado de consumo (losango) com o estado END. Então, por não apresentar diferenciação entre o modo de entrar na primeira região promotora e de entrar nas demais (se existirem), este modelo também foi descartado.

Uma terceira abordagem possível seria utilizar a habilidade do HMM para modelar *gramática* [6]. Muitos problemas de análise de seqüências biológicas possuem uma estrutura gramatical. A idéia básica é definir 'palavras' a partir dos sinais conhecidos existentes nas seqüências e a partir disto determinar as frases válidas. No nosso caso, poderíamos dizer que a região promotora tem que vir antes de um códon de início e a frase começaria antes da região promotora e terminaria no códon de início.

Existem abordagens parecidas, mas que não criam realmente uma 'gramática' [14]. A idéia é colher informações como quantidade de regiões relevantes e distâncias relativas entre elas, usando um software. De posse destas informações, um HMM é criado. Este HMM seria composto de vários pequenos HMM que representam cada um uma determinada região relevante. A informação de distância relativa seria utilizada no momento da criação do HMM maior, posicionando os pequenos HMMs de forma a respeitar as distâncias encontradas. O funcionamento básico do HMM seria: cada parte do HMM pode reconhecer uma região relevante. Existem ligações entre estas partes e para passar de um pequeno HMM para outro, é necessário que o atual HMM reconheça sua entrada. No final, o resultado é um escore, de posse neste escore pode-se decidir se a seqüência pertence ou não à família de genes.

Esta última abordagem apresenta bons resultados e é uma prova de que a abordagem com HMM realmente funciona. Esta seria uma boa abordagem para se implementar, porém ela é complexa e necessitaria de um tempo maior para ser realizada. Outro motivo para não utilizá-la é o desejo de testar uma nova abordagem, que será mostrada na próxima subseção, cujas similares são citadas na literatura, mas não são encontradas em aplicações.

4.2 Abordagem utilizada

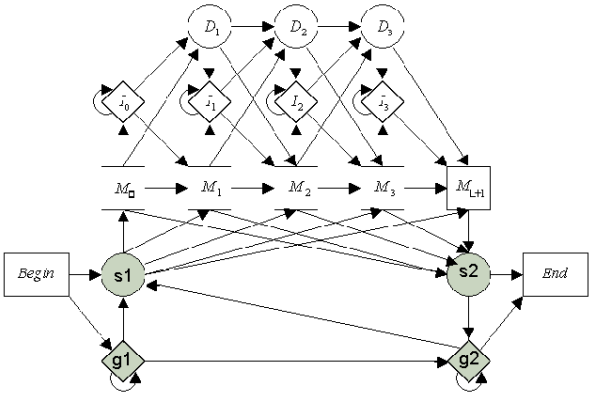

Figura 6: Modelo utilizado no projeto

A abordagem escolhida para tentar resolver o problema de reconhecer regiões promotoras de genes de reparação dos sacaramídeos foi uma mistura das abordagens que utilizavam HMM, apresentadas na seção anterior. Um modelo é apresentado na **Figura 6**.

É possível notar que existe uma ligação entre os estados de consumo (losangos) e também existe uma ligação entre o segundo losango e o primeiro círculo, tornando possível encontrar mais de uma região promotora. Com isto, o modelo fica similar ao segundo modelo apresentado na subseção anterior. Ao mesmo tempo, o nosso modelo apresenta dois modos de entrar na região promotora, o primeiro e o segundo losango, respectivamente. Deste modo, podemos escolher probabilidades diferentes para encontrar a primeira região promotora (vindo do primeiro losango) e para encontrar as demais regiões, se existirem, (vindo do segundo losango). Assim, o modelo também é parecido com a abordagem do primeiro modelo da subseção anterior.

Com as alterações realizadas, nós esperamos satisfazer o problema de encontrar mais de uma região apresentado pelo primeiro modelo, e esperamos resolver também o problema de usar a mesma probabilidade para encontrar a primeira e as demais regiões promotoras apresentado pelo segundo modelo. Com isto pretendemos tornar o nosso modelo mais apto para encontrar regiões promotoras.

O modelo é composto de um perfil HMM, usado para realizar o alinhamento local com a região promotora que desejamos encontrar, e de seis estados externos. O perfil HMM possui os estados de *match*, *insertion* e *deletion*, como descrito na seção anterior. Os elos

de ligação seguem as descrições da mesma seção e só os estados *match* e *insertion* podem emitir símbolos, lembrando que os estados M_0 e M_{L+1} são exceções e não emitem símbolos.

Os seis estados externos são:

- *BEGIN* (bg) - estado de início do programa.
- *greedy1* (g_1) - estado que serve para consumir os símbolos que ficam antes da região promotora. É representado pelo losango da esquerda na Figura 6.
- *silent1* (s_1) - estado que serve de passagem para entrar no perfil HMM. É representado pelo círculo da esquerda.
- *silent2* (s_2) - estado que serve de passagem para sair do perfil HMM. É representado pelo círculo da direita.
- *greedy2* (g_2) - estado que serve para consumir os símbolos depois da região promotora. Pode sair do programa ou entrar em uma nova região promotora. É representado pelo losango da direita.
- *END* - estado de fim do programa.

Dentre os seis estados externos os únicos que emitem símbolos são: *greedy1* e *greedy2*. A probabilidade de emissão é $p(b)$.

O estado *BEGIN* possui dois elos de ligação, um para o estado *silent1* e outro para o estado *greedy1*. O primeiro elo representa a possibilidade de termos uma região promotora no início da seqüência de entrada. O segundo representa a possibilidade de existir uma subseqüência antes da região que queremos encontrar.

Os estados *silent1* e *silent2* apesar do mesmo nome e funções parecidas não possuem as mesmas ligações. O estado *silent1* possui L+2 elos de saída para os estados *match*. O primeiro elo vai para o estado M_0, o segundo vai para M_1, o terceiro para M_2 e assim por diante até que o último elo se ligue a M_{L+1}. Estes elos representam a possibilidade de entrarmos na região promotora. Se escolhermos entrar por M_0, estamos fazendo um alinhamento com toda região promotora, se escolhermos um estado *match* na metade do perfil, por exemplo $M_{\frac{L}{2}}$, estamos alinhando com metade da região. Se o estado M_{L+1} for escolhido, não fazemos nenhum alinhamento. Já o estado *silent2* possui dois elos de saída: um para o estado *greedy2* e outro para o estado *END*. Estes elos representam, respectivamente, a possibilidade de exitir uma subseqüência depois da região promotora e a possibilidade de não existirem mais símbolos, logo o alinhamento terminou.

O estado *greedy1* possui três elos de saída. O primeiro vai para *silent1* e representa a possibilidade de existirem símbolos antes da região promotora e então decidirmos entrar na região promotora. O segundo elo é um laço para o próprio *greedy1*, representando a idéia de ficar na região anterior à promotora. O último elo liga *greedy1* a *greedy2* e representa a possibilidade de não existir região promotora na seqüência de entrada, então passamos de uma subseqüência antes da região promotora para uma subseqüência depois.

O estado *greedy2* também possui três elos de saída: um para *silent1*, o segundo para o próprio *greedy2* e o último para o estado *END*. O primeiro elo serve para fornecer a possibilidade de existir mais de uma região promotora dentro da seqüência de entrada. O segundo elo serve para consumir regiões depois da região promotora. O último elo é utilizado quando a seqüência foi totalmente consumida e temos que terminar o alinhamento.

Dado um perfil HMM Υ de tamanho L, uma seqüência $X = (x_1, x_2, \ldots, x_T)$ e os estados externos, os escores logarítmicos de Viterbi para cada estado externo são:

- $v_1^s(t)$ - escore do melhor caminho para casar a seqüência $X = (x_1, x_2, \ldots, x_t)$ com o nosso modelo, terminando no estado s_1, sem emitir nenhum símbolo.
- $v_1^g(t)$ - escore do melhor caminho para casar a seqüência $X = (x_1, x_2, \ldots, x_t)$ com o nosso modelo, terminando com x_t emitido pelo estado g_1.
- $v_2^s(t)$ - escore do melhor caminho para casar a seqüência $X = (x_1, x_2, \ldots, x_t)$ com o nosso modelo, terminando no estado s_2, sem emitir nenhum símbolo.
- $v_2^g(t)$ - escore do melhor caminho para casar a seqüência $X = (x_1, x_2, \ldots, x_t)$ com o nosso modelo, terminando com x_t emitido pelo estado g_2.

O algoritmo Viterbi que é utilizado para o cálculo do melhor caminho continua sendo realizado para os estados do perfil HMM do mesmo modo como mostrado na seção anterior, com exceção de M_0 e M_{L+1}. O cálculo dos estados externos e das exceções é mostrado abaixo:

1. Início:

$$v_{begin}(0) = 0 \tag{25}$$

2. Recursão :

$$v_0^M(t) = v_1^s(t) + \log\left(a_{s_1,M_0}\right) \tag{26}$$

$$v_1^g(t) = \log\left(\frac{e_{g_1}(x_t)}{p(x_t)}\right) + \max\left\{\begin{array}{l} v_{begin}(t-1) + \log\left(a_{begin,g_1}\right) \\ v_1^g(t-1) + \log\left(a_{g_1,g_1}\right) \end{array}\right. \tag{27}$$

$$v_2^g(t) = \log\left(\frac{e_{g_2}(x_t)}{p(x_t)}\right) + \max\left\{\begin{array}{l} v_2^s(t-1) + \log\left(a_{s_2,g_2}\right) \\ v_2^g(t-1) + \log\left(a_{g_2,g_2}\right) \\ v_1^g(t-1) + \log\left(a_{g_1,g_2}\right) \end{array}\right. \tag{28}$$

$$v_1^s(t) = \max\left\{\begin{array}{l} v_{begin}(t) + \log\left(a_{begin,s_1}\right) \\ v_1^g(t) + \log\left(a_{g_1,s_1}\right) \\ v_2^g(t) + \log\left(a_{g_2,s_1}\right) \end{array}\right. \tag{29}$$

$$v_{L+1}^{M}(t) = \max \begin{cases} v_L^M(t) + \log\left(a_{M_L, M_{L+1}}\right) \\ v_L^I(t) + \log\left(a_{I_L, M_{L+1}}\right) \\ v_L^D(t) + \log\left(a_{D_L, M_{L+1}}\right) \end{cases} \tag{30}$$

$$v_2^s(t) = \max\left(v_j^M(t) + \log\left(a_{M_j, s_2}\right) \right), \quad onde\ 0 \leq j \leq L+1 \tag{31}$$

3. Término:

$$P^* = \max \begin{cases} v_2^s(T) + \log\left(a_{s_2, END}\right) \\ v_2^g(T) + \log\left(a_{g_2, END}\right) \end{cases} \tag{32}$$

A complexidade deste modelo pode ser calculada do seguinte modo: nós calculamos $O((L+6) \cdot T)$ valores, gastando $O(1)$ (visto que nós precisamos considerar os escores de no máximo três predecessores) para calcular a maioria dos valores, porém o cálculo de s_2 leva $O(L+2)$. A complexidade do tempo total é $O(L^2 \cdot T)$ e a complexidade do espaço é $O(L \cdot T)$.

4.3 Implementação

A implementação do programa foi feita na linguagem PERL. Foi desenvolvido um modelo como o apresentado na subseção anterior. Existe um perfil HMM e seis estados externos.

O perfil HMM foi treinado com 18 seqüências conhecidas de região promotora de sacaramídeos para poder reconhecer regiões similares em outras seqüências. Seu treinamento utilizou o procedimento usado por Bastos, Falcão e Guimarães [17] anteriormente, porém em um novo contexto. Ao invés de treinarmos o HMM para famílias de proteínas, o treinamos para seqüências de nucleotídeos. As probabilidades de transição de estados e de emissão de símbolos foram obtidas automaticamente através do mesmo procedimento.

Também neste passo foram encontradas as freqüências *background* para o programa. Foi contado o número de ocorrência de cada base, somou-se todas as ocorrências e o valor da freqüência foi o número de ocorrências de uma base dividido pela soma das ocorrências de todas as bases. Formalmente,

$$p(b) = \frac{\sum b}{\sum\limits_{i \in \Sigma} i}$$

Os estados *greedy1* e *greedy2* usaram estes valores como probabilidades de emissão de símbolos.

As probabilidades de transição dos estados externos foi a parte mais difícil de se conseguir no trabalho. Esta dificuldade aconteceu porque diferentemente do perfil HMM cujas probabilidades foram encontradas através de Bastos et al. [17], que utiliza o algoritmo

forward-backward, as probabilidades dos estados externos foram determinadas de maneira manual. Esta decisão foi tomada por dois motivos: o primeiro foi que a implementação do algoritmo *forward-backward* para a obtenção das probabilidades dos estados externos estava fora do escopo do trabalho (como dito na Seção 3.4); e o segundo é que este algoritmo é difícil de implementar [4, 15]. Por estes motivos, num primeiro momento resolvemos testar valores que segundo nossa experiência e nossa suposição de como o modelo funciona levariam a resultados satisfatórios.

Os valores de transição dos estados externos foram decididos baseados na seguinte lógica: considere uma seqüência de entrada contendo uma região promotora. Sabe-se que a região promotora é, geralmente, bem menor do que a seqüência, logo é preciso consumir uma grande parte da seqüência até que se chegue à região desejada. Neste momento, entraria-se no perfil HMM, que está treinado para reconhecer esta região. Depois que se sai do perfil ter-se-ia três opções: a seqüência acabou e, conseqüentemente, o programa acabou; existem outros símbolos depois da região promotora, então ou consumimos todos os símbolos até o final da seqüência ou encontramos outra região promotora.

Pela idéia apresentada acima percebemos que a transição de *greedy1* para ele mesmo possui uma grande probabilidade, muito maior do que as duas outras probabilidades que *greedy1* possui.

A partir da mesma idéia, pode-se perceber que ao sair do perfil HMM é grande a probabilidade de ainda existirem símbolos da seqüência, logo deseja-se que a probabilidade de ir de *silent2* para *greedy2* seja grande.

E estando no estado *greedy2*, existe uma probabilidade maior de continuar neste estado do que voltar para uma região promotora.

Esta é idéia na qual nos baseamos para decidir os valores das probabilidades de transição dos estados externos. A idéia é válida e interessante, porém ao lidarmos com modelos probabilísticos, a escolha das decisões é feita nas casas decimais. Dito isto, é possível que o modelo não apresente resultados tão bons quanto se fosse usado o algoritmo *forward-backward* [1, 2, 3, 4, 5]. Porém, mesmo assim é válido fazer os testes e registrá-los para futuras comparações.

5 Testes

Foram usadas 18 seqüências conhecidas de regiões promotoras de sacaramídeos para treinar o perfil HMM.

Foram separados três conjuntos de testes: o primeiro contendo 33 seqüências com o comprimento de 15 aminoácidos, o segundo contendo 15 seqüências com o comprimento 100 aminoácidos e o terceiro contendo 15 seqüências com o comprimento de 1200 aminoácidos. Todos as seqüências de testes eram de sacaramídeos obtidas no SGD [20], algumas contendo a região promotora e outras não.

5.1 Primeiro Conjunto de Testes

Este primeiro conjunto de testes foi dividido em 5 (cinco) fases e possui o objetivo de encontrar um conjunto de probabilidades para o perfil HMM e ajustar as probabilidades de transição dos estados externos. As três primeiras serviram para encontrar qual conjunto de probabilidades de transição e de emissão apresentava melhores resultados para o perfil HMM, e as duas últimas fases tentaram encontrar valores para as probabilidades dos estados externos que apresentassem resultados significativos.

Serão especificadas em cada teste realizado as probabilidades de transição de : a_{begin,s_1}, a_{g_1,s_1}, a_{g_1,g_2}, a_{s_1,M_0}, $a_{s_1,M_{L+1}}$, a_{M_l,s_2} (probabilidade de ir de um estado *match* para o estado *silent2*), a_{s_2,g_2}, a_{g_2,s_1}, $a_{g_2,END}$. As probabilidades não especificadas podem ser deduzidas a partir destas.

O primeiro conjunto possui as seguintes seqüências:

- 10 seqüências usadas para treinar o perfil HMM;
- 18 seqüências conhecidas de regiões promotoras de sacaramídeos não usadas no treinamento;
- 5 seqüências de partes de genes de sacaramídeos, que não são regiões promotoras. Os genes utilizados foram : **YLR383W**, **YGR258C**, **YER142C**, **YDR263C** e **YMR137C**.

Nas tabelas que aparecerão a seguir a seguinte notação será utilizada:

- Grupo 1 - são as 10 seqüências usadas para treinar o perfil HMM;
- Grupo 2 - são as 18 seqüências conhecidas não usadas no treinamento;
- Grupo 3 - são as 5 seqüências de partes de genes de sacaramídeos, que não são regiões promotoras.
- Seqüências - Seqüências que foram utilizadas, podem ser do **Grupo 1**, **Grupo 2** ou **Grupo 3**.
- Quantidade - Quantidade de seqüências utilizadas nos testes.
- Acertos - Quantas regiões foram corretamente reconhecidas, em porcentagem.
- Falsos Positivos - Porcentagem de regiões que foram ditas positivas e não eram.
- Falsos Negativos - Porcentagem de regiões que deveriam ser assinaladas como positivas e não foram.

Fase 1

O primeiro conjunto de probabilidades utilizado para o perfil foi o primeiro conjunto obtido através do programa de Bastos et al. [17] em um novo contexto. Este conjunto

apresenta valores completamente randômicos. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.05$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0025$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	10	0%	0%	100%
Grupo 2	18	0%	0%	100%
Grupo 3	5	60%	40%	0%

Tabela 1: Tabela da Fase 1 do Primeiro Teste

É possível perceber na **Tabela 1** que este conjunto de probabilidades não está treinado para o perfil HMM, pois não reconhece nenhuma seqüência usada no treinamento, realmente apresentando valores completamente randômicos.

Fase 2

O segundo conjunto de probabilidades utilizado para o perfil foi o quarto conjunto obtido através do programa de Bastos et al. [17] em um novo contexto. Este conjunto apresenta valores da quarta iteração do algoritmo *forward-backward*. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.05$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$

- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0025$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	10	70%	0%	30%
Grupo 2	18	38,9%	0%	61,1%
Grupo 3	5	100%	0%	0%

Tabela 2: Tabela da Fase 2 do Primeiro Teste

Nesta fase do teste, pode ser visto claramente na **Tabela 2** o algoritmo *forward-backward* fazendo o perfil HMM ser treinado para as regiões promotoras que foram usadas como conjunto de treinamento. E pode ser percebido também que o perfil começa a reconhecer regiões promotoras similares (grupo 2).

Fase 3

O terceiro conjunto de probabilidades utilizado para o perfil foi o sétimo conjunto obtido através do programa de Bastos et al. [17] em um novo contexto. Este conjunto apresenta valores da sétima iteração do algoritmo *forward-backward*. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.05$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0025$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	10	90%	0%	10%
Grupo 2	18	61,1%	0%	38,9%
Grupo 3	5	100%	0%	0%

Tabela 3: Tabela da Fase 3 do Primeiro Teste

Nesta fase do teste, pode ser percebido pela **Tabela 3** que o perfil está bem treinado para as seqüências de treinamento e está com um valor aceitável para as seqüências que são regiões promotoras e não foram usadas no treinamento (grupo 2). Pode-se ver que o modelo apresenta bons resultados para seqüências que possuem o tamanho das regiões promotoras. Este conjunto de probabilidades para o perfil HMM será utilizado para os outros testes. Precisamos agora determinar valores para as transições dos estados externos.

Fase 4

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.25$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0625$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	10	90%	0%	10%
Grupo 2	18	66,7%	0%	33,3%
Grupo 3	5	100%	0%	0%

Tabela 4: Tabela da Fase 4 do Primeiro Teste

Nesta fase do teste, foram alterados os valores de a_{g_1,s_1} para 0.25 e de a_{g_2,s_1} para 0.0625. Os resultados dos testes estão na **Tabela 4**. Esta idéia de aumentar a probabilidade de

greedy1 para *silent1* e de *greedy2* para *silent1* pode ser interessante para seqüências maiores. Podemos perceber que houve um aumento na quantidade de seqüências não treinadas reconhecidas.

Fase 5

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.5$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.25$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	10	100%	0%	0%
Grupo 2	18	77,8%	0%	22,2%
Grupo 3	5	100%	0%	0%

Tabela 5: Tabela da Fase 5 do Primeiro Teste

Nesta fase do teste, foram alterados os valores de a_{g_1,s_1} para 0.5 e de a_{g_2,s_1} para 0.25. Os resultados dos testes estão na **Tabela 5**. De novo, é possível se perceber um aumento na quantidade de seqüências não treinadas sendo reconhecidas.

Os valores das probabilidades dos estados externos utilizados nas fases 3, 4 e 5 serão utilizados nos testes das próximas subseções. É válido lembrar que muitas probabilidades dos estados externos não foram alteradas e elas também podem influenciar nos resultados obtidos.

5.2 Segundo Conjunto de Testes

Este segundo conjunto de testes foi dividido em 3 (três) fases e possui o objetivo de testar os resultados das probabilidades encontradas na subseção anterior.

Serão especificadas em cada teste realizado as probabilidades de transição de : a_{begin,s_1}, a_{g_1,s_1}, a_{g_1,g_2}, a_{s_1,M_0}, $a_{s_1,M_{L+1}}$, a_{M_l,s_2} (probabilidade de ir de um estado *match* para o estado *silent2*), a_{s_2,g_2}, a_{g_2,s_1}, $a_{g_2,END}$. As probabilidades não especificadas podem ser deduzidas a partir destas.

Este segundo conjunto de teste possui as seguintes seqüências com o comprimento de 100 aminoácidos:

- 5 seqüências usadas para treinar o perfil HMM. As seqüências utilizadas foram: **YMR137C**, **YDR263C**, **YGR258C**, **YLR383W** e **YMR201C**. A região promotora está dentro destas 100 bases;
- 5 seqüências conhecidas de regiões promotoras de sacaramídeos não usadas no treinamento. As seqüências utilizadas foram: **YDR076W**, **YML095C**, **YCR066W**, **YOR386W** e **YBR114W**. A região promotora está dentro destas 100 bases.
- 5 seqüências de sacaramídeos, que não possuem regiões promotoras. Os genes utilizados foram : **YIL128W**, **YIL143C**, **YGL058W**, **YLR288C** e **YDR076W**.

Nas tabelas que aparecerão a seguir a seguinte notação será utilizada:

- Grupo 1 - são as 5 seqüências usadas para treinar o perfil HMM;
- Grupo 2 - são as 5 seqüências conhecidas não usadas no treinamento;
- Grupo 3 - são as 5 seqüências que não possuem regiões promotoras.
- Seqüências - Seqüências que foram utilizadas, podem ser do **Grupo 1**, **Grupo 2** ou **Grupo 3**.
- Quantidade - Quantidade de seqüências utilizadas nos testes.
- Acertos - Quantas regiões foram corretamente reconhecidas, em porcentagem.
- Falsos Positivos - Porcentagem de regiões que foram ditas positivas e não eram.
- Falsos Negativos - Porcentagem de regiões que deveriam ser assinaladas como positivas e não foram.

Fase 1

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$

- $a_{g_1,s_1} = 0.05$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0025$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	5	0%	0%	100%
Grupo 2	5	0%	0%	100%
Grupo 3	5	100%	0%	0%

Tabela 6: Tabela da Fase 1 do Segundo Teste

Nesta fase do teste, pode ser percebido que os valores dos estados externos não estão ajustados para o reconhecimento de regiões promotoras em seqüências grandes. Pode ser percebido que mesmo apresentando seqüências que possuem regiões promotoras nenhuma delas foi reconhecida (grupos 1 e 2). Os resultados do teste nesta fase estão na **Tabela 6**.

Fase 2

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.25$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0625$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	5	0%	0%	100%
Grupo 2	5	0%	0%	100%
Grupo 3	5	100%	0%	0%

Tabela 7: Tabela da Fase 2 do Segundo Teste

- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Olhando-se a **Tabela 7**, pode-se perceber que nenhuma região foi reconhecida, mesmo existindo regiões promotoras nos dois primeiros grupos. Um fato interessante para ser registrado é que as seqüências foram consumidas pelo estado *greedy2* e não pelo estado *greedy1* como era esperado. Esta constatação reforça a hipótese de que é realmente necessário um algoritmo ou um modo automático para se determinar as probabilidades dos estados externos, estados estes que afetam diretamente o alinhamento do modelo contra uma seqüência com maior quantidade de bases.

Fase 3

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.5$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.25$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

No grupo 1, houve apenas uma região indicada e ela realmente era uma região promotora. Isto ocorreu em uma única seqüência. Nas outras seqüências utilizadas no treinamento, não foram reconhecidas outras regiões.

No grupo 2, o modelo reconheceu corretamente um região promotora e não reconheceu quatro outras regiões válidas. O modelo indicou como região promotora a parte de uma seqüência que não era região promotora.

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	5	20%	0%	80%
Grupo 2	5	20%	20%	80%
Grupo 3	5	100%	0%	0%

Tabela 8: Tabela da Fase 3 do Segundo Teste

No grupo 3 nenhuma região foi reconhecida. Isto é bom já que realmente não existem regiões promotoras nestas seqüências.

O resultado do teste desta fase é mostrado na **Tabela 8**.

Dois fatos interessantes foram notados. O primeiro foi que todas as subseqüências que foram reconhecidas como regiões promotoras apresentavam a terminação **TGAAA**. Isto pode indicar que o conjunto de treino apresentava seqüências com esta terminação específica. Ao se examinar o conjunto de treino esta sugestão é comprovada. O segundo fato interessante foi que o modelo não seguiu o caminho imaginado (veja seção anterior), e sim, foi direto para o estado *greedy2* e depois entrou no perfil HMM.

De posse destes dois fatos pode-se tirar algumas sugestões: o conjunto de testes tem que ser revisto e modificado para possuir uma quantidade mais representativa de regiões promotoras; as probabilidades dos estados externos têm que ser revista e melhor analisada para entendermos o funcionamento do modelo. A utilização do algoritmo *forward-backward* para a obtenção destas probabilidades seria de grande utilidade como já foi mencionado.

5.3 Terceiro Conjunto de Testes

Este terceiro conjunto de testes foi dividido em 3 (três) fases e possui o objetivo de testar os resultados das probabilidades encontradas na primeira subseção e avaliar a influência do tamanho da seqüência nos resultados.

Serão especificadas em cada teste realizado as probabilidades de transição de : a_{begin,s_1}, a_{g_1,s_1}, a_{g_1,g_2}, a_{s_1,M_0}, $a_{s_1,M_{L+1}}$, a_{M_l,s_2} (probabilidade de ir de um estado *match* para o estado *silent2*), a_{s_2,g_2}, a_{g_2,s_1}, $a_{g_2,END}$. As probabilidades não especificadas podem ser deduzidas a partir destas.

Este terceiro conjunto de teste é composto das seguintes seqüências:

- 5 seqüências usadas para treinar o perfil HMM. As seqüências utilizadas foram: **YMR137C**, **YDR263C**, **YGR258C**, **YLR383W** e **YMR201C**. A região promotora está dentro destas 100 bases;

- 5 seqüências conhecidas de regiões promotoras de sacaramídeos não usadas no treinamento. As seqüências utilizadas foram: **YDR076W**, **YML095C**, **YCR066W**, **YOR386W** e **YBR114W**. A região promotora está dentro destas 100 bases.

- 5 seqüências de sacaramídeos, que não possuem regiões promotoras. Os genes utilizados foram : **YIL128W**, **YIL143C**, **YGL058W**, **YLR288C** e **YDR076W**.

Nas tabelas que aparecerão a seguir a seguinte notação será utilizada:

- Grupo 1 - são as 5 seqüências usadas para treinar o perfil HMM;
- Grupo 2 - são as 5 seqüências conhecidas não usadas no treinamento;
- Grupo 3 - são as 5 seqüências que não possuem regiões promotoras.
- Seqüências - Seqüências que foram utilizadas, podem ser do **Grupo 1**, **Grupo 2** ou **Grupo 3**.
- Quantidade - Quantidade de seqüências utilizadas nos testes.
- Acertos - Quantas regiões foram corretamente reconhecidas, em porcentagem.
- Falsos Positivos - Porcentagem de regiões que foram ditas positivas e não eram.
- Falsos Negativos - Porcentagem de regiões que deveriam ser assinaladas como positivas e não foram.

Fase 1

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.05$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0025$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Esta fase do teste apresenta os mesmos resultados apresentados na Primeira Fase do Segundo teste. Pode ser percebido que estes valores para os estados externos não estão ajustados para o reconhecimento de regiões promotoras em seqüências grandes. Pode ser percebido que mesmo apresentando seqüências que possuem regiões promotoras nenhuma delas foi reconhecida (grupos 1 e 2). Os resultados desta fase estão na **Tabela 9**.

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	5	0%	0%	100%
Grupo 2	5	0%	0%	100%
Grupo 3	5	100%	0%	0%

Tabela 9: Tabela da Fase 1 do Terceiro Teste

Fase 2

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.25$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.0625$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	5	0%	0%	100%
Grupo 2	5	0%	0%	100%
Grupo 3	5	100%	0%	0%

Tabela 10: Tabela da Fase 2 do Terceiro Teste

Pode-se perceber que nenhuma região foi reconhecida, mesmo existindo regiões promotoras nos dois primeiros grupos. Estes resultados são idênticos aos da Segunda Fase do Segundo teste. Isto vem comprovar que estes valores das probabilidades dos estados externos não funcionam bem com seqüências grandes. Um outro ponto a ser levantado é o tamanho das seqüências, que são 10 vezes maiores do que as utilizadas no teste 2 e quase 100 vezes maior daquelas do teste 1. Ainda não é possível afirmar se apenas as probabilidades dos estados externos influenciam nos resultados ou se o tamanho das seqüências também é importante. De maneira similar ao apresentado na Fase 2 do segundo teste os símbolos das

seqüências foram completamente consumidos pelo estado *greedy2*. Os resultados podem ser vistos na **Tabela 10**.

Fase 3

O conjunto de probabilidades utilizado para o perfil foi o conjunto utilizado na Fase 3 do Primeiro Teste. Os valores das probabilidades dos estados externos são:

- $a_{begin,s_1} = 0.95$
- $a_{g_1,s_1} = 0.5$
- $a_{g_1,g_2} = 0.0001$
- $a_{s_1,M_0} = 0.9$
- $a_{s_1,M_{L+1}} = 0.00625$
- $a_{M_l,s_2} = 0.001$
- $a_{s_2,g_2} = 0.9$
- $a_{g_2,s_1} = 0.25$
- $a_{g_2,END} = 6.25 \cdot 10^{-6}$

Seqüências	Quantidade	Acertos (%)	Falsos Positivos (%)	Falsos Negativos (%)
Grupo 1	5	0%	0%	100%
Grupo 2	5	0%	0%	100%
Grupo 3	5	100%	0%	0%

Tabela 11: Tabela da Fase 3 do Terceiro Teste

Nesta fase, pode ser percebido que os valores utilizados não conseguiram reconhecer nenhuma região promotora. Isto comprova que não apenas as probabilidades dos estados externos influenciam nos resultados, mas também o tamanho das seqüências a serem avaliadas.

Seriam necessários mais testes com este tamanho de seqüências para que conclusões mais significativas fossem encontradas. De posse dos fatos apresentados pode-se afirmar que as probabilidades dos estados externos são um fator importante para o reconhecimento das regiões promotoras. Outra conclusão é que o tamanho das seqüências também influencia no reconhecimento das regiões que desejamos encontrar. Os resultados estão na **Tabela 11**.

6 Conclusão e Trabalhos Futuros

É possível notar pelo trabalho que um modelo matemático proposto no fim dos anos 60 e que foi inicialmente utilizado no problema de reconhecimento da fala, mostra-se promissor para a área de bio-informática. O HMM pode ser utilizado em diversas aplicações como por exemplo na predição de genes e na classificação de proteínas em famílias, entre outras. A aplicação mais comum do HMM é como perfil HMM, forma esta que foi utilizada neste trabalho.

O objetivo deste trabalho era criar um HMM para encontrar regiões promotoras em seqüências de sacaramídeos. Foi desenvolvido um modelo que apresenta uma abordagem diferente das apresentadas na literatura, utilizando perfil HMM e seis estados externos. Estes estados servem, basicamente, para consumir símbolos antes e depois da região promotora. O modelo foi proposto para tentar solucionar dois problemas: o problema de existir mais de uma região promotora dentro de uma seqüência de entrada e o problema de fornecer diferentes probabilidades para encontrar estas regiões. O modelo proposto foi criado e implementado como previsto.

As dificuldades para a criação e implementação do modelo aconteceram principalmente pela falta de aplicações e de códigos fontes que representassem modelos similares ao apresentado. Na literatura existem alguns livros que sugerem como pode ser feito um modelo assim, mas a transformação da teoria na prática sempre é trabalhosa. Contribuiu também para as dificuldades a determinação das probabilidades de transição dos estados externos. O modo mais eficiente para determiná-las seria usando o algoritmo *forward-backward*, porém é dito na literatura que este é o algoritmo mais complicado de ser implementado [4, 15]. Sendo assim, as probabilidades de transição foram determinadas de forma empírica.

Os resultados dos testes realizados foram realmente bons para seqüências com o tamanho de uma região promotora (15 aminoácidos). O modelo apresentou resultados promissores para seqüências com o comprimento de 100 aminoácidos, reconhecendo algumas regiões promotoras, mas mostrando uma tendência de reconhecer um tipo específico de terminação. Para seqüências realmente grandes (mais de 1000 aminoácidos) o modelo não apresentou resultados significativos, mas é possível tirar a seguinte conclusão: tanto as probabilidades dos estados externos quanto o tamanho das seqüências influenciam no reconhecimento de uma região promotora.

A influência das probabilidades dos estados externos era esperada, já a influência do tamanho das seqüências era uma possibilidade, mas esperava-se que o modelo pudesse encontrar as regiões promotoras independentemente deste tamanho. Uma possível solução seria a utilização do algoritmo *forward-backward*. Ao fazer uso deste algoritmo, seqüências de treinamentos maiores teriam que ser usadas (neste trabalho foram utilizadas seqüências com o tamanho da região promotora). Isto garantiria uma melhor adequação das probabilidades dos estados externos às seqüências do conjunto de treinamento, porém não garantiria que o modelo realmente conseguiria reconhecer regiões promotoras em seqüências muito grandes. Usar o algoritmo *forward-backward* para treinar todo o modelo seria uma melhoria do trabalho e serviria para verificar se, realmente, o algoritmo apresentaria resultados melhores.

Outro trabalho futuro seria modificar outras probabilidades dos estados externos, pois neste trabalho apenas a_{g_1,s_1} e a_{g_2,s_1} foram alteradas. De posse dos resultados obtidos nos testes realizados, sabe-se que as probabilidades dos estados externos influenciam no reconhecimento das regiões procuradas, assim seria uma possível extensão, modificar outras probabilidades e comparar os resultados.

Uma maneira de comprovar e validar o trabalho seria implementar as duas outras abordagens apresentadas no início da Seção 4. Ao se fazer isto, poderia-se confirmar nossas suposições de que os outros modelos apresentam os problemas sugeridos e que o nosso modelo consegue resolver estes problemas, e seria uma forma válida de se realizar uma comparação entre os modelos.

Outra possível extensão do trabalho seria a utilização de informações extras para construção do modelo. Poder-se-ia utilizar informações como: posição relativa das regiões promotoras, verificação de outros sinais biológicos conhecidos, etc. Estas informações poderiam ser reconhecidas por pequenos HMMs. Desta forma, seria criado um tipo de HMM composto de partes de HMM cujo funcionamento básico seria: cada parte pode reconhecer uma informação específica; as partes se relacionam, e para passar de um HMM para outro é necessário que o HMM atual reconheça sua entrada. Desta forma, o HMM funcionaria como um autômato reconhecedor de uma linguagem específica. Estas informações podem ser obtidas na própria biologia que já conhece muitos dos sinais utilizados em certos processos biológicos. Porém para se conseguir estas informações seria necessária uma parceria com a biologia. Este trabalho levaria bastante tempo, mas poderia produzir resultados significativos.

De uma maneira geral, o trabalho apresentou bons resultados com a sugestão e implementação de um novo modelo para reconhecimento de regiões promotoras, abrindo caminhos para derivações e expansões. Esta nova abordagem foi utilizada para reconhecer regiões promotoras de genes de reparação dos sacaramídeos, mas pode ser usada no reconhecimento de outras regiões promotoras dos sacaramídeos, no reconhecimento de regiões de outros organismos e pode até ser utilizado para reconhecer domínios de proteínas. O modelo pode e deve ser melhorado no processo de obtenção das probabilidades de transição de forma automática, podendo isto ser realizado em trabalhos futuros.

7 Glossário

- **Base**
  Adenina, citosina, guanina, timina e (apenas para RNA) uracila. São chamadas bases porque são alcalinas (básicas) na estrutura ácida do DNA. As bases são as letras que determinam o código genético. No DNA, os códigos das letras são A, C, G e T correspondentes aos nomes citados acima, respectivamente. Nos pares de base, adenina sempre aparece com a timina e a citosina sempre aparece com a guanina.

- **Códon**
  A seqüência de três nucleotídeos em uma região codificadora de proteína que espe-

cifica um aminoácido individual ou representa os sinais de início (*start codon*) ou término (*stop codon*) da síntese da proteína.

- **Cromossomo**
  Estrutura no núcleo da célula que contém todo o DNA celular junto com muitas proteínas que compactam e empacotam o DNA. Ele serve para a armazenamento e transmissão de informação genética.

- **Domínio**
  Uma região da seqüência de aminoácidos de uma proteína que possui significado evolucionário, estrutural ou funcional. A seqüência de aminoácidos de um domínio determina a estrutura 3D de uma proteína.

- **Downstream**
  Identifica seqüências que se situam na direção da expressão do gene, por exemplo, a região codificante está *downstream* a partir do *start codon*, na direção 3' de uma molécula de RNA mensageiro.

- **Enhancer**
  Uma seqüência que aumenta a utilização de (algunss) *promoters* nos eucariontes e pode funcionar nas duas orientações e em qualquer localização (*upstream* ou *downstream*) relativa ao *promoter*.

- **Expressão do Gene**
  A conversão de uma informação a partir do gene para uma proteína, via transcrição e tradução.

- **Gene**
  Classicamente, uma unidade de hereditariedade. Na prática, um gene é um segmento de DNA dentro do cromossomo que codifica uma proteína e todas as seqüências reguladoras exigidas para controlar a expressão daquela proteína.

- **Motif**
  Um curta região conservada numa seqüência de proteína. Frequentemente, são partes altamente conservadas de domínios.

- **Promoter**
  Regiões na molécula de DNA envolvidas pelo RNA polimerase para iniciar a transcrição. *Promoters* são seqüências no lado 5' do gene nas quais o RNA polimerase se liga quando a transcrição começa. Em todos os grupos de organismos, foram encontrados *promoters* alternativos para muitos genes. Estes *promoters* foram classificados em seis classes. Certos tipos de *promoters* alternativos tornam possível que a transcrição comece em diferentes pontos do gene em diferentes casos e que o produto da transcrição tenha diferentes códon de iniciação em diferentes posições do cromossomo. Então, o gene pode produzir mais de um tipo de moléculas de RNA mensageiro, codificando mais de uma proteína.

- **Regulação da expressão do gene**
  Qualquer um dos processos pelos quais fatores nucleares, citoplasmáticos ou intercelulares influenciam o controle diferencial da ação do gene no nível de transcrição ou tradução. Estes processos incluem ativação e indução gênica.

- **Região promotora**
  Região do DNA, geralmente anterior à seqüência codificante do gene, que liga e direciona o RNA polimerase para o correto sítio transcriptacional de início e, então, permite o início da transcrição. Geralmente, é uma região altamente conservada dentro do *promoter*.

- **Start Codon** (Códon de Início)
  Conjunto de três nucleotídeos numa molécula de RNA mensageiro no qual o ribossomo começa o processo de tradução. O mais comum *start codon* nos eucariontes é o AUG (ATG).

- **Stop Codon** (Códon de Parada)
  Conjunto de três nucleotídeos para os quais não há nenhuma molécula correspondente de RNA tradutor para inserir um aminoácido na cadeia de proteína. A síntese protéica é então terminada e a proteína completa é liberada pelo ribossomo. Os três *stop codons* encontrados são: UAA (TAA), UAG (TAG) e UGA (TGA).

- **Tradução**
  Processo unilateral que se realiza nos ribossomos pelo qual a informação genética presente no RNA mensageiro é convertida em uma seqüência correspondente de aminoácidos de uma proteína.

- **Transcrição**
  Processo pelo qual a informação genética codificada numa seqüência linear de nucleotídeos de uma fita de DNA é copiada em uma seqüência complementar de RNA.

- **Upstream**
  Identifica seqüências localizadas na direção oposta à direção da expressão do gene, por exemplo, a região promotora está *upstream* a partir do *start codon*, na direção 5' na molécula de RNA mensageiro.

Referências

[1] SHAMIR, RON, lecturer; WEIN, RON; AVRAHAMI, NIR, scribes. *Algorithms for Molecular Biology - Lecture 6: January 10, 1999.* Tel Aviv University, Fall Semester 1998.

[2] SHAMIR, RON, lecturer; GVIRTZER, OPHIR; GANON, ZOHAR, scribes. *Algorithms for Molecular Biology - Lecture 6: December 4, 2000* http://www.math.tau.ac.il/~rshamir/algmb/00/scribe00/html/lec06/lec06.html

[3] BALDI, PIERRE; BRUNACK, SOREN. *Bioinformatics - The Machine Learning Approach.* MIT Press, 2nd edition, 2001.

[4] RABINER, LAWRENCE R. *A Tutorial on Hidden Markov Models Selected Applications in Speech Recognition. Proceedings of IEEE*, vol. 77, no.2, February 1989.

[5] DURBIN, R; EDDY, S; KROGH, A; MITCHISON, G. *Biological sequence analysis - Probabilistic models of proteins and nucleic acids.* Cambridge University Press, 1998.

[6] KROGH, ANDERS. *Chapter 4 - An Introduction to Hidden Markov Models for Biological Sequences*, in *Computationial Methods in Molecular Biology* edited by S. L. Salzberg, D. B. Searls and S. Kasif, pages 45-63. Elsevier, 1998.

[7] Oxford University Press. *Nucleic Acids Research.*
http://nar.oupjournals.org/

[8] MARY CHITTY, Cambridge Healthtech Institute . *Genomics glossaries and taxonomies.* http://www.genomicglossaries.com/
Última revisão 28 de janeiro de 2003.

[9] *bioinfo_glossary.html* . Bioinformatics Glossary.
http://www.library.csi.cuny.edu/~davis/Bioinfo_326/bioinfo_glossary.html

[10] NATHALIE CASTELLS-BROOKE. *Beginner's Guide to Molecular Biology.*
http://www.rothamsted.bbsrc.ac.uk/notebook/courses/guide/.
Última atualização 22 de outubro de 2002.

[11] Cambridge Research Laboratory. *HTK - Hidden Markov Model Toolkit – speech recognition research toolkit.*
http://htk.eng.cam.ac.uk/

[12] L. ALLISON. *Minimum Message Length - MML.* School of Computer Science and Software Engineering, Monash University.
http://www.csse.monash.edu.au/~lloyd/tildeMML/Structured/HMM.html

[13] Sanger Institute. *Pfam: Pfam Home Page.*
http://www.sanger.ac.uk/Software/Pfam/

[14] PAVLIDIS, PAUL; FUREY, TERRENCE S.; LIBERTO,MURIEL; HAUSSLER,DAVID; GRUNDY, WILLIAM NOBLE. *Promoter region-based classification of genes. Proceedings of the Pacific Symposium on Biocomputing*, January 3-7, 2001. pp. 151-163.

[15] University of Leeds. *Hidden Markov Models.*
http://www.scs.leeds.ac.uk/scs-only/teaching-materials/HiddenMarkovModels/html_dev/main.html

[16] GREEN, SHELDON. *Hypertext Help with LaTeX*. http://www.giss.nasa.gov/latex/index.html.

[17] BASTOS S., GUSTAVO; FALCÃO, TACIANA PONTUAL DA R.; GUIMARÃES, KATIA S.; *An HH-Based Protein Family Classifier*. II Encontro Regional de Matemática Aplicada e Computacional (ERMAC 2002). pg. 51 - 52.

[18] GRUNDY, WN; BAILEY, TL; ELKAN, CP; BAKER, ME. *Meta-MEME: Motif-based hidden Markov models of protein families*. *CABIOS*, 13:397-406, 1997.

[19] UCSD; SDSC; NBCR. *The MEME/MAST System - Motif Discovery and Search*. http://meme.sdsc.edu/meme/website/intro.html. Version 3.0.

[20] US National Institutes of Health. *Saccharomyces Genome Database*. http://genome-www.stanford.edu/Saccharomyces/

[21] GREGORY, T. RYAN. *The human genome for not-so-dummies*. http://www.genomesize.com/rgregory/genome.html